Semester project - An analysis of bidding strategies for the Trading Agent Competition

Jonas Vautherin
Master Student
Ecole Polytechnique Fédérale
de Lausanne

Ludek Cigler *(supervisor)*
Artificial Intelligence Laboratory
Ecole Polytechnique Fédérale
de Lausanne

Abstract— **The Trading Agent Competition/Ad Auctions game (TAC/AA), taking place annually since 2009 at *University of Michigan*, offers an opportunity to compare the bidding strategies of the different participants. The simulated market environment of this competition is therefore a good place to study dynamics of ad auctions.**

This paper reports simulations made on a strategy using no advanced prediction models as well as the testing environment created to analyze the obtained data. It then compares this simple, model-light strategy to more advanced ones such as winners of last years' competitions and those developed by previous students at the Swiss Federal Institute of Technology (EPFL).

I. INTRODUCTION

A. The Competition

Advertisement represents a non-negligible part of the overall revenues generated on the Internet. More specifically, sponsored search [6] designates this form of advertising where an *advertiser* pays a given amount of money to have its advertisement appearing on some webpages in certain conditions. The publisher then displays those ads on its website when a user is searching for the corresponding set of keywords. This way the advertiser can target on which kind of pages it wants its advertisements to be integrated.

Whenever a user clicks on an advertisement, the advertiser has to pay a fixed amount of money. From now on, it will be called the *Cost-per-Click (CPC)*. The CPC is determined by a *generalized second-price auction*: advertisers each offer to pay a certain amount of money per click for a set of keywords. The advertiser with the highest bid (i.e. the one that offers to pay the most for each click) wins the auction and hence the slot for its ad. This model is actually a very good approximation of the exact mechanism followed by major search-engines or social-networks on the Web (such as Yahoo, Google or Facebook).

In the Trading Agent Competition [5], the universe of keywords is represented by the combinations of three different manufacturers selling three different product components. An agent can decide to bid on either the manufacturer, the product, both or none of them. The more specific an ad is (e.g. the ad targets users searching for "Apple, Computer"), the bigger the chance that the user seeing the ad will actually click on it to get more information about the product (in this illustration, the user searches for "Apple, Computer" on its web engine and sees an advertisement for Apple computers on the result page). But the less users will see the ad (indeed, only those users who searched for those exact keywords will see it). Inversely, the less specific an ad is (i.e. no keyword is specified), the smaller the probability that the user will be interested but the more users will see the ad. Choosing the focus of an ad therefore is important to maximize its utility.

During a *game* (i.e. a simulation representing the evolution of a fictive market for a few days), agents have to bid for each possible set of keywords each day. Eight agents compete in a game. For each query, each day, the five highest bidders win a slot for an ad (i.e. their ad will appear on the associated page in the decreasing order of the bids). Up to a point, the lowest the position (the highest bid gets the lowest position, which is the first place), the highest the probability to reach users. But maybe the lowest position is not the best: imagine the case where the first position is three times more expensive than the second position, but both appear at the very top of the page. Is it really worth paying for the first place?

Finally, when an ad appears on a page, the probability that a user actually clicks on the ad is called the

Jonas Vautherin is in Federal Institute of Technology (EPFL), Lausanne

Click-Through Rate (CTR). The action of following (clicking) an ad and then buying a product on the advertisers page is called a *conversion*. Trivially, an advertiser wants to maximize the number of conversions (which represent benefits) with respect to the number of clicks (which represent costs).

B. Previous Work

All the agents that once participated in the tournament are available on the repository of the Trading Agent Competition[1]. A few of them were used in the occasion of this project to run the multiple simulations. However, the study of those more advanced strategies focused on the team which won every year (from Austin, Texas, [1]) and on the team from Brown University, Rotterdam ([2]). The former team explicitly uses particle filters while the latter only mentions machine learning algorithms. Nevertheless, the team from Rotterdam makes a very interesting point when it comes to the optimization strategies.

Indeed, they differentiate two kinds of strategies: one kind is *model-light*, the other is *model-heavy*, in reference to each strategy's dependance on the prediction models. The first model-light strategy they implement is called `EquateROI` (it will later be referred to as "**the** *model-light strategy*") and was implemented in this project (more details are given in Section III). This strategy has been previously analyzed in the context of online advertisement in [4].

In the model-heavy strategy, they solve the problem of optimization by classifying it as a *multiple-choice Knapsack problem* (MCKP). The idea is the following: given that one can get really accurate predictions, which one of the 5 slots would give the highest utility? This problem was mentioned before because winning the first position is not always the best idea. They hence predict the behavior of the other agents to estimate their future bids, solve the system using MCKP and choose the best bid according to this. It apparently works well.

Apart from those two teams, more modest student semester projects took place at EPFL during the last semesters. When one project studied particle filters ([9]), Kalman filters were concluded as more efficient for this application in [10].

C. Organization of the Paper

The first and main part of this project was - after having studied a bit the two previously mentioned teams - to create a testing environment which would be convenient to test the different strategies. Once it was done, several games using the model-light strategy were run and analyzed and, finally, performances were compared to those of a few other model-heavy strategies.

Section II describes in some details the testing environment which was created to analyze the strategies in a systematic way.

Section III depicts the actual strategy and Experiment setup. Results are then presented and analyzed in Section IV.

[1] `http://www.sics.se/tac/showagents.php`

II. Testing Environment

The course of a single game depends on multiple factors, may they be due to the randomness of the population or to the influence of other agents. Hence, one cannot rely only on a few games to conclude anything; the performances of a strategy strongly vary from one game to another. For this reason, it is important to run as many games as possible in order to get a tendency before drawing conclusions about a strategy. A testing environment is required to automatize all the tasks necessary to achieve this. Multiple scripts created in the occasion of this project allow to create competitions on the server and treat the results from the parsing to the actual plotting of graphs. Because of the structure of the environment offered by the University of Michigan for this annual competition, the agent code also needs to be adapted to the testing environment. A more complete documentation is given as an appendix.

III. Experiment

A so-called *model-light strategy* - a strategy that was not using any prediction algorithm - was implemented and run with different parameters in order to quantify their incidence on the results. This strategy inspires from [2] and is called `EquateROI`. It basically only relies on the conversion probability $PrConv$. Different properties such as the global utility (profit), global cost, the total number of impressions, clicks and conversions, the click-through rate, the conversion rate and the return on investment were considered and their behavior will be presented in Section IV.

Let's now define more precisely **Algorithm 1**, which is the one used by `EquateROI`. Using the models presented in [2], we define the cost, revenue and sales as follows:

$$cost_q(b_q) = nbClicks \cdot CPC_q \quad (1)$$

$$rev_q(b_q) = sales_q(b_q) \cdot USP_q \quad (2)$$

$$sales_q(b_q) = nbClicks \cdot PrConv_q \quad (3)$$

assuming a bid $b_q \in \mathbf{R}$ on query $q \in \mathbf{Q}$.

Moreover, we define the *return on investment* (ROI) as follows:

$$ROI_q(b_q) = \frac{rev_q(b_q) - cost_q(b_q)}{sales_q(b_q)} \quad (4)$$

According to [2] and the equimarginal principle depicted in [3], utility (profit) is maximized when the ROI is equated across all possible queries. Consequently, we can consider a fixed ROI value which we will call $targetROI$; it is our desired ROI.

With respect to equations (1), (2), (3) and (4), we can derive:

$$\begin{aligned} ROI_q(b_q) &= \frac{rev_q(b_q) - cost_q(b_q)}{sales_q(b_q)} \\ &= USP_q - \frac{CPC_q}{PrConv_q} \end{aligned} \quad (5)$$

and, trivially:

$$CPC_q = (USP_q - ROI_q(b_q))PrConv_q \quad (6)$$

Knowing USP_q and $PrConv_q$, we can replace $ROI_q(b_q)$ by the *desired* $targetROI$ and we get an estimation of CPC_q. We choose the bid as being a slightly different version of CPC_q, being $CPC_q + \epsilon$. Trivially, the higher the $targetROI$, the lower the bid and the smaller the probability to win a slot. Therefore, increasing the $targetROI$ will result in a decrease of our sales and inversely.

One should still introduce a last constraint to the problem: the *distribution capacity* (denominated C). Each agent is given a distribution capacity which is *the maximum number of products an agent can sell in a given period before getting penalty on the sales*. It can be formalized as follows:

$$\sum_{q \in Q} sales_q(b_q) \leq C \quad (7)$$

As a conclusion, the number of sales depends on the $targetROI$ and is bounded by C. The idea of the strategy is hence to find the $targetROI$ resulting in a number of sales close to C. This is formalized in **Algorithm 1**: as long as the number of sales is lower than the capacity, we increase $targetROI$ using a factor called INC_ROI. When the number of sales exceeds the capacity, $targetROI$ is decreased using the same factor.

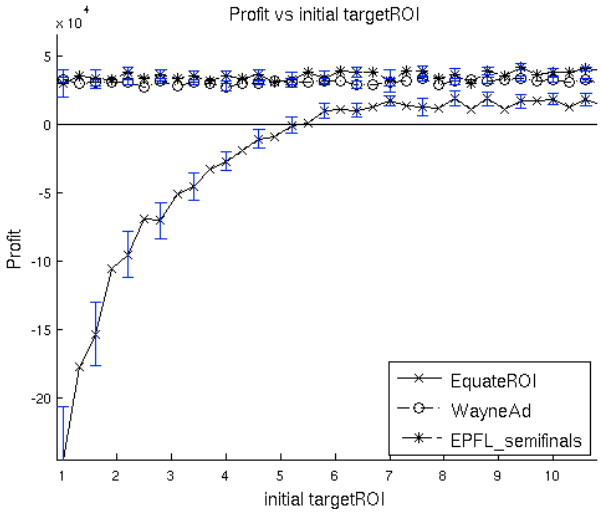
Algorithm 1 EquateROI [2]. C *represents the distribution capacity (i.e. the maximum number of products an agent can sell for a day),* d *is the day.*

Input: $sales(d), targetROI(d), C$
Output: $\forall q, bid_q$
  if $sales(d) > C$ **then**
    $targetROI(d+1) \Leftarrow targetROI(d) * INC\_ROI$
  else if $sales(d) < C$ **then**
    $targetROI(d+1) \Leftarrow targetROI(d) / INC\_ROI$
  end if
  for all q **do**
    $CPC_q \Leftarrow (USP_q - targetROI(d+1)) * PrConv_q$
    $bid_q \Leftarrow CPC_q + \epsilon$
  end for

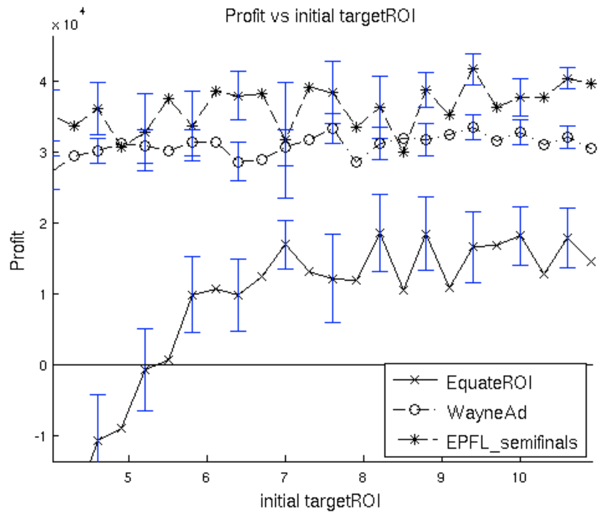
One can quickly realize that the `initial targetROI` and `INC_ROI` are the two parameters that potentially greatly influence the behavior of the algorithm. Consequently, several games were run with this strategy, always keeping the same configuration but varying each of those two parameters. Results obtained with this procedure are shown in Section IV. From now on, the terms *targetROI* and *initial targetROI* will be used interchangeably to refer to the *initial targetROI*.

IV. RESULTS

Two main experiences were conducted, basically studying the influence of two parameters: the *initial targetROI* and the increment of the *targetROI* (*INC_ROI*). The very same group of eight agents always participated to the games: twice *EPFL_semifinals [8]*, three times *WayneAd* and twice *Schezaml* [11]. Only one instance of **EquateROI** was used. For the experiment making the initial *targetROI* vary, more than 50 games where run for each variation of the parameter, for a total of around 2500 games (one game representing 60 days). When it comes to the increment (*INC_ROI*), more than 60 games where run for each variation of the parameter, for a total of around 1500 games (one game representing 60 days as well). The confidence interval appearing on all the graphs is always 0.95. In the graphs presented in this section, the *distribution capacity* of the agent corresponding to the curve always is 450. It



(a) Complete graph



(b) Zoom on the zone of "positive-profit" for **EquateROI**

Fig. 1. Comparison of the profit between different agents while the initial *targetROI* of **EquateROI** varies. An overall graph shows that EquateROI seemingly doesn't have a strong influence on the competing agents, and a zoom on the zone of profit allows a more accurate comparison.

should be noticed that when the distribution capacity of one agent is 450, it is not necessarily the case for the other agents. This is due to the implementation of the server, and should not bias the results. Finally, when not used as the varying parameter, the *initial targetROI* is set to 8.2 and the *INC_ROI* is set to 1.05.

A. *Initial targetROI*

Let's now look more precisely at some of the most relevant results obtained for the experiment with *targetROI*. Firstly, let's observe the evolution of the profit as shown in Fig. 1(a). With a small *initial targetROI*,

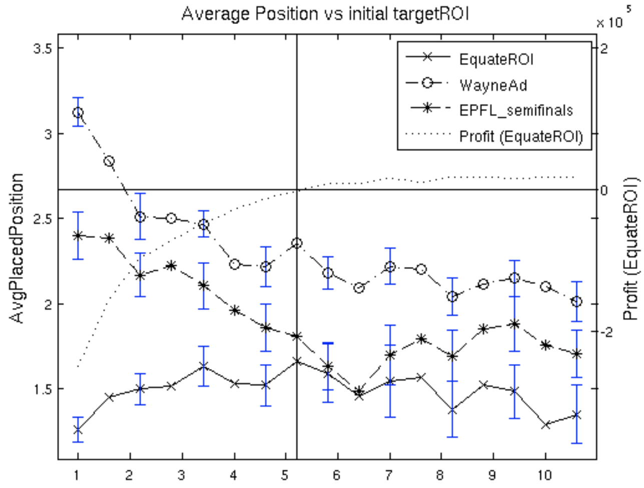


Fig. 2. *Average Placed Position* versus the initial *targetROI* of **EquateROI**. Comparison with other agents. The profit of EquateROI and its intersection with the null-axis are shown for interpretation purpose.

EquateROI loses money. By increasing it, it quickly goes up to a positive profit. Once the area of positive profit is reached, the profit stops increasing, which seems logical: it is easier to lose more than to win more. We can hence conclude that a minimum value of *initial targetROI* is required for the strategy to perform good.

Interestingly, the profit of the competing agents is not influenced by the profit - or even the huge losses - of this one. Fig. 1(b) is a zoom on the area where all three agents make profit, in order to compare their performances. *EPFL_semifinals* clearly outperforms both of the others, also with the distribution capacity set to 300 and 600 (those results are not presented in this paper).

Another interesting point is the average position of the agents with respect to the bids. As previously explained, there are five slots for advertisements each day, and getting the first position is not necessarily the best idea. Fig. 2 presents the average position of the agents versus the *targetROI*. The profit of **EquateROI** is shown as well for interpretation purpose. It should be noticed that the average position of both competing agent goes down while **EquateROI**'s profit goes up, and stabilizes when it reaches the positive-profit area. This is probably explained by the fact that the model-light strategy always gets the first position when it loses money (the bids are way too high when *targetROI* is low) and not when it makes profit.

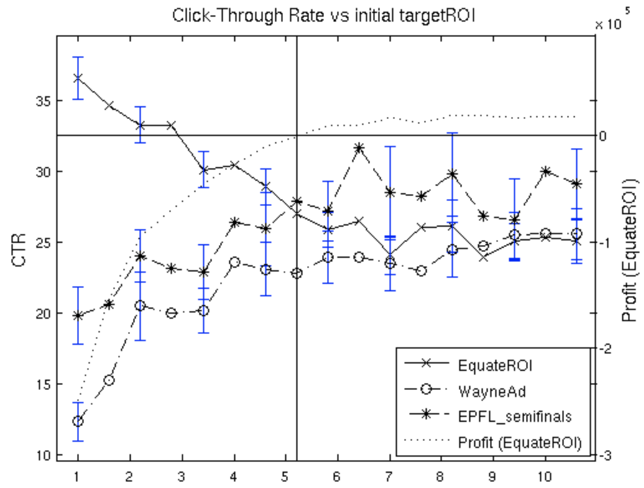


Fig. 3. *Click-Through Rate* versus the initial *targetROI* of **EquateROI**. Comparison with other agents. The profit of EquateROI and its intersection with the null-axis are shown for interpretation purpose.

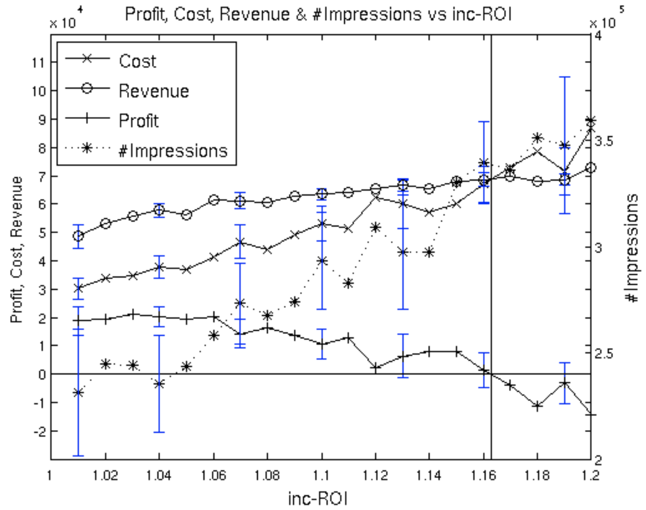


Fig. 4. Comparison of the *Cost*, *Revenue*, *Profit* and *number of Impressions* for **EquateROI** when the increment parameter "*Inc-ROI*" varies. An horizontal line shows the 0 for the *Cost*, *Revenue* and *Profit* when a vertical line enlightens the fact that the *Profit* is null when the *Cost* equals the *Revenue*.

Another relevant point is that *EPFL_semifinals* always gets a better position than WayneAd and apparently outperforms it.

When it comes to the Click-Through Rate, same kind of observations can be made. *EPFL_semifinals* always is higher than WayneAd. This is probably partly due to the average position, but maybe also by the choice of focused/general ads depending on the keywords. Both CTR increase as long as the CTR of **EquateROI** decreases and stabilize afterwards, and so does the number of conversions (not shown here). One should also be aware that, interestingly, while the number of impressions of the model-light strategy decreases with the increase of *targetROI* (because bids become lower), the number of impressions of the other competing agents does not have a clear tendency, but always is highly variable. One could therefore conclude that the "quality" of the ads increases when **EquateROI** gets a "rational behavior" and stops losing so much money.

B. Value of the increment (INC_ROI)

The *initial targetROI* value is important because the furthest it is from a value where a positive profit can be earned, the longest it will be to reach this value. But this time also depends on the *INC_ROI* parameter. The bigger the increment, the quicker we "move" on the scale of the *targetROI*, but at some point we could simply "miss" the ideal value by making too big "jumps".

As it can be seen on Fig. 4, the bigger the increment, the smaller the profit. Both the revenue and the cost increase with the increment (because the number of impressions increases as well), but the cost increases faster than the revenue, which is a bad point. Consequently, the smaller the increment, the better.

But this graph shows the evolution when *targetROI* equals 8.2, which has been shown as being a good initial value. With a too low *initial targetROI*, a bigger increment could be necessary to quickly reach the so-called "positive-profit area".

V. Conclusion and Further Work

The Trading Agent Competition / Ad Auctions allows to compare one's strategy to others in an international range. This is really constructive considering the complexity of the problem. However, it has limitations. For instance, the competition creates its own model of the world which is not necessarily always accurate. But it is still a good first test and training ground for a strategy before trying to study it in the real world.

Now that the testing environment has been implemented and allows to run and analyze big competitions, further work should definitely include other, model-heavy strategies. Nevertheless, **EquateROI** could still be used to study the impact of the focus/general ads and how to choose the focus of an ad. Moreover, the *INC_ROI* parameter could be variable instead of constant in order to move faster when the profit is highly negative and slower when the bids seem to be good.

APPENDIX

A. Linked documents

- Complete documentation of the testing environment
- Brief manual of how to create a new agent
- Overview of the LogParser API

B. Extra analysis: particles filter

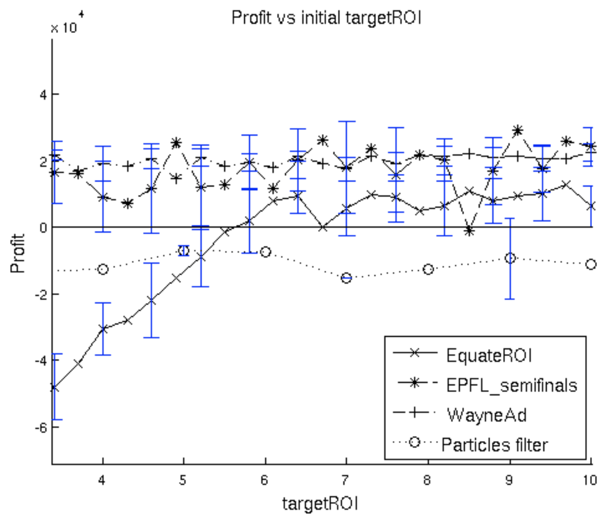


Fig. 5. Profits comparison between the tested agents. *"Particles filter"* is the agent from [9]. This graph shows the results when **the distribution capacity is 300**.

Three possibly relevant graphs could not be integrated into the report and where therefore included here at the end of the document. They concern the models made in [9], using **particle filters**.

Because of some technical issues, it hasn't been possible to run this agent on enough simulations to get precise results and only 100 games where considered, shared between the three different *distribution capacities*. Consequently, more measures should be done before getting definitive conclusions. As observed in the more accurate experiments that were led with other agents, the profit of the competing agents apparently is constant versus the *initial targetROI* of **EquateROI**. This is verified for the particle filters in those graphs. Interestingly, the distribution capacity has a drastic influence on the profit of this agent. With the smallest one (Fig. 5), the profit is even negative. It is almost null with the middle distribution capacity (Fig. 6) and slightly positive with the highest one (Fig. 7). For all of them, it is not really clear if the strategy studied in [9] performs better than the model-light strategy presented in this paper.

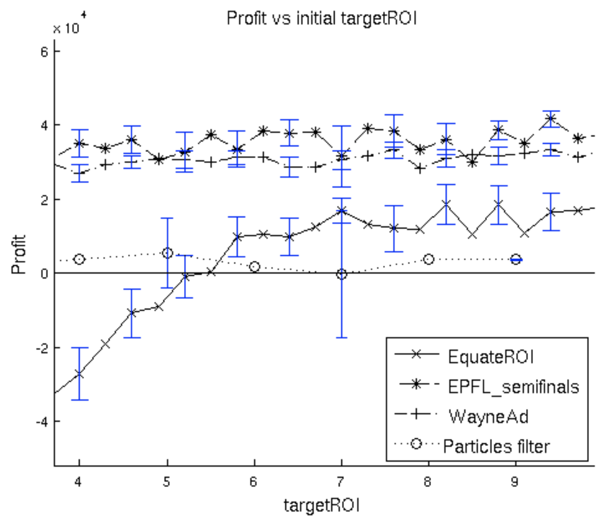


Fig. 6. Profits comparison between the tested agents. *"Particles filter"* is the agent from [9]. This graph shows the results when **the distribution capacity is 450**.

Even though a deeper experiment wasn't made to prove these observations, it is considered as probably worth mentioning it here.

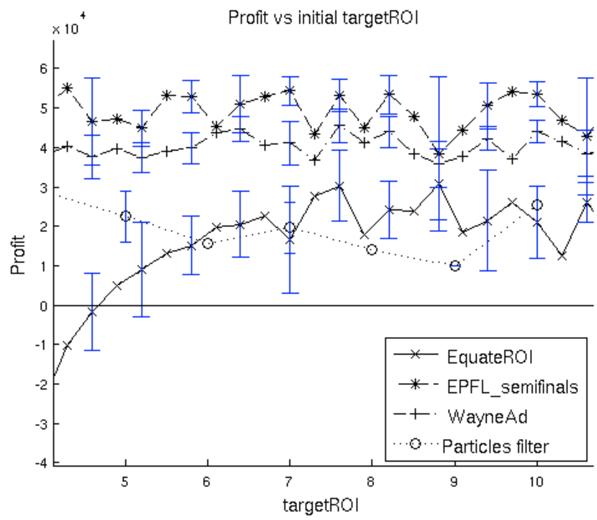


Fig. 7. Profits comparison between the tested agents. *"Particles filter"* is the agent from [9]. This graph shows the results when **the distribution capacity is 600**.

REFERENCES

[1] D. Pardoe, D. Chakraborty, and P. Stone. TacTex09: A champion bidding agent for ad auctions. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, 2010.

[2] J. Berg et al. A First Approach to Autonomous Bidding in Ad Auctions, In *ACM*, 2010

[3] T. Odean, V. Naroditskiy, A. Greenwald, and J. Donaldson. Marginal bidding: An application of the equimarginal principle to bidding in tac scm. In *Workshop on Trading Agent Design and Analysis*, July 2007.

[4] C. Bors et al. Bid optimization in online advertisement auctions. *Microsoft Research, Redmond, WA, USA*.

[5] Patrick R. Jordan et al. The Ad Auctions Game. *University of Michigan, Ann Arbor, MI 48109-2121 USA*, 2010.

[6] S. Lahaie et al. Sponsored search auctions. In *T. Roughgarden, N. Nisan , E. Tardos, and V. Vazirani, editors, Algorithmic Game Theory. Cambridge University Press*, 2007.

[7] M. Sanjeev Arulampalam et al. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. In *IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 50, NO. 2*, February 2002.

[8] L. Cigler. Semester Project: Bidding strategy for the Trading Agent Competition. *Swiss Federal Institute of Technology, Lausanne*.

[9] E. Mirza, L. Cigler. Semester Project: Bidding Agent for Advertisement Auctions. *Swiss Federal Institute of Technology, Lausanne*.

[10] ?, L. Cigler. Semester Project: Improve Prediction Model of Trading Agent. *Swiss Federal Institute of Technology, Lausanne*.

[11] TAC agents repository.
`http://www.sics.se/tac/showagents.php`

[12] TAC/AA webpage
`http://aa.tradingagents.org/`