BlueWolf Agent Design for SCML Standard Track

Zun Li

July 2021

1 Introduction

This manuscript presents our agent design for SCML'2021, standard track competition.

Our agent directly adopt SupplyDrivenProductionStrategy and PredictionBasedTradingStrategy for scheduling production and decide needs, with the following main differences: (1) we only accept negotiation when our agent is at the first level or the last (2) we adopt the ObedientNegotiator to retract negotiation management back to the agent class itself, and use a similar Boulware-styled negotation strategy as adopted by the OneShotGreedyAgent.

We now define some notations.

u_i is the utility function of player i. It is a function of $AGG\_CONTRACTS$.

$BEST\_PRICE(o)$: the best price (highest for L_0 and lowerest for L_1) encountered during the bargaining process with the opponent o today so far.

$BEST\_PRICE$: the best price across all opponents today so far.

$ACC\_BEST\_PRICE(o, w)$: the best price of the negotiation results toward opponent o in the past w agreements.

$MIN\_QUANTITY(o, t)$: the minimum quantity proposed by opponent o in the current bargaining thread in the past t rounds.

We next elaborate our propose strategy (when asked to propose an offer) and respond strategy (when asked to respond to an opponent's counter-offer).

1.1 Respond

We first reject any offer whose time issue is more than 3 days ahead of the current time step.

Otherwise, we check whether the price issue is a *good price*. If it is, then we accept, otherwise reject.

We will specify what do we mean by *good price* in Sec 1.3.

1.2 Propose

First we determine the quantity issue to be proposed, as we consider it crucial. We let the quantity be $\min(MIN\_QUANTITY(o,3), \max(2/3 \times DEMAND, 1))$. Our interpretation is: the proposed quantity should be at most some proportion of its own left demand (we choose 2/3); meanwhile to increase chance of being accepted, it shouldn't be too much larger than the opponent's left demand (which we use $\min(MIN\_QUANTITY(o,3)$ as an indicator.)

After have chosen this quantity, we first find a *good price* (Sec 1.3). Then starting from this *good price*, we increasing it til the *best price* (which is the maximum prices for L_0 and minimum for L_1), until the one that makes the current marginal utility positive. If we have not found such price, we just set the price as the best price. Then we return this offer as the proposal. For time issue, we just randomly select a time from current step and current step + 3.

1.3 Good Price and Price Concession Strategy

Now we define what do we mean by a *good price* at a certain round of bargaining process. Intuitively, for an L_0 agent its acceptation/proposing price should be lower and lower as the negotiation continues (higher and higher for L_1). Being consist with GreedyOneShotAgent, we use a concession factor e to model such concession effect, combining with the prices information encountered so far.

To be more specific, for an L_0 agent, we consider a range of price $[mn, mx]$ where mx are the maximum price possible and mn is dynamically changed. We let $mn = \max\{(1-\zeta_1)mx, \min\{(1-\zeta_2)BEST\_PRICE(o,d), (1-\zeta_3)BEST\_PRICE(d)), (1-\zeta_4)BEST\_ACC\_PRICE(o,d)\}\}$. I.e., the prices should be at least some proportion of the best prices encounted so far.

For now we let $\zeta_1 = 0.1, \zeta_2 = 0.2, \zeta_3 = 0.52, \zeta_4 = 0.3$

And define $th = ((T-t)(T))^e$, then we say a price p is good if $p - mn \geq th \cdot (mx - mn)$, where t is the round index of a bargaining and T being the maximum round of bargaining.

And when proposing, we define a good price as $mn + th(mx - mn)$.

For L_1 it is similarly defined.