Ecole Polytechnique Fédérale de Lausanne - EPFL

CS-430 Intelligent Agents

Mission 5

Auction Agent

Armand Bosquillon de Jenlis
Vladimir Somers

Group 27

Professor : Boi Faltings

Academic Year 2015 - 2016

1 Introduction

In this report, we first talk about the problem properties we observed in the auctions. These properties guided us toward the choice of our strategy. Finally, we will explain our strategy and the results it gives against other strategies.

2 Our theoretical discovery

First of all, it is good to remember that the goal we want to achieve in the end is to finish with more money than the opponent. More formally, lets define **Reward1** $= \sum bid1 - cost(task1)$ where task1 are all the tasks won by agent 1 and $\sum bid1$ is the sum of the bids for the tasks won by agent 1. For the rest of the report, we state that **our agent** is agent **1** and the **opponent** agent is agent **2**. So what we want is that at the end **Reward1** $<$ **Reward2** $\Leftrightarrow \Delta(R) = Reward1 - Reward2 > 0$. Therefore, at each task:

- **Our agent** should try to find the best bid in order to **MAXIMIZE** $\Delta(R)$.
- **Our opponent** should try to find the best bid in order to **MINIMIZE** $\Delta(R)$.

2.1 Bidding in the middle

Our first discovery was that it is important to adapt our bid in function of the advantage that the opponent has to take the task. In Figure 1, we show that if we are able to know how much taking a task will impact the final value **V1** of Reward1 and the final value of Reward2 **V2**, then the **best bid** is $\frac{V1+V2}{2}$. It captures the idea that I prefer to lose 4.9 CHF than to let my opponent win 5.1 CHF, and of course this statement is also true for him.
As we can see, there is always two options: I win the task or my opponent wins the task. The best possible opponent will always pick the action that minimizes $\Delta(R)$: this is why all the possible values for $\Delta(R)$ in function of the value of my bid follow the red lines. As I want to optimize $\Delta(R)$, I can see that in any case the best choice for my bid is: bid $= \frac{V1+V2}{2}$.

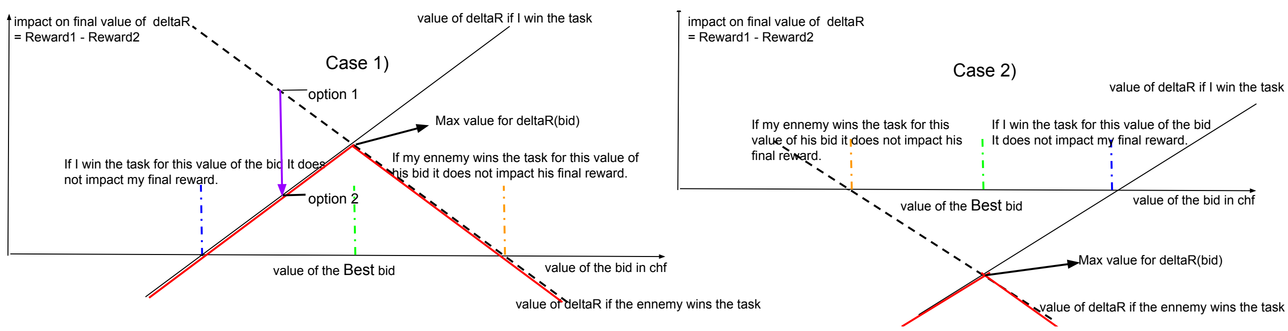

Figure 1: Bidding in the middle

2.2 Bidding at the mean cost per task

Our second finding was that it is a very good idea to always bid close to the "final" mean cost per task. For this section we assume that:

- There are a total of **N** tasks that are going to be distributed.
- The cost of delivering x tasks is always constant, no matter what these x tasks are.
- **Mean(x)** means the cost of delivering x tasks divided by x. This function decreases with x. It is the mean cost per task for delivering x tasks.
- The marginal cost of a task when we have no tasks is always bigger than Mean(N), as Mean(1)<Mean(N)

With these few -quite realistic- assumptions, we quickly see the advantage of **bidding close to Mean(N)**. For example, if we fight against an agent that always bid at his marginal cost, we are going to take all the tasks and end up with a reward slightly positive (we bid just above Mean(N)).
Figure 2 summarizes the advantages of bidding just above Mean(N) under our assumptions. First of all, with this strategy we are sure that if we win all the bids, we will end up with $\Delta(R) > 0$.

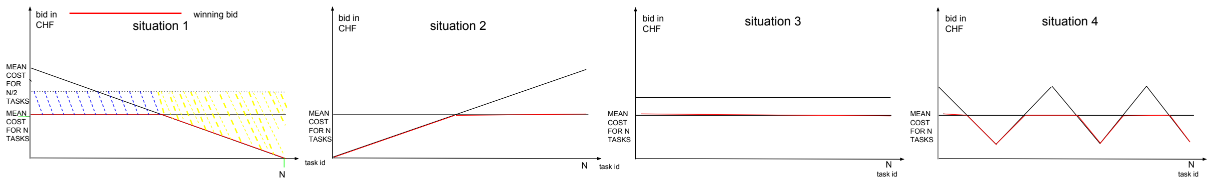

Figure 2: Biding at the mean cost per task : different situations

Situation 1 shows that if the enemy starts pessimistic and then starts to be optimistic then $\Delta(R) > 0$. You can see that on the Figure: the final profit of the two agents will be negative, but the negative profit of the agent biding at the mean cost per task (the blue area) will be smaller than the negative profit of the other agent. The agent biding at mean cost will therefore win! The three other situations shows different kind of biding strategy, but we can observe that the agent biding at mean cost always wins!

2.3 Combining our two properties

Under the assumptions of section 2.1, we can see in Figure 3 that our simple strategy of bidding around Mean(N) can be beaten by a more complex and precise strategy:

In this figure, we can see that the opponent (orange line) bids just bellow our bids (black line), until 3*4 of the total task amount, and then bids just above us. In this case, we both bids bellow the mean cost per task, so we both loose money. But we can see that the amount of money lost by the opponent (yellow area) is smaller than the amount of money we lost (blue area). Therefore, the opponent will win!

To avoid such situations, we should be clever than him: If we change our biding strategy and bid a bigger value (green line) after 3/4 of the total amount of tasks, then the opponent will loose in any cases! This situation shows how it's important to be aware of the opponent's bidings and estimated profit, in order to optimize our strategy.

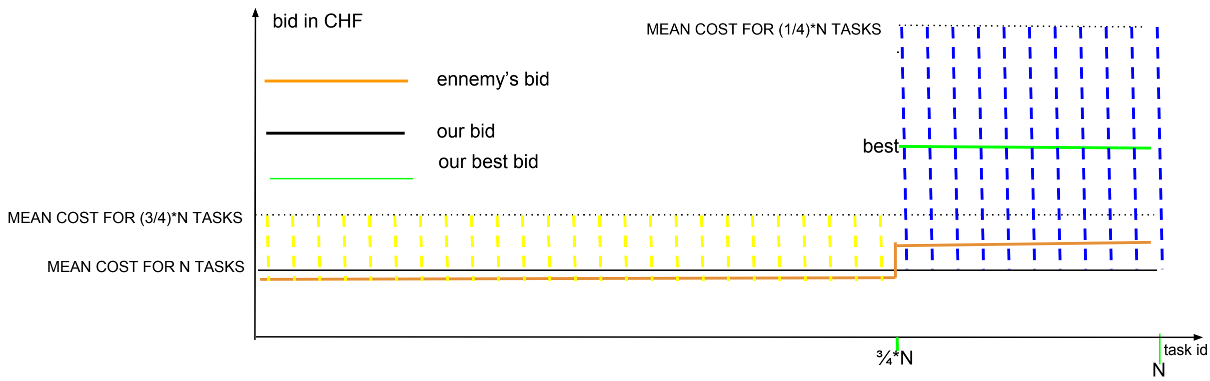

Figure 3: A tricky situation

3 Final Strategy

You can see on Figure 4 the graphical representation of our strategy. During the setup stage, we compute Mean(N) using the N most probable tasks. We recompute this mean cost at each step of the auction, by adding the *N-t* most probable tasks to the *t* tasks that our agent already acquired. We then compute the maximal marginal cost *maxMarginalCost*, which is two times the biggest distance between two cities.

At each step of the auction, we also compute the maximal bid *maxBid* we allow our agent to make. This maxBid is simply located at $x\%$ of the distance between *maxMarginalCost* and *meanCostPerTask*. As you can see on the Figure, this ratio x starts at 0 and reach his constant final value $x = 40\%$ after 10 tasks. We do the same kind of computation for the *minBid*: it is equal to $y\%$ of the *meanCostPerTask*. This ratio y starts at 90% and reaches 60% after 10 tasks.

This two values, *minBid* and *maxBid* defines the interval in which we will make our bid. However, this final bid depends on two last important parameters : our marginal cost for the current task and the estimated marginal cost of the opponent for this task. As we explained earlier, we should always bid between the marginal cost of the ennemy and our own marginal cost. In order to limit our bid to the interval we have just defined, we first compute the marginal cost of the enemy *MC2* and our own marginal cost *MC1* and do a mapping of these marginal costs on the interval $[minBid, maxBid]$. These two mapped costs are called respectively *M2* and *M1*. We finally compute the bid we will make as $\frac{M1+M2}{2}$.

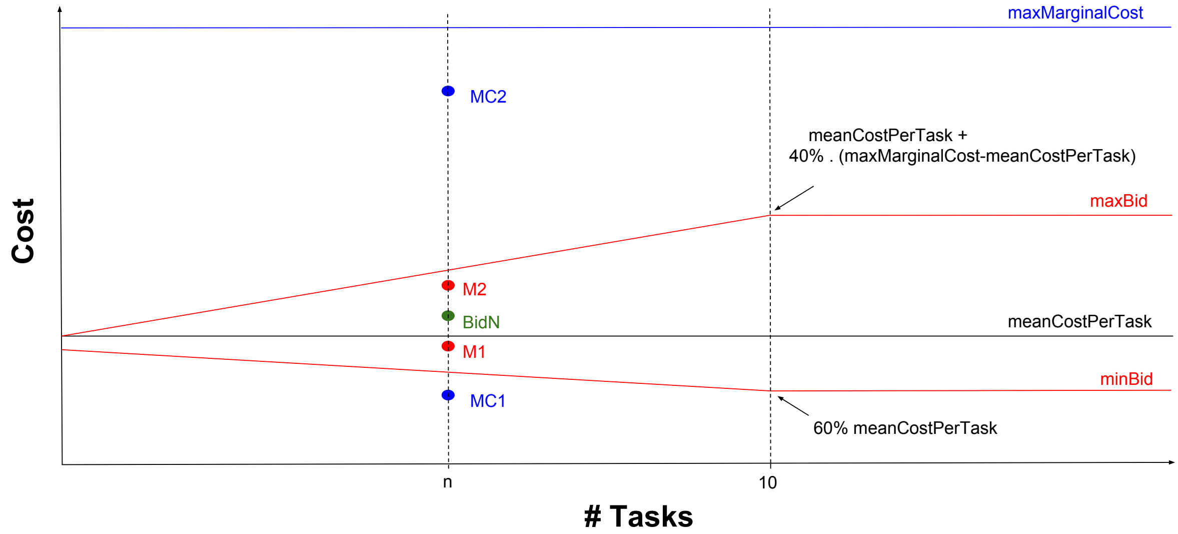

Figure 4: Final strategy

3.1 Estimation of Mc2

One of the difficult points was the estimation of the marginal cost of the opponent *Mc2*, as we do not know the configuration of the opponent. To do so, we used three random possible configurations for the opponent and we give a weight to each of them. In order to estimate the marginal cost, we do as follow: if configuration *a* gives *Mc2a*, configuration *b* found *Mc2b*, configuration *c* found *Mc2c*, then we estimate that $Mc2 = \frac{wa*Mc2a+wb*Mc2b+wc*Mc2c}{wa+wb+wc}$.

3.1.1 Choice of the weights

At the beginning, all the weight have the value 1. Each time auction result is called, we compute for each configuration *conf* the ratio $\frac{bid2}{Mc2}$ where Mc2 is the marginal cost of the opponent for the configuration *conf*. We keep all these ratios and the previous ones in a table. Thanks to these tables, we compute the variance of the ratios for each strategy. Finally we augment the weight of the configuration that has the smallest variance and decreases a little bit the others. The way we adapt our weights is a bit more complicate than this but it is the main idea.

4 Results

Our final agent gives good results. We are able to beat easily any dumb agent, such as the given Template or an agent making random bids. We are also able to beat some more complex and smart agents. The first complex agent we can beat is an agent which always returns a constant bid *B*. He computes this constant *B* as the mean cost per task for the 15 most probable tasks. The second smart agent we can beat is an agent which bids just between his marginal cost and the estimated marginal cost of its opponent.

5 Conclusion

This project was much more complicated than the previous one in the sense that an Auction is a very complex situation where each agent has very few informations. This is why we can create a very large range of strategies, where none of them is optimal.