Rchan: An Agent Submitted to the ANAC 2025 SCML OneShot Track

Shota Takayama
Tokyo University of Agriculture and Technology
s243395u@st.go.tuat.ac.jp

June 15, 2025

Abstract

This paper presents **Rchan**, an autonomous negotiation agent developed for the ANAC 2025 SCML OneShot Track. Rchan builds upon *CautiousOneShotAgent*, the champion agent from the previous year, which featured concentrated proposal distributions toward high-performing partners and acceptance decisions based solely on time-sensitive quantity thresholds.

Our key contributions are twofold. First, we replace heuristic partner targeting with a *probabilistic first proposal strategy* that dynamically selects up to three partners based on past acceptance outcomes, optimizing expected fulfillment. Second, we introduce a *score-aware acceptance strategy* that evaluates trade-offs using real profit components, including shortage and disposal penalties.

These improvements allow Rchan to adapt more effectively to negotiation environments with unreliable or inflexible partners—addressing a critical weakness in the original approach. By learning from first-proposal history and making decisions that reflect actual reward structure, Rchan achieves robust performance even in adversarial or stochastic settings.

1 Introduction

Rchan is a negotiation agent developed for the ANAC 2025 SCML OneShot track. It is based on *CautiousOneShotAgent*, the winning agent from ANAC 2024, which focused on two key strategies: (1) concentrated first proposals toward historically cooperative partners, and (2) time-dependent acceptance thresholds based on quantity mismatch.

While effective in many scenarios, the baseline agent had a critical limitation—it aggregated all accepted quantities, including both first and counter offers, into a single metric for proposal distribution. This ignored the qualitative difference between proactive offers and reactive acceptances, and led to degraded performance in environments where partners rarely accepted first proposals.

Rchan addresses these limitations with two key innovations:

- A **probabilistic first proposal strategy** that leverages success history from only first proposals, allowing dynamic and expectation-driven targeting of up to three partners.
- A **score-based acceptance mechanism** that evaluates offer sets using a realistic cost function, considering not just quantity gaps but also production limits, shortfall penalties, and disposal costs.

Together, these improvements enable Rchan to tailor its negotiation behavior based on the reliability of each partner. In particular, by distinguishing between proactive and reactive offer success, and by evaluating offers using realistic cost trade-offs, Rchan maintains strong performance even in environments with uncooperative or inconsistent partners—settings in which CautiousOneShotAgent often failed to secure sufficient agreements.

2 The Design of Rchan

2.1 Negotiation Choices: First Proposal Strategy

Previously, agents determined proposal distribution based on total past agreement quantities, regardless of whether they were accepted from first or counter proposals. Rchan improves upon this by focusing *only* on success history from **first proposals**.

For each negotiation round, Rchan selects up to three partners with the highest success rate for first proposals. It calculates the expected acceptance quantity for each combination using a Beta-distribution-based model.

Formally, for each candidate agent i and quantity q, the expected acceptance probability $P_i(q)$ is estimated using:

$$P_i(q) = \mathbb{E}[\mathrm{Beta}(\alpha_i^q + 1, \beta_i^q + 1)]$$

where α_i^q and β_i^q are the counts of success and failure for past first proposals of quantity q.

For each negotiation round, Rchan identifies up to three partners who have historically accepted first proposals with the highest probability. Let $k = \min(3, \text{number of such partners})$, and define the maximum allowed offer per partner as Q. For each selected partner $i \in \{1, \ldots, k\}$, Rchan considers possible quantities $q_i \in \{0, 1, \ldots, Q\}$.

For every combination $\vec{q} = (q_1, \ldots, q_k)$, the expected total accepted quantity is estimated as:

$$\sum_{i=1}^{k} P_i(q_i) \cdot q_i$$

Rchan then selects the combination $\vec{q}$ whose expected total is closest to the required quantity.

When the expected total is insufficient (e.g., no good combinations exist), Rchan estimates the remaining unmet quantity and distributes this residual demand among the other available partners.

The size of this residual is dynamically adjusted based on recent negotiation performance: if, in the last 10 rounds, proposals frequently failed to meet the required quantity, Rchan increases the residual allocation to hedge against under-fulfillment. Conversely, if most past proposals were sufficient, it reduces this buffer. This adaptive adjustment allows Rchan to manage uncertainty and maintain a balanced trade-off between over-ordering and shortfall.

2.2 Negotiation Choices: Acceptance Strategy

In the baseline agent, acceptance decisions were based on minimizing the mismatch (difference between needed and proposed quantity), without regard to outcome profitability.

Rchan introduces a profit-based evaluation. For every combination of offers (subset of current proposals), Rchan computes a score:

$$\text{Score} = \text{Revenue} - (\text{Production Cost} + \text{Shortfall Penalty} + \text{Disposal Cost})$$

Then it compares the best score among combinations that exceed the required quantity (over-acceptance) and those that fall short (under-acceptance).

By evaluating real profit impact, Rchan can choose to accept offers that would previously be rejected (e.g., slightly over-supplying is better than risking shortage when shortfall penalty is high), or avoid accepting costly oversupply when disposal cost dominates.

This results in more conservative and situation-aware decisions, especially beneficial under variable penalty settings.

3 Evaluation

To evaluate Rchan, I tested it in OneShot tournaments against six agents: CautiousOneShotAgent (the winner of SCML2024), EpsilonGreedyAgent, SuzukaAgent, QuantityOrientedAgent, MatchingPennies, and DistRedistAgent.

Each tournament was configured with $n\_steps = 150$ and $n\_configs = 5$, and I ran five such tournaments to compute the average performance. The results are shown in Table 1.

Conclusions

We presented Rchan with key improvements in its proposal and acceptance strategies. By estimating acceptance probabilities from past first proposals and using profit-aware evaluation for incoming offers, Rchan adapts more effectively to varying partner behaviors. Experimental results showed that Rchan

Agent	Average Score
Rchan	**1.0825**
EpsilonGreedyAgent	1.0798
CautiousOneShotAgent	1.0788
DistRedistAgent	1.0784
MatchingPennies	1.0761
SuzukaAgent	1.0646
QuantityOrientedAgent	1.0515

Table 1: Average scores across 5 tournaments ($n\_steps = 150$, $n\_configs = 5$)

consistently outperformed baseline agents, particularly in environments with low-cooperation opponents. These results highlight the benefit of adaptive, history-based negotiation in dynamic multi-agent settings.