PonponAgent: A Proactive Strategy Agent submitted to the ANAC 2025 SCM League

Kento Fukuda, Takanobu Otsuka[1]

[1]Nagoya Institute of Technology, Aichi, Japan
fukuda.kento@otsukalab.nitech.ac.jp, otsuka.takanobu@nitech.ac.jp

June 10 2025

Abstract

This paper presents the design and evaluation of PonponAgent, an autonomous agent developed for the ANAC 2025 Supply Chain Management League (SCML). PonponAgent builds upon ProactiveAgent by implementing a suite of adaptive negotiation strategies tailored to both buyer and seller roles. As a buyer, PonponAgent aggressively procures materials in early simulation steps while avoiding overstocking and contracts with distant delivery times. As a seller, it dynamically manages inventory constraints, gradually increases concession levels as simulation progresses, and employs strict upper bounds to prevent over-contracting. Across both roles, the agent leverages partner trust scores and penalty-aware heuristics to refine its decision-making. Experimental results demonstrate that PonponAgent achieves stable and competitive performance in diverse configurations.

1 Introduction

In supply chain management, maximizing profit requires balancing timely procurement, appropriate sales volumes, and risk management. In the SCML competition environment, the key cost factors are inventory holding and shortfall penalties. PonponAgent is designed to flexibly manage negotiation strategies as both buyer and seller, leveraging the features of ProactiveAgent as a base. The agent introduces heuristics for contract timing, quantity, and price, focusing on avoiding both over-purchasing and overselling, while exploiting available production capacity and reliable partners.

2 Design of PonponAgent

PonponAgent extends ProactiveAgent and leverages its utility functions for negotiation context awareness, but adds several important strategies:

2.1 Buyer Strategy

PonponAgent's BUYER strategy focuses on balancing early procurement efficiency with inventory risk control, using the following mechanisms:

1. **Frontloaded Purchasing**
   In the early stages of the simulation (approximately the first 15% of steps), PonponAgent actively seeks purchasing opportunities to secure raw materials early. This ensures input readiness for production while prices tend to be more favorable. This behavior is implemented within both `respond()` and `propose()` by checking the current simulation step.

2. **Inventory-Aware Rejection**
   After the initial phase, new purchases are restricted unless the current inventory falls below a defined safety threshold. The agent uses inventory checks to avoid unnecessary overstocking, and purchase offers are rejected dynamically in `respond()` when the threshold is exceeded.

3. **Delivery Window Constraint**
   The agent only accepts contracts with delivery deadlines that are 2 to 4 steps ahead of the current step. This window ensures a balance between timely fulfillment and inventory control. Contracts outside this range—especially those with very short or distant deadlines—are typically rejected to avoid stock misalignment. This constraint is enforced in `respond()`.

4. **Dynamic Price Evaluation**
   The acceptability of a contract's price is evaluated using the `is_good_price()` function, which considers both simulation progress and recent negotiation outcomes. This enables adaptive price tolerance as market conditions evolve.

2.2 Seller Strategy

PonponAgent's SELLER strategy is designed to ensure profitability while minimizing risk. It consists of the following components:

1. **Minimum Profit Margin Guarantee**
   PonponAgent only accepts sales contracts that guarantee at least a 2% profit margin above the estimated cost. Offers priced below 1.02 times the cost are automatically rejected in the `respond()` function. This rule filters out low-margin deals and ensures consistent profitability.

2. **Aggressive Clearance under Overstock Conditions**
   If the current inventory exceeds 120% of the defined target inven-

tory level (`target_inventory`), the agent adopts an aggressive selling posture. In this mode, it increases the quantity offered and allows for slight price concessions. This behavior is implemented in the `good_offer()` function to effectively reduce excessive inventory.

3. **Hard Upper Limit on Sales Quantity**
   To avoid over-selling and overloading its production capacity, PonponAgent imposes a strict upper bound (`max_sell_qty`) on the quantity it can sell in a single step. This limit is dynamically calculated based on current inventory, outstanding contracts, and production capability, and is enforced via the `max_safe_sales_today()` function.

4. **Delivery Date Diversification**
   While the agent generally prefers contracts with near-term delivery dates, it introduces controlled randomness to spread deliveries across multiple steps. In `good_offer()`, delivery dates are selected from early steps but occasionally include later ones to prevent scheduling bottlenecks.

3 Experiments

To evaluate the agent's performance, we compared PonponAgent against baseline agents under the following settings:

- n_steps = 30
- n_processes = 3
- n_configs = 5
- n_repetitions = 1

We added Proactive_Agent and SyncRandomStdAgent as competitors. The scores of each agent across five configurations and their averages are shown in Table 1. This table shows that PonponAgent consistently achieved the highest scores.

Table 1: Experiment Results

Experiments	PonponAgent	Proactive_Agent	SyncRandomStdAgent
1	1.04	0.59	0.48
2	1.06	0.68	0.59
3	1.09	0.55	0.54
4	0.91	0.89	0.48
5	1.15	0.39	0.51
Average	1.05	0.62	0.52

As shown in the table, PonponAgent consistently achieved a mean score above 1.0, demonstrating stable profit generation and improved robustness over baseline agents.

4 Conclusions

PonponAgent achieves high and stable performance in the SCML environment by combining proactive planning, dynamic risk management, and adaptive negotiation strategies. Inheriting from ProactiveAgent enables modular extension and flexible experimentation with advanced tactics.