

Agent01Agent: Portfolio-Based Negotiation for SCML OneShot

Team 403

ANAC 2026 SCML OneShot Track

Abstract

Agent01Agent is a synchronous negotiation agent for the SCML OneShot track. The main design choice is to treat the simultaneous bilateral negotiations within a day as a single portfolio decision. Instead of accepting or rejecting offers independently, the agent evaluates subsets of current offers using the official SCML utility function and accepts the portfolio with the highest utility. This draft report documents the submitted baseline, the local validation protocol, and the initial ablation results used to decide which mechanisms to keep in the live submission.

1 Motivation

In SCML OneShot, an agent negotiates with multiple suppliers and consumers in parallel while facing coupled quantity constraints. A supply-side agreement is valuable only if it can be converted into profitable sales, and a sales agreement is risky if the agent cannot secure enough input. This coupling makes purely local offer evaluation brittle. Agent01Agent therefore uses a global utility function to score sets of contracts rather than individual contracts.

The submission emphasizes robustness and low computational overhead. It does not use external data, large language models, persistent state across runs, or disk/network access during execution. The current candidate keeps the original portfolio utility core and adds a learned, very conservative penalty-risk tie-breaker only for first-level roles, where local tournament logs showed a consistent positive signal.

2 Agent Design

The agent subclasses the synchronous OneShot controller. At each negotiation callback it collects all current-step offers and searches for the best subset. For typical partner counts the search is exact over all subsets. A beam-search fallback is implemented for unusually large numbers of concurrent offers. The selected offer set is evaluated by the official utility function with already signed contracts included, so the decision accounts for existing supply and sales commitments.

For counter-offers, the agent estimates the remaining quantity needed on the supply and sales sides separately. It distributes requested quantities across active partners and uses a time-dependent concession curve for unit prices. Early in the negotiation it over-orders mildly to hedge against failures; near the deadline it reduces over-ordering to avoid excess inventory or unmatched commitments. The submitted class is:

Field	Value
Agent name	agent01
Module path	agent01.core.agent
Class name	Agent01Agent
Submitted strategy	CoReLearnedPenaltyAgent
Uses LLM	No

3 Validation Protocol

Local validation used the official SCML package tournament runner `anac2024_oneshot`, which is the available runner in the current SCML release used by the skeleton. Each candidate was compared against common baseline agents including `SyncRandomOneShotAgent`, `EqualDistOneShotAgent`, `GreedySyncAgent`, and `RandomOneShotAgent`. We used fixed seeds for reproducibility and varied the number of simulation steps and generated configurations to check whether apparent gains persisted across worlds.

The local experiments are not a replacement for the live leaderboard because the official opponent pool and hidden configurations may differ. They are used only as a filter: a mechanism is promoted only if it gives a clear and repeated improvement over the submitted baseline.

4 Ablation Summary

Several variants were tested. The full version added a regret gate, marginal shadow pricing, and partner reliability tracking. Additional post-submission variants explored capacity-aware allocation, market-price anchored counters, responsive allocation based on current partner prices, and risk-adaptive concession speed. These mechanisms are plausible, but their measured gains were small or unstable in local tournaments.

Agent or variant	Mean score	Comment
Portfolio-only baseline	1.08506	Current live submission
Responsive mild	1.08580	Small, unstable gain
Responsive tiny	1.08553	Small, unstable gain
Selective responsive	1.08307	Not promoted
SyncRandom baseline	1.04782	External baseline
EqualDist baseline	1.04861	External baseline

The main conclusion is that portfolio-level acceptance is the most reliable component so far. Most complex modules improved individual seeds but were not stable enough to replace the baseline. A later log-matched analysis found that a tiny penalty-aware tie-breaker was useful on first-level roles, leading to the current learned hybrid candidate.

5 Planned Work

Future work will focus on stronger evidence rather than adding many heuristics. The next iteration will use larger multi-seed tournaments and, once available, leaderboard feedback from the official live tournament. Promising directions include separate policies for supply and sales sides, adaptive over-ordering based on penalties and level position, and safer tuning of concession curves. Any

new version will be submitted only after it clearly beats the current portfolio-only baseline under repeated local validation.