CARC2025: An Agent for Solving Sequential Multi-deal Negotiations

Tianzi Ma[1], **Ruoke Wang**[1], **Hongji Xiong**[1], **Yulin Wu**[1] and **Xuan Wang**[1]
[1]Harbin Institute of Technology, Shenzhen
mtz982437365@gmail.com, 24S151159@stu.hit.edu.cn, xionghj@stu.hit.edu.cn,
yulinwu@cs.hitsz.edu.cn, wangxuan@cs.hitsz.edu.cn

Abstract

We propose CARC2025, a negotiation agent designed for one-to-many sequential multi-deal negotiations. As the central agent, it maximizes global utility across sub-negotiations by either decomposing the global problem into local optimizations (when the utility function is independent across sub-negotiations) or evaluating globally compatible joint bids when dependencies exist. As an edge agent, it employs time-based concession to balance utility and agreement likelihood. All agents adopt a dynamic concession strategy that adjusts both with time and the agent's position in the negotiation sequence. An opponent modeling component further refines proposals by estimating the opponent's preferences and utility trends, improving agreement rates and overall performance.

1 Introduction

The Automated Negotiating Agent Competition (ANAC) Automated Negotiation League (ANL) focus on developing automated negotiation agents and strategies for diverse scenarios. The 2025 challenge focuses on sequential multi-deal negotiation, where a central agent engages in a sequence of bilateral negotiations with multiple opponents [Baarslag, 2024]. The agent's performance depends on the combined outcomes of all agreements. This setting extends prior work by emphasizing complex one-to-many interactions through sequential bilateral deals. The ANL will use the NegMAS platforms, a Python-based framework for automated negotiation research [Mohammad *et al.*, 2021; Lin *et al.*, 2014].

This scenario introduces several significant challenges:

- The need to coordinate offers and responses across multiple sub-negotiations, ensuring coherent and mutually beneficial agreements.
- Limited information about opponents, as the agent does not have prior knowledge of their preferences or strategies, requiring adaptive and robust negotiation strategy.

Our Method. We propose CARC2025, a negotiation agent designed to maximize global utility across sequential sub-negotiations. As the central agent, CARC2025 dynamically adapts its strategy based on issue dependencies. When utility depends on interdependent issues, we construct a joint bid space consistent with past agreements to estimate the global utilities for each candidate bid. These estimates guide bid selection toward globally beneficial outcomes.

For edge agent, which optimize utility in a single sub-negotiation, CARC2025 uses time-based concession strategy: starting with a strict threshold and easing it as negotiation proceeds.

For both roles, agents use a dynamic concession strategy that accounts for negotiation time and position in the sequence. An opponent modelling module is also employed to predict preferences, allowing for adaptive bidding.

Paper Organization. The core ideas of sequential one-to-many negotiation are introduced in Section 2. Section 3 presents the detailed strategy design for the center agent. Section 4 describes the bidding and acceptance strategies for edge agent. Experimental results and evaluations are reported in Section 5. Finally, Section 6 concludes the paper.

2 Preliminary

2.1 Notations

Sequential multi-deal negotiation. In a one-to-many sequential negotiation with a fixed order [Aydoğan *et al.*, 2017], there exists a central agent, denoted as C, and N edge agents, denoted as $E_1, E_2, \ldots, E_N$. The central agent C engages in a sequence of N sub-negotiations, each conducted independently with an edge agent E_i in the fixed order. This setting is particularly relevant in scenarios where a central agent needs to coordinate multiple partial agreements across independent bilateral tracks [Florijn, 2024].

Each sub-negotiation involves a set of m issues, denoted as $I_1, I_2, \ldots, I_m$. Every issue I_j has a discrete domain of k possible values, denoted as $v_1, v_2, \ldots, v_k$.

The joint bid of the center agent across all N sub-negotiations is denoted as $O = (o_1, o_2, \ldots, o_N)$, where each $o_i = (v_1^{(i)}, v_2^{(i)}, \ldots, v_m^{(i)})$ is a tuple representing the agreed-upon outcome in the i-th sub-negotiation, with $v_j^{(i)}$ being the selected value for issue I_j. Here, o_i denotes a finalized agreement between C and E_i, while b_i denotes a tentative bid that is currently being proposed or responded to and has not yet become a final outcome.

Finally, the central agent maps the joint outcome O to a real-valued utility score using its utility function $U_C(O)$. The edge agent i, with the sub-negotiation result o_i, receives a utility score denoted as $U_{E_i}(o_i)$, where U_{E_i} is the utility function of agent E_i that maps the agreed bid o_i to a real value utility.

Outcome space. For edge agents, they only participate in a single sub-negotiation, each involving m issues with k discrete values per issue. Therefore, their outcome space is of size $k^m + 1$ (an additional special outcome representing no agreement). For the central agent, the joint outcome space is of size $(k^m + 1)^N$.

2.2 Handling Sequential One-to-Many Negotiations with a Predefined Order

Dynamic Strategy Evolution. As the sequential negotiation progresses, the strategic positions of the center and edge agents evolve. In the earlier sub-negotiations, the center agent can afford to be more selective, as there are still many remaining opportunities to reach favorable agreements. This gives the center agent room to adopt a stricter stance, while the edge agents are expected to make greater concessions in order to secure early deals.

In contrast, during the later stages of the sequence, the center agent faces increasing pressure to finalize agreements, especially if previous sub-negotiations do not have sufficient outcomes. Consequently, the center agent becomes more flexible and willing to concede, while edge agents may tighten their demands.

Algorithm 1 Preference Bids List in Sub-negotiation

Require: Current sub-negotiation index i, prefix agreements $(o_1, \dots, o_{i-1})$, global outcome space $\mathcal{O}$, utility function U_C, tolerance ε

Ensure: Preference Bids List at index i: `pref_list`

// Select the candidates
Initialize `util_dict` $\leftarrow \{\}$
for all $o \in \mathcal{O}$ **do**
  if $o_{1:i-1} == (o_1, \dots, o_{i-1})$ **then**
    `util_dict`$[o] \leftarrow U_C(o)$
  end if
end for
`max_util` $\leftarrow \max($`util_dict.values`$)$
`candidates` $\leftarrow \{o \mid U_C(o) \geq$ `max_util` $- \varepsilon\}$
// Sort the candidates
for all $o \in$ `candidates` **do**
  // Given that $o = (o_1, \dots, o_{i-1}, o_i, o_{i+1}, \dots, o_N)$
  $o^{\texttt{pessimistic}} \leftarrow (o_1, \dots, o_{i-1}, o_i, \texttt{None}, \dots, \texttt{None})$
  `score`$[o] \leftarrow U_C(o^{\texttt{pessimistic}})$
end for
Sort `candidates` in descending order of `score`
`pref_list` $\leftarrow$ deduplicated o_i from `candidates`
return `pref_list`

Preference bids in sub-negotiation. As the center agent, before each sub-negotiation, we construct the set of all joint outcomes compatible with the current agreement prefix. To manage this efficiently, we maintain a dictionary `util_dict` that stores the global utility $U_C(o)$ for each such outcome o.

To reduce overhead and focus on promising outcomes, we filter `util_dict` to retain only those within a predefined tolerance of the optimal utility.

We then rank the remaining candidates using a conservative estimate of their minimum attainable utility, assuming the current sub-negotiation succeeds while all future ones fail. This prioritizes both global potential and local robustness.

Finally, we derive a preference list over offers in the current sub-negotiation from the sorted candidates to guide bidding and concession. The full procedure is shown in Algorithm 1.

Given a known utility function, if sub-negotiation outcomes are independent, the center agent can simplify bid generation by focusing solely on the current sub-negotiation (size $k^m + 1$), instead of the full space (size $(k^m + 1)^N$). This decomposition reduces computational complexity, especially in large one-to-many negotiation domains where the outcome space grows exponentially [Koça *et al.*, 2024].

3 Implementation of the Center Agent

3.1 Bidding Strategy

The bidding strategy of the center agent dynamically adjusts based on the current negotiation time and the utility value of candidate bids. It mainly consists of the following two stages:

Early Stage (Relative Time $t \leq 0.9$). During the early phase of negotiation, the agent prioritizes bids that are optimal or near-optimal. Specifically, candidate bids must have utilities at least 95% of the best utility. The agent applies a Softmax-like exponential weighting to the candidate bids, assigning higher probabilities to those with greater utility. This approach also prevents overly predictable bidding that could be exploited by opponents.

Late Stage (Relative Time $t > 0.9$). As negotiation approaches the final stage, the center agent gradually concedes, adjusting bids to increase the likelihood of reaching an agreement. The detailed approach is as follows:

- **Opponent Preference Prediction.** The agent predicts the opponent's utility for each candidate bid b_i using an opponent model $U_{\text{opp}}(b_i)$. For details of opponent modelling, refer to Section 3.3.
- **Composite Scoring of Candidate Bids.** Each candidate bid b_i is assigned a composite score that balances the agent's own utility and the opponent's estimated utility:

$$\text{score}(b_i) = \alpha \times U_C(o) + (1 - \alpha) \times U_{\text{opp}}(b_i)$$

  Here, $o = (o_1, o_2, \dots, o_{i-1}, b_i, \ None, \dots, \ None)$ represents the partially constructed joint bid, where $o_1, \dots, o_{i-1}$ are agreements already reached, b_i is the current candidate bid, and the $None$ denotes undertermined bids for future sub-negotiations.

  The concession weight α is a time-dependent parameter, defined by the exponential decay function $\alpha = e^{-2t}$, where t represents the relative negotiation time.

  As time progresses, α decreases, gradually shifting the agent's focus from maximizing its own utility toward accommodating the opponent's preferences.

- **Bid Selection Strategy.** The agent selects the final bid b^* by performing a weighted random sampling over the candidate bids according to their composite scores.

$$b^* \sim P(b_i) \propto \text{score}(b_i)$$

This concession strategy based on opponent-modelling better balances the interests of both parties. This approach steers bids toward mutually acceptable agreements, preventing concessions that provide no benefit to the opponent, and thereby increasing the possibility of reaching a deal.

3.2 Acceptance Strategy

The center agent's response to an incoming offer depends primarily on the current negotiation time and the offer's utility relative to the agent's preferences.

Early Stage (Relative Time $t \leq 0.9$). At the early stage, the agent only accepts offers that provide a high utility—specifically, those that reach at least 95% of the maximum utility expected in the current sub-negotiation.

Late Stage Response (Relative Time $t > 0.9$). In the final stage of negotiation, the center agent adopts an acceptance strategy that allows flexible concessions to avoid deadlock. The decision logic is outlined as follows:

- **Immediate Acceptance from Preference List.** If the received offer appears in the center agent's `pref_list`, it is accepted immediately.
- **Dynamic Utility Threshold.** If the offer is not on the preference list, the agent computes its utility and compares it against a dynamically adjusted threshold. This threshold is calculated as follows:

$$U_{\text{min}} = 0.6 \times U_{\text{max}} \times \left(1 - 0.5 \times \frac{i}{N}\right)$$

$$\text{threshold}(t) = U_{\text{min}} + (U_{\text{max}} - U_{\text{min}}) \cdot (1 - (t')^{0.4})$$

  where:
  - $t' = \frac{t-0.9}{1-0.9} \in [0,1]$ is the re-normalized time,
  - U_{max} is the maximum utility of bids in `pref_list`,
  - i is the index of the current sub-negotiation,
  - N is the total number of sub-negotiations, $\frac{i}{N} \in [0,1]$ represents the relative progress in the overall negotiation sequence.

  As time progresses, the threshold gradually decreases, allowing acceptance of lower-utility offers to increase the chance of agreement.
- **Final Decision.** If the utility of the offer exceeds the computed threshold, the offer is accepted. Otherwise, it is rejected.

Overall, this strategy balances a strong preference for high utility early on with an increasing willingness to concede and reach agreement as time progresses, while concessions according to the importance of each sub-negotiation.

3.3 Opponent Modeling

Opponent modeling plays a crucial role in automated negotiation [Baarslag *et al.*, 2016]. To predict the opponent's utility function during multi-issue negotiation, we adopt a Bayesian inference framework that incrementally estimates a linearly weighted utility model. In sub-negotiation i, given an offer $o_i = (v_1^{(i)}, v_2^{(i)}, \ldots, v_m^{(i)})$, the predicted opponent utility is defined as:

$$\hat{U}_{\text{opp}}(o_i) = \sum_{j=1}^{m} w_{\text{opp}}^j \cdot v_{\text{opp}}^j(v_j),$$

where w_{opp}^j denotes the estimated weight of issue j, and $v_{\text{opp}}^j(v_j) \in [0,1]$ represents the predicted utility of option v_j within that issue.

We design a set of likelihood functions based on commonly observed behavioral consistency assumptions, such as monotonic concession trends, early-stage utility concentration, and preference-indicative response behaviors. These behavioral models are encoded into a unified likelihood term $\mathcal{L}_{\text{behav}}$, which is combined with a prior distribution $P(\theta)$ over utility parameters to form the posterior strength:

$$p_t \propto \mathcal{L}_{\text{behav}} \cdot P(\theta),$$

where p_t reflects the confidence of a candidate utility function at time t, conditioned on the opponent's observed behavior.

This posterior is used to guide the iterative refinement of option-level utility estimates. For each option $v_j^{(i)}$ in issue j, we apply the following update rule:

$$v^{\text{new}} = v^{\text{old}} + p_t \cdot \Delta,$$

where Δ captures the directional correction derived from observed frequencies, behavioral patterns, or pairwise comparisons, and p_t controls the update magnitude based on posterior confidence.

Through this inference framework, the agent continuously learns and updates its estimation of the opponent's utility function, thereby supporting adaptive bidding and response decisions throughout the negotiation process.

4 Implementation of the Edge Agent

4.1 Bidding Strategy

The preprocessing logic of the edge agent is similar to that of the center agent. The edge agent enumerates the outcome space of the sub-negotiation, which constitutes its entire negotiation.

When bidding, it then identifies all bids whose utilities are close to the maximum value and selects among them using a Softmax-weighted sampling strategy.

4.2 Acceptance Strategy

The edge agent's acceptance strategy is primarily governed by the utility of the incoming offer and the current negotiation time. It adopts a time-dependent concession mechanism, becoming more flexible at the end of the negotiation.

Early Stage (Relative Time $t \leq 0.9$). In the early phase of negotiation, the agent is highly selective. It only accepts offers that appear in its pre-computed list of top bids, referred to as `pref_list`, which represent outcomes with near-maximal utility.

Late Stage (Relative Time $t > 0.9$). At the late stage, the edge agent initiates a flexible concession strategy. The process involves the following steps:

- **Dynamic Utility Threshold.** The agent computes a threshold as follows:

$$U_{\text{min}} = 0.6 \times U_{\text{max}} \times \frac{2}{\pi} \arctan(i)$$

$$\text{threshold}(t) = U_{\text{min}} + (U_{\text{max}} - U_{\text{min}}) \cdot (1 - (t')^{0.5})$$

  where:
  - $t' = \frac{t-0.9}{1-0.9} \in [0, 1]$ is the re-normalized time,
  - U_{max} is the maximum utility of bids in `pref_list`,
  - i is the index of the edge agent in the sub-negotiation sequence, mentioning that an edge agent only knows its index, but does not know the amount of all sub-negotiations.

  As time progresses, the threshold decreases, enabling acceptance of offers with lower utility, particularly for agents positioned later in the sequence.

- **Final Decision.** If the utility of the received offer exceeds the calculated threshold, the agent accepts the offer. Otherwise, it is rejected.

This strategy ensures early caution and late-stage flexibility, while incorporating the agent's position in the sequence to modulate concession levels appropriately.

5 Experiments

Metrics. The performance of agents is evaluated through a final tournament, including a range of scenarios to ensure generality. Each agent's final score A is computed as the average of its utility as center agent (μ_c) and as edge agent (μ_e):

$$A = \frac{\mu_c + \mu_e}{2}$$

Online and offline experiments. We conducted extensive offline experiments under various settings, varying the number of edge agents, utility functions, and negotiation round limits. These helped evaluate the robustness and effectiveness of our strategies (the result is shown in Figure 1).

In the official online testing system, our agent performed consistently well across diverse random scenarios, ranking among the top teams on the leaderboard.

6 Conclusion

This paper proposes an agent for one-to-many sequential negotiations, employing dynamic concession strategies (incorporating both temporal factors and negotiation sequence positioning) and a Bayesian-based opponent modelling approach. It achieves efficient coordination in both center agent (global utility optimization) and edge agent (local utility maximization) scenarios. Experimental results demonstrate that this method significantly improves agreement rates and overall utility performance.

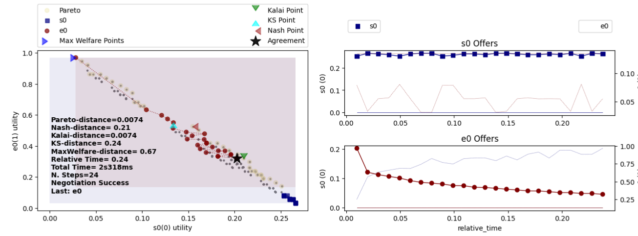

Figure 1: Offline experiment results.

References

[Aydoğan *et al.*, 2017] Reyhan Aydoğan, David Festen, Koen V Hindriks, and Catholijn M Jonker. Alternating offers protocols for multilateral negotiation. *Modern approaches to agent-based complex automated negotiation*, pages 153–167, 2017.

[Baarslag *et al.*, 2016] Tim Baarslag, Mark JC Hendrikx, Koen V Hindriks, and Catholijn M Jonker. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems*, 30:849–898, 2016.

[Baarslag, 2024] Tim Baarslag. Multi-deal negotiation. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2668–2673, 2024.

[Florijn, 2024] Tamara CP Florijn. Negotiation strategies for combining partials deals in one-to-many negotiations. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 2734–2736, 2024.

[Koça *et al.*, 2024] Thimjo Koça, Dave De Jonge, and Tim Baarslag. Search algorithms for automated negotiation in large domains. *Annals of Mathematics and Artificial Intelligence*, 92(4):903–924, 2024.

[Lin *et al.*, 2014] Raz Lin, Sarit Kraus, Tim Baarslag, Dmytro Tykhonov, Koen Hindriks, and Catholijn M Jonker. Genius: An integrated environment for supporting the design of generic automated negotiators. *Computational Intelligence*, 30(1):48–70, 2014.

[Mohammad *et al.*, 2021] Yasser Mohammad, Shinji Nakadai, and Amy Greenwald. Negmas: A platform for automated negotiations. In *PRIMA 2020: Principles and Practice of Multi-Agent Systems: 23rd International Conference, Nagoya, Japan, November 18–20, 2020, Proceedings 23*, pages 343–351. Springer, 2021.