PatientAgent

Chris Mascioli and Amy Greenwald
Brown University

June 17, 2022

1 Introduction

`PatientAgent` is an agent designed to take full advantage of the concurrency in the SCML OneShot world. Most agents from last year's competition were based on `AdaptiveAgent` and `LearningAgent`, two agents who use an aspirational concession strategy [3] that learn from interactions on a given day of trading and/or their interactions with opponents in previous negotiations. `PatientAgent` also inherits from `AdaptiveAgent`. In a nutshell, it is hard headed until the end of a negotiation, which ensures that it either makes a very profitable deal early in the negotiation, or negotiations go on as long as possible at which point it seeks the best possible deal for itself. It also uses some basic opponent detection to allow it to better respond to the agents the environment provides (`SingleAgreementAgent`, `GreedyOneShotSyncAgent`, and `GreedyOneShotAgent`).

1.1 World Information

`PatientAgent` focuses on getting as close to matching its exogenous contracts as it can on each day, because of how utility is calculated in the SCML world. The day's utility u is a function of an ordered pair (q, p), where q is the quantity from a contract and p is the price. In general, $u(0, 0) \ll u(q_{\text{exog}}, p_{\text{worst}}) < u(q_{\text{exog}}, p_{\text{best}})$, so the main thing `PatientAgent` does each day is try its best to quantity match the exogenous contracts it receives in order to minimize the loss of utility that comes along with each unit it fails to secure. In the SCML world, the agents vastly prefer to trade at the worst price than to fail to come to any agreement.

1.2 Agent Behavior

The best performing agents from last year's competition were modified versions of `AdaptiveAgent` and `LearningAgent`, so `PatientAgent` was designed with their behavior in mind. Both of these agents tend to focus almost exclusively on price, conceding aspirationally between $(p_{\min}, p_{\max})$. They set their concession rate based on how negotiations have gone (`AdaptiveAgent` sets it based on how negotiations are going during the current day, while `LearningAgent` sets this behavior based on a combination of the prices it has seen on the current day and how its negotiation partner has traded in the past). In a world of just these agent's (without the base agents included in the tournament world) all negotiations proceed until step 20, and the agent making the final offer has an advantage, as the agent's have a discontinuity in their acceptance criteria so will accept almost any offer made at the last time step. None of these base agents take the concurrency of the world into account, which leads them to over offer/accept, and pay penalties.

The agent's provided by the environment all have distinct behavior patterns. `SingleAgreementAgent` often comes to an agreement early and then ends negotiations with all other partners. `GreedySyncOneShotAgent` waits until it has an offer from all partners before sending any replies. `GreedyOneShotAgent` accepts any offer abov a certain utility threshold. Although all three agents have distinct behaviors that can be best responded to, `PatientAgent` only detects and has special behavior for the first of these agents.

2 `PatientAgent`

`PatientAgent` has two separate strategies: one for when it is making the first (and last) offer, and the other for when it is the recipient of the first and last offer.

2.1 First Offer Strategy

When `PatientAgent` is on the side of making the first offer, it simply adopts the strategy of `AdatpiveAgent`. `AdaptiveAgent` is generally nicer than `LearningAgent`, as it isn't severely punishing to other agents for offers they made when desperate in the past, and it adapts better to the changes in daily conditions.

2.2 Last Offer Strategy

Proposal Strategy `PatientAgent`'s primary objective is to make it to the last round, so its quantity offer is always $\min\{q_{\text{exog}}, q_{\text{partner's last offer}}\}$, while its price offer is always p_{best} (i.e., $p_{\min}$ when buyer, and $p_{\max}$ when seller. Various agent strategies were tested that made some concessions in later rounds to induce larger concessions from its partners, but they were found to be ineffective as most agents do not pay attention to changes in offers beyond using them to remember what the best price they were offered, following `LearningAgent` and `AdaptiveAgent`.

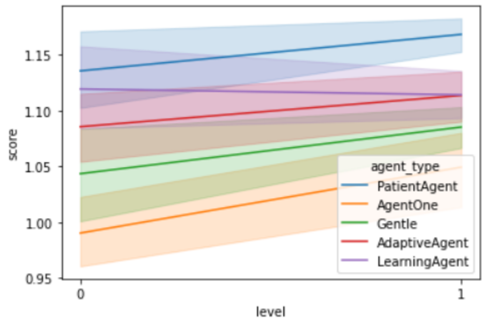
Response Strategy `PatientAgent` has a simple response strategy. Before the last step of the negotiation, it just rejects all offers not at p_{best} (essentially, $AC^{\mathcal{I}}_{const}(1)$ [2]). In the last step, it gathers the final offers from each of the other agents, determines the set with the best utility for itself, and accepts the offers in that set, similar to the patient look-ahead open loop strategy [1]. If there was a de-sync due to other agents using the `wait` response, it re-computes its preferred set among the remaining offers as additional final offers come in, accepting some offers before it runs out of its allotment of `wait` responses (world size + 1).

2.3 Opponent Identification

`PatientAgent` has some very basic opponent modeling. It tries to identify `SingleAgreementAgents` and then it tries to reach fast agreements with them. It identifies these agents by looking for those who end negotiations early without first significantly decreasing the quantity of their offers When successful, this strategy allows `PatientAgent` to dispense with one of its competitors' potential negotiation partners and hence decrease the utility the others can gain this round. In the future, we hope to detect the other two environmental agents as well, and add appropriate response strategies.

3 Results

`PatientAgent` performed well against a test suite of agents, including `AdaptiveAgent`, `LearningAgent`, `Gentle`, and `AgentOne` (the two top agents in last year's competition).



References

[1] Tim Baarslag, Tijmen Elfrink, Thimjo Koça, and Faria Nassiri Mofakham. Bargaining Chips: Coordinating One-to-Many Concurrent Composite Negotiations. page 8, 2021.

[2] Tim Baarslag, Koen Hindriks, and Catholijn Jonker. Acceptance Conditions in Automated Negotiation. In Takayuki Ito, Minjie Zhang, Valentin Robu, and Tokuro Matsuo, editors, *Complex Automated Negotiations: Theories, Models, and Software Competitions*, volume 435, pages 95–111. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. Series Title: Studies in Computational Intelligence.

[3] Peyman Faratin, Carles Sierra, and Nick R. Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3):159–182, September 1998.