IIAI Open Conference Publication Series
IIAI Letters on Informatics and Interdisciplinary Research
Vol. 003, LIIR074
DOI: https://doi.org/10.52731/liir.v003.074

Divide-and-Conquer in Automated Negotiations Through Utility Decomposition

Zongcan Li [*], Rafik Hadfi [*], Takayuki Ito [*]

Abstract

The success of a negotiation depends largely on the actors and the negotiation domain. It is common that negotiators rely on an agenda to simplify the process and reach better deals. This is particularly the case when the preferences of the negotiators are complex and when multiple issues are at stake. Using an agenda to explore and decompose the interdependence relationships between the issues is one way to address this problem. In this paper, we propose to address this challenge by applying the classical divide-and-conquer approach to automated negotiations through means of utility decomposition and bottom-up agenda construction. The approach does not impose an agenda from the top level of the negotiations, but builds it bottom-up given the individual utility functions of the agents and the relationships between the issues. Our approach reduces the cost of exploring the utility spaces of the agents and the resulting bidding processes. We implement the approach in a novel protocol called the Decomposable Alternating Offers Proto-col (DAOP). The experimental results show that our divide-and-conquer algorithm makes a positive influence on the global performance of an automated negotiation system.

Keywords: Agent Development, Automated Negotiation, Human-Agent Interaction, Multiagent Systems, Preference Elicitation

1 Introduction

Negotiation is a process by which several rational and self-interested parties attempt to reach an agreement [14]. The involved parties may compromise to achieve their purposes. This process of beneficial exchange happens in every aspect of people's daily lives. While negotiating with intelligent opponents such as humans, the outcome is not always guaranteed. In order to achieve a relatively optimal outcome, it is crucial to understand each other's requirements so that the negotiator can make concession appropriately.

* Department of Social Informatics, Graduate School of Informatics, Kyoto University

Automated negotiation has already been addressed in different ways in research [3, 14]. The general setting involves at least two parties, where each party has preferences, strategies, and possibly an agenda [15]. A negotiation strategy is a mapping from the state of the negotiations, as understood by the negotiator, to the actions allowed by the negotiation protocol (sometimes called a mechanism)[20]. Preferences are usually represented using a function that maps the bids to their value. In the following, we define an agenda as subsets of issues that the ne-gotiators negotiate about. For instance, in the case of online trading, the agenda usually consists of the price, the quantity and the delivery methods. In the whole process, the preferences of the self-interested agents have great influence on the decision-making process.

The dynamics of the negotiations are thoroughly studied in negotiation research. An unnoticeable variation of action sequence could cause huge deviation in the expected result. The reason for this, is that often times, the issues under consideration are usually not independent when it comes to a purchase or a sale, for example, of a bundle of items. Besides, if the agenda of the negotiation can be modeled appropriately, the value space of some issues could be so enormous that the negotiation becomes cognitively and computationally intractable [9]. To tackle the problem that most real-world negotiation face, previous research proposed to decompose contracts [7]. By evaluating the interdependence among issues in contracts, the issues could be grouped and an optimal bid selected by the mediator between two negotiators will be the final bid for both negotiators. Other methods relied on structured search and the use of agenda to simplify the contract spaces [28, 18]

This paper contributes to the negotiation research by proposing a novel negoti-ation protocol that uses the classical divide-and-conquer paradigm [24]. The basic idea of the divide-and-conquer paradigm is to decompose a given problem into two or more similar, but simpler, sub-problems, to solve them in turn, and to com-pose their solutions to solve the given problem. The divide-and-conquer approach is also a design pattern that can be applied to all kinds of computational problems such as optimization, prediction, and matching tasks [29, 8, 21].

Our approach is not only applied to the elicitation phase of the negotiation, but allows the division of the complete bidding space into several sub-bids that are later negotiated individually with the opponents. The approach optimizes the global efficiency and intuitively simplifies a complex utility space [10]. Our pro-posed protocol, namely Decomposable Alternating Offers Protocol (DAOP), allows the negotiators to negotiate over sub-bids and the utility functions of all negotia-tors is used to correctly compute the utilities of the sub-bids. This requires the decomposition of utility functions that are difficult to handle in the presence of nonlinear relationships for instance [16, 5].

The main mechanism in DAOP requires all the parties to participate in several rounds of negotiations toward one single agreement. First, the protocol will divide the complete bid into sub-sets of issues. Then, agents negotiate over these sub-sets of issues in order until reach all sub-agreements. Finally, the DAOP will combine the sub-agreements into a complete agreement. The negotiation is finished. In addition, at any time point if the negotiation reaches the time limit, the negotiation will be terminated without any agreement.

The paper is structured as follows. In the next section, we cover some of the key concepts in automated negotiation Then, we introduce our divide-and-conquer framework. In section 4, we provide the experimental results. Finally, we conclude and highlight the future directions.

2 Related Work

2.1 Automated Negotiation

The general interaction model of in agent-based negotiation is shown in figure 1. The process starts from two humans trying to reach a deal about a possible complex negotiation domains. Two agents will be used to firstly elicit their preferences, and then engage in bilateral negotiation concluded with an agreement, or not.

The use of artificial agents for this task takes advantage of the rapid develop-ment of computing technologies such as Deep Learning [23] or Smart Contracts [17]. Though the agents in negotiations seem to be competing, there are benefits to cooperation through means of concessions. This is the case when negotiations aim at win-win deals for all parties, while simultaneously reducing the time and effort, and adding significant value to system as a whole.

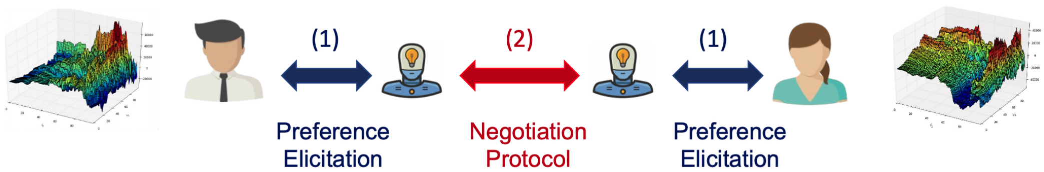

Figure 1: In a bilateral automated negotiation, artificial agents (1) elicit human preferences, and then (2) negotiate on their behalf

The agents are used to reduce the burden of eliciting complex preferences and then exchanging bids at a rate that humans cannot generally cope with. The process of querying humans to learn their preferences could be costly though and additional methods need to be used [4].

2.2 Utility Functions

The preferences of the negotiators are required for the agents to properly evaluate the offers of the opponents and propose offers that benefit their interests. The utility functions are therefore acquired from the negotiators' preferences. It is common practice to assume that the domains of the issues are discrete to avoid large utility spaces, which will consequently complexify the negotiations [9]. The size of the domains is only one of the obstacles on the way to put automated negotiation into real use. There are other considerations such as the shapes of utility functions. When the issues are independent, agents can aggregate the utilities of the issues by simple summation, producing linear utility functions [13]. Linear utility functions are commonly more tractable because of its regularity with respect to the value of issues.

2.3 Interdependent Issues

Interdependence between the issues is an aspect that characterizes most complex negotiations. The relationships between the issues are usually unknown across agents and even for the agent itself. This assumption is one that we take in our proposed protocol (DAOP) in the sense that each agent starts from an empty box of preferences, and needs to gradually extract the relationships between the issues given its own preferences. The approach here is to query the utility function of the agent and based on the result, the agent could explore these relationships.

There is abundant research on interdependence extraction in negotiation [11, 1]. For example, a method grouping highly interdependent issues by collecting meta-level information about the agenda in prior to the negotiation was developed and evaluated in [27]. A method was then proposed for using meta-negotiation. To better understand the influence of the input preference structure on the nego-tiators' performance, five parameters were defined to capture the topological and the interdependent characteristics of the preference structure.

The approach in [27] showed promising results on the effectiveness and compu-tational cost aspects. However, the process of meta-negotiation is not very intuitive comparing to real-world negotiations since the collection of meta-information and meta-negotiation cost extra time and resources. More importantly, as an offline method, the updates can hardly be incorporated at real-time level efficiency.

2.4 Negotiation Protocols

Negotiation protocols are often defined as the rules of interaction during a nego-tiation. These protocols are based for example on alternating offers [2], in which negotiators take turns as they exchange possible deals. In argumentation-based protocols, agents can exchange logical sentences intended to persuade one another to change their states of mind [26]. Other protocol rely on agendas [6]. In the stacked alternating offers protocol (SAOP) all of the negotiators around the table get a turn per round; turns are shuffled randomly. One of the negotiating parties starts the negotiation with an offer that is observed by all others. Whenever an offer is made, the next party in line can take the following actions: accept the offer, reject the offer, and repeat the process. Our proposed mechanism is an adaptation of SAOP, called the decomposable alternating offers protocol (DAOP). Although all of these protocols can hardly influence the process of negotiation, they can still be regarded as mediators between the negotiators since they work between them, which also make them an essential component of any negotiation.

2.5 Negotiation Platforms

There are many simulation platforms for the simulation and study of automated negotiation. The most prominent platforms are NegMAS [19] and GENIUS [12].

GENIUS is a Java-based tool that facilitates research in the area of bilateral multi-issue negotiation. It was developed comparably earlier than NegMAS, which led to abundant agent libraries [12]. GENIUS supports all kinds of bilateral negotiation mechanisms.

NegMAS is a Python-based toolkit that is currently being used to study negotiations for the supply chain management (SCM) domain . In this setting, a supply chain is a sequence of processes by which raw materials are converted into finished goods. This process is managed by multiple independent entities (agents), whose coordination is performed according to a situated negotiation. The automated negotiating agents represent sellers and buyers locating in the supply chain's upstream and downstream [20]. These two main platforms provide little support for interdependent negotiation sessions, which are required to model situations that involve concurrent negotiations and various utility functions. NegMAS sup-ports more general protocols and the negotiation is not limited to the bilateral case. Moreover, the dynamical features of NegMAS allow the manipulation of all the parameters involved in the negotiation. For this reason, we choose to develop our protocol on NegMAS.

3 The Divide-and-conquer Approach

3.1 Decomposable Alternating Offers Protocol (DAOP)

The utility space of each agent could easily become intractable with complex domains. For instance, even if each issue domain is discrete with 5 values. For a negotiation over 10 issues, there will be 5^{10} different possibilities for the negotiators to explore and evaluate the bids. Here the assumption is that our divide-and-conquer algorithm will make it easier for the negotiators to come up with good offers re-garding the efficiency and total gain utilities for all the negotiations. The approach is illustrated in figure 2.

The negotiators first negotiate over the subsets of issues. After reaching agreement over all subsets, the protocol will combine these sub-agreements into complete agreement. Note that the subsets of issues could overlap in the sense that they mutually affect the overall utility function. To implement the DAC paradigm, we propose an adaptation of SAOP and call it the decomposable alternating offers protocol (DAOP). The difference between DAOP and SAOP is that, negotiations over sub-bids are permitted in DAOP and that the stacked scheme is not required since we focus on bilateral negotiations. In SAOP, all of the negotiators around the table get a turn per round; turns are shuffled randomly. One of the negotiating parties starts the negotiation with an offer that is observed by all others immediately. Whenever an offer is made, the next party in line can take the following actions: accept the offer, reject the offer, and repeat the process [2]. The mechanism of DAOP is shown in figure 2 (bottom). DAOP, follows the same process with the exception that it is firstly applied to sub-offers before reaching a full agreement in the end.

3.2 Decomposition Mechanism

When the agents attempt to negotiate over subsets of issues, the protocol divides the complete bidding space into *m* sub-bidding space. The negotiators will then evaluate each sub-space and make decisions of accepting or declining sub-offers. The functions used for evaluating the complete bid space are of course different from the ones we use for sub-bids. To infer the utility functions for evaluating sub-bids, we decompose the original utility function. The agents originally use a

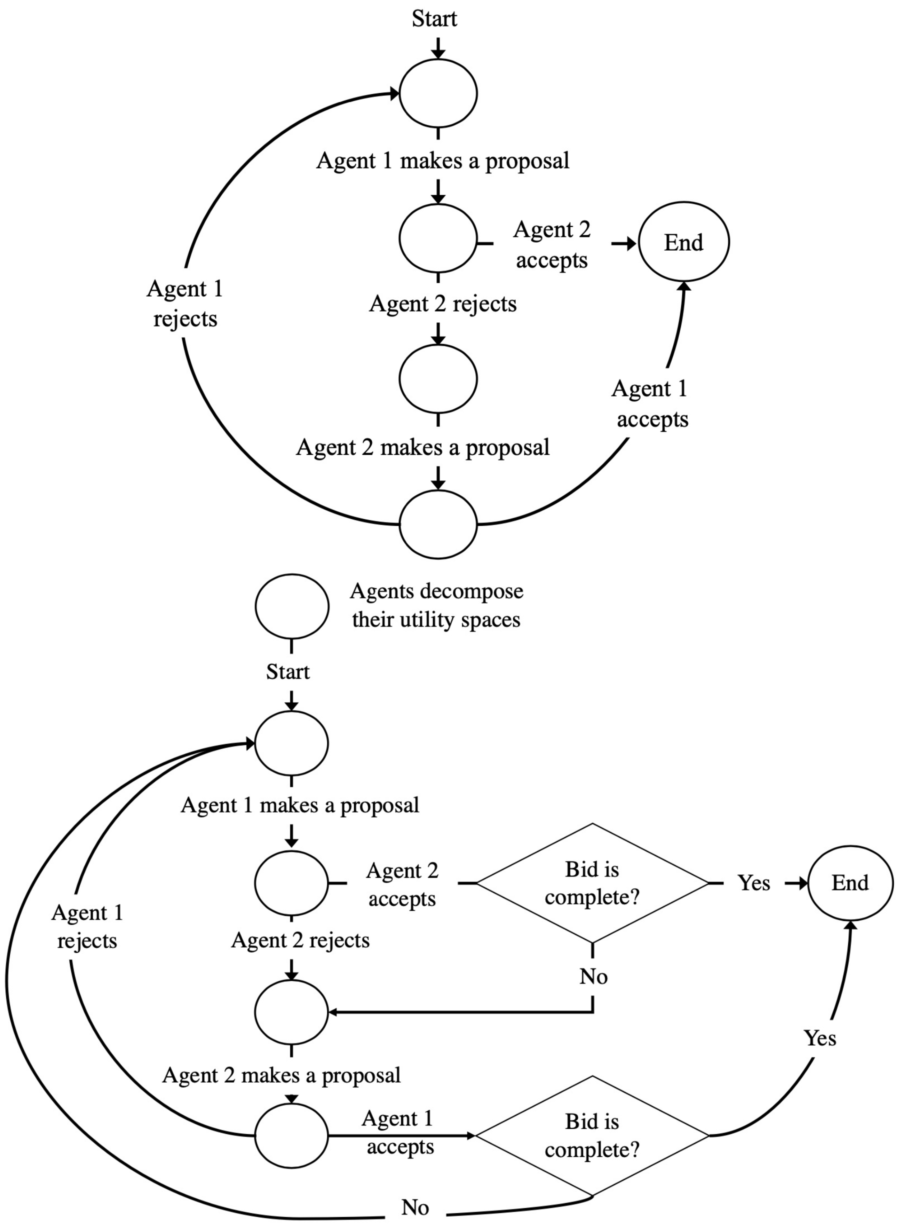

Figure 2: Implementing the divide-and-conquer approach by adapting the SAOP (top) into DAOP (bottom)

classical linear utility 1 as their preferences model to evaluate the bids.

$$U(o) = \sum_{k=1}^{n} w_k \times u_k(v_k) \tag{1}$$

Where o is an offer valued as $o = (v_1, v_2, \ldots, v_k, \ldots, v_n)$ defined as a valuation for issue $\{i_k\}_{1 \leq k \leq n}$. Each issue k takes values $v_k \in [v_k^{min}, v_k^{min}]$ weighted by $w_k \in [0,1]$.
While decomposing the linear utility space, the agents first recognize the issues they are bidding. Then form the utility functions for issues in the sub-bids into sub-utility functions. The sub-bids are evaluated by the agents using corresponding sub-utility

functions. The decomposition of issues $\{i_k\}_{1 \leq k \leq n}$ into a set $\{C_j\}_{1 \leq j \leq m}$ would imply a new representation (2).

$$U(o) = \sum_{j=1}^{m} u_{C_j}(s) \tag{2}$$

with $s \subseteq j\ o$, meaning that the sub-utility u_{Cj} is applied to a subset of issues s within partition, or cluster C_j . Clusters C_j are lumped together depending on how they affect utility U. Since, the utility (*1*) is linear additive in terms of the issues, we take the assumption that $|C_j| = c, \forall j$ with $c < m$. We take this assumption for the sake of the experimental section as it will be omitted if the shape of the utility (*1*) is unknown or nonlinear. After decomposing their utility spaces, the agents possess the ability to negotiate over sub-bids and reach sub-agreements. The protocol records the results of the sub-agreements and combines the sub-agreements into a complete agreement at the end.

4 Experiments

As an initial proof of concept, experiments were performed to prove that even without the complex process of interdependence extraction and utility decomposition, the divide and conquer technique still has its effect on the results of automated negotiation.

4.1 Experimental Settings

Basic automated negotiating agents are put into use during the experiments for proof of concept. Our simulations were developed using the NegMAS platform to implement and test DAOP. NegMAS supports many protocols and the negotiation is not limited to bilateral [19, 20]. Moreover, the dynamical feature of NegMAS allow the manipulation of all the parameters that affect the negotiation.

Agent strategy	Bidding and Accepting
Protocol	DAOP
Number of issues	Between 3 and 10
Time limit	100s
Utility functions	Linear and additive
Dividing strategy	Bids are divided into m subsets of equal sizes

Table 1: Experimental settings.

4.2 Agent Settings

In the simulations, we used two agents with the same bidding and acceptance strategies. When the protocol requests the agent to make a proposal to the opponent, the agent first generates an offer randomly from the bidding space, and then proposes it to the opponent. When responding to an offer, the agent evaluates the offer using its utility functions, then accepts the offer if the utility is satisfied, or declines if it is not. Adapted to our DAOP protocol, the agents have the ability to propose partial offers and respond to partial proposals. The offer the agent generates can be converted into a partial offer. The utility functions the agent uses for the evaluations can be decomposed with the mechanism we introduced in the previous sections when agents negotiating over subsets of issues.

4.3 Results

We start by looking at the utility and social welfare values for different numbers of issues. In figure 3, when there were only a few issues to negotiate over, the issue space was small enough for the agents to come up with the optimal offer. The agents finally agreed upon high-utilities and high-social welfare agreements. As the number of issues gradually increased, the agents started to offer sub-optimal even bad offers. This directly caused the significant drop of the utilities and social welfare.

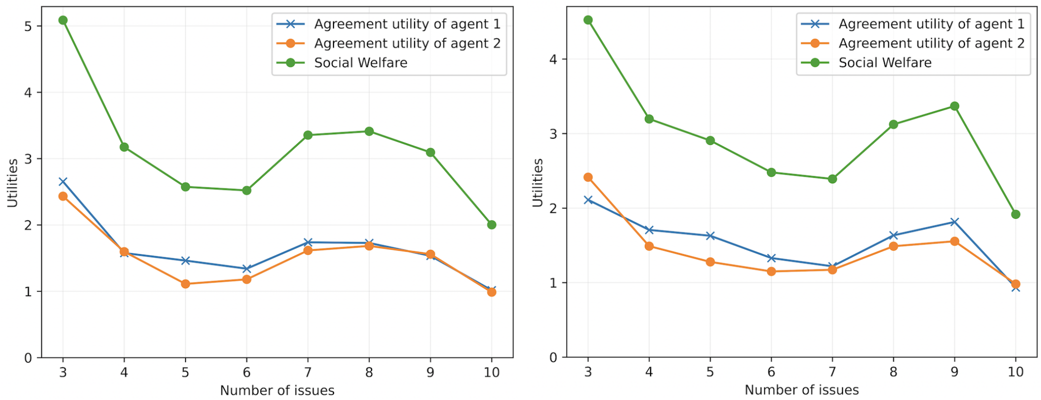

Figure 3: Utilities with normal settings

After applying the divide-and-conquer approach, we obtain figure 4. The performances became more stable and the utilities and social welfare of the agents were able to keep a comparatively high level.

Moreover, the agents using divide-and-conquer strategy were able to achieve significantly better social welfares during 100 simulations than the agents without the strategy as shown in figure 5.

4.4 Discussion

By taking a look at the result, we will know that the divide-and-conquer algorithm enhances the performances of the agents. But whether it makes the whole system more efficient still remains unknown. There are several possible improvements that can be made with respect to several aspects.

Measurement of the global efficiency. From the perspective of environment settings, it will be useful to measure the relationship of time cost and agents' performance in the meantime we measure the global performance. Thus, the experi-ment results will help us to get a better knowing about the influence the variables have on the global performance. If the divide and conquer algorithm helps with the agents' performance but dramatically

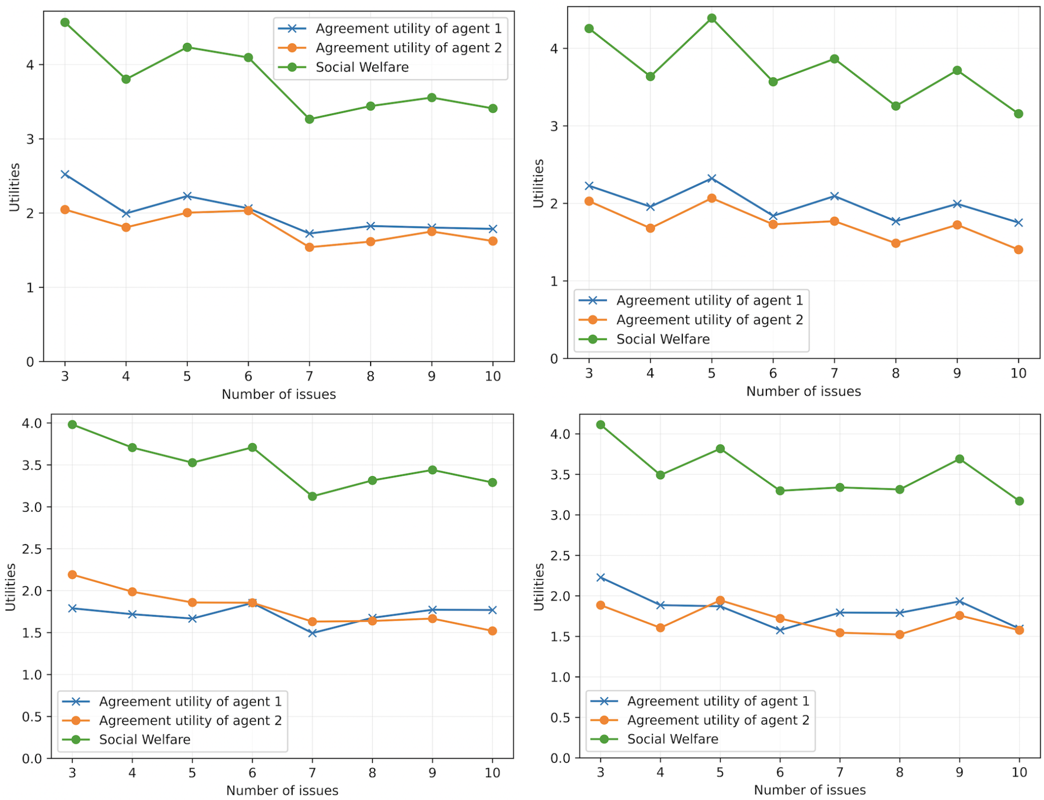

Figure 4: Applying Divide-and-conquer to Utilities

increases the time cost, then there should be some extra consideration about the concept. There are few important limitations of this work tat should highlighted.

Complexity of agents. The agents are simple. In order to prove the concept with least influence of other factors, we picked the random agents as the negotia-tors of the experiments. Apparently, it is not very appropriate to connect realistic negotiation with the current experiments. If we are going to imitate the real-world situations, not only we need a suitable environment setting for situated negotia-tion, we also need rational and more intricate entities as the components of the negotiation.

Interdependent issues. The issues are still independent. In order to test the effectiveness of divide-and-conquer algorithm on the interdependent issues based negotiation, in the next step of this research, we will add more constraints to the issue settings to make the issues interdependent.

5 Conclusion and Future Work

In this paper, we put the basics of new protocol for bilateral automated negotiation that can decompose the preferences of the agents and apply a divide-and-conquer approach. Preliminary results show that the protocol works under some assumptions of linearity and independence. Future investigations will firstly focus on us-ing different agents and then running the simulations with different domains. Our approach also sets the basis for

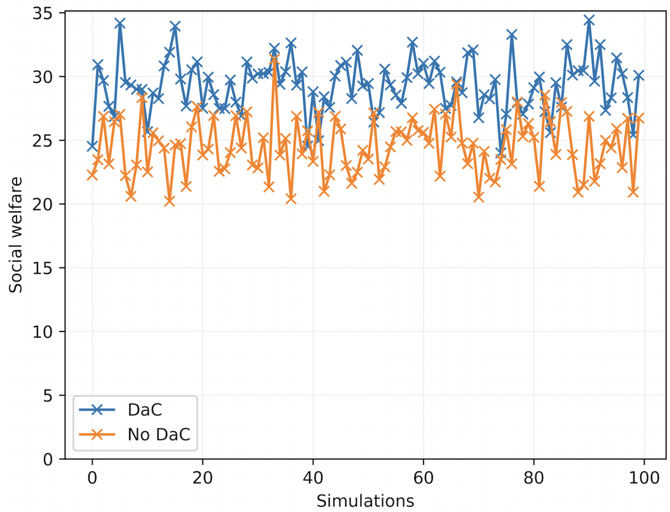

Figure 5: Social welfare among simulations

the possibility to combine our divide-and-conquer protocol with meta-negotiations, which as some related applications in many do-mains [22, 25, 28].

Acknowledgements

This research was partially supported by JSPS Kakenhi Grant Number JP20K11936, and JST CREST Grant Number JPMJCR20D1.

References

[1] Muddasser Alam, Enrico H Gerding, Alex Rogers, and Sarvapali D Ramchurn. A scalable interdependent multi-issue negotiation protocol for energy exchange. In *Twenty-Fourth International Joint Conference on Artificial Intelli-gence*, 2015.

[2] Reyhan Aydogan, David Festen, Koen Hindriks, and Catholijn Jonker. *Alter-nating Offers Protocols for Multilateral Negotiation*, volume 674, pages 153–167. 04 2017.

[3] Tim Baarslag, Koen Hindriks, Catholijn Jonker, Sarit Kraus, and Raz Lin. The first automated negotiating agents competition (anac 2010). In *New Trends in agent-based complex automated negotiations*, pages 113–135. Springer, 2012.

[4] Tim Baarslag and Michael Kaisers. The value of information in automated negotiation: A decision model for eliciting user preferences. In *Proceedings of the 16th conference on autonomous agents and multiagent systems*, pages 391–400, 2017.

[5] Maxime Bouton, Kyle Julian, Alireza Nakhaei, Kikuo Fujimura, and Mykel J Kochenderfer. Utility decomposition with deep corrections for scalable planning

under uncertainty. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 462–469, 2018.

[6] Shaheen Fatima, Michael Wooldridge, and Nicholas R. Jennings. Optimal agendas for multi-issue negotiation. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, page 129–136, New York, NY, USA, 2003. Association for Computing Ma-chinery.

[7] Katsuhide Fujita, Takayuki Ito, and Mark Klein. An approach to scalable multi-issue negotiation: Decomposing the contract space. *Computational Intelligence*, 30, 02 2014.

[8] Ali Ugur Guler, Emir Demirović, Jeffrey Chan, James Bailey, Christopher Leckie, and Peter J Stuckey. A divide and conquer algorithm for predict + optimize with non-convex problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3749–3757, 2022.

[9] Rafik Hadfi and Takayuki Ito. Addressing complexity in multi-issue negotiation via utility hypergraphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28, 2014.

[10] Rafik Hadfi and Takayuki Ito. On the complexity of utility hypergraphs. In *Recent Advances in Agent-based Complex Automated Negotiation*, pages 89–105. Springer, 2016.

[11] James Hale, Peter Kim, and Jonathan Gratch. Preference interdependencies in a multi-issue salary negotiation. In *Proceedings of the 22nd ACM International Conference on Intelligent Virtual Agents*, pages 1–8, 2022.

[12] Koen Hindriks, Catholijn M Jonker, Sarit Kraus, Raz Lin, and Dmytro Tykhonov. Genius: negotiation environment for heterogeneous agents. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 1397–1398, 2009.

[13] Takayuki Ito, Hiromitsu Hattori, and Mark Klein. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. pages 1347–1352, 01 2007.

[14] Nicholas R Jennings, Peyman Faratin, Alessio R Lomuscio, Simon Parsons, Carles Sierra, and Michael Wooldridge. Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2):199–215, 2001.

[15] Ahmed Kattan and Shaheen Fatima. Evolving optimal agendas and strategies for negotiation in dynamic environments: a surrogate based approach. In *Proceedings of the 14th annual conference companion on Genetic and evolutionary computation*, pages 1435–1436, 2012.

[16] Frank Kelly, Peter Key, Neil Walton, et al. Incentivized optimal advert assignment via utility decomposition. In *ACM Conference on Economics and Computation*. Citeseer, 2014.

[17] Desen Kirli, Benoit Couraud, Valentin Robu, Marcelo Salgado-Bravo, Sonam Norbu, Merlinda Andoni, Ioannis Antonopoulos, Matias Negrete-Pincetic, David Flynn, and Aristides Kiprakis. Smart contracts in energy systems: A systematic review of fundamental approaches and implementations. *Renew-able and Sustainable Energy Reviews*, 158:112013, 2022.

[18] Ivan Marsa-Maestre, Miguel A Lopez-Carmona, Juan A Carral, and Guillermo Ibanez. A recursive protocol for negotiating contracts under non-monotonic preference structures. *Group Decision and Negotiation*, 22(1):1–43, 2013.

[19] Yasser Mohammad, Shinji Nakadai, and Amy Greenwald. *NegMAS: A Platform for Situated Negotiations*, pages 57–75. 05 2021.

[20] Yasser Mohammad, Enrique Viqueira, Nahum Ayerza, Amy Greenwald, Shinji Nakadai, and Satoshi Morinaga. *Supply Chain Management World*, pages 153–169. 10 2019.

[21] Sriram Narayanan, Ramin Moslemi, Francesco Pittaluga, Buyu Liu, and Man-mohan Chandraker. Divide-and-conquer for lane-aware diverse trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15799–15808, 2021.

[22] Raquel Ros and Carles Sierra. A negotiation meta strategy combining trade-off and concession moves. *Autonomous Agents and Multi-Agent Systems*, 12(2):163–181, 2006.

[23] Ayan Sengupta, Yasser Mohammad, and Shinji Nakadai. An autonomous negotiating agent framework with reinforcement learning based strategies and adaptive strategy switching mechanism, 02 2021.

[24] Douglas R Smith. The design of divide and conquer algorithms. *Science of Computer Programming*, 5:37–58, 1985.

[25] Xun Tang, Ahmed Moustafa, and Takayuki Ito. The design of meta-strategy that can obtain higher negotiating efficiency. In *2018 Thirteenth International Conference on Knowledge, Information and Creativity Support Systems (KICSS)*, pages 1–6. IEEE, 2018.

[26] Gerhard Weiss. *MULTIAGENT SYSTEMS*. MIT Press, 2013.

[27] Xiaoqin Zhang and Mark Klein. Hierarchical negotiation model for complex problems with large-number of interdependent issues. volume 2, pages 126–133, 12 2012.

[28] Xiaoqin Zhang, Mark Klein, and Ivan Marsá-Maestre. Scalable complex con-tract negotiation with structured search and agenda management. *Proceedings of the National Conference on Artificial Intelligence*, 2:1507–1513, 06 2014.

[29] Yicheng Zou, Hongwei Liu, Tao Gui, Junzhe Wang, Qi Zhang, Meng Tang, Haixiang Li, and Daniel Wang. Divide and conquer: Text semantic matching with disentangled keywords and intents. *arXiv preprint arXiv:2203.02898*, 2022.