MONTY HALL: BROWN UNIVERSITY'S AGENT FOR THE SUPPLY CHAIN MANAGEMENT LEAGUE OF THE 2020 AUTOMATED NEGOTIATING AGENTS COMPETITION

Enrique Areyan-Viqueira, Edward Li, Daniel Silverston, Amrita Sridhar, James Tsatsaros, Andrew Yuan, and Amy Greenwald

OVERVIEW. At a high level, MontyHall's strategy is to buy inputs, immediately convert them into outputs, and then sell said outputs. Monty begins its day by computing a business plan. For each day into the future (up to some predefined horizon T), this plan dictates how many inputs to aim to buy and sell so that, in expectation (over many games), it maximizes its profit. After negotiating to fulfill the business plan, Monty signs a subset of its agreements, again with the intent of maximizing profit, but accounting for its past relationship with its negotiating partners.

UNCERTAINTY MODEL. For $x \in \mathbb{N}$, we write $[x] = \{0, 1, \ldots, x\}$, and we assume t ranges over the game's days.

We model the market uncertainty our agent faces by joint distribution $G^t \doteq G^t_{P_{\text{IN}},Q_{\text{IN}},P_{\text{OUT}},Q_{\text{OUT}}}$, where $G^t_{P_{\text{IN}},Q_{\text{IN}},P_{\text{OUT}},Q_{\text{OUT}}}(P^t_{\text{IN}} \leq p_{\text{IN}}, Q^t_{\text{IN}} \leq q_{\text{IN}}, P^t_{\text{OUT}} \leq p_{\text{OUT}}, Q^t_{\text{OUT}} \leq q_{\text{OUT}})$ is the probability that at time t at most a *total* of q_{IN} units of the input were sold at an *average* price of at most p_{IN} per-unit, and at most a *total* of q_{OUT} units of the output were sold at an *average* price of at most p_{OUT}. We denote by P^t_{OUT} (respectively, $Q^t_{\text{OUT}}, P^t_{\text{IN}}$, and Q^t_{IN}) the marginal distribution of output prices (respectively, output quantities, input prices, and input quantities). We bound $Q^t_{\text{OUT}} \in [q_{\max}]$ and $Q^t_{\text{IN}} \in [q_{\max}]$, for all t, assuming hyperparameter $q_{\max} \in \mathbb{N}$.

We take the simplest possible approach to building a model of market uncertainty, which is to build empirical marginal distributions from many simulation of the game by counting the relevant values. For example, $P(Q^t_{\text{IN}} = k) = {}^{k \text{ inputs were traded across all games on day } t}\big/_{\text{total number of games in the logs}}$. An important and interesting future research direction is to experiment with more expressive probabilistic modeling, for example, using hidden Markov models, to model a factored joint (rather than simply marginal) distribution(s).

BUSINESS PLAN. Given our model of market uncertainty, we compute the expected value of the input and output prices (i.e., we set $p^t_{\text{IN}} = \mathbb{E}[P^t_{\text{IN}}]$ and $p^t_{\text{OUT}} = \mathbb{E}[P^t_{\text{OUT}}]$), and focus on deciding how many inputs to buy and outputs to sell. Taking inspiration from the newsvendor model, we assume the agent can sell (respectively, buy) at most some random quantity of the output (respectively, input). Consequently, if the agent decides to sell y_t units at time t, the actual number of units sold is modeled by the random variable $\min(y_t, Q^t_{\text{OUT}})$. Likewise, if the agent decides to buy x_t units at time t, the actual number of units bought is modeled by the random variable $\min(x_t, Q^t_{\text{IN}})$.

Let $\alpha_{tk} = \mathbb{E}[\min(k, Q^t_{\text{OUT}})]$ and $\beta_{tk} = \mathbb{E}[\min(k, Q^t_{\text{IN}})]$. All values α_{tk} and β_{tk} can be computed in $O(q_{\max})$ time, but the details of this computation (a dynamic program) are left for a longer description of our agent. Now, define the decision variables $I_{tk} \in \{0,1\}$ and $O_{tk} \in \{0,1\}$, for $t \in [T]$ and $k \in [q_{\max}]$, with the interpretation:

$$I_{tk} = \begin{cases} 1 & \text{if the plan is to buy } k \text{ inputs at time } t \\ 0 & \text{if the plan is } \textit{not} \text{ to buy } k \text{ inputs at time } t \end{cases}, \quad O_{tk} = \begin{cases} 1 & \text{if the plan is to sell } k \text{ output at time } t \\ 0 & \text{if the plan is } \textit{not} \text{ to sell } k \text{ outputs at time } t \end{cases}$$

The following ILP (1) solves for the number of inputs to buy and sell on each day of horizon T so as to maximize expected profit, given our uncertainty model:

$$\begin{aligned}
\max_{I_{tk}, O_{tk} \in \{0,1\}} \quad & \sum_{t=0}^{T} \left[\sum_{k=0}^{q_{\max}} \beta_{tk} p^t_{\text{OUT}} O_{tk} - \sum_{k=0}^{q_{\max}} \alpha_{tk} p^t_{\text{IN}} I_{tk} \right] \\
\text{s.t.} \quad & \sum_{k=0}^{q_{\max}} I_{tk} \leq 1 && \forall t \in [T] \\
& \sum_{k=0}^{q_{\max}} O_{tk} \leq 1 && \forall t \in [T] \\
& \sum_{k=0}^{q_{\max}} k O_{tk} \leq \sum_{s=0}^{t-1} \left[\sum_{k=0}^{q_{\max}} k I_{sk} - \sum_{k=0}^{q_{\max}} k O_{sk} \right] && \forall t \in [T] \setminus \{0\} \\
& O_{0k} = \begin{cases} 1, & \text{if } k = i \\ 0, & \text{otherwise} \end{cases} && \forall k \in [q_{\max}]
\end{aligned} \tag{1}$$

The first and second constraints guarantee that the plan can only target one quantity in the set $[q_{\max}]$ on each day. The third constraint is an inventory feasibility constraint—it guarantees that enough inputs are available to convert to outputs today, or enough outputs are leftover from previous days, to meet the day's target. The fourth constraint defines the initial output inventory level $i \in \mathbb{N}$. (We assume all inputs are converted to outputs immediately.)

Finally, note that given an optimal solution $\{I^*_{tk}, O^*_{tk}\}_{t\in[T],k\in[q_{\max}]}$ to ILP (1), we can recover an optimal business plan $\{x^*_t, y^*_t\}_{t\in[T]}$, where x^*_t (y^*_t) is the optimal number of inputs (outputs) to buy (sell), by setting:

$$x^*_t = \sum_{k=0}^{q_{\max}} kI^*_{tk} \quad \text{and} \quad y^*_t = \sum_{k=0}^{q_{\max}} kO^*_{tk}$$

Our agent solves ILP (1) at the beginning of each day using python's pulp mathematical program solver (pypi.org/project/PuLP/). Empirically, we found that it takes less than one second to solve for a typical plan. Our agent then uses its plan to set a negotiation agenda, specifically a range of quantities based on the plan. Currently, our agent sets a range of prices based on catalog prices. An interesting future direction would be to expand our business planer to account for the full uncertainty over prices modeled by P^t_{IN} and P^t_{OUT}, and not just the expected values of these distributions. Alternatively, we might carry out a parallel strategy in addition to the current one, taking expectations of Q^t_{IN} and Q^t_{OUT} as input, and outputting a range of prices as a negotiation agenda.

CONTRACT SIGNER. We now describe how our agent decides which subset of agreements to sign.

Let A be the set of agreements about which the agent needs to make a signing decision. An agreement $a \in A$ is defined as $a = \langle q_a, p_a, \tau_a, d_a, \rho_a \rangle$, where $q_a \in \mathbb{N}$ is the quantity, $p_a \in \mathbb{R}_+$ is the price per unit, $\tau_a \in \{t+1, t+2, \ldots, T\}$ is the delivery date, $d_a \in \{\text{BUY}, \text{SELL}\}$ is the "direction", and ρ_a is the negotiating partner. Conditioned on our partner also signing, an agreement becomes a binding contract.

Our agent decides whether or not to sign an agreement based on an assessment of its partner's likelihood of signing. Let $\pi(\rho) \in [0,1]$ be the probability that partner ρ will sign an agreement. As a first approximation, we assume that the signing probability $\pi(\rho)$ is the same across all of our agreements with ρ, and independent of any previous or future signing decisions. Moreover, signing probabilities are independent across partners.

Our initial signing probability estimates are simple counts of the form $\pi(\rho) = {}^{\text{\# of agreements reached and signed by } \rho}/_{\text{total \# of agreements reached with } \rho}$. However, if a partner does not sign agreements early on, we run the risk of never signing agreements with the partner again. To remedy this situation, we model each signing event with partner ρ as a Bernoulli random variable with parameter $\pi(\rho)$, and maintain a confidence interval $\hat{\pi}(\rho) \pm \varepsilon$, via Hoeffding's inequality, where $\hat{\pi}(\rho)$ is the empirical signing probability. Finally, we use the upper bound $\hat{\pi}(\rho) + \varepsilon$ as a proxy for the true signing probability $\pi(\rho)$.

ILP (2) selects a subset of agreements in A that maximizes expected profit subject to feasibility constraints, where the expectation is taken over the negotiating partner's signing probabilities. Feasibility means that there will be enough outputs available for each sell agreement the agent signs, among initial inventory, new inputs from buy contracts signed today, and leftover inputs and outputs from previous days. As usual, the formulation assumes that every input is turned into an output immediately. Further, $i \in \mathbb{N}$ is the initial inventory, defined as the total quantity of inputs and outputs already in storage when making the signing decisions.

$$\begin{aligned} \max_{z_a \in \{0,1\}} \quad & \sum_{a \in A_{\text{SELL}}} q_a p_a \pi(\rho_a) z_a - \sum_{a \in A_{\text{BUY}}} q_a p_a \pi(\rho_a) z_a \\ \text{s.t.} \quad & \sum_{\substack{s \in A_{\text{SELL}}: \\ \tau_s = t}} z_s q_s \leq i + \sum_{\substack{b \in A_{\text{BUY}}: \\ \tau_b < t}} z_b q_b - \sum_{\substack{s \in A_{\text{SELL}}: \\ \tau_s < t}} z_s q_s \quad \forall t \in \mathscr{T} \end{aligned} \tag{2}$$

In ILP (2), for all $a \in A$, $z_a \in \{0,1\}$, with $z_a = 0$ when the agent is not going to sign agreement a, and $z_a = 1$ when it will sign agreement a. Further, $A_{\text{BUY}} = \{a \in A \mid d_a = \text{BUY}\}$, and $A_{\text{SELL}} = \{a \in A \mid d_a = \text{SELL}\}$. (Note $A = A_{\text{BUY}} \cup A_{\text{SELL}}$.) Finally, $\mathscr{T}$ is the set of all times for which the agent has a sell agreement.

Our agent solves ILP (2) using python's pulp mathematical program solver (pypi.org/project/PuLP/). Empirically, we found that it takes less than one second to solve ILP (2).