Agent Description Report for Automated Negotiation Leagues 2025

Agent Name: SacAgent
Team Name: University of Tehran
Team Members: Ali Jahani, Hosssein Savari

June 15, 2025

Abstract

This report describes the negotiation agent submitted to the Automated Negotiation Leagues for the year 2025. The agent uses a reinforcement learning-based strategy built on the Soft Actor-Critic (SAC) algorithm. The negotiation setting involves a central agent negotiating sequentially with multiple edge agents. Our model outputs two parameters at each step to govern bidding and acceptance decisions. The reward is based on the final utility achieved in each negotiation. To ensure generalization, training was conducted using both generated and test scenarios.

1 Introduction

In this report, we present the design of our negotiation agent for the Automated Negotiation Leagues 2025. The competition format requires agents to participate in a multi-deal negotiation scenario where a central agent negotiates with multiple edge agents in sequence.

Our agent is built using the Soft Actor-Critic (SAC) reinforcement learning framework. The SAC policy learns to negotiate by treating each full negotiation as a single decision-making step. The objective is to maximize the final utility outcome from each complete episode, rather than optimizing behavior at intermediate steps. During training, the agent learns to act both as a central negotiator and as an edge agent.

2 Bidding Strategy

In the `propose` phase, the agent must decide on a utility value to offer. The SAC model outputs two continuous values, denoted as a_1 and a_2. To ensure a valid range, we swap them if $a_1 > a_2$. We then compute exponential terms as follows:

$$\exp_{\min} = e^{a_1}, \quad \exp_{\max} = e^{a_2}$$

$$u_{\min} = 1 - t^{\exp_{\min}}, \quad u_{\max} = 1 - t^{\exp_{\max}}$$

where $t = \frac{n}{m}$ is the normalized time within the negotiation (step n out of m total steps). The agent samples an outcome with utility uniformly from the interval $[u_{\min}, u_{\max}]$ and proposes it.

3 Acceptance Strategy

In the `respond` phase, the agent receives a proposal and must decide whether to accept it. As in the bidding phase, the SAC model provides two values which are transformed to compute:

$$\exp_{\min} = e^{a_1}, \quad u_{\min} = 1 - t^{\exp_{\min}}$$

If the utility of the received offer exceeds $u_{\min}$, the agent accepts the offer; otherwise, it rejects it.

4 Training

In our approach, each environment step in SAC corresponds to a **complete negotiation** session between two agents—one acting as central, and the other as edge. A full training episode consists of a series of such negotiations, where the central agent engages sequentially with multiple edge agents, consistent with the competition setup.

At each step, the agent observes a vector of features describing the context of the negotiation:

- Normalized index of the current deal
- Total number of deals in the episode
- Total number of steps in a single negotiation
- Boolean flag indicating whether the agent is in the central role

Importantly, during training, our agent is exposed to **both roles**—acting as the central agent in some negotiations and as an edge agent in others. This dual-role training enhances the agent's ability to generalize and react strategically in both positions.

The reward structure is sparse: all intermediate rewards are zero, and only the final utility at the conclusion of each episode is used as the reward signal. This design encourages the agent to focus on outcomes of complete negotiations rather than optimizing isolated actions.

We trained the model using a mixture of generated scenarios and official test scenarios to ensure both coverage and robustness in unseen environments.