A Bidding Agent for Advertisement Auctions: An Overview of the CrocodileAgent 2010

Irena Siranovic, Tomislav Cavka, Ana Petric and Vedran Podobnik
University of Zagreb, Faculty of Electrical Engineering and Computing
Zagreb, Croatia
irenasiranovic@gmail.com, {tomislav.cavka, ana.petric, vedran.podobnik}@fer.hr

Abstract

Sponsored search is a popular form of targeted online advertising and the most profitable online advertising revenue format. Online publishers use different formats of unit price auctions to sell advertising slots. In the Trading Agent Competition Ad Auctions (TAC/AA) Game, intelligent software agents represent publishers which conduct keyword auctions and advertisers which participate in those auctions. The publisher is designed by game creators while advertisers are designed by game entrants. Advertisers bid for the placement of their ads on publisher’s web page and the main challenge placed before them is how to determine the right amount they should bid for a certain keyword. In this paper, we present the CrocodileAgent, our entry in the 2010 TAC AA Competition. The agent’s architecture is presented and a series of controlled experiments are discussed.

1 Introduction

Internet advertising provides significant income streams for online publishers. Revenue of major search engines such as Google1, Yahoo!2 and MSN3 amounts to tens of billions of dollars annually (e.g., in 2010, Google reported total advertising revenues over USD $28 billion4). According to a report published by the Interactive Advertising Bureau5 and Price-waterhouse Coopers LLP6, sponsored search is the most profitable online advertising revenue format which accounted for 47% of the total Internet advertisement revenue in the USA in the first half of 20107. In sponsored search (i.e., keyword advertising) publishers (i.e., search engines) use different formats [Feng et al., 2007] of unit price auctions (e.g., keyword auctions) to sell advertising slots (i.e., positions in the list that contains search results) [Chen et al., 2010].

1http://www.google.com/
2http://www.yahoo.com/
3http://www.msn.com/
4http://investor.google.com/financial/tables.html
5http://www.iab.net/
6http://www.pwc.com/
7http://www.iab.net/media/file/IAB_report_1H_2010_Final.pdf

In a keyword auction advertisers bid for the placement of their ads (i.e., rank of the ad in the results of the sponsored search) which are then displayed on publisher’s web page. The format of sponsored search results is very much alike the format of generic search results. Usually, it is comprised of the title of the ad, short description and a hyperlink to the advertiser’s web page or the web page of the advertised product. An ad is chosen for a specific keyword(s) from the user’s query, thus targeting users interested in advertiser’s products.

A publisher conducts keyword auctions and solicits bids. When a user submits a query containing one or more keywords, sponsored ads are displayed to the user alongside the results of a generic search mechanism. At the end of an auction, the publisher ranks advertisers’ bids (i.e., determines the placement of ads and cost-per-click (CPC) of those ads) [Easley and Kleinberg, 2010]. CPC is the price that an advertiser pays to the publisher each time its ad is clicked on.

According to earlier studies on user behaviour (e.g., [Joachims et al., 2005]) the higher the position of the search result (e.g., ad, document) is, the more likely the users will click on it. This phenomenon, where the probability that the result will be clicked depends not only on its relevance, but also on the position of the result, is known as the position bias. Several models of the position bias have been proposed and the cascade model gave the best explanation for position bias in early ranks [Craswell et al., 2008]. However, latest research has shown that 46% of users do not click sequentially (i.e., start from the best ranked result and continue to lower ranked ones) and 57% of them do not behave as suggested by the cascade model (i.e., first click on higher and afterwards on lower positioned results) [Jeziorski and Segal, 2010].

A challenge placed upon publishers evolves around the selection of a mechanism which will result with highest profits. There are two frequently used mechanisms for ranking solicited bids: i) rank-by-bid, and, ii) rank-by-revenue. As the name of the rank-by-bid mechanism states the bids are sorted in a descending order (i.e., bids offering higher CPC get higher ranked positions), while the rank-by-revenue mechanism multiplies the offered CPC with the ad’s expected click-through rate (i.e., the percentage of users that will click on the ad once it is displayed to them) and afterwards sorts offers in a descending order of the calculated product [Lahaie, 2006]. In addition, some mechanisms offer advertisers the possibility to target a certain group of users by specifying the context.
for viewing ads (e.g., user’s location, time of day, etc.) and to control exposure by limiting the number of times an ad should be displayed to users [Lahaie et al., 2008].

From the advertisers’ point of view, the question is how to determine the right amount they should bid for a certain keyword(s) since the probability that their ad will be better ranked than other ads rises as they place higher bids [Varian, 2009]. The complexity of the answer to this question increases as the availability of information about user behaviour, as well as bidding behaviour of other advertisers decreases.

The paper is organized as follows. Section 2 describes the Trading Agent Competition / Ad Auctions (TAC/AA) game characteristics. A brief overview of research in the area of TAC/AA bidding strategies is given in Section 3. Section 4 presents the CrocodileAgent 2010, our entrant for the 2010 TAC/AA Tournament. Section 5 presents the conducted experiments and the obtained results, while Section 6 concludes the paper and gives an outline for future work.

2 Trading Agent Competition / Ad Auctions

Researchers test advertisers’ bidding strategies by using the designed market simulators which provide a risk free environment [Acharya et al., 2007]. The TAC/AA game [Jordan et al., 2009; Jordan and Wellman, 2010], which was released in 2009, is based on such a market simulator. The game enables advertisers to bid on multiple queries and to define budget constraints in an information-lacking and competitive environment. Eight intelligent program agents which represent online advertisers participate in the game. Each advertiser sells 9 different products which are specified by a manufacturer (Flat, Lioneer and PG) and a component (TV, Audio and DVD). Furthermore, each advertiser is specialized for one manufacturer and for one component type.

The ad ranking mechanism varies between rank-by-bid and rank-by-revenue [Lahaie and Pennock, 2007] and it is chosen at the beginning of each game. Users can generate 16 different queries: specifying both manufacturer and component (i.e., F2 level query), specifying only manufacturer or component (i.e., F1 level query), or neither (i.e., F0 level query). Correspondingly, conversion probability increases with more keywords specified. Users can find themselves in one of the following states: i) non-searching, ii) searching, and, iii) transacted; while user population in each state (and sub-state) is modelled as a Markov chain.

In the TAC/AA game scenario, when a user submits a query, an auction for the given keyword(s) starts. The auction is an instance of the repeated generalized second price auction (i.e., the price that the advertiser pays for the position of its ad is determined by the price which did the winner of the next-best position offered in its bid) and the first position is allocated to the bidder with the highest bid. An advertiser sends a bid bundle (i.e., one bid for each query class) to the publisher every day. A bid consists of: i) the CPC an advertiser is ready to pay, ii) the chosen ad which can be generic or targeted (i.e., specified manufacturer and/or component), iii) budget limits for each query class, and, iv) budget limit for all queries altogether for the following day (optional). The publisher uses the information from the bid when it runs ad auctions for received user queries. Advertisers receive daily reports about the outcomes of the prior (i.e., two days old) auctions and use those information to generate new bids. Daily reports include: i) query report, ii) account status report, and, iii) sales report. The game lasts 60 virtual days.

3 Related Work

The bidding strategy for keyword auctions has been a great challenge for researchers that conducted various simulations and empirical analyses on this matter. [Berg et al., 2010] have presented autonomous bidding strategies for ad auctions based on click probability and CPC estimations. The algorithms for bid optimization come in two parts: i) rule-based algorithms, and, ii) greedy multiple choice Knapsack algorithms. [Pardoe and Stone, 2010] have shown a particle filter that can be used for estimating other agents’ bids given a periodic ranking of their bids. The particle filter uses bidding behaviour models of other advertisers. Since the information revealed about competitor advertisers is limited, they have shown how such models can learn from past bidding data. [Cigler, 2009] has described several bidding strategies (Return-on-investment (ROI), Knapsack ROI, Balanced Best-response, online Knapsack) and evaluated their performance empirically and against omniscient strategy. The best performing strategies were ROI and Balanced Best-response. Furthermore, Cigler presented a new profit maximizing strategy for multiple keyword ad campaigns. The strategy takes into account the budget constraint which has shown to be a competitive strategy particularly for small budgets.

4 CrocodileAgent 2010

With an intention to better investigate strategic approaches to ad auctions mechanisms, an advertiser agent CrocodileAgent has been designed. A bidding strategy was chosen among several strategies (profit maximization, linear regression [Witten and Frank, 2005] and analysed through controlled experiments. The strategy was compared with already existing in literature and practice by running TAC/AA games. As a result, the CrocodileAgent setup which produced most satisfactory performance was identified. The Crocodile Agent’s bidding strategy for participating in ad auctions can be divided into three logical segments: i) the ad generator, ii) the CPC generator, and, iii) the daily spend limit generator. Figure 1 shows the bidding model.

4.1 The ad generator model

The algorithm for ad generation is shown in Algorithm 1. The chosen method implements ad generation based on query focus levels. This method was chosen by analysing final results in game simulations with several different implemented algorithms for ad generation. The observed conversion probability is the highest when user submits a query with focus level F2. Since there are 9 different products and each user has a preference for just one, it is highly probable that a user will click on the ad that matches the query when he/she submits an F2 query. Furthermore, we have concluded that if an agent
Figure 1: CrocodileAgent’s bidding model

provides a generic ad for submitted F2 query, the probability that user clicks on the ad decreases significantly, especially if competitor agents choose a targeted advertisement.

Algorithm 1: CrocodileAgent’s ad generation algorithm

```plaintext
if query focus level = F0 or F1 then
    generated ad = generic
else
    generated ad = targeted (F2 manufacturer, F2 component)
end
```

For a submitted query F1, CrocodileAgent generates generic ads - if CrocodileAgent generated targeted ad, it would have to decide which component or manufacturer to add alongside specified keyword (i.e., manufacturer or component, respectively). The negative aspect of the latter approach is that targeted ad has to match user preference in order to agent receive a conversion (a probability that generated ad matches user preference is 1/3). If the ad does not match, click probability decreases according to the targeting factor [Jordan et al., 2009], and vice versa. Moreover, if the ad does not match user preferences, the user may click on the ad, but will not convert and thus increase only the CPC. On the other hand, the benefit of the approach that results with targeted ad manifests in a case when ad matches submitted query - positive aspects of this approach are: i) additional profit gained by component (CSB) or manufacturer specialization bonus (MSB), ii) increased odds of converting based on CSB, and, iii) increased targeting effect.

4.2 The CPC generator model

The CPC bid is defined for every query type with an intention to maximize the profit (profit = revenue - cost). Taking into consideration the two days delay of reports, the algorithm for generating the CPC for the first two days of the game contains fixed values of CPC. Values for all query types are defined based on matches between component and manufacturer in a query and agent specialties as shown in Algorithm 2.

The more accurate the match is, the higher is the bid. Additionally, values are scaled by agent’s capacity (i.e., the bid is slightly decreased in cases of medium and low capacity). Definitions of matching values are listed in Table 1, while Table 2 contains the parameter values. All parameter values were optimized using the heuristic approach. The decisions were based on conducted game simulations (locally and in the TAC/AA 2010 qualifying rounds) by analysing and comparing the CrocodileAgent’s profit for different sets of parameters. The used decision making method could be best compared to the bisection method.

A method for defining CPC bids in the rest of the game is based on calculating the conversion rate (the ratio of average

Algorithm 2: CrocodileAgent’s initial bidding algorithm

```plaintext
result = match between query and agents specialties;
switch result do
    case miss
        bid = α;
    case miss-neutral
        bid = β;
    case miss-hit
        bid = γ;
    case neutral
        bid = δ;
    case neutral-hit
        bid = ε;
    case hit
        bid = ξ;
endsw
where β < γ < α ≤ δ ≤ ε < ξ;
```

Table 1: Definitions of specialty matching values

<table>
<thead>
<tr>
<th>Value</th>
<th>Query type</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>miss</td>
<td>F2</td>
<td>Component and manufacturer from query do not match agent’s specialties.</td>
</tr>
<tr>
<td>miss-neutral</td>
<td>F1</td>
<td>Component or manufacturer from query does not match agent’s specialties, second keyword is null.</td>
</tr>
<tr>
<td>miss-hit</td>
<td>F2</td>
<td>Component or manufacturer from query does not match agent’s specialties, second keyword is a match.</td>
</tr>
<tr>
<td>neutral</td>
<td>F0</td>
<td>A query without specified manufacturer and component.</td>
</tr>
<tr>
<td>neutral-hit</td>
<td>F1</td>
<td>Component or manufacturer from query matches agent’s specialties, second keyword is null.</td>
</tr>
<tr>
<td>hit</td>
<td>F2</td>
<td>Component and manufacturer from query are agent’s specialties.</td>
</tr>
</tbody>
</table>
received on the latest report. The bid is later adjusted depending
on the query focus level and the CrocodileAgent’s specialties. The
CPC generator model is shown in Algorithm 3. The last part of the CPC bidding model is adjusting the bid according
to the agent’s capacity (i.e., if the capacity is low, the bid
decreases).

In defining CPC bids the most important aspect is the
"quality" of an ad, which is measurable through agent’s profit.
Quality can be defined as an estimated value that a click event
will turn into conversion, which is based on previous report
data. Note that CrocodileAgent’s bidding strategy does not
gave the most satisfactory results in a conducted experiment.

### 4.3 The spend limit manager

#### General spend limit

Too many conversions lead to decrease of possible conver-
sions in the future due to stock shortage, as defined by
the game rules [Jordan and Wellman, 2010]. The stock manag-
ing policy is necessary in a scenario when a certain amount of
product must be immediately available in order to avoid profit loss. On the other hand, excessive product storage leads to
"dead capital". Additionally, the other driver for using spend
limits are the clicks of informational searchers (F0).

The general spend limit manager algorithm is shown in Al-
gorithm 4, while corresponding optimal parameter values are
listed in Table 4. In the first five days of the game the limit is
fixed and determined based on agent’s capacity. In the con-
trolled experiment we have concluded that the lack of spend
limit has the most significant (negative) impact on Crocodile
Agent’s profit when the capacity is low.
Algorithm 4: General spend limit manager

if first five days then
switch capacity do
  case low
    spend limit = spend limit low fixed;
  case medium
    spend limit = spend limit medium fixed;
  case high
    spend limit = spend limit high fixed;
endsw
else
switch capacity do
  case low
    spend limit = spend limit low;
  case medium
    spend limit = min(spend limit medium, custom profit);
  case high
    spend limit = min(spend limit high, custom profit);
endsw
end

Table 4: Optimal value of parameters for general spend limit

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td>spend limit low fixed</td>
<td>700</td>
</tr>
<tr>
<td>spend limit medium fixed</td>
<td>1000</td>
</tr>
<tr>
<td>spend limit high fixed</td>
<td>1200</td>
</tr>
<tr>
<td>spend limit low</td>
<td>750</td>
</tr>
<tr>
<td>spend limit medium</td>
<td>1150</td>
</tr>
<tr>
<td>spend limit high</td>
<td>1350</td>
</tr>
</tbody>
</table>

Query spend limit

The query spend limit manager defines a mechanism for optimal distribution of investments in order to maximize the profit. The mechanism ensures that the click number is limited based on the calculated quality of an ad. Considering the report delay, the spend limits for the first two days are fixed. They are generated based on matching the CrocodileAgent’s specialties with keywords in query. In the rest of the game spend limit is defined according to the click and conversion ratio, as well as previous profits. Low ratio corresponds to

Algorithm 5: Query spend limit manager

n_click = average number of clicks per day;
\( n_{conversion} \) = average number of conversions per day;
mod = \( n_{click}/n_{conversion} \);
result = a match between query and agent’s specialties;

if first two days then
switch result do
  case miss
    limit = a;
  case miss-neutral
    limit = b;
  case miss-hit
    limit = c;
  case neutral
    limit = d;
  case neutral-hit
    limit = e;
  case hit
    limit = f;
endsw
else
  if mod < ratio lower bound then
    limit = min (minimum limit, revenue/2);
  else if ratio lower bound \( \leq \) mod \( \leq \) ratio middle value then
    limit = min (minimum limit, revenue/3);
  else
    limit = default limit;
  end
  limit = SpecialtyMatchingLimit (result, limit)
end

function SpecialtyMatchingLimit (result, limit)

switch result do
  case neutral
    limit = neutral \cdot limit;
  case neutral-hit
    limit = neutral – hit \cdot limit;
  case hit
    hit : limit = hit \cdot limit;
endsw

the successful advertisement. Therefore, the lower is the ratio, the higher is the limit. The query spend limit manager algorithm is shown in Algorithm 5, while corresponding optimal parameter values are listed in Table 5.

Table 5: Optimal value of parameters for general spend limit

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>60</td>
<td>ratio lower bound</td>
<td>5</td>
</tr>
<tr>
<td>b</td>
<td>80</td>
<td>ratio middle value</td>
<td>10</td>
</tr>
<tr>
<td>c</td>
<td>100</td>
<td>minimum limit</td>
<td>20</td>
</tr>
<tr>
<td>d</td>
<td>120</td>
<td>default limit</td>
<td>40</td>
</tr>
<tr>
<td>e</td>
<td>140</td>
<td>neutral</td>
<td>1.10</td>
</tr>
<tr>
<td>f</td>
<td>180</td>
<td>neutral-hit</td>
<td>1.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hit</td>
<td>1.30</td>
</tr>
</tbody>
</table>

5 Controlled experiment

In order to evaluate performance of the CrocodileAgent, which placed 6th in the TAC/AA 2010 Competition Finals, an experiment was conducted by repeating games with fixed parameters. Based on the analysis of the results CrocodileAgent’s deficiencies were identified and guidelines for future improvements were set. The participants in the
experiment were the following agents which competed in the TAC/AA 2010 Competition TacTex, Mertacor, Schlemazl, CrocodileAgent, tauagent, EPFLAgent. Additionally, due to lack of TAC/AA 2010 agents in the official agent repository, two agents from TAC/AA 2009 Competition, AstonTAC and WayneAd, were included in experiment. The average results from 40 games played are shown in Table 6.

The games in the controlled experiment were configured to ensure fair capacity distribution among competing agents each agent played with high capacity ten times, twenty times with medium capacity and ten times with low capacity. In a specific game, two agents had low capacity, two had high capacity and four of them had medium capacity. The goal of the competition was to observe agents’ behaviour in respect with the assigned capacities. The results of these observations are shown in Figure 2.

In the graph from Figure 2 bars represent the ratio between the average result of a single agent in games with the specified capacity (i.e., high, medium and low) and the average result of the same agent in all games. We call this ratio the intra-agent relative profitability. On the other hand, the horizontal lines represent the ratio between average results of all agents in games with the specified capacity (i.e., high, medium and low) and the average results of all agents in all games. This measure represents the average intra-agent relative profitability of all agents. Finally, squares, triangles and diamonds represent the ratio between the average result of a single agent in games with the specified capacity (i.e., high, medium and low, respectively) and the average result of all agents in games with the same capacity. This graph allows us to compare single agent’s average profit achieved in games with different assigned capacities with: i) its average profit in all games, and, ii) the average profit of all agents’ in games with the same capacity. While the former measure enables us to compare the profitability of agent’s strategies in games with different capacities with the agent’s overall profitability, the latter measure provides relative benchmarking among different agents.

If we take a look at the average values of profits achieved in games with low capacity, we can notice that the CrocodileAgent has the highest intra-agent relative profitability among all agents in the controlled experiment. However, the CrocodileAgent has smaller intra-agent relative profitability than the average intra-agent relative profitability of all agents both in medium and high capacity games. At the same time, we can also notice that, when comparing the average score of different agents, the CrocodileAgent has 12% better score than all agents’ average in games with low capacity, while its performance in both medium and high capacity games is equal to all agents’ average.

Table 6: Average results in the conducted competition

<table>
<thead>
<tr>
<th>Position</th>
<th>Agent</th>
<th>Game score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>TacTex</td>
<td>57 848</td>
</tr>
<tr>
<td>2.</td>
<td>Mertacor</td>
<td>53 998</td>
</tr>
<tr>
<td>3.</td>
<td>Schlemazl</td>
<td>53 933</td>
</tr>
<tr>
<td>4.</td>
<td>CrocodileAgent</td>
<td>49 435</td>
</tr>
<tr>
<td>5.</td>
<td>tauagent</td>
<td>47 789</td>
</tr>
<tr>
<td>6.</td>
<td>AstonTAC</td>
<td>45 104</td>
</tr>
<tr>
<td>7.</td>
<td>EPFLAgent</td>
<td>44 179</td>
</tr>
<tr>
<td>8.</td>
<td>WayneAd</td>
<td>36 456</td>
</tr>
</tbody>
</table>
Figure 3: Profit distribution among query classes

From this analysis we can identify certain CrocodileAgent’s deficiencies improvement of those CrocodileAgent’s drawbacks in future versions could significantly increase its profits. Namely, we can conclude that the CrocodileAgent should examine the possibility of using other strategies for achieving profits in medium and high capacity games in order to increase its profit in those games. Possibly, such an approach could significantly (positively) affect the final result of the CrocodileAgent in the TAC/AA 2011 competition.

Another interesting fact we can learn from Figure 2 is that the relative intra-agent profitability of the TacTex agent, the best agent in competition, is approximately equal to the average intra-agent profitability of all competing agents (for all three capacity allocations). Furthermore, it is attention-grabbing to note that the WayneAd agent, who placed last in the competition, has the highest relative intra-agent profitability in games with high capacity. However, WayneAd’s weakest absolute results in medium and low capacity games are the reason for its the last place in the competition.

After analysing the impact of capacity on agents’ achieved profit, we have also analysed the correlation of the achieved profit and the query category. As mentioned earlier, there are three types of queries (i.e., F0, F1 and F2) that users generate. Each advertiser selects an ad for display for each query type, choosing between generic or targeted ad mentioning a particular product [Jordan et al., 2009]. The agents’ profits from all games in the experiment were grouped based on the type of user query. The graph presented in Figure 3 shows the distribution of profit achieved from transactions originating from different query types.

In the graph from Figure 3, the horizontal lines represent the ratio between agent’s average result for queries of specified type (i.e., F0, F1 and F2) and agent’s average result for queries of all types.

If we look at the average values of profits originating from different query types, we can notice that agents, who have better final result, also achieve higher profits on targeted ads. The last three agents in the competition AstonTAC, EPFLAgent and WayneAd achieved the lowest relative profits from focus level F2 queries and the highest relative profits from focus level F1 queries.

Expressed in percentages and in co-relation to average results for queries of all types, TacTex agent made 78% of its total profit from focus level F2 queries. Consequently, TacTex achieved 22% of its total profit from focus level F0 and focus level F1 queries. Agent Schlemazl, who has the highest relative score for F2 queries, achieved 86% of its total profit from focus level F2 queries.

Another interesting fact we can learn from Figure 3 is that overall profits are not highly correlated with the fraction of clicks received under manufacturer specialty (i.e., focus level F2-hit). Therefore, we conclude that shifting spend towards queries focusing on manufacturer specialty is not a guarantee of greater profitability for agents, despite bonuses they get if such transactions take place.

If we analyse profit distribution for CrocodileAgent, depending on the type of user query, we can notice that CrocodileAgent is achieving approximately 85% of its total profit from focus level F2 queries. This percentage is higher for CrocodileAgent than for two best-placed agents, TacTex and Mertacor. On the other hand, CrocodileAgent achieved lower relative profits than TacTex and Mertacor from focus level F1 queries. We can conclude that CrocodileAgent should try to redistribute a few percent of its profit obtained from F2-queries to F1-queries, while maintaining the share of F0-queries.
6 Conclusion and future work

Trading Agent Competition Ad Auctions (TAC/AA) Game enables scholarly community and advertising industry to analyse effects of various bidding strategies through simulation of sponsored search scenarios. The fact that sponsored search is the most profitable online advertising revenue format gives great importance to ad auction research as well.

In this paper, we presented bidding strategies of the CrocodileAgent, the representative of University of Zagreb in the TAC/AA 2010 Competition. Furthermore, we conducted controlled experiment with best-ranked agents from 2010 and 2009 TAC/AA Finals. Based on an analysis of the controlled experiment, we: i) explained some reasons why certain agents performed better than others, and, ii) identified mechanisms that should be improved in order to boost CrocodileAgent’s performance.

For future work we plan to enhance CrocodileAgent’s performance by implementing guidelines for improvements defined based on controlled experiment analysis. Namely, we will: i) redesign strategies for achieving profits in medium and high capacity games, and ii) redistribute a part of relative CrocodileAgent’s profit from F2-queries based profit to F1-queries based profit.

Acknowledgments

The authors acknowledge the support of research project “Content Delivery and Mobility of Users and Services in New Generation Networks” (036-0362027-1639), funded by the Croatian Ministry of Science, Education and Sports of the Republic of Croatia.

References


