RTAgent

Reita Kaneko and Yuko Sakurai
Nagoya Institute of Technology

April 14, 2024

1 Introduction

RTAgent is an agent that rethinks decision from evaluation of past negotiations. Preliminary experiments demonstrated that SimpleAgent, which only considers quantity, is the most effective base agent. Thus, we improve upon SimpleAgent by Introducing rethinking elements.

In the proposal strategy, we adopt a "improvement strategy" that takes into account the proposal quantity from the negotiation partner. By using this improvement strategy, it is possible to make a balanced proposal between the other party's offer and our own request, thus increasing the chances of a successful negotiation. In the acceptance strategy, we adopt a "rethinking strategy". Specifically, we simply evaluate past negotiations and use the average of those negotiations as a benchmark. This rethinking strategy determine if the other party has made a concession.

2 RTAgent

2.1 Symbol Definitions

The definitions of the Symbols are as follows :

i : opponent number

$offer[...]$: offer to the negotiating opponent

q_{need} : current required quantity

$h_{eval}[i]$: all evaluations of opponent i (Update throughout the day)

$f(q)$: function for evaluating offer quantity
We definit $f(q)$ as follows:

$$f(q) = \begin{cases} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(q-q_{need})^2}{2\sigma^2}) & if\ q \leq q_{need}+1 \\ 0 & if\ q > q_{need}+1 \end{cases}$$

We assign any value to parameter σ^2. This function takes a larger value the closer q is to q_{need}, and a lowest value when q is excessive.
To evaluete quantitiy and price collectively, RTAgent stores the following values in $h_{eval}[i]$:

$$\begin{cases} f(offer[quantitiy]) * offer[price] & if\ opponent\ i\ is\ buyer \\ -(1 - f(offer[quantity])) * offer[price] & if\ opponent\ i\ is\ seller \end{cases}$$

In both cases, it takes large value when the offer is better for RTAgent.

2.2 Agent Behavior (Offer Strategy)

In the offer strategy, RTAgent determines the offer to propose to the negotiating opponent by comparing $h_[i]$ and q_{need} according to the following:

$$offer[quantity] = \begin{cases} q_{need}+1 & if\ average(h_{eval}[i]) = 0 \\ q_{need} + k(\in [-2, 0]) & if\ average(h_{eval}[i]) \neq 0 \end{cases}$$

$$offer[price] = \begin{cases} max\_value & if\ current\_step/n\_steps < 0.7 \\ min\_value & if\ current\_step/n\_steps \geq 0.7 \end{cases}$$

If average of $h_{eval[i]}$ equals zero, RTAgent determins the quantity of the other party is excessive, and concede only one quantity. When not so, RTAgent propose the acceptable quantity such that the average improves. In addition, RTAgent sets the price according to the current step to increase the success rate of negotiations.

2.3 Agent Behavior (Acceptance Strategy)

In the acceptance strategy, RTAgent rethinks whether to accept or reject the offer rejected by SimpleAgent according to the following formula:

$$\begin{cases} Accept & if\ average(h'_{eval}[i]) > average(h_{eval}[i]) \\ Reject & if\ average(h'_{eval}[i]) \leq average(h_{eval}[i]) \\ EndNegotiation & if\ q_{need} \leq 0 \end{cases}$$

$$offer \in h'_{eval}[i] \cap offer \notin h_{eval}[i]$$

If current offer improves the average of evaluation values of the offer, RTAgent determines that the other party has made a concession, and then accepts the offer. If not so, RTAgent rejects the offer. However, if the required quantitiy is less than or equal to zero, RTAgent terminates negotiations.

3 Evaluation

We compare between our proposed RTAgent and base agents by varying the number of days. Table1 show the average obtained utility of RTAgent and SimpleAgent. RTAgent outperformed SimpleAgent in most cases.

Table 1: Table1: Experimental Results

Agent \ Num of days	50days	125days	200days
RTAgent	1.092	1.114	1.121
SimpleAgent	1.091	1.116	1.117

4 Conclusions

We proposed RTAgent by incorporating "improvement" and "rethinking" concepts to SimpleAgent which obtained the bast result among the base agents in this year's problem setting. In our experiments, we showed that RTAgent performed better than SimpleAgent. Future work will focus on developing more sophicated agents capable of evaluating an opponent's offer effectively and adjusting the offer based on these evaluations.