Godfather(Goldfish)

Jackson de Campos, Ben Fiske, Chris Mascioli, and Amy Greenwald

August 10, 2021

1 Introduction

The Godfather agent is Brown's University's entrant in the Automated Negotiating Agents Competition (ANAC) Supply Chain Management 2021 OneShot League (SCML OneShot). As the name suggests, all ANAC leagues concern automated negotiation. SCML in particular is characterized by *simultaneous negotiations*. Thus, the main challenge in SCML is to design an agent that can negotiate effectively with multiple other agents simultaneously. Godfather is expressly designed to tackle this challenge.

Godfather's negotiation strategy is built upon a negotiation heuristic that has been shown to be very effective in ANAC over the years, namely the time-based aspiration strategy [1]. This strategy, which aspires to achieve an agreement with a desired utility value, was designed for 1-on-1 negotiations, not simultaneous negotiations. Moreover, it is not possible to use an aspiration negotiator directly, because utility in a 1-on-1 negotiation in SCML is undefined—it depends on the outcomes of not just one but all negotiations! The Godfather agent addresses this challenge by reinterpreting the SCML utility function as multiple dependent marginal utility functions, one per negotiation. It can then apply an aspiration negotiation strategy.

2 Marginal Utility Functions

At the heart of the Godfather agent's strategy is a method for finding a set of marginal utility functions, one per negotiation, which together represent the SCML utility function. This method relies on an oracle which, for each negotiation x, predicts the outcomes $\Omega^{\neg x}$ in all other negotiations $\neg x$. With such an oracle, the marginal utility of agreeing to a contract ω^x in negotiation x is simply the utility of $\Omega^{\neg x} \cup \{\omega^x\}$ less the utility of $\Omega^{\neg x}$. More formally, if we assume an oracle $P^{\neg x} : \Omega \to [0, 1]$ that makes probabilistic predictions over outcomes in other negotiations, the expected marginal utility μ of an outcome ω^x in negotiation x is:

$$\mu(\omega^x; P) = \mathop{\mathbb{E}}_{\Omega \sim P^{\neg x}} \left[u(\{\omega^x\} \cup \Omega) - u(\Omega) \right] \tag{1}$$

where u is a utility function, such as the SCML utility function, which in general depends on a set of outcomes. We estimate the expected marginal utility of an outcome ω^x in negotiation x using a sample average, as described in Algorithm 1 (see Appendix A.1). Further, we estimate the expected marginal utility function for all possible outcomes: i.e., in SCML, for all price–quantity pairs.

We consider three types of outcome predictors, when predicting the outcome of a negotiation, which vary in the amount of information they use to form their predictions. Our first and most basic—the static predictor—uses only information about the world (e.g., our agent's exogenous contract) and about previous days of the simulation (e.g., the outcomes of past negotiations with this opponent). The second—the dynamic predictor—also uses information about the current negotiation trace. Our third and most sophisticated—the introspective predictor—also uses information about the marginal utility functions in the other negotiations.

A **static outcome predictor** takes as input some information about the world (e.g., our agent's exogenous contract) and about previous days of the simulation (e.g., the outcomes of past negotiations with this opponent), but no information about the current state of the negotiation or about the agent's marginal utility functions in the other negotiations. With a static model, we need only compute marginal utility functions once, at the beginning of each negotiation, as nothing changes as the negotiation proceeds.

Figure 1 (see Appendix A.1) illustrates the outcome predictions and expected marginal utility functions generated by our empirical model, which is mostly static (see Section 4 for details). Our agent, a buyer, is negotiating with three sellers: LearningAgent (00Lea@0), GreedyOneShotAgent (01Gre@0), and another Godfather Agent (02GPE@0). The Godfather agent under discussion learns that it is likely to secure valuable contracts when negotiating the other Godfather agent, and is unlikely to reach any agreement when negotiating with any other agent types. Its marginal utility functions reflect this information, as they ascribe very high value to high-quantity, high-price outcomes, and to low-quantity and especially low-price outcomes, all with the other Godfather agent, and much less value to outcomes with the other agent types.

A **dynamic outcome predictor** also takes as input the current negotiation trace. With a dynamic model, outcome predictions change every time the trace changes: i.e., after each round of the negotiation. Thus, it is necessary to recompute marginal utility functions after each round of the negotiation. Moreover, to the extent that there is error in the model's predictions, the agent's marginal utility functions change from one round to the next. Thus, the agent executes its negotiation strategy with respect to a changing utility function.

An **introspective outcome predictor** takes an additional input as well, namely the agent's marginal utility functions in the other negotiations. The marginal utility function in a negotiation affects our agent's behavior in that negotiation, which in turn affects the outcome, which in an introspective model affects the outcome predictions in other negotiations. Hence, an introspective model creates a circular dependence (Figure 2; see Appendix A.1).

We call a set of marginal utility functions *consistent* with an introspective outcome predictor if the outcome predictor, given the marginal utility functions, produces a set of predictions which in turn generates the same marginal utility functions per Algorithm 1. Given an outcome predictor, there may be many sets of marginal utility functions with which it is consistent. Our agent aims to identify just one such set, by iterating to convergence. Although convergence is not guaranteed, when eyeballing the marginal utility functions and outcome predictions, it seemed they usually converged in just a few iterations.

3 Pareto Aspiration Negotiation Strategy

Our bilateral negotiation strategy is a variant of the classic time-based aspiration strategy [1]. It makes offers that it believes will be favorable to its opponent as well as itself by aspiring along the Pareto frontier towards the Nash bargaining solution (see Appendix A.2 for formal definitions of these concepts). Note that this strategy requires a prediction of the opponent's utility function, as well as knowledge of its own.

Aspiration Strategy Our agent calculates the Pareto frontier and the Nash bargaining solution in a single negotiation using its *marginal* utility function and an estimate of the opponent's *marginal* utility function. It then calculates an aspiration value, which serves as a threshold indicating how much utility the agent aspires to achieve at this point in the negotiation. The aspiration value ranges from the maximum possible utility (on the first negotiation round) down toward the utility of the Nash bargaining point

(on the last negotiation round). Specifically, it approaches a target utility X% of the way between the disagreement utility and the utility of the Nash bargaining point. Experiments led us to choose $X = 75$.

The agent only accepts offers above its target utility. When proposing, it makes the offer along the estimated Pareto frontier that minimally surpasses that target. That is, among all offers that achieve its aspiration value, our agent attempts to maximize its opponent's utility.

Opponent's Utility Function We ran experiments to determine which inputs to the SCML utility function had the greatest impact on its value. These experiments confirmed our hypothesis that "desired quantity," which we defined as the difference between the exogenous quantity and the total quantity of contracts already secured, is the most important factor. The second most important factor was production cost; all other factors were negligible.

In light of this, as an estimate of the opponent's marginal utility function, we use the SCML utility function with the following parameters: the agent's level is set exactly; contracts secured in other negotiations are set to none; exogenous quantity is set to the quantity of the opponent's most recent offer (hence the "Goldfish" moniker, due to a goldfish's short memory); all other parameters are set to the means of the distributions from which they are drawn.

When inspecting negotiation traces (see, for example, Figure 3; Appendix A.2), we found that this method of estimation usually resulted in offers much closer to the actual Pareto frontier, as intended, as compared to offers made by a basic linear aspiration agent, which does not attempt to estimate the opponent's utility function or remain on the Pareto frontier.

4 Outcome Prediction

We built two outcome predictors to train and test our agent, one static and the other introspective. Both make the (false) assumption that the outcomes across the simultaneous negotiations are independent. The first, the empirical model, is a simple baseline predictor based on historical data. The second, the gradient boostng machine (GBM), makes its predictions using information about the current environment, the current negotiation trace, and the current iteration of estimated marginal utility functions.

Empirical Model Our empirical model takes as input the outcomes of past negotiations with a given opponent, and outputs a probability distribution over the possible outcomes of the current negotiation. Specifically, it outputs the empirical distribution over those outcomes, with a constant prior weight on the disagreement outcome. Because its predictions do not change with the state of the negotiation, our empirical model is a static model.

GBM Model We simulated a large number of worlds consisting of multiple instances of our agent playing against itself using the aforementioned empirical model so that we could collect data to train a more advanced model. The data we collected fell into three broad categories:

- Information about the world: the day, our level, the size of the different layers, the "competitiveness" of the market, as determined by the exogenous contract quantities (e.g., the competitiveness of a world with total exogenous input contracts of 12 and total exogenous output contracts of 15 is -3)
- Information about the current negotiation: what price and quantity was offered at each step
- Information our agent generates: the distribution over quantity and a point estimate for price from the empirical model and the marginal utility function, compactly represented as a polynomial of degree 3

Using this information, we created 80 different models: one to generate a distribution over quantities and one to generate a point estimate for price, for each of the 40 possible rounds of a negotiation. The models were built using `xgboost` in order to keep train/score time as low as possible (using our intuition/experience that tree-based models generally outperform neural networks on tabular data). The price model had an average RMSE of approximately 3 and the quantity models had an average AUC of around 0.95. The quantity model outputs were then normalized to produce a probability distribution.

Model Hacks We apply two adjustments to make our models more accurate. The first is that when a negotiation ends, we assign probability 1 to its outcome. The second is that if if is our turn, and if our agent plans to respond in such a way as to end the negotiation immediately, then we assign probability 1 to the imminent outcome. These adjustments effectively convert any static model into a dynamic model.

Future Work The main weakness of our models is that they are based on play against ourselves, so the accuracy against agents with different negotiation strategies will most likely be significantly lower. We had planned to mix the output of our two prediction models, weighing each according to its accuracy, but did not have time to test this idea. In future work, we might also improve our model by adding time-series indexing over the negotiation traces, so that we can classify each of our opponent's individual behaviors and then use a model specifically for them.

5 Experiments

We ran experiments to test Godfather's performance against the baseline SCML OneShot agents. In an environment full of baseline agents, Godfather performs very well against the weaker baselines and nearly as well as BetterAgent and LearningAgent. Here are the final scores from a comprehensive experiment facing Godfather agents off against the baseline agents:

Agent	Score
GodfatherEmpirical	1.072
GodfatherGBM	1.126
Better	1.397
Learning	1.241
Sync	0.568
SimpleSingle	0.662
SimpleAgent	0.713
Adaptive	1.011
GreedyInd	0.727

References

[1] Peyman Faratin, Carles Sierra, and Nick R Jennings. Negotiation decision functions for autonomous agents. *Robotics and Autonomous Systems*, 24(3-4):159–182, 1998.

[2] Ariel Rubinstein, Zvi Safra, and William Thomson. On the interpretation of the nash bargaining solution and its extension to non-expected utility preferences. *Econometrica*, 60(5):1171–1186, 1992.

A Appendix

A.1 Marginal Utility Functions

Algorithm 1 Estimate the expected marginal utility of the outcome ω^x of negotiation x

Input: ω^x, u, n, $P^{\neg x}$
Output: marginal utility $\mu^x \doteq \mu(\omega^x)$ of outcome ω^x

$\mu^x \leftarrow 0$
for $i \in \{1, \ldots, n\}$ **do**
  sample outcomes $\Omega^{\neg x}$ from $P^{\neg x}$
  compute $u(\omega^x \cup \Omega^{\neg x})$
  compute $u(\Omega^{\neg x})$
  add $u(\omega^x \cup \Omega^{\neg x}) - u(\Omega^{\neg x})$ to μ^x
end for
divide μ^x by n (in place)
return μ^x

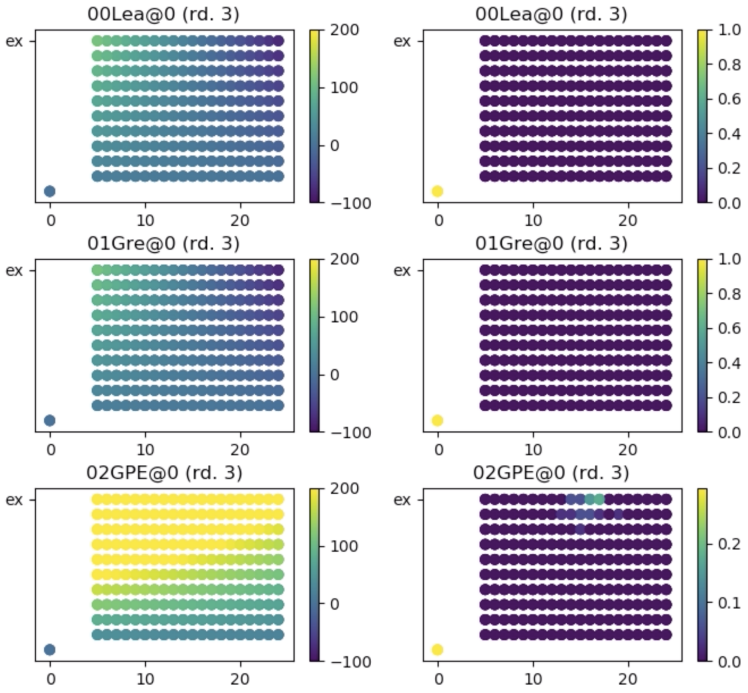

Figure 1: Outcome predictions (right) and corresponding marginal utility functions (left) for each of three opponents, generated by our empirical model. All plots show price on the x-axis and quantity on the y-axis, ranging from 1 to 10. On the left, color represents utility; on the right, probability. In both cases, yellow denotes the highest values, and purple, the lowest.

Algorithm 2 Find marginal utility functions consistent with an introspective outcome predictor

Input: An introspective outcome predictor

Initialize outcome predictions (for example, using a non-introspective predictor)
for $i \in \{1, \ldots, n\}$ **do**
  Estimate expected marginal utility functions per Algorithm 1
  Update outcome predictions per the introspective predictor
end for

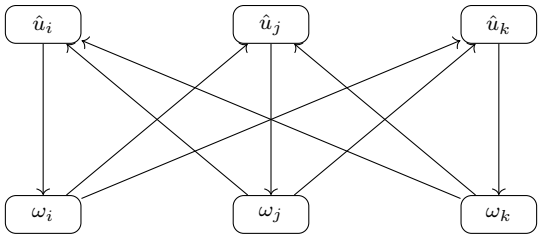

Figure 2: Outcome predictions ω and estimated marginal utility functions $\hat{u}$ for three bilateral negotiations (i, j, and k). Arrows indicate dependencies in the computations.

A.2 Pareto Aspiration Strategy Details

A.2.1 Aspiration Framework

- **Relative time estimate**: Assume the agents' "turns" (propose and respond) are equally spaced, and that the first propose step (of either agent) happens at $t = 0$ while the last respond step (of either agent) happens at $t = 1$. Our agent computes the expected time elapsed $\bar{t}$ as

$$\bar{t} = \max\left\{0, \frac{2k - 0.5}{n}\right\} \tag{2}$$

  Here k is the number of offers our agent made so far in the negotiation, and n is the total number of rounds (40, in SCML 2021). For example, when our agent is pondering its first proposal, $\bar{t} = 0$. While pondering its first response, $\bar{t} = 1.5$, because either two or three turns may have already transpired, and just before its last response, $\bar{t} = 39.5$.

  Note: SCML 2021 limits negotiations both by total number of turns (40) and by total time (120 seconds). An improved relative time estimate would take both of these measures into account.

- **Concession rate**: Our agent calculates its concession rate as follows: $\rho = 1.0 - \bar{t}^{\lambda}$. This function is strictly decreasing for $\lambda > 0$. Additionally, $0 \leq \bar{t}^{\lambda} \leq 1$ for $\lambda > 0$, since $0 \leq \bar{t} \leq 1$. Therefore, $0 \leq \rho \leq 1$.

- **Target utility**: The utility our agent aspires to is a weighted average $u_{\text{asp}} = \rho u_{\max} + (1 - \rho)u_{\text{tol}}$, where $u_{\max}$ is the maximum possible utility our agent can achieve, and u_{tol} is the minimum utility our agent is willing to accept.

A.2.2 Pareto definitions

Let u_i and u_j represent the agent's and its opponent's utility functions, respectively, and let ω_d denote the disagreement outcome, so that $u_i(\omega_d)$ and $u_j(\omega_d)$ represent the agents' reservation values, respectively.

Definition 1 (Pareto Frontier)**.** *Given two agents with utility functions u_i and u_j, respectively, the Pareto frontier Π^* is the set of outcomes s.t. neither agent can be made better off without making the other agent worse off. Formally,*

$$\Pi^* = \left\{\omega \in \Omega \mid \textit{There does not exist an } \omega' \in \Omega \textit{ s.t. } u_i(\omega') \geq u_i(\omega) \textit{ and } u_j(\omega') > u_j(\omega) \textit{ or vice versa}\right\} \tag{3}$$

Definition 2 (Nash Bargaining Solution)**.** *Given a Pareto frontier Π^*, the Nash bargaining solution N^* is defined as follows[2]:*

$$N^* = \arg\max_{\omega \in \Pi^*} \left(u_i(\omega) - u_i(\omega_d)\right)\left(u_j(\omega) - u_j(\omega_d)\right) \tag{4}$$

A.2.3 Parameter Settings

We chose to set $\lambda = 0.5$, which means that our agent concedes more quickly at the start of a negotiation and more slowly near the end. With this value, our agent concedes fast enough to be able to make quick agreements in some negotiations, eliminating uncertainty sooner and avoiding a last-minute scramble, and slow enough as to not trade away potential value and reach undesirable agreements early in a negotiation.

As our agent's minimum acceptable utility, we interpolated between its utility at the Nash bargaining point ω_{n^*} and its utility at the disagreement point ω_d: i.e., we set $u_{\text{tol}} = (0.75)u(\omega_{n^*}) + (0.25)u(\omega_d)$.

A.2.4 Negotiation Traces

Figure 3 depicts two visualizations of a single negotiation between an agent using our negotiation strategy and one using classic linear aspiration. In this negotiation, our agent's strategy is more stable/predictable than the opponent's, and the agents reach agreement very close to the Nash bargaining point.

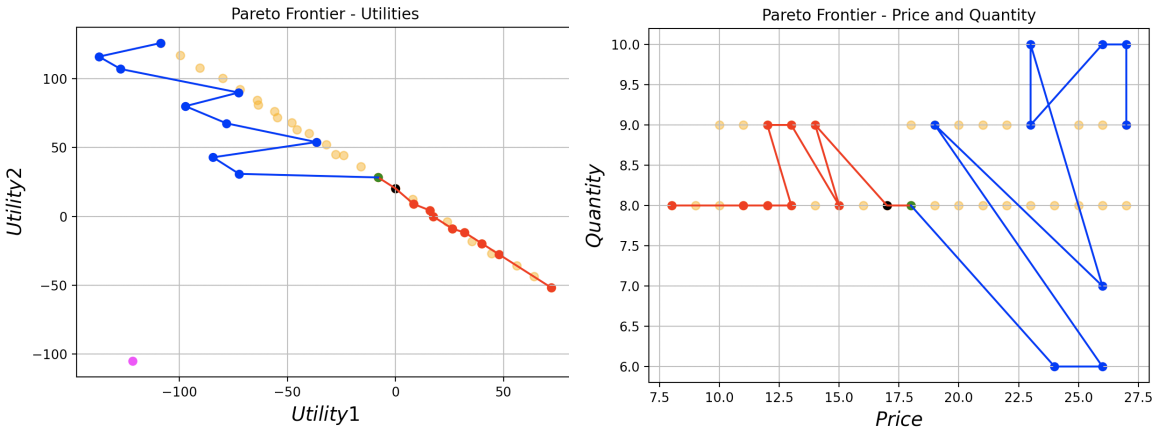

Figure 3: A single negotiation during which the buyer, depicted in red, is using our Pareto aspiration strategy, while the seller, depicted in blue, is using a classic linear aspiration strategy. Both figures show the Pareto frontier in orange circles, as well as the agents' progression of offers, the final agreement, in green, the Nash bargaining point, in black, and the disagreement point, in pink. On the left, the axes are the agents' utilities, while on the right, they are price and quantity.