The 13th International Automated Negotiating Agent Competition Challenges and Results

Reyhan Aydoğan[1,2](✉), Tim Baarslag[3,4], Katsuhide Fujita[5,6], Holger H. Hoos[7,8,9], Catholijn M. Jonker[2,8], Yasser Mohammad[6,10], and Bram M. Renting[2,8]

[1] Özyeğin University, Computer Science, Istanbul, Turkey
reyhan.aydogan@ozyegin.edu.tr
[2] Delft University of Technology, Interactive Intelligence Group, Delft, The Netherlands
c.m.jonker@tudelft.nl
[3] Centrum Wiskunde and Informatica, Amsterdam, The Netherlands
T.Baarslag@cwi.nl
[4] Utrecht University, Utrecht, The Netherlands
[5] Tokyo University of Agriculture and Technology, Tokyo, Japan
katfuji@cc.tuat.ac.jp
[6] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
[7] RWTH Aachen University, Aachen, Germany
hh@aim.rwth-aachen.de
[8] Leiden University, Leiden, The Netherlands
B.M.Renting@liacs.leidenuniv.nl
[9] University of British Columbia, Vancouver, Canada
[10] NEC Corporation, Tokyo, Japan

Abstract. An international competition for negotiating agents has been organized for years to facilitate research in agent-based negotiation and to encourage the design of negotiating agents that can operate in various scenarios. The 13th International Automated Negotiating Agents Competition (ANAC 2022) was held in conjunction with IJCAI2022. In ANAC2022, we had two leagues: Automated Negotiation League (ANL) and Supply Chain Management League (SCML). For the ANL, the participants designed a negotiation agent that can learn from the previous bilateral negotiation sessions it was involved in. In contrast, the research challenge was to make the right decisions to maximize the overall profit in a supply chain environment, such as determining with whom and when to negotiate. This chapter describes the overview of ANL and SCML in ANAC2022, and reports the results of each league, respectively.

R. Hadfi et al. (Eds.): IJCAI 2022, SCI 1092, pp. 87–101, 2023.
https://doi.org/10.1007/978-981-99-0561-4_5

1 Introduction

Negotiation is one of the processes aiming to form alliances and find mutually acceptable solutions when stakeholders have conflicts of interest or objectives. It can be considered as a search problem in which we are looking for a decision that the majority/all stakeholders are pleased with. Research in the field of negotiation originates from various disciplines, including economics, social sciences, game theory, and artificial intelligence. The artificial intelligence community focuses on designing and developing negotiating agents that can automatically negotiate with their partners. That requires understanding the negotiation problem, reasoning on the given objectives and preferences, making strategic decisions leading to profitable consequences, and adapting behaviour based on their opponent's moves and environmental conditions such as remaining negotiation time. At this point, the International Automated Negotiating Agents Competition (ANAC) plays a vital role in developing effective negotiation strategies and providing a benchmark for the community. Consequently, the organizers of this competition aimed to encourage the design of agents that can negotiate proficiently in various circumstances and objectively assess the performance of different bargaining strategies designed by researchers worldwide. In addition, we aim to collect and make available state-of-the-art negotiating agents, negotiation domains, and preference profiles for the negotiation research community.

ANAC has studied various negotiation problems and research challenges in this field since its inception in 2010 [5]. It has focused on bilateral negotiations with reservation values and discount factors [4,5,12], large and varying domains [7,13], multilateral and non-linear settings [6], and repeated [2] negotiations. Since 2017, ANAC has had different leagues with their own challenges. In 2022, the two leagues were set up as follows:

– **Automated Negotiation League (ANL)**: Designing a negotiation agent for bilateral negotiation that can learn from previous encounters while the tournament progresses.
– **Supply Chain Management League (SCML)** [9]: Designing factory agents aiming to maximize their profit in a competitive market environment. Therefore, agents must decide with whom and when to negotiate to get the necessary sources to produce their products which will be sold to the end customers.

In negotiation, there are a variety of research challenges spanning from reasoning on incomplete information to learning about the opponent's preferences or strategies and adapting behaviour accordingly. In previous years of ANL, we introduced the challenge of learning across negotiation sessions which took much attention from our participants; however, the setting had some limitations due to the framework constraints and security concerns. The framework allowed agents to store only some structured data from their previous negotiation sessions, such as the utility distribution of their offer exchanges. However, agents may use other types of information to get a better deal. This year, agents can store any information from their previous negotiation sessions and utilize the learned knowledge in their subsequent negotiations. Regarding the SCML, the problem description and challenges were too complicated to deal with; therefore, this year, the rules were mainly simplified to streamline the challenges of maximizing profit by negotiating trades with other agents simultaneously. Consequently, agent designers could focus on one particular challenge.

The remainder of this chapter is organized as follows. Section 2 provides an overview of ANL in ANAC2022. In Sect. 3, we present the setup of SCML in ANAC2022. Section 4 discusses the results of ANL and SCML, respectively. Finally, Sect. 5 outlines our conclusions and plans for future competitions.

2 Automated Negotiation League

The Automated Negotiating Agents Competition (ANAC) originally consisted of a single challenge [3]. Since 2017, new challenges have been added, and the original competition was renamed to the Automated Negotiation League (ANL). The ANL is focused purely on the development of negotiation strategies for agents negotiating with other agents, where each year, a specific challenge is introduced by the organizers. We start by laying out the background knowledge before introducing the 2022 challenge and evaluation method. The competition results will be discussed in Sect. 4.1.

2.1 Background

In ANL, we focus on bilateral negotiations where two agents negotiate on a particular scenario to reach a consensus. Agents exchange offers by following the Alternating Offers Protocol (AOP) [1,11], where agents take turns having three possible actions: making a (counter) offer, accepting the previous offer, or walking away from the negotiation. Usually, a deadline is used to prevent agents from negotiating indefinitely, and in ANL 2022, we have set a deadline of 60 s in wall-clock time. Agents must reach an agreement before the deadline passes. Failing to reach an agreement results in a reservation utility, mostly a utility of zero, for both agents involved.

2.2 Negotiation Problem

The negotiation problem, also known as the negotiation domain, defines the set of negotiation issues and their possible values, the space Ω in which an outcome $\boldsymbol{\omega} \in \Omega$ of the negotiation must be agreed upon by the agents. Such a domain generally consists of a set of sub-problems or issues $I \in \mathcal{I}$; for instance, when negotiating over buying new computing facilities, not only the price is important, but also delivery times, hardware specs, brand, installation costs, etc. In this league, we make the simplifying assumption that all the issues are discrete. For each issue, there is thus a fixed set of values $I = \{v_1, \cdots, v_k\}$. The Cartesian product of all the issues comprises the outcome space $\Omega = I_1 \times \cdots \times I_n$ of the negotiation. The agents try to agree upon an outcome $\boldsymbol{\omega} = (\omega_1, \omega_2, \cdots, \omega_n) \in \Omega$, where n is the cardinality of the set of issues, and $\omega_i \in I_i$.

The agents have preferences over the outcome space that are considered private information. Their preferences are represented through a utility function that maps an outcome to a value $u : \Omega \rightarrow [0, 1]$, where 1 is the utility value that the agent can get in case of reaching the best possible outcome. In this league, we use linear additive utility functions, in which each issue I_i has an associated weight w_i, where $\sum_{i=1}^{n} w_i = 1$. The preference over the values within an issue is expressed through the value function $v : I \mapsto [0, 1]$. The overall utility of an outcome is calculated by the utility function given in Eq. 1.

$$u(\boldsymbol{\omega}) = \sum_{i=1}^{n} w_i \cdot v_i(\omega_i) \tag{1}$$

The negotiation domain and preferences, expressed by utility functions, are randomly generated. That is, we generated negotiation domains that have between 4–10 issues and have a size between 200–10 000. The preferences over the negotiation problem are also generated randomly. The code that produced the negotiation scenarios can be found in the public repository at[1].

2.3 Challenge

Each agent submitted to the league competed against each other agent in bilateral negotiation setups. Every opponent was encountered 50 times in succession on a randomly generated negotiation scenario. The results were averaged and sorted based on two evaluation criteria: *individual utility*, and *social welfare*. Social welfare is measured by the sum of the utilities of both agents involved in a negotiation and is a more social measure compared to individual utility.

What made the 2022 edition of ANL special is that participants were challenged to learn during the course of the tournament. All agents were provided with a storage location, where they could save any data they wanted, in order to deal more effectively with repeated encounters with the same opponents. In a real-world case, we might find ourselves negotiating with the same partners in, e.g. calendar scheduling scenarios or smart-grid energy trading. As we do, agents can exploit their previous experiences in their current negotiations to find better deals.

2.4 Method

As previously mentioned, every submitted agent competed against each other on 50 randomly created negotiation scenarios in succession. This succession is essential, as agents need to be able to learn from previous encounters with opponents. However, running every negotiation session in this tournament in succession is intractable, as a single tournament with 19 submitted agents would require 8550 negotiation sessions. Considering the deadline, this would have led to an upper bound of approximately 6 days in negotiation time. As explained later, multiple repetitions of the tournament needed to be run, which would have further increased the computational effort.

We opted to let every submitted agent negotiate against each other in parallel. A new round is started only when all previous sessions are finished, and we

[1] https://github.com/brenting/ANL-2022-example-agent.

repeat this 50 times. This ensures that a single opponent is never negotiated multiple times simultaneously and that all agents have comparable knowledge about the tournament at the start of a new round. An illustration of this procedure is given in Fig. 1.

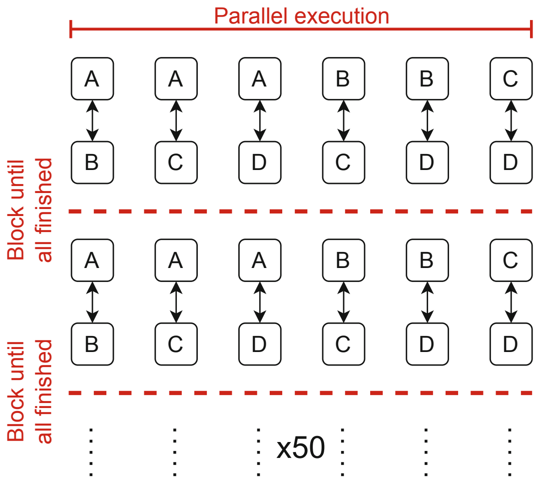

Fig. 1. Run schedule for a tournament with 4 agents: $\{A, B, C, D\}$

Negotiation scenarios were randomly generated causing a stochastic influence. Preferences can be skewed in favour of one of the agents, and the maximum obtainable social welfare can be lower for some problems. To fix the potentially skewed preferences, we repeated the tournament once more, while changing sides on the negotiation scenario by switching the utility functions. We made sure to wipe the data storage of all agents before doing so, in order to prevent unfair advantages. Furthermore, we reduced the stochastic influence by repeating the entire tournament 5 times. A total of 85,500 negotiation sessions were run to obtain the results of this competition.

2.5 Submissions

In total, there were 19 valid submissions; code and reports submitted by the participants are available online[2]. We provide an overview of the agents submitted to ANL 2022 as follows:

- Agent007, Bar Ilan University
- Agent4410, College of Management Academic Studies
- AgentFish, Tokyo University of Agriculture and Technology
- AgentFO2, Tokyo University of Agriculture and Technology

[2] https://tracinsy.ewi.tudelft.nl/pubtrac/GeniusWebThirdParties/browser/ANL2022.

– BIU_agent, Bar Ilan University
– ChargingBoul, University of Tulsa
– CompromisingAgent, Bar Ilan University
– DreamTeam109Agent, College of Management Academic Studies
– GEAAgent, College of Management Academic Studies
– LearningAgent, Bar Ilan University
– LuckyAgent2022, Babol Noshirvani University of Technology
– MiCROAgent, IIIA-CSIC
– Pinar_Agent, Siemens
– ProcrastinAgent, University of Tulsa
– RGAgent, Bar Ilan University
– SmartAgent, College of Management Academic Studies
– SuperAgent, Bar Ilan University
– ThirdAgent, College of Management Academic Studies
– Tjaronchery10Agent, College of Management Academic Studies.

3 Supply-Chain Management League

The Supply Chain Management League (SCML hereafter) has been one of the ANAC leagues since 2019. The main goal of SCML is to provide a realistic business-like environment for developing and evaluating negotiation strategies situated in a dynamic environment. The *SCM world* simulates a supply chain consisting of multiple factories that buy and sell products from one another. The factories are represented by autonomous agents that act as factory managers. Each agent decides which other agents to buy and sell from, and then negotiates with them. Their goal is to turn a profit, and the agent with the highest profit (averaged over multiple simulations) wins.

The simulation proceeds in discrete time steps, which we refer to as days. During each day, multiple simultaneous negotiations transpire, and outputs are manufactured from inputs. The game is intended to further research on agent negotiation; as such, the design emphasizes negotiation and de-emphasizes operations (e.g., scheduling). *Factories* in the SCM world convert *products* into other products by running *manufacturing processes* on their *production lines*. All processes take one day to complete. Factories store the inputs and outputs of manufacturing processes in their *inventories*, and their funds in their *accounts*.

Each factory has multiple production lines, each of which is assigned a profile specifying the cost at which it can execute the various manufacturing processes. In general, these costs can vary from factory to factory and may vary from line to line. Prior to SCML 2022, however, each factory had a set of identical production lines, each of which can run only a single manufacturing process. Factory costs are private information: i.e., no factory manager knows the cost of any other factory.

The production graph is assumed to be directed and acyclic, with products and manufacturing processes as its nodes. An edge from a product to a process node indicates that this product is an *input* to this process. An edge from a process to a product node indicates that this product is an *output* of this process. (Note that there are no edges between product nodes or between process nodes.) Figure 2 depicts a sample production graph for the game.

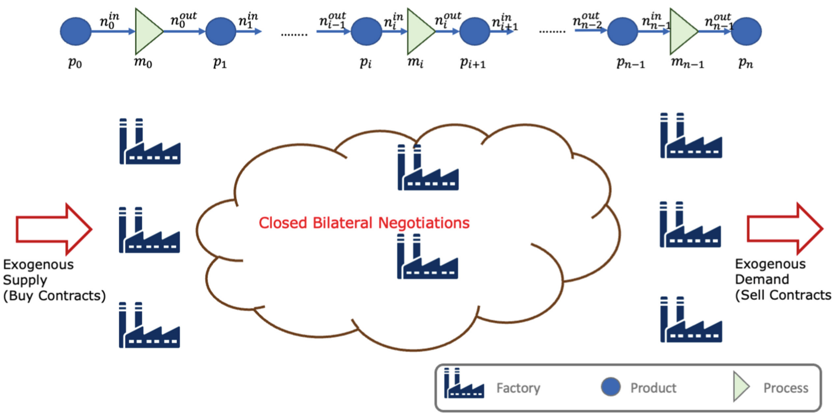

Fig. 2. Example of a world in SCML showing the production graph (top). Demand and supply are controlled through exogenous contracts while all trade in non-terminal products is conducted through closed concurrent bilateral negotiations

The agents in the SCM world function as *factory managers*. In addition to managing production, they negotiate with other agents to reach agreements to buy and sell products. Such agreements are generated via bilateral negotiations using the *alternating offers protocol* typically used in ANAC competitions [8,9]. Each offer specifies a buyer, a seller, a product, a quantity, a delivery time, and a unit price. The sequences of offers and counteroffers in a negotiation are private to the negotiating parties.

The SCM world does not endow agents with arbitrary utility functions[3]. On the contrary, all utility functions are endogenous, meaning they are engendered by the simulation dynamics and agents' interactions with other agents. Endogenous utility functions are a distinguishing feature of SCML. It is an agent's job to assign utilities to potential contracts, given its unique production capabilities, and then to negotiate with other agents to secure those which are most favorable to them.

Agents consuming raw materials are endowed with exogenous buy contracts but no exogenous sell contracts, while agents producing the finished product will be endowed with exogenous sell contracts but no exogenous buy contracts. No other agents will be endowed with any exogenous contracts. By design, *no agent can turn a profit without negotiating successfully*, since no agent is endowed with both exogenous buy and exogenous sell contracts. SCML had three tracks with gradually increasing difficulty.

[3] See the three tracks later for more information about this point.

Each simulation of the SCM world runs for multiple (say, 1000) days. Before the first day, each agent is assigned a *private* manufacturing profile. In addition, the bulletin board is populated with the production graph information and catalog prices, an initial balance is deposited into each agent's account, and agents are endowed with exogenous contracts. Then, during each day:

1. Agents can engage in multiple (say, 100) rounds of negotiations with their negotiating partners. They can also read the bulletin board, and request negotiations with other agents (for the next day).
2. All contracts that have come due are executed: i.e., products are moved from the seller's inventory to the buyer's, and money is moved from the buyer's account to the seller's.
3. The manufacturing processes on all lines in all factories are run: i.e., inputs are removed from inventory, outputs are stored in inventory, and production costs are subtracted from the factories' accounts.

3.1 The OneShot Track

The simplest track was the OneShot track, which focused on *concurrent negotiation* with a well-defined utility function that is given to the participants. In this track, the production graph had exactly three products (i.e. two manufacturing processes). Agents could either be on the buying end or the selling end of all their negotiations, but not both[4]. All products in OneShot are perishable (i.e., no inventory is carried over to the next day), and agents start with an initial endowment of money that guarantees that they can never go bankrupt. Exogenous contracts are revealed day by day. Moreover, all buyer agents negotiate with all seller agents every day, and all agreements reached are immediately binding (i.e., no separate signing step) and agents can—in principle—get all their needs from a single supplier (or sell all their products to a single consumer). Taken together, these features guarantee that the profit for a day is completely independent of what happened in the past or what will happen in the future given the agreements of that day and makes the utility function (profit) easy to define.

Utility Function. Agent a's utility u_a can now be defined as a's profits, i.e., its revenue less its costs and penalties:

$$u_a(C_a^{\text{in}}, C_a^{\text{out}}) = \underbrace{\sum_{c \in C_a^{*\text{out}}} p_c q_c^{*\text{out}}}_{\text{revenue}} - \underbrace{\sum_{c \in C_a^{\text{in}}} p_c q_c^{\text{in}} - m_a\, Q^{*\text{out}}}_{\text{costs}} - \underbrace{\left(\alpha_a \operatorname{tp}(\rho_a^{\text{in}}, d)\, Q_a^{\text{excess}} + \beta_a \operatorname{tp}(\rho_a^{\text{out}}, d)\, Q_a^{\text{shortfall}}\right)}_{\text{total penalties}}, \tag{2}$$

[4] Each submitted strategy was played in both roles.

where ρ_a^{in} and ρ_a^{out} are factory a's input and output products, respectively, and $\text{tp}(\rho, d)$ is the **trading price** of product ρ on day d which is a weighted average of previous actual trading prices of the product.

$\sum_{c \in C_a^{*\text{out}}} p_c q_c^{*\text{out}}$ The total **revenue** it earns by selling its outputs.
$\sum_{c \in C_a^{\text{in}}} p_c q_c^{\text{in}}$ The total **cost** it incurs to buy its inputs.
$m_a \, Q_a^{*\text{out}}$ The production **cost**. Note that factories produce exactly what they can sell on the current day, as inventory does not carry over from one day to the next.
$\alpha_a \, \text{tp}(\rho_a^{\text{in}}, d) \, Q_a^{\text{excess}}$ The total buy-side **penalty**, which is incurred on any output products that are not sold. Note that these penalties depend on the trading price of the input product.
$\beta_a \, \text{tp}(\rho_a^{\text{out}}, d) \, Q_a^{\text{shortfall}}$ The total sell-side **penalty** incurred by the factory for failing to deliver its output product. Note that these penalties depend on the trading price of the output product.

3.2 The Standard Track

In the standard track, products do not perish (i.e., the inventory is carried to the next day), exogenous contracts are revealed days before their due, and negotiators can reach agreements about deliveries any day in the future. Agreements are only binding once signed at the end of each day. Agents can commit breaches and may go bankrupt.

There are several challenges here. Firstly, the agent needs to take into account not only the current set of concurrently running negotiations but also future negotiations. Secondly, the agent negotiates with its suppliers and consumers concurrently. Thirdly, exogenous contracts are not forced, and agreements are not binding until they are *signed* by the end of the day. This entails that an agent cannot be sure that an agreement it has will actually be signed and must model the signing probability of its agreements. Fourthly, the agent directly controls production and must decide what to produce. Fifthly, agents may go bankrupt, affecting other agents with whom they signed contracts (like in the real world), and this actually happens frequently enough that agents should not ignore it. Finally, it is not possible to know the profit of a set of contracts on the day they are signed (as with the OneShot track), which means that each agent has to define its own – uncertain – utility function.

Breach Processing: When a contract comes due, the simulator tries to execute it (i.e., move products from the seller's inventory to the buyer's, and move money from the buyer's account to the seller's). If this execution fails, either because of insufficient funds on the part of the buyer, or insufficient products on the part of the seller, a breach of contract occurs. In both cases, the contract is executed to the extent possible, and the agent in breach of contract is penalized and reported to the breach list.

Bankruptcy Processing: If an agent is unable to meet its financial obligations, it is declared bankrupt. The assets of bankrupt agents are liquidated, and their factories are closed (no further production can transpire). They can no longer participate in negotiations. The simulator takes over their outstanding contracts and fulfils them to the extent possible.

The Spot Market: exists so that agents who would otherwise be in breach of contract for insufficient products (funds) can instead buy (sell) as necessary on the spot market at buy (sell) prices, which are always above (below) *trading prices*—an average over the historic prices at which products are traded.

3.3 The Collusion Track

In the Standard and OneShot tracks, at most one instance of each team's agent runs in each simulation, together with an unknown mix of agents prepared by other participants and agents prepared by the organizing committee. In the *collusion* track, multiple instances of the same team's agent run during a single simulation. In this track, it is perfectly legal for instances of the same agent to collude with one another to try to corner the market or exhibit other collusive behaviours. This is the main challenge added by the collusion track.

3.4 Competition Mechanics

The competition was conducted in two main phases: (1) an online phase, in which agents could be submitted to https://scml.cs.brown.edu and were automatically checked for being runnable within the simulation environment then entered into tournaments with other submitted agents with a leaderboard that is kept up to near the submission deadline for the official competition. (2) the official competition phase, which is further divided into a qualification and finals. Only top-performing agents in the qualifications were allowed to run in the finals.

For both rounds of the competition, we applied a factorial t-test (i.e., t-tests between every pair of agents) with Bonferroni's multiple comparison correction and considered two agents to have different ranks only if the differences between their scores were statistically significant.

We received a total of 76 registrations for the online competition, of which only 25 agents were submitted to the official competition; of these, 13 were selected as finalists in the three tracks (OneShot: 8, Standard: 3, Collusion: 2). Moreover, the winning agents from last year with no updated strategy this year were automatically entered into the official competition (OneShot: 2, Standard: 1). Because we only had 2 submissions in the Collusion track, both agents qualified automatically for the finals.

4 Competition Results

In this section, we first report the results of the ANL (Sect. 4.1), then present the results of the SCML (Sect. 4.2).

4.1 Automated Negotiation League Results

We present the winners of ANL 2022 in Table 1. Full results of the tournament can be found in Table 2. Surprisingly, "DreamTeam109Agent" won in both the individual utility and the social welfare categories. Optimising for individual utility generally hurts social welfare [10], but it seems that this did not prevent the agent from obtaining the highest score in both categories. The first and second places in individual utilities are close, but there is quite a significant gap between the second and third places. The difference in social welfare score between "Agent007" and "CompromisingAgent" was almost negligibly in favour of "Agent007", resulting in a close third position of the latter in terms of social welfare. Lastly, "LuckyAgent2022" had a bug in its learning mechanism, which caused it to obtain a low ranking.

Table 1. Winners of the Automated Negotiation League (ANL)

Rank	Individual utility	Social welfare
1st	DreamTeam109Agent	DreamTeam109Agent
2nd	ChargingBoul	Agent007

4.2 Supply-Chain Negotiation Results

Table 3 shows the results of the OneShot track. All qualified agents could outperform the top agents from SCML 2021, showing progress in solving the challenge. Moreover, all qualified agents except AdamAGent and EveAgent could achieve some profit (i.e. a score higher than $10,000$), with the top strategy (PatientAgent) achieving 12.2% profit[5]. The OneShot track was run twice (once with the winners from SCML2021 and once without them). Table 3 shows that agent ranks did not change in the two runs. Moreover, removing the weakest agents did improve the profit of all other agents in the environment. The winner of this track was PatientAgent from Brown University. The three winner agents could achieve positive profits in both runs of the finals.

Table 4 shows the results of the Standard track. All newly qualified agents could outperform the second-place agent from last year (ArtisanKangaroo)[6]. Only the winner of the qualifications round (Lobster) could make a profit in this more challenging environment, and it only made 0.72%. No agents in the standard track could generate positive profits suggesting, that the challenge is still too hard for the methods explored so far. The winner agent for this track was Lobster, from Nagoya Institute of Technology, Japan that also won the qualifications.

[5] The simulations were designed so that a beneficial dictator optimizing agreements can achieve a 15% profit.

[6] The top agent from SCML2021 was modified and resubmitted to SCML2022 as M5.

Table 2. Results from the Automated Negotiation League (ANL), where **bold** is best and <u>underline</u> is worst. The measures are averaged over all the negotiation sessions that agents participated in.

Agent	Individual utility	Opponent utility	Nash product	Social welfare	Number of offers	Agreement ratio
DreamTeam109Agent	**0.725**	0.736	0.541	**1.460**	**6459**	0.943
ChargingBoul	0.724	0.670	0.551	1.393	5705	0.866
SuperAgent	0.704	0.576	0.505	1.280	6173	0.791
CompromisingAgent	0.686	0.771	0.550	1.456	1233	0.925
RGAgent	0.682	0.713	**0.567**	1.395	1209	0.850
LearningAgent	0.668	0.535	0.481	1.203	1272	0.733
Agent007	0.642	0.814	0.527	1.456	2554	0.956
AgentFO2	0.641	0.727	0.543	1.367	3244	0.851
ProcrastinAgent	0.640	0.376	0.357	1.016	5302	0.665
MiCROAgent	0.627	0.502	0.453	1.130	6144	0.696
Pinar_Agent	0.618	0.517	0.456	1.136	5303	0.702
BIU_agent	0.608	0.526	0.463	1.133	1422	0.685
ThirdAgent	0.591	0.721	0.508	1.313	963	0.833
Agent4410	0.581	0.818	0.514	1.399	1190	0.907
Tjaronchery10Agent	0.578	0.509	0.418	1.087	848	0.682
GEAAgent	0.576	0.727	0.505	1.303	<u>45</u>	0.826
AgentFish	0.569	**0.871**	0.507	1.440	2380	**0.957**
SmartAgent	0.553	0.356	0.339	0.909	1442	0.574
LuckyAgent2022	<u>0.301</u>	<u>0.250</u>	<u>0.221</u>	<u>0.551</u>	1310	<u>0.338</u>

In the collusion track, we only had two agents (CharliesAgent and M5) so a single round was conducted. The M5 agent from Tokyo University of Agriculture and Technology, Japan was able to achieve 0.7% extra profit due to its collusion strategy (i.e., over what it could achieve when collusion was turned off), winning it an honourable mention.

Taken together, these results suggest that the strategies submitted to SCML 2022 were an improvement over the ones submitted to SCML2021 in all tracks. The more straightforward challenge of the OneShot track was almost met, with a maximum profit of around 12% of a theoretical expectation of no more than 15%, with still some room for improvement. The Standard and Collusion challenges are still proving too hard, with hardly any profit being made even by top agents. This highlights the difficulty in translating advances in automated negotiation research to the complexity of the real world.

Table 3. Results for the SCML OneShot track ordered by the score in the finals. Agents with statistically insignificant score differences according to the factorial t-test are given the same rank. The finals round shows two scores for the two runs with and without SCML2021's winners.

Agent	Qualifications			Finals					
	Rank	Instances	Score	Rank		Instances		Score	
				First	Second	First	Second	First	Second
PatientAgent	1	1200	11,209	1	1	5,000	3,500	11,430	10,991
NewGentle	2	1200	10,320	2	2	5,000	3,500	10,625	10,463
AgentSAS	2	1200	10,400	3	3	5,000	3,500	10,593	10,399
AgentNeko	2	1200	10,611	4	4	5,000	3,500	10,403	10,115
EVEAgent	7	1200	9,902	5	4	5,000	3,500	10,328	9,968
LearningAdaptive	2	1200	10,381	6	6	5,000	3,500	10,131	9,608
AgentRM	2	1200	10,215	7	7	5,000	3,500	9,705	9,302
AdamAgent	8	1200	9,658	7	8	5,000	3,500	9,620	8,555
Agent112*	9	1200	9,473	9		5,000	3,500	8740	
Agent74*	10	1200	9,353	10		5,000	3,500	8317	
AdaptivePercentile	11	1200	9,036						
UCOneshot	12	1200	8,273						
AdaptiveQIAgent	12	1200	8,092						
MMMPersonalized	14	1200	7,517						
Agent125	14	1200	7,347						

Table 4. Results for the SCML Standard track ordered by the score in the finals. Agents with statistically insignificant score differences according to the factorial t-test are given the same rank.

Agent	Qualifications			Finals		
	Rank	Instances	Score	Rank	Instances	Score
Lobster	1	5,000	72	1	1,000	−96
M5	2	5,000	−264	2	1,000	−155
ArtisanKangaroo*	4	5,000	−386	3	1,000	−186
CharliesAgent	3	5,000	−354	4	1,000	−358
SkyAgent	5	5,000	−851	5	1,000	−1,041
SmartAgent	6	5,000	−991			
SalesAgent	6	5,000	−1,080			

5 Conclusion and Discussion

This chapter describes the 13th annual ANAC, held in 2022 and reports the results of the completion. The competition comprised two leagues where ANL focused on incorporating learning from past negotiations and SCML focused on strategic decision-making on whom to negotiate with and how to negotiate to maximize the overall profit in a supply chain environment.

The chapter makes the complete setup of ANAC 2022 available to the broader negotiation research community. We hope that the addressed challenges in both leagues will drag the attention of more participants and they can get benefit from insights given by the winning agents. We plan to organize the next competition in conjunction with AAMAS 2023, in London.

Acknowledgments. ANAC2022 was supported by sponsors (NEC-AIST AI Cooperative Research Laboratory, BIRD INITIATIVE, TU Delft, and CWI). This research was (partly) funded by the Vidi project COMBINE (VI.Vidi.203.044) and the Hybrid Intelligence Center(024.004.022), programmes funded through the Netherlands Organisation for Scientific Research, and by EU H2020 projects "Humane AI Net" under contract # 952026 and TAILOR under GA No 952215.

References

1. Aydoğan, R., Festen, D., Hindriks, K.V., Jonker, C.M.: Alternating offers protocols for multilateral negotiation. In: Fujita, K., et al. (eds.) Modern Approaches to Agent-based Complex Automated Negotiation. SCI, vol. 674, pp. 153–167. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51563-2_10
2. Aydoğan, R., Fujita, K., Baarslag, T., Jonker, C.M., Ito, T.: ANAC 2018: repeated multilateral negotiation league. In: The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, Japan (2019)
3. Baarslag, T., Aydoğan, R., Hindriks, K.V., Fuijita, K., Ito, T., Jonker, C.M.: The automated negotiating agents competition, 2010–2015. AI Mag. **36**(4), 115–118 (2015). http://www.aaai.org/ojs/index.php/aimagazine/article/view/2609
4. Baarslag, T., et al.: Evaluating practical negotiating agents: results and analysis of the 2011 international competition. Artif. Intell. **198**, 73–103 (2013). https://doi.org/10.1016/j.artint.2012.09.004
5. Baarslag, T., Hindriks, K., Jonker, C.M., Kraus, S., Lin, R.: The first automated negotiating agents competition (ANAC 2010). In: Ito, T., Zhang, M., Robu, V., Fatima, S., Matsuo, T. (eds.) New Trends in Agent-based Complex Automated Negotiations, Series of Studies in Computational Intelligence, pp. 113–135. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-24696-8_7
6. Fujita, K., Aydoğan, R., Baarslag, T., Hindriks, K., Ito, T., Jonker, C.: The sixth automated negotiating agents competition (ANAC 2015). In: Fujita, K., et al. (eds.) Modern Approaches to Agent-based Complex Automated Negotiation. SCI, vol. 674, pp. 139–151. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-51563-2_9
7. Fujita, K., Aydoğan, R., Baarslag, T., Ito, T., Jonker, C.: The fifth automated negotiating agents competition (ANAC 2014). In: Fukuta, N., Ito, T., Zhang, M., Fujita, K., Robu, V. (eds.) Recent Advances in Agent-based Complex Automated

Negotiation. SCI, vol. 638, pp. 211–224. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30307-9_13

8. Jonker, C.M., Aydoğan, R., Baarslag, T., Fujita, K., Ito, T., Hindriks, K.: Automated negotiating agents competition (ANAC). In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-2017), pp. 5070–5072. AAAI Press (2017)
9. Mohammad, Y., Viqueira, E.A., Ayerza, N.A., Greenwald, A., Nakadai, S., Morinaga, S.: Supply chain management world. In: Baldoni, M., Dastani, M., Liao, B., Sakurai, Y., Zalila Wenkstern, R. (eds.) PRIMA 2019. LNCS (LNAI), vol. 11873, pp. 153–169. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33792-6_10
10. Nowak, M.A.: Five rules for the evolution of cooperation. Science **314**(5805), 1560–1563 (2006). https://doi.org/10.1126/science.1133755
11. Rubinstein, A.: Perfect equilibrium in a bargaining model. Econometrica **50**(1), 97–109 (1982). http://www.jstor.org/stable/1912531
12. Williams, C.R., Robu, V., Gerding, E.H., Jennings, N.R.: An overview of the results and insights from the third automated negotiating agents competition (ANAC2012). In: Marsa-Maestre, I., Lopez-Carmona, M.A., Ito, T., Zhang, M., Bai, Q., Fujita, K. (eds.) Novel Insights in Agent-based Complex Automated Negotiation. SCI, vol. 535, pp. 151–162. Springer, Tokyo (2014). https://doi.org/10.1007/978-4-431-54758-7_9
13. (Ya'akov) Gal, K., Ilany, L.: The fourth automated negotiation competition. In: Fujita, K., Ito, T., Zhang, M., Robu, V. (eds.) Next Frontier in Agent-based Complex Automated Negotiation. SCI, vol. 596, pp. 129–136. Springer, Tokyo (2015). https://doi.org/10.1007/978-4-431-55525-4_8