AgentFO2

Hyuga Matsuo

Tokyo University of Agriculture and Technology
matsuo@katfuji.lab.tuat.ac.jp

1 Learning Method

1.1 Data to Store

My agent stores the following data.

- History of utility received from the opponents.
- History of hamming distance from the previous bid.
- Who accepted, the utility at that time, and the utility of the supposed Pareto Front at that time.
- Estimated strategy of opponent and value of parameter *min*.

My agent uses them for strategy of estimating the opponent.

1.2 Supposed Pareto Front

My agent predicts Pareto front for domain estimation. It supposes that the utility of the first bid proposed by the opponent has 1 for the opponent. Then, it supposes that when it sends a bid with my utility of 1 to the opponent, the opponent's utility equals my utility of the first bid proposed by the opponent. It supposes that the line segment connecting these two bids is Pareto Front. This Supposed Pareto Front is defined as:

$$y = -x + 1 + u_{first}$$

when x is my utility, y is the opponent's utility, and u_{first} is my utility of the first bid proposed by the opponent.

1.3 Strategy Estimation of Opponent and My Policy

Using the Hamming distance, my agent classifies the opponent's strategy into three types: time-dependent, random, and others. Also, it decides my policy according to each strategy.

vs. Time-Dependent
If a bid Hamming distance is 2 or more comes only after 20 bids from the beginning, it is judged that the opponent's strategy is time-dependent. At this time, my agent changes the policy depending on who accepted the bid previous negotiation.

If my agent accepted the bid at the previous negotiation, the parameter min is defined as:

$$min = \min\left(u_{preAcc}, U_{suppose}(u_{preAcc})\right)$$

when u_{preAcc} is the accepted utility of the previous negotiation and $U_{suppose}(u)$ is the estimated utility for the opponent when my utility is u.
If the opponent accepted the bid at the previous negotiation, the parameter min is defined as:

$$min = \min\left(\max(u_{preAcc}, U_{suppose}(u_{preAcc})), 0.9\right)$$

vs. Random
If a bid Hamming distance is 2 or more comes only before 20 bids from the beginning, it is judged that the opponent's strategy is random. At this time, my agent changes the policy depending on who accepted the bid previous negotiation.
If my agent accepted the bid at the previous negotiation, the parameter u_{goal} is defined as:

$$u_{goal} = \max(UtilityLog) \tag{1}$$

when $UtilityLog$ is the history of utility received at previous negotiation. Also, $RandomMax$ is defined as:

$$RandomMax = \max(U(\omega_t)) \left(t \leq \frac{1}{e}\right)$$

when ω_t is a bid at a certain time t and $U(\omega)$ is the utility obtained from bid ω. This equation obtains the maximum utility of all bids received at the time less than $\frac{1}{e}$ with reference to the secretary problem (this e is Napier's constant.)
If the opponent accepted the bid at the previous negotiation, the parameter u_{goal} is defined as Equation (1).

vs. Others
If the opponent is neither time-dependent nor random, it is judged that the opponent's strategy is others. At this time, my agent changes the policy depending on whether the previous negotiation is achieved the agreement or not.
If the agreement was reached at the previous negotiation, the parameter min is defined as:

$$min = \min(u_{preMin} + 0.05, 0.8)$$

when u_{preMin} is value of min at the previous negotiation.
If the agreement was not reached at the previous negotiation, the parameter min is defined as:

$$min = \max(u_{preMin} - 0.05, 0.4)$$

2 Bidding Strategy

My agent uses a simple time-dependent strategy. The equation is defined as:

$$f(t) = min + (max - min) * t^{\frac{1}{e}}$$

when e is 0.4 and max is 1. The utility of the proposing bid is randomly determined from $[f(t) - 0.05, \min(1.0, f(t) + 0.05)]$ using $f(t)$.

3 Acceptance Strategy

My agent uses an acceptance strategy that combines multiple conditions. It is defined by the following equation.

$$
\begin{aligned}
1.02 * U(\omega_t) + 0.04 &\geq U(\omega_{t+1}) \\
U(\omega_t) > u_{goal} \ \& \ t &> t_{notAccept} \\
U(\omega_t) > RandomMax \ \& \ t &> t_{notAccept} \\
U(\omega_t) > 0.85 \ \& \ U(\omega_t) &\geq U_{suppose}(\omega_t) \\
t > 0.95 \ \& \ U(\omega_t) &\geq U_{suppose}(\omega_t)
\end{aligned}
$$

If any of these equations are True, it accepts the bid.

4 Opponent Modeling

My agent does not use any opponent modeling.