Social Networks

Luigi Giugliano[1], Steven Rosario Sirchia[1]

[1]Università degli studi di Salerno

July 1, 2016

Introduction

The purpose of our work is to test different algorithms in the three fundamental areas for assembling any search engine offering a Sponsored Search system:

- **Ranking** of web documents
- **Matching** of words inside documents
- **Auctions** for acquiring advertisement slots.

We will briefly talk about the proposed algorithms, and then compare running times and results obtained from their execution more in detail, suggesting what combination of algorithms seems to be the best for realizing a new search engine.

Overview

Overview

Page Rank

Page Rank

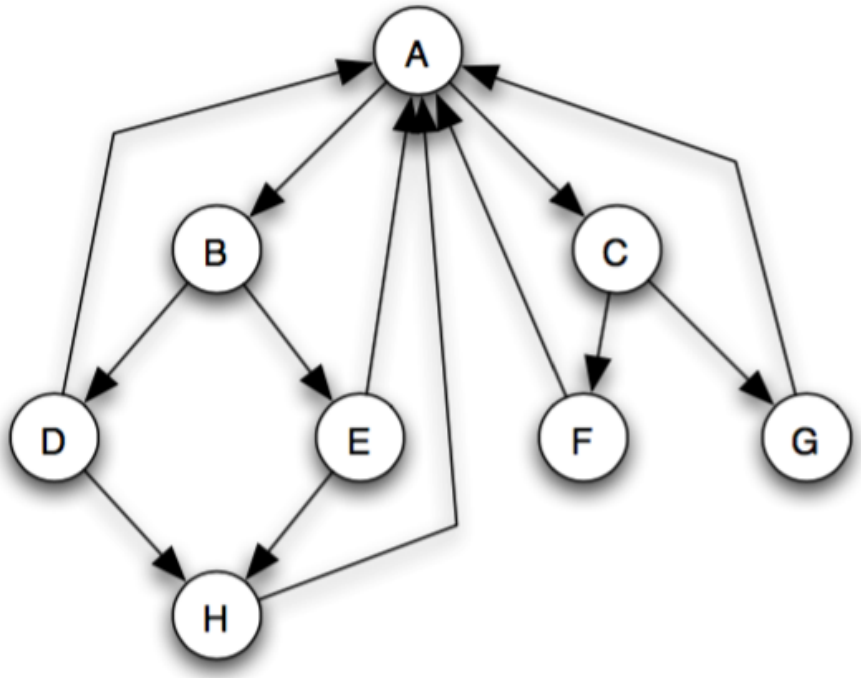
The intuition behind *Page Rank* is:

> "a page is important if it is cited by other important pages".

This intuition rises from the usual endorsement mode, for example, among academic or governmental pages, among bloggers, or among personal pages more generally. It is also the dominant mode in the scientific literature.

Algorithm

We can think of PageRank as a kind of "fluid" that circulates through the network, passing from node to node across edges, and pooling at the nodes that are the most important.



Algorithm Steps

- In a network with n nodes, we assign all nodes the same initial PageRank, set to be 1/n.

Algorithm Steps

- In a network with n nodes, we assign all nodes the same initial PageRank, set to be 1/n.
- We choose a number of steps k.

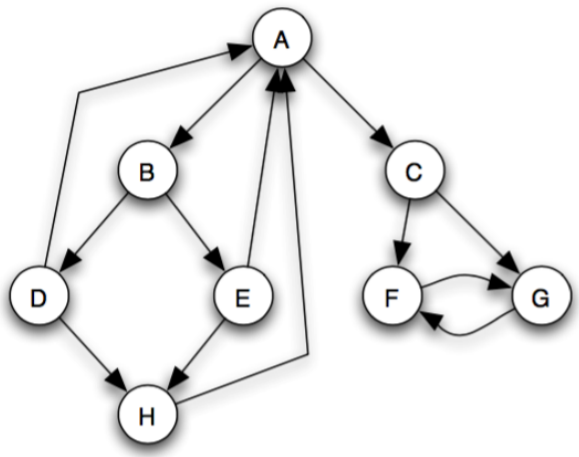
Algorithm Steps

- In a network with n nodes, we assign all nodes the same initial PageRank, set to be 1/n.
- We choose a number of steps k.
- We then perform a sequence of k updates to the PageRank values, using the following rule for each update:

> **Basic PageRank Update Rule**: Each page divides its current PageRank equally across its out-going links, and passes these equal shares to the pages it points to. (If a page has no out-going links, it passes all its current PageRank to itself.) Each page updates its new PageRank to be the sum of the shares it receives.

The "Wrong" nodes

There is a difficulty with the basic definition of PageRank, however: in many networks, the "wrong" nodes can end up with all the PageRank.



The Wrong nodes are a small sets of nodes that can be reached from the rest of the graph, but have no paths back.

Scaled PageRank

We can use the mechanism of fluid presented before: there is a counter-balancing process preventing that all the water stands only on downhill places on the earth.

Scaled PageRank Update Rule

First apply the Basic PageRank Update Rule.

Then scale down all PageRank values by a factor of s, shrinking the total from 1 to s.

We divide the residual $1 - s$ units of PageRank equally over all nodes, giving$(1 - s)/n$ to each.

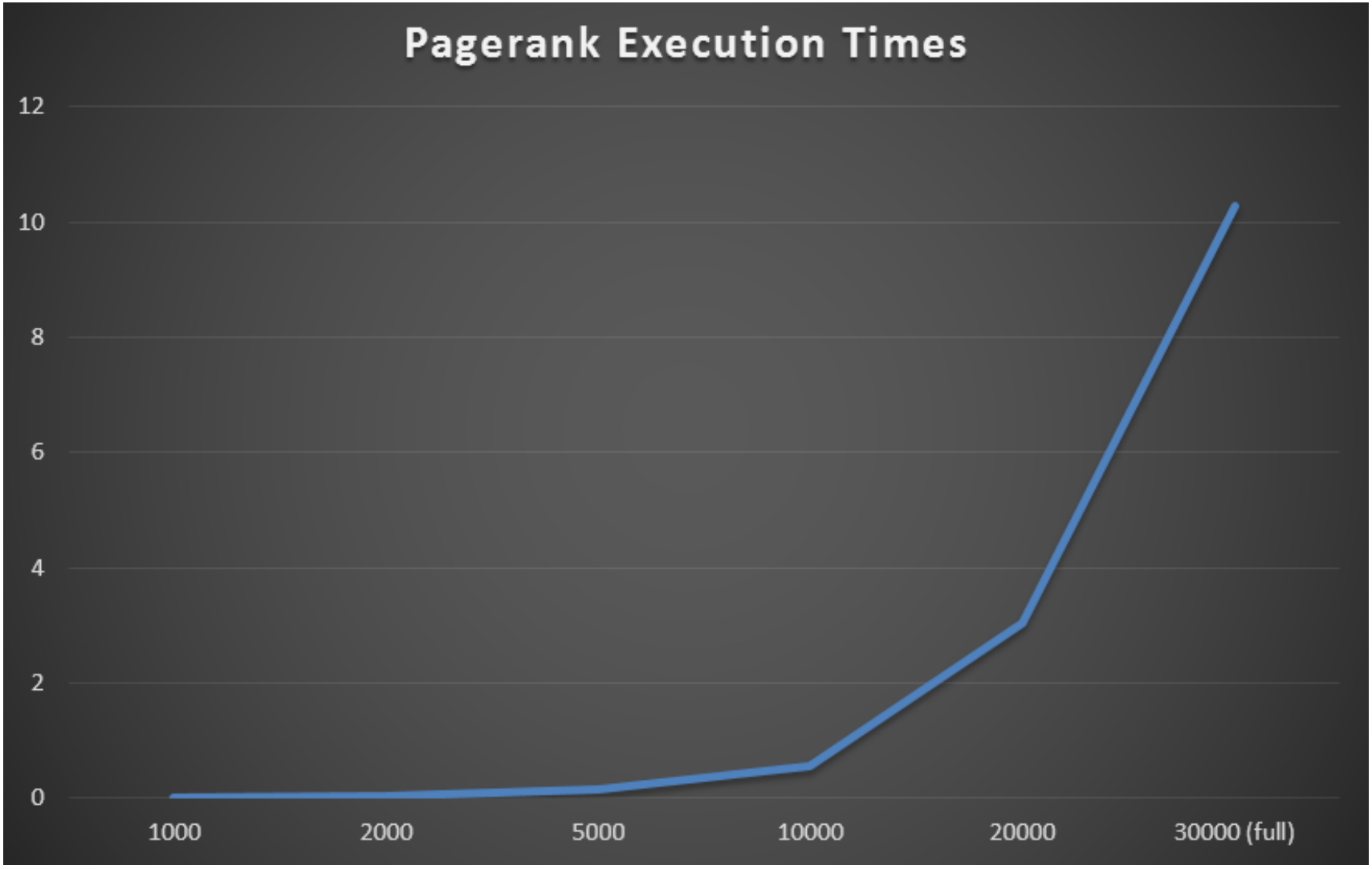
Results

We are going to present the result of the experiment, comparing the execution time of PageRank on following inputs:

- Graph of 1000 nodes
- Graph of 2000 nodes
- Graph of 5000 nodes
- Graph of 10000 nodes
- Graph of 20000 nodes
- Full Graph (30000 nodes)

All the graphs are generated by chunking the Full Graph.

Graph of Times



HITS

This hubs-and-authorities algorithm, sometimes called HITS (*hyperlink induced topic search*), was originally intended not as a preprocessing step before handling search queries, as PageRank is, but as a step to be done along with the processing of a search query, to rank only the responses to that query.

This kind of approach is used by the Ask search engine.

The Intuition Behind HITS

HITS views important pages as having two different type of importance.

- Certain pages are valuable because they provide information about a topic. These pages are called **authorities**.
- Other pages are valuable not because they provide information about any topic, but because they tell you where to go to find out about that topic. These pages are called **hubs**.

HITS Algorithm

For calculating the HITS values for the pages, we shall assign two scores to each Web page. One score represents the *hubbiness* of a page, that is the degree to which it is a good hub, and the second score represents the degree to which the page is a good authority.

These values are then calculated as:

- **Hubbiness**: the sum of the Authority value of the outgoing nodes.

HITS Algorithm

For calculating the HITS values for the pages, we shall assign two scores to each Web page. One score represents the *hubbiness* of a page, that is the degree to which it is a good hub, and the second score represents the degree to which the page is a good authority.

These values are then calculated as:

- **Hubbiness**: the sum of the Authority value of the outgoing nodes.
- **Authority**: the sum of Hubbines value of the incoming nodes.

HITS Algorithm

For calculating the HITS values for the pages, we shall assign two scores to each Web page. One score represents the *hubbiness* of a page, that is the degree to which it is a good hub, and the second score represents the degree to which the page is a good authority.

These values are then calculated as:

- **Hubbiness**: the sum of the Authority value of the outgoing nodes.
- **Authority**: the sum of Hubbines value of the incoming nodes.

These values are normalized so that the largest value is 1.

Results

We are going to present the result of the experiment, comparing the execution time of HITS on following inputs:

- Graph of 1000 nodes
- Graph of 2000 nodes
- Graph of 5000 nodes
- Graph of 10000 nodes
- Graph of 20000 nodes
- Full Graph (30000 nodes)

All the graphs are generated by chunking the Full Graph.

Tuning for improving performance

On the first attempts of running the algorithm on the full graph we observed that one iteration takes about 30 minutes, due to the nature of the algorithm.
In each iteration we explore all the graph and calculate the incoming nodes for the current node . . .

Considering that the graph never changes, we precomputed all the incoming nodes for each node so we can obtain the incoming nodes in $O(1)$.

Tuning for improving performance

In the HITS algorithm there are two stop rules:

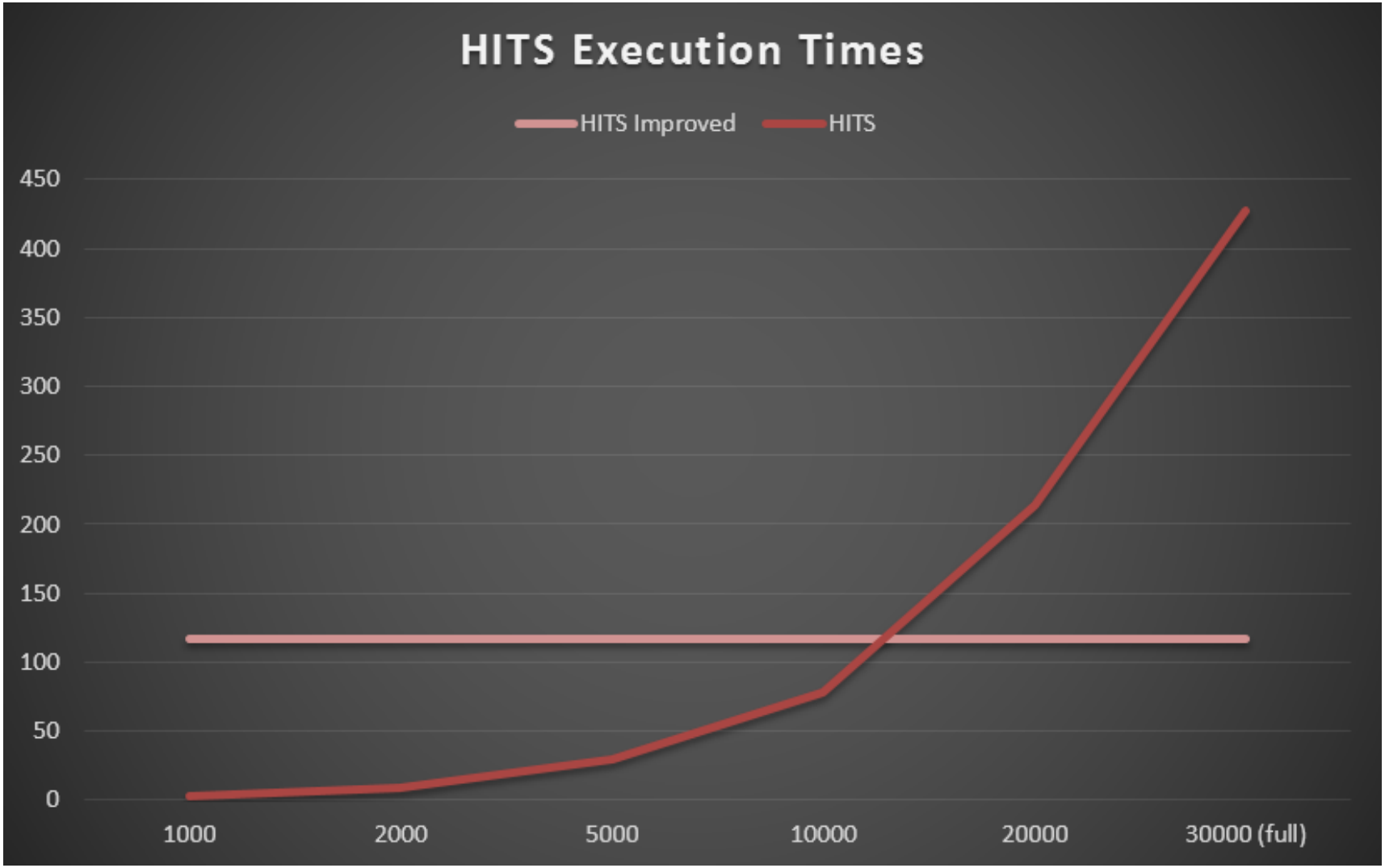
- Max number of iteration
- Min confidence on the errors reached

We noticed that in some cases the Algorithm runs until the maximum number of steps is reached, and when this happens the relative error trends to stabilize on a fixed level. We decided then to add a new stop rule:

- If two successive relative errors are distant at maximum a small ϵ, then stop the algorithm.

We tried this improvement on the full graph, we plotted as a pink line . . . the difference is embarrassing

Graph of Times



Parallel Version

We also implemented the parallel version, but the performance resulted to be worst compared with the non-parallel version: this is due to overhead of coping all the structures needed for the calculation of scores.

We now present times of 10 iteration of both version:

- **Non-Parallel**: 4,15"
- **Parallel**: 53,41"

Comparing the Results

On the time side, the results are incomparable since they the HITS time is up to 40 time the PageRank time.

The values for the nature of the algorithms are impossible to compare even if we tried.

Overview

The idea of Best Match

Given a query q, containing n query words, and a set of documents S, we define Best Match as a method that finds a subset of document S' such that:

- each document s_i of S' has a "reasonable" number of query words in it

According to this definition the basic Best Match consists of:

- counting how many query words documents have
  - we call this value "score" of a document
  - it is at maximum n
- ordering in decreasing order of score the documents (optional)
- return all documents whose score is "reasonable"
  - we use a threshold to define what is "reasonable"

Refining the Best Match

Two are the basic refinements to have a more efficient Best Match:

1. using an inverted index
   - in the form (word -> list of documents containing the word)
   - than the keys of the dataset are the query words
   - we can have in O(1) all the documents with a determined word
2. using the frequency instead of assigning score 1 to each query word found
   - defined as number of occurrences in document d / length(d)
   - requires precalculation of occurrences for all words and all S
   - it represents the **relevance** of documents to a particular word or query

Improved Best Match

We implemented also the following improved version of Best Match:

1. Sort documents in each inverted index in order of frequency of the term at which the inverted index refers
2. For every query term define its possible impact on the score as the frequency of the most frequent document in its index
3. Sort the query terms in decreasing order of impact
4. Consider the first 20 documents in the index of the first query term (if the first query term has an index with less than 20 documents, then complete with the first documents in the index of the next query term)

Improved Best Match

5. Compute the score for each of these documents
6. Consider the first term in which there are documents that have not been scored
7. Consider the first non-scored document in the index of this term
8. If the frequency of the current term in the current document plus the sum of the impact of next terms is larger than the score of the 20-th scored document, then score this document and repeat from 7, otherwise consider the next

Creating the Dataset

Our experiments ran on a set of approximatively 30000 pages created this way:

- we choose a web-page for each of the 15 categories listed in https://www.dmoz.org/
- for every of these web pages we crawled 2000 pages by using the Wibbi online crawler
- Moreover, from each pair of sets of 2000 pages, we choose at random 10 pairs of vertices (u, v) with u being a page in the first set and v being a page in the second set and added a link from u to v (if this link was absent)

Creating the Dataset

Here is the complete list of the websites chosen.

Category	Website	Description
Arts	www.imdb.com	The Internet Movie Database
Business	www.moodys.com	Corporate finance, banking
Computers	www.ibm.com	International Business Machines Corporation.
Games	www.ign.com	Videogame news
Health	www.who.int	World Health Organization
Home	www.cooks.com	Recipe search
Kids	www.cartoonnetwork.com	The home of cartoons online

Category	Website	Description
News	www.foxnews.com	Breaking News
Recreation	www.lego.com	Producer of bilding blocks.
Reference	www.britannica.com	Encyclopaedia Britannica Online.
Science	www.nasa.gov	Comprehensive, world-class center for aeronautics
Shopping	www.amazon.com	Most know shopping website
Society	www.un.org	Daily United Nations news, documents and publications
Sports	www.nba.com	The official site of the National Basketball Association
Regional	www.lonelyplanet.com	Offers travel advice, detailed maps, travel news

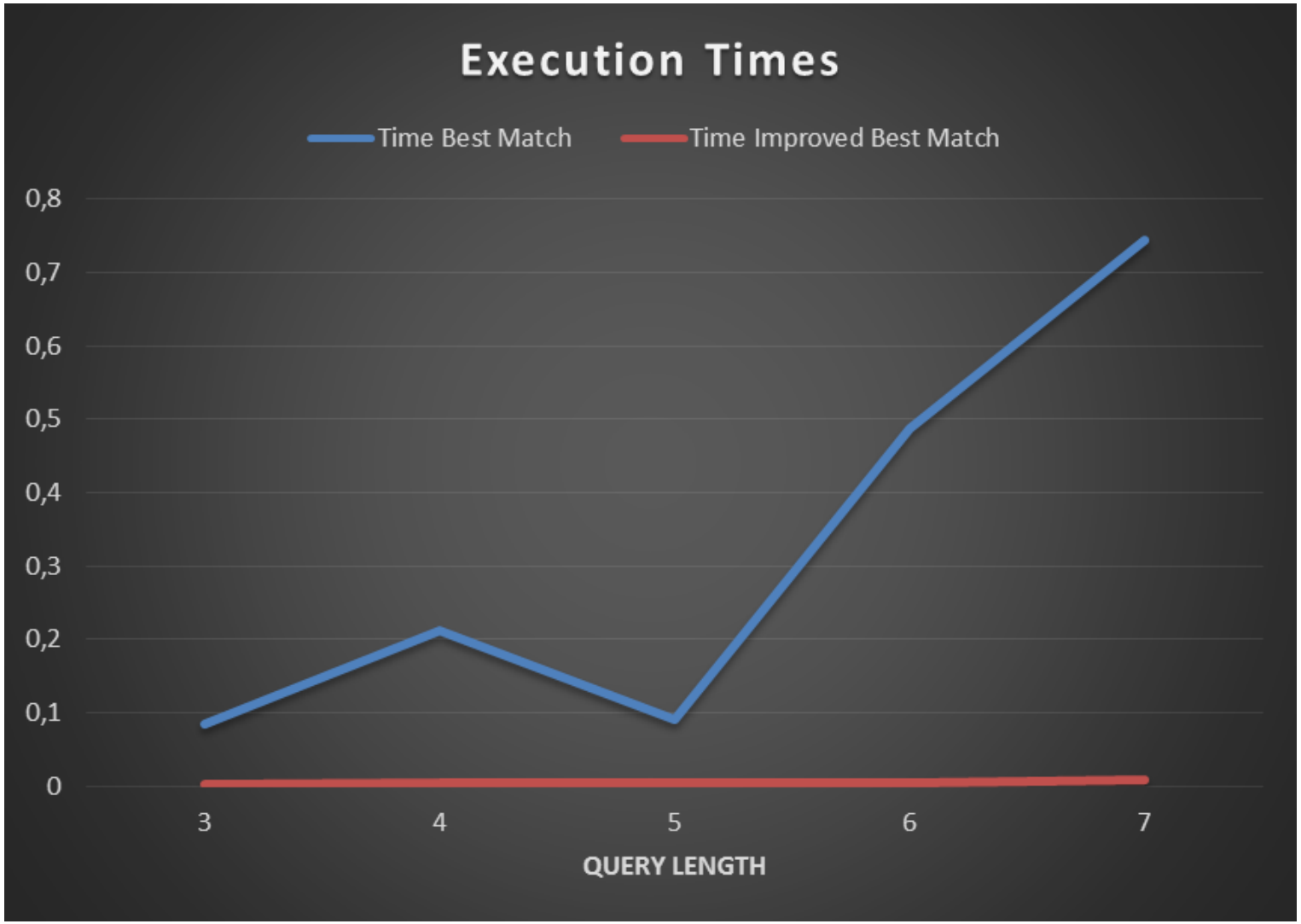
Experiment configuration

For obtaining the times comparison between BestMatch and ImprovedBestMatch we run both algorithms on 25 random queries of different length :

- 5 for each length from 3 to 7

Then we mean the results and plotted them on a graph

Time comparison BestMatch vs ImprovedBestMatch



Time comparison BestMatch vs ImprovedBestMatch

First of all Improved Best Match "wins" on all query lengths, moreover it seems to be constant. This is due to the fact that there is a huge difference on the times of the algorithm at least of 1 order of magnitude.

We can notice that times of query of length 5 in best match are smaller of the one on length 4; this can be attributable to the random generation of queries, in fact, if we take 5 very uncommon words there are few documents that have one of these word so the algorithm ends quickly.

Overview

Search Engine

The implemented Search Engine combine the algorithms seen before.
Given a query:

1. Find the documents that match the query using:
   - Best Match
   - Improved Best Match
2. Order these documents by:
   - Page Rank Score
   - HITS authority

Running configurations

- We ran the experiment for a total of 25 queries
- 5 for each length between 3 and 7
  - In each group of 5 queries 1 of them is manually constructed by us for having a high likelihood with a real query
  - The other 4 are randomly generated among the dataset
- The manual generated query are used not only for calculate the execution time, but also for fully analysing the output of search engine

Best Match Considerations

We notice that there is an high correlation within the first positions on the documents ordered by PageRank and HITS, this means that event if the value of the ranking algorithms are not comparable absolutely, tent to have a relative comparability.

Overview

First Price Auction

In this kind of auction, bidders submit simultaneous "sealed bids" to the seller. The terminology comes from the original format for such auctions, in which bids were written down and provided in sealed envelopes to the seller, who would then open them all together.

The highest bidder wins the object and pays the value of her bid.

Non - truthfulness of FPA

In a sealed-bid first-price auction, the value of your bid not only affects whether you win but also how much you pay.

Bidding your true value is not a dominant strategy. By bidding your true value, you would get a payoff of 0 if you lose (as usual), and you would also get a payff of 0 if you win, since you'd pay exactly what it was worth to you.

As a result, the optimal way to bid in a first-price auction is to "shade" your bid slightly downward, so that if you win you will get a positive payoff. Determining how much to shade your bid involves balancing a trade-off between two opposing forces.

Generalized Second Price Auction

Also called Vickrey auctions. Bidders submit simultaneous sealed bids to the sellers;

The highest bidder wins the object
and pays the value of the second-highest bid.

These auctions are called Vickrey auctions in honor of William Vickrey, who wrote the first game-theoretic analysis of auctions. Vickery won the Nobel Memorial Prize in Economics in 1996 for this body of work.

Truthfulness of GSP

Truthful bidding is a dominant strategy in a sealed-bid second-price auction. The heart of the argument is the fact noted at the outset: in a second-price auction, your bid determines whether you win or lose, but not how much you pay in the event that you win.

So in a second-price auction, it makes sense to bid your true value even if other bidders are overbidding, underbidding, colluding, or behaving in other unpredictable ways.

Description of Bots

We used the following bots:

- **Best_response** with balanced tie-breaking rules
- **Best_response_competitive** submit the highest possible bid that gives the preferred_slot
- **Best_response_altruistic** submit the lowest possible bid that gives the preferred_slot
- **Competitor** always submit a bit grater than the highest bid
- **Budget_saving** always submit the minimum between last-non winning bid and advertiser value

Description of Bots

We used the following bots:

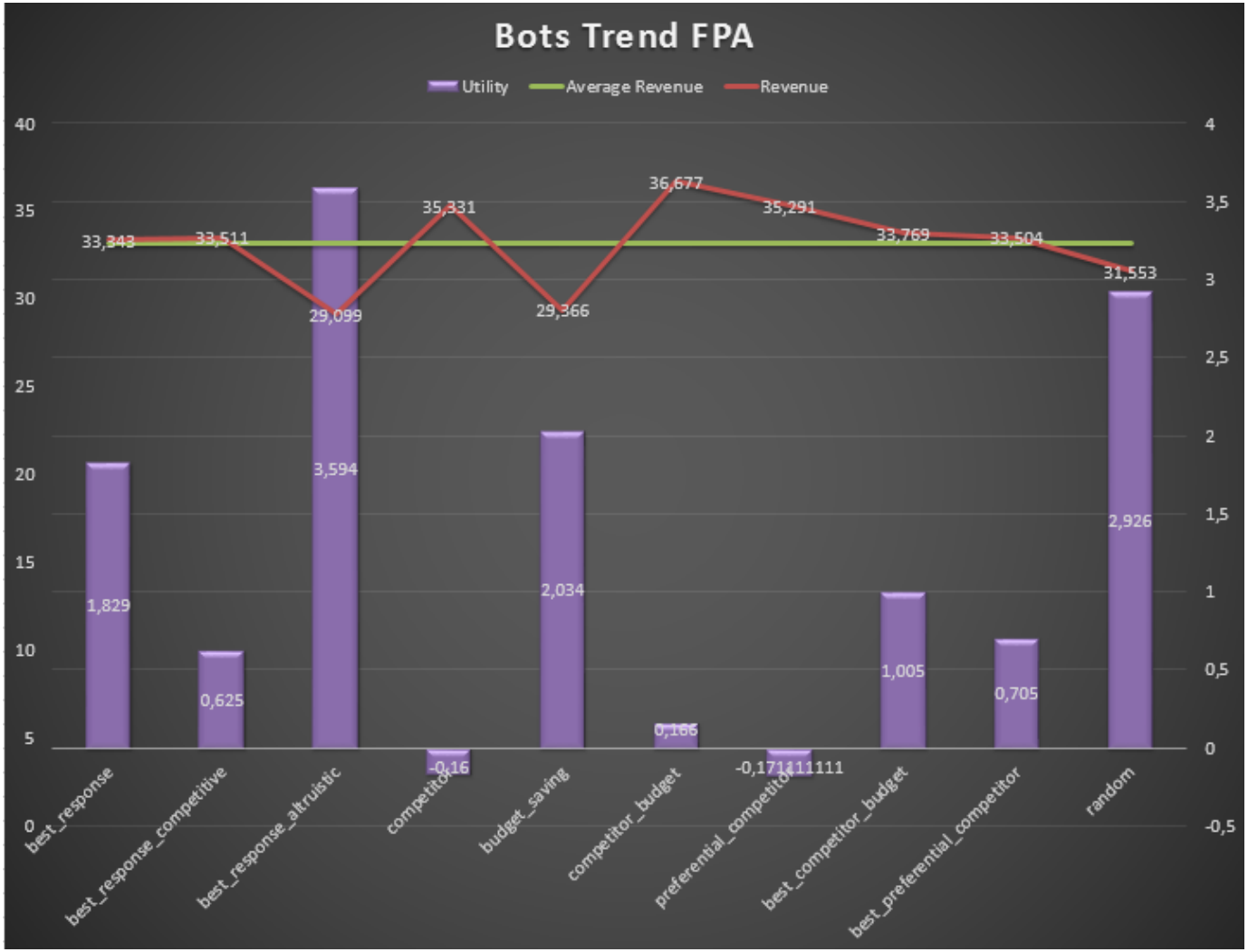
- **Competitor_budget** Use competitor when budget is more than its half, best_response otherwise
- **Preferential_competitor** Use competitor when value is more than a threshold, budget_saving otherwise
- **Best_competitor_budget** Use best_response_competitive when budget is more than its half, best_response otherwise
- **Best_preferential_competitor** Use best_response_competitive when value is more than a threshold, budget_saving otherwise
- **Random** bids randomly

Selection of interesting Bots

The configuration of the experiment for founding the most interesting bots is the following:

- **Number of Query Words** 1
- **Number of Auction** 10
- **Slot 1 click-through** 0.4
- **Slot 2 click-through** 0.15
- **Number of Advertiser** 3
  - 1 bot to test and 2 enemies (of the same kind)
- **Values** 7
- **Budgets** 25
- **Number of runs** 10000

FPA Utility


Bots Trend FPA
Utility
Average Revenue
Revenue
40
35
30
25
20
15
10
5
0
4
3,5
3
2,5
2
1,5
1
0,5
0
-0,5
33,343
33,511
29,099
35,331
29,366
36,677
35,291
33,769
33,504
31,553
1,829
0,625
3,594
-0,16
2,034
0,166
-0,171111111
1,005
0,705
2,926
best_response
best_response_competitive
best_response_altruistic
competitor
budget_saving
competitor_budget
preferential_competitor
best_competitor_budget
best_preferential_competitor
random

Considerations on FPA advertiser-side

Observing FPA graph it seems that *best_response_altruistic* is the best bot, yielding the highest utility.

This is due to the fact the bots that try to save their budgets, offer the lowest bid possible. Since these bids are "far" from the advertiser's values and FPA is not truthful, they have a high chance to have a good slot at low price.

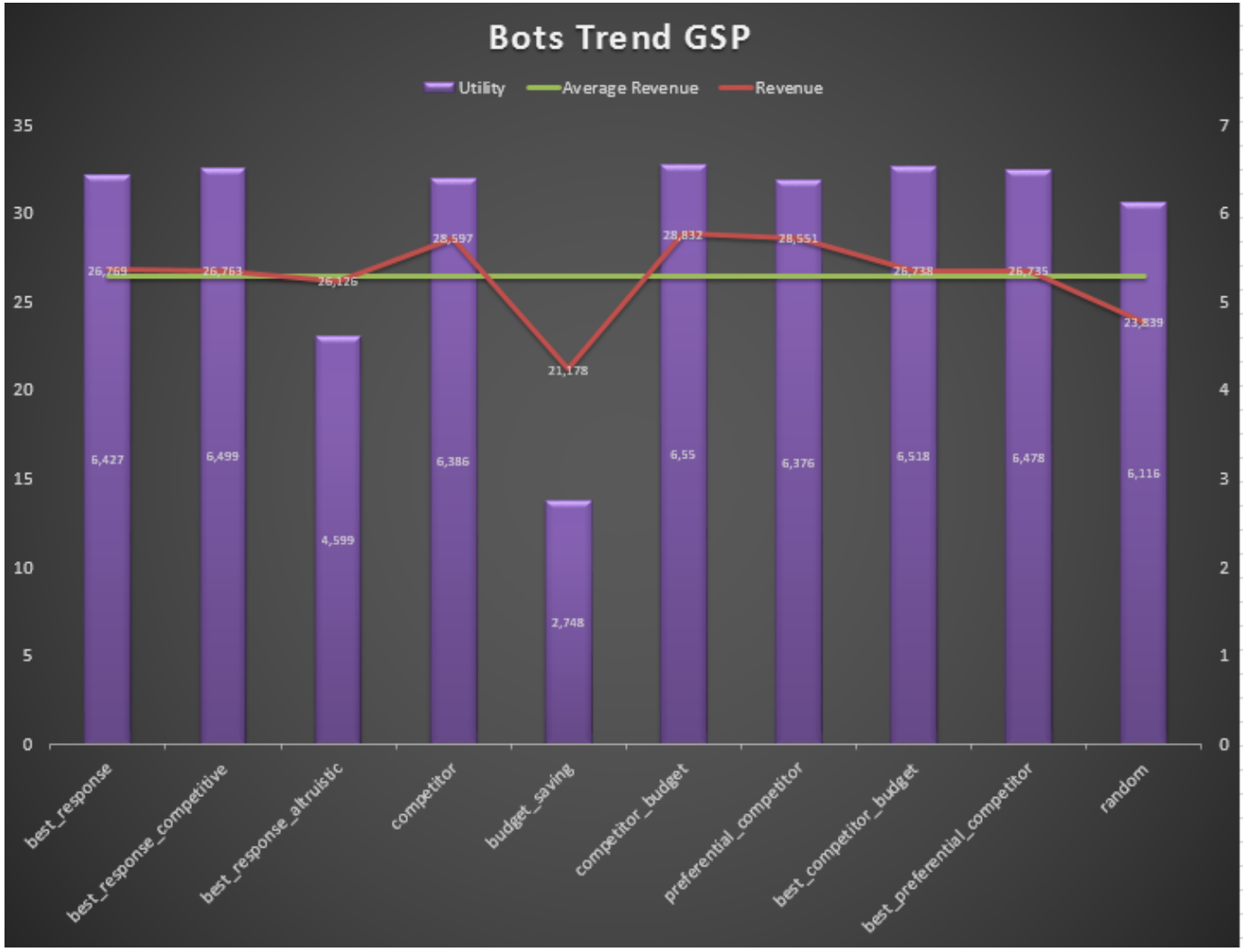
Considerations on FPA seller-side

The bots who tries to save their budget on seller-side are the worst, because they lower not only their own bids, but also the bids of the whole auction.

It follows that the other kinds of bot which raise their bids are more lucrative for us.

For the reasons explained before we need a balance between high revenue and high utility, and the bot that seems to accomplish this is **best_response**.

GPS Utility


Bots Trend GSP
Utility
Average Revenue
Revenue
35
30
25
20
15
10
5
0
7
6
5
4
3
2
1
0
26,769
26,763
26,126
28,597
21,178
28,832
28,551
26,738
26,735
23,839
6,427
6,499
4,599
6,386
2,748
6,55
6,376
6,518
6,478
6,116
best_response
best_response_competitive
best_response_altruistic
competitor
budget_saving
competitor_budget
preferential_competitor
best_competitor_budget
best_preferential_competitor
random

Considerations on GSP advertiser-side

We notice that, since GPS is truthful, the bots that offer bids near to their "real" value perform better, obtaining a higher utility.

For this reason all the bots that tent to lower the bids, stray from the "real" value, perform a lot worst, obtaining a lower utility.

Considerations on GSP seller-side

We notice that there are a lot of bots with utility $\sim 6,5$.

But we also notice there are a lot of fluctuation on the revenue, difficult to explain.

For this reason we decided to calculate an heuristic about the quantity and quality of slots obtained.

Our heuristic

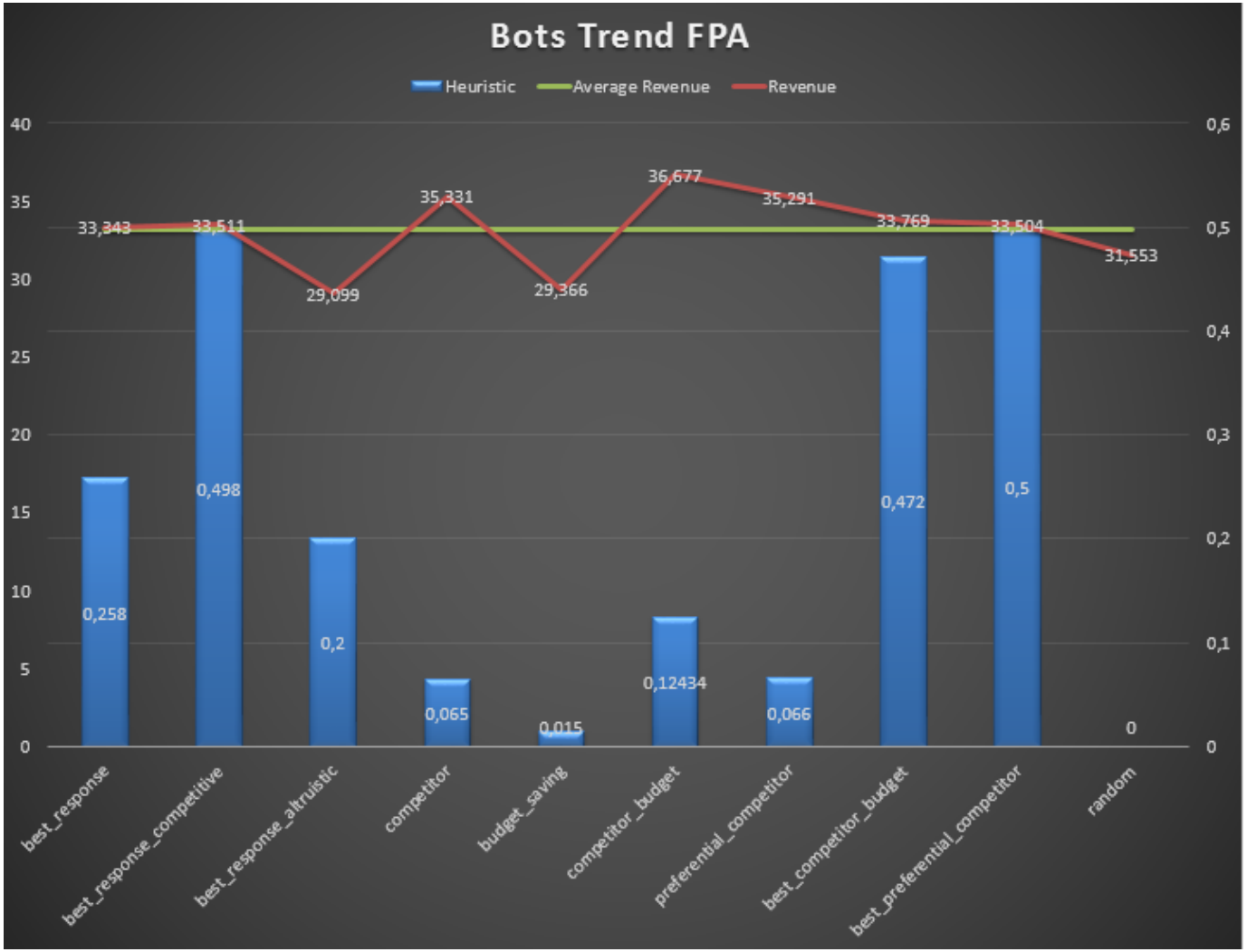
The heuristic for a bidder is calculated as follow:

On each step of an auction, given the **preferred_slot** and the result of that step.

- Add 1 if the bidder obtains the **preferred_slot**, or he wants nothing and obtain nothing
- Add 2 if the bidder obtains a better slot, since he bids for a worse slot.
- Subtract 1 if the bidder obtain a worse slot, since he bids for a better slot.

We mediate this score on the auction step.
Straddling in the range $[-1; 2]$

Heuristic on FPA



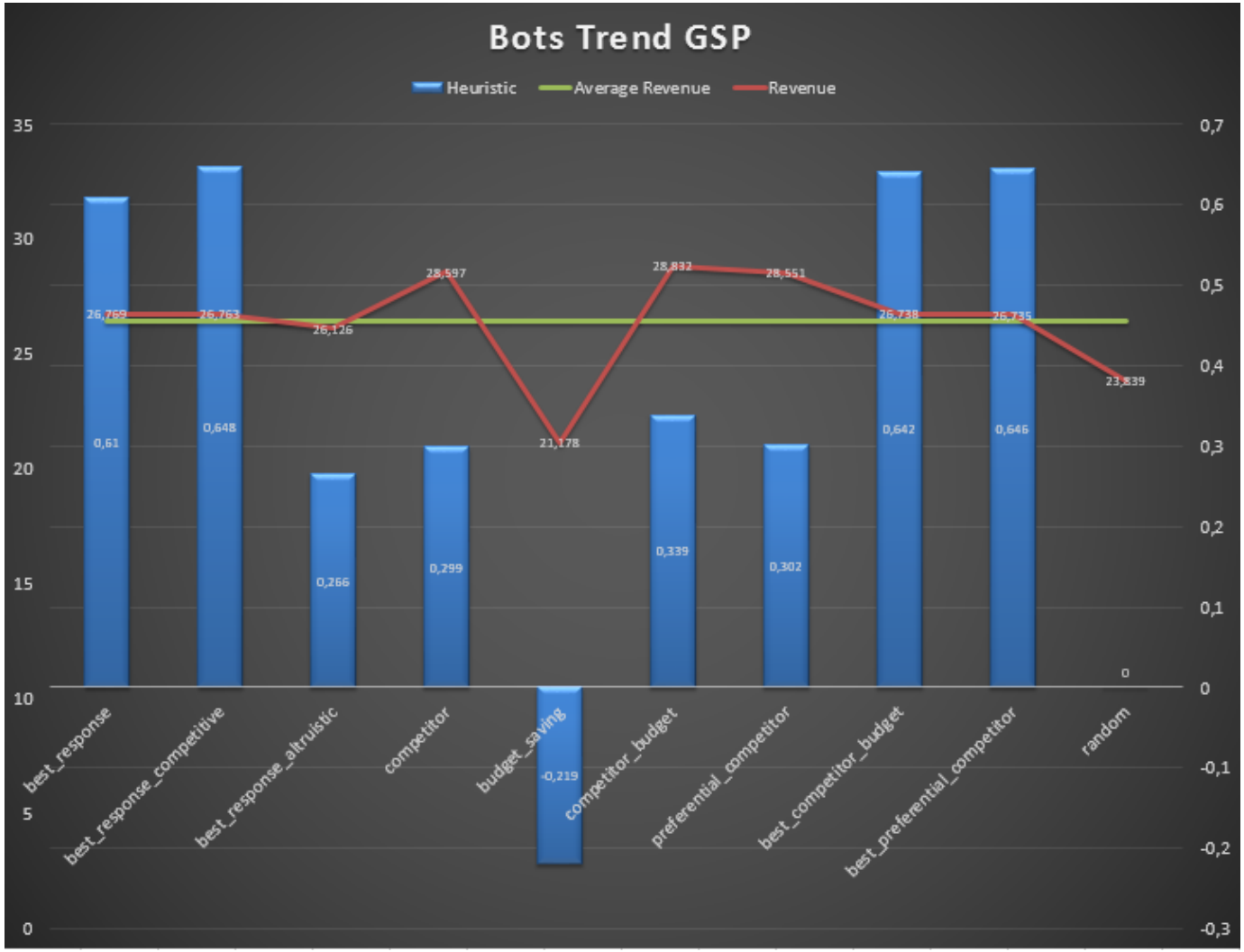
Considerations on FPA heuristic

Considering the bots the have the higher utility (best_response_altruistic, budget_saving) we notice that they have a very low heuristic score, this is due to the fact they tent the obtain the slots that the don't "want" but a very low price resulting in a high utility.

On the other hand aggressive bidders, tent to have a better heuristic score but a worse utility, since they pay a lot for the slot they obtain.

For the FPA the revenue is not dependant on the heuristic score, but on the "nature" of the bot.

Heuristic on GSP



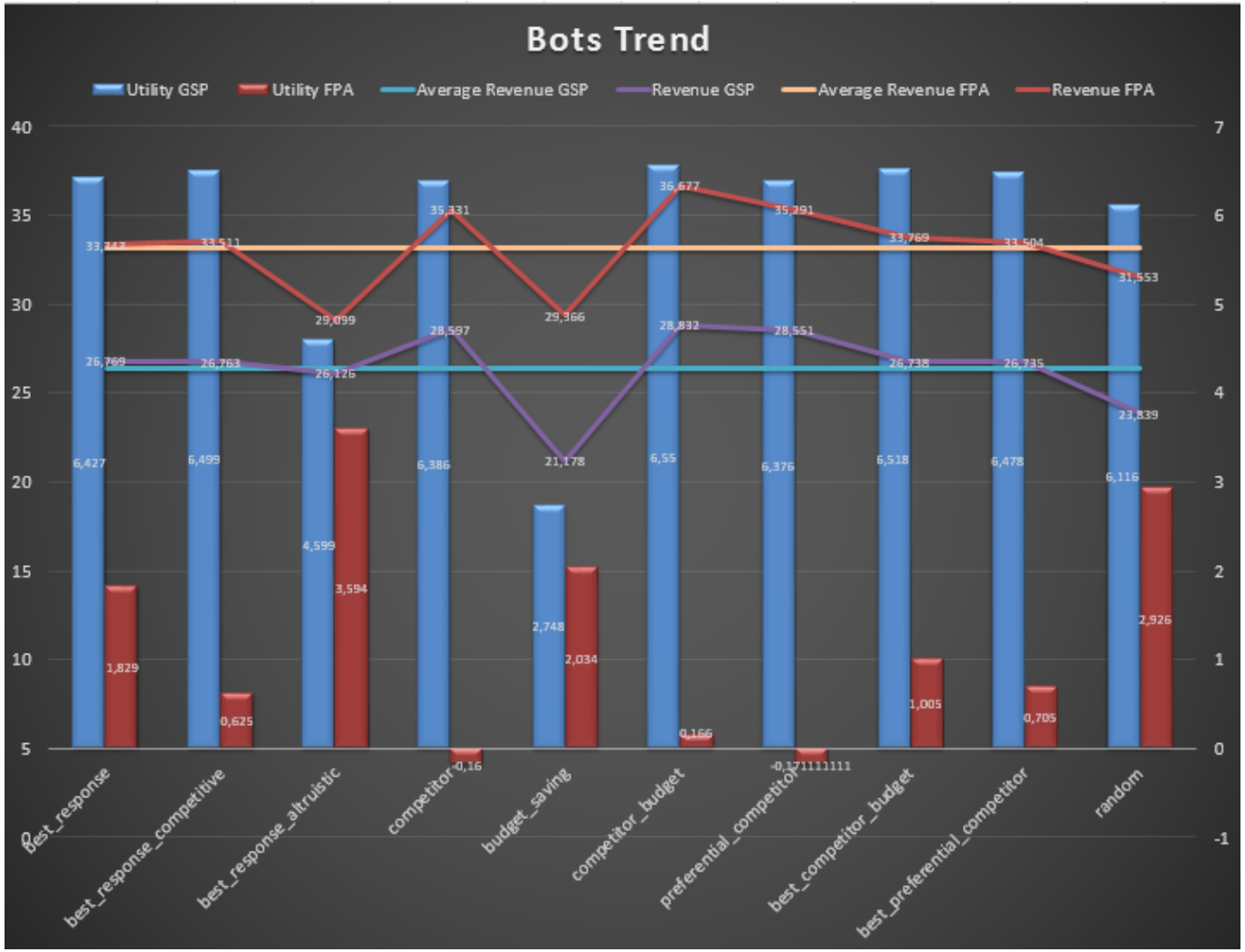
Considerations on GSP heuristic

Thanks to the heuristic score we can divide the bots in several groups, the “blind” competitors, the “wise” competitors and the “stingy”.

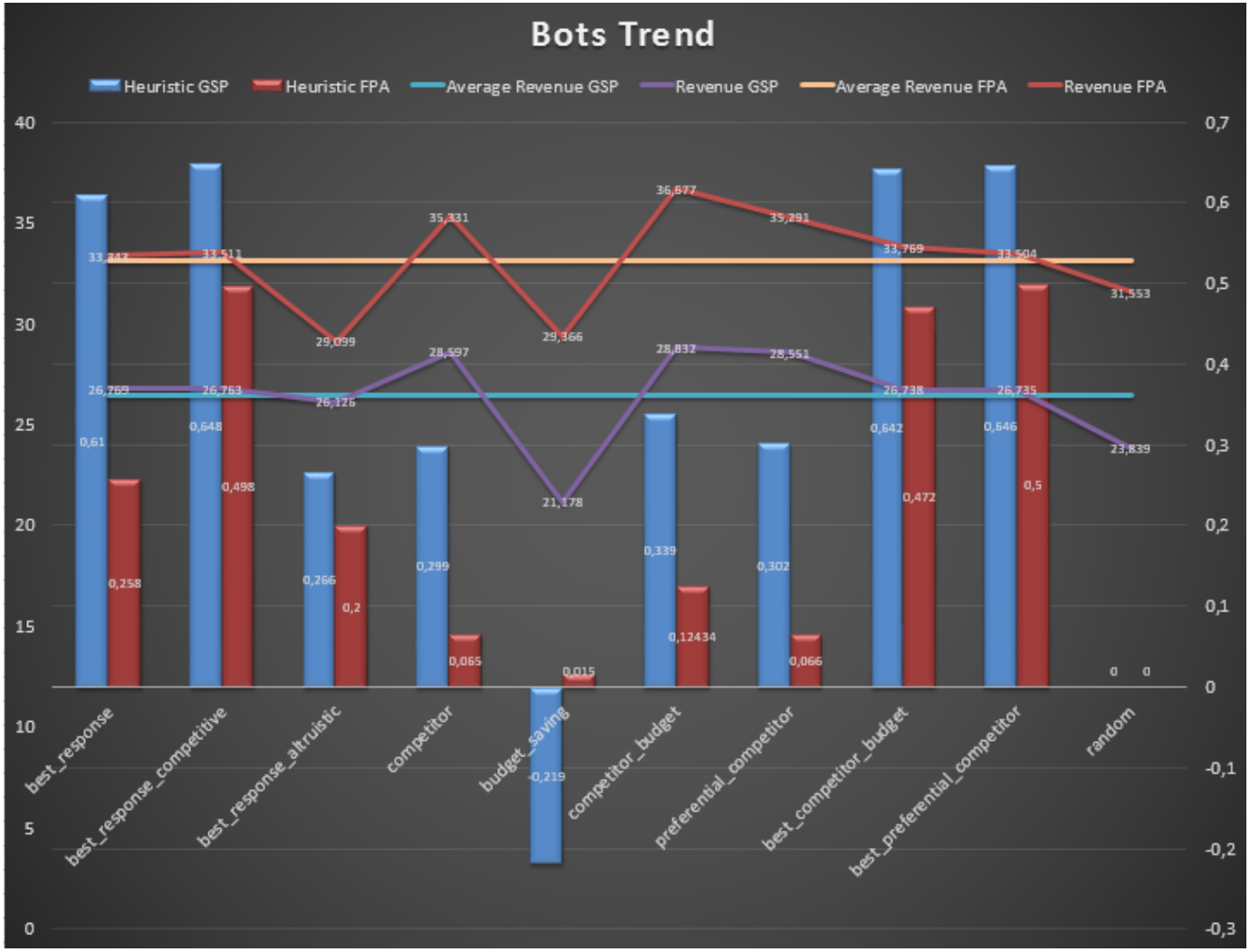
	Utility	**H_score**	**Revenue**
"Blind" competitor	High	Low	High
"Wise" competitor	High	High	Average
"Stingy"	Low	Low	Low

Table : Resume

Comparing FPA and GPS results



Comparing FPA and GPS results



Comparing FPA and GPS results

Comparing the auction types, we notice that on a "fixed" auction they have the following behaviour:

- FPA auction produce the highest revenue for the seller, this follow from the fact that you pay exactly what you bid. As for clients satisfaction, we notice that they have bot at low heuristic and utility compared to GPS.
- GPS instead produce lowest revenue and high utility and heuristic score for the client, this is due the nature of GPS:
  the winner of a slot can offer a very high bid without having too much impact on what he pays, because he doesn't hid bid bud the bids immediately lower than his.

The choice depends of what we want to "favour"

Revenue equivalence Theorem

As we notice the revenue are different, this is due to the fact that we stopped the auction at 10 steps and moreover the budget of the advertiser are different so using balance algorithms, sometime they end up with not enough budget for bidding.

Bug setting the max step to: 100, and giving an higher budget equal to all the advertiser we have this result:

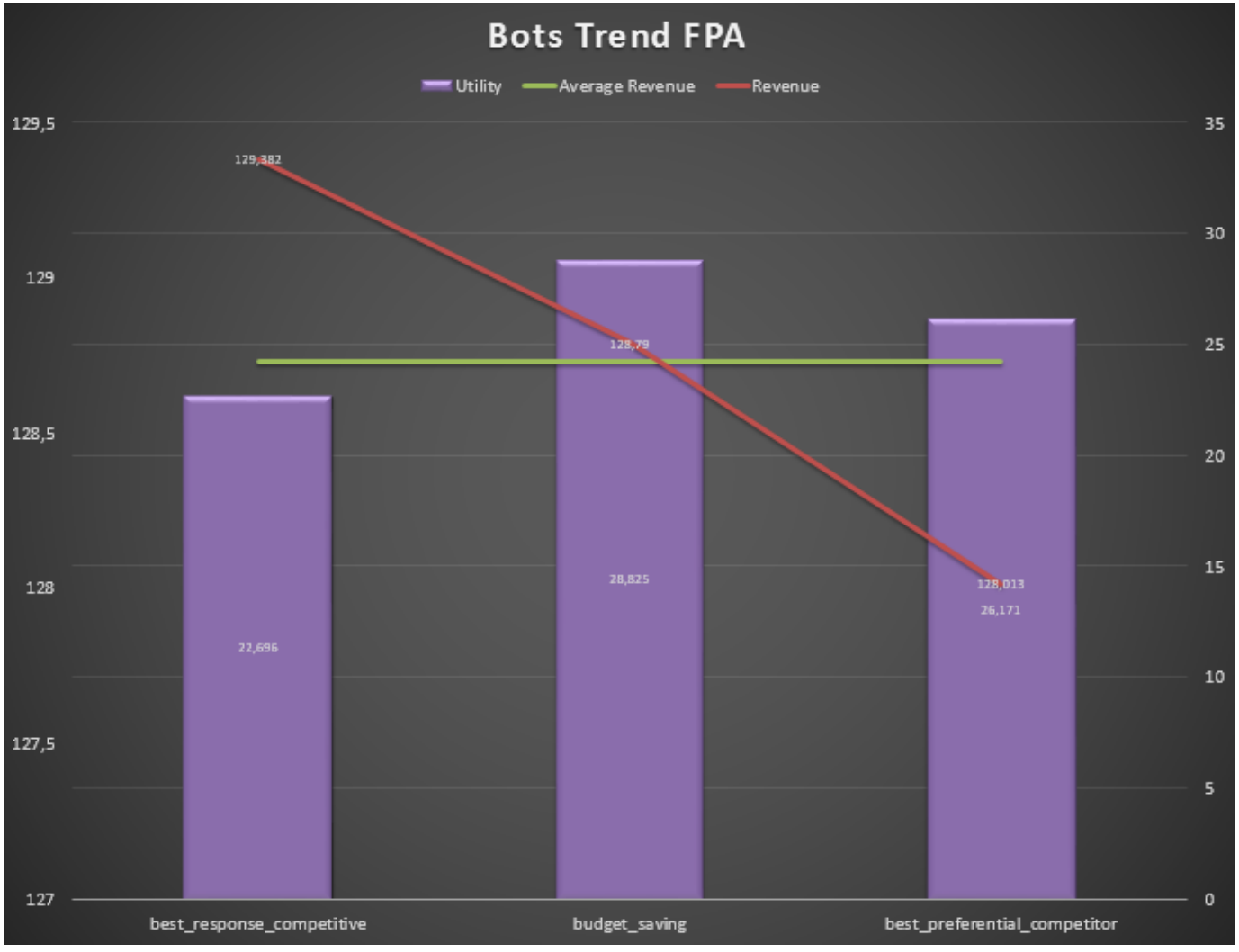
```
55
{'prova': {'adv_slots': {'id2': 'z', 'id1': 'y'}, 'adv_bids': {'prova': {'y': 7.0, 'x': 7.0, 'z': 7.0}}, 'adv_pays': {'y
': 7.0, 'z': 7.0}}}
Totale Val:             0.672727272727
Totale Uti:             9.85714285714
Totale Rev:             756.4
AUCTION: fpa I: best_response ENEMIES: best_response
55
{'prova': {'adv_slots': {'id2': 'z', 'id1': 'y'}, 'adv_bids': {'prova': {'y': 7.0, 'x': 7.0, 'z': 7.0}}, 'adv_pays': {'y
': 7.0, 'z': 7.0}}}
Totale Val:             0.672727272727
Totale Uti:             9.85714285714
Totale Rev:             756.4
```

Running Configuration

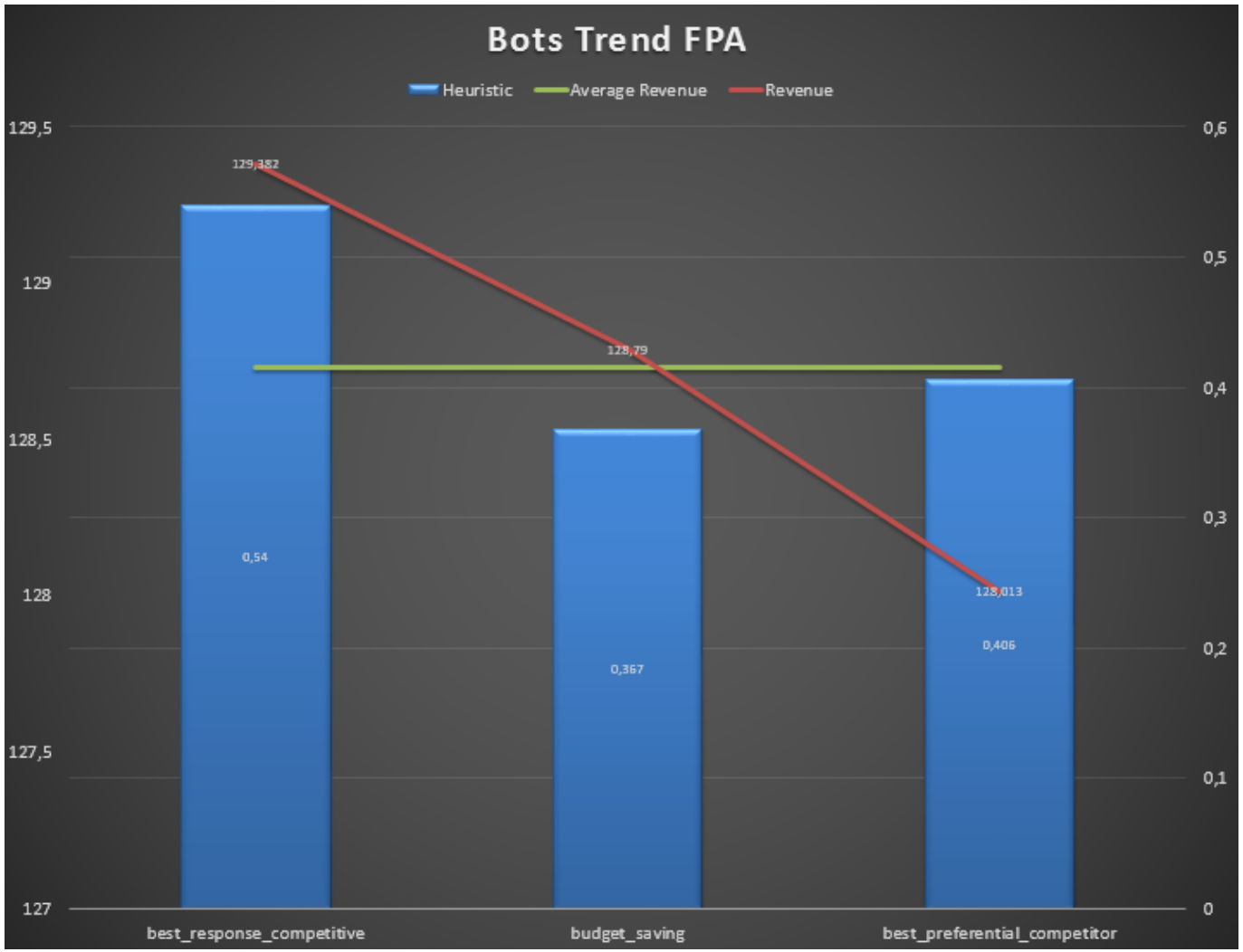
- **Number of Query Words** 10
- **Number of Auction** 20
- **Number of Slot for each word** [2, 4]
- **Slot i click-through** [0,1]
- **Number of Advertiser** 6
  - 1 bot to test and 5 enemies (of the same kind)
- **Values** [0, 20]
- **Budgets** [10, 50]
- **Minimum Interest Threshold** [5, 15]
- **Number of runs** 500

We choose two bots that have the best result among all the auction, and the worst one.

Utility FPA


Bots Trend FPA
Utility
Average Revenue
Revenue
129,5
129
128,5
128
127,5
127
35
30
25
20
15
10
5
0
129,382
128,79
128,013
22,696
28,825
26,171
best_response_competitive
budget_saving
best_preferential_competitor

Heuristic FPA

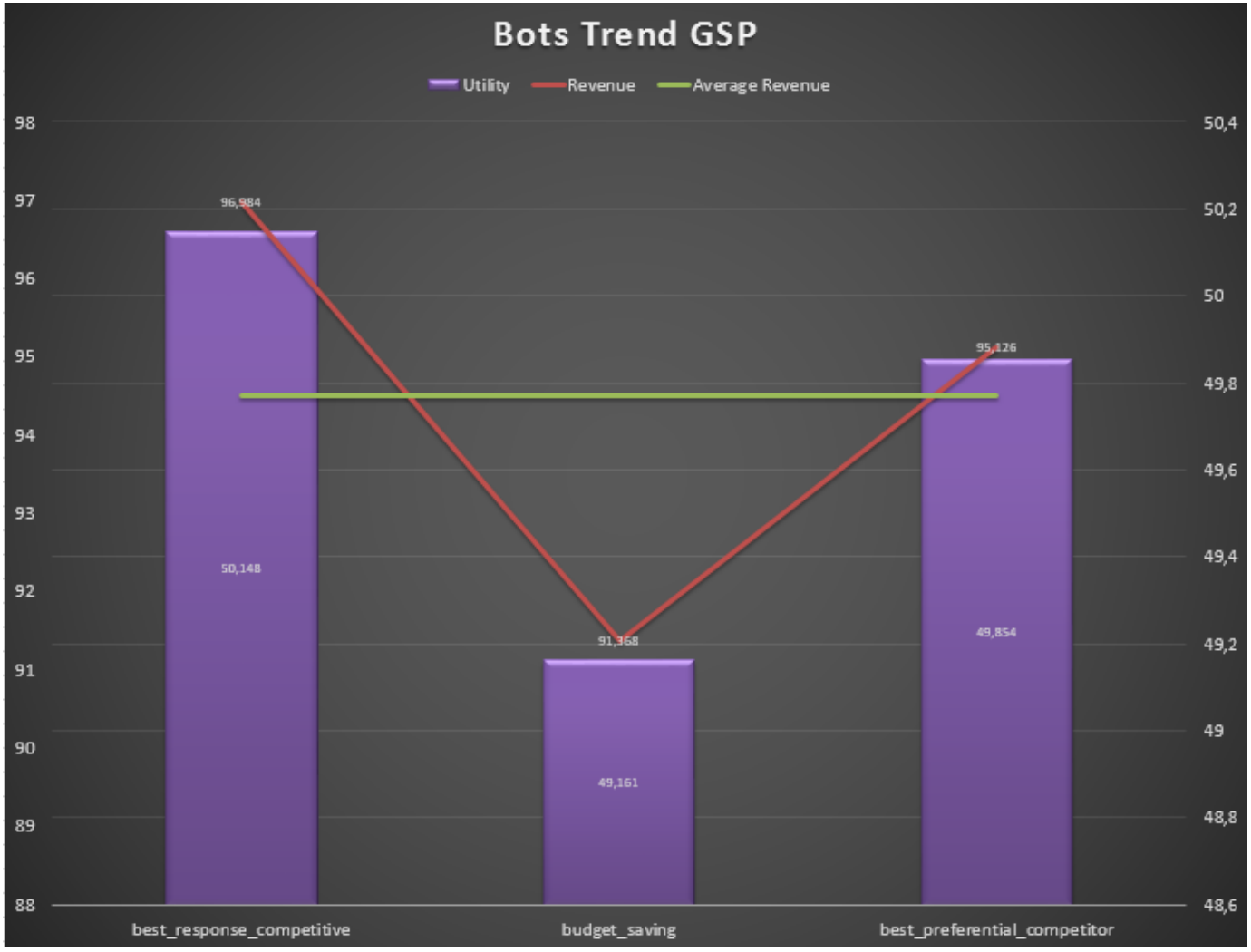


Consideration on FPA

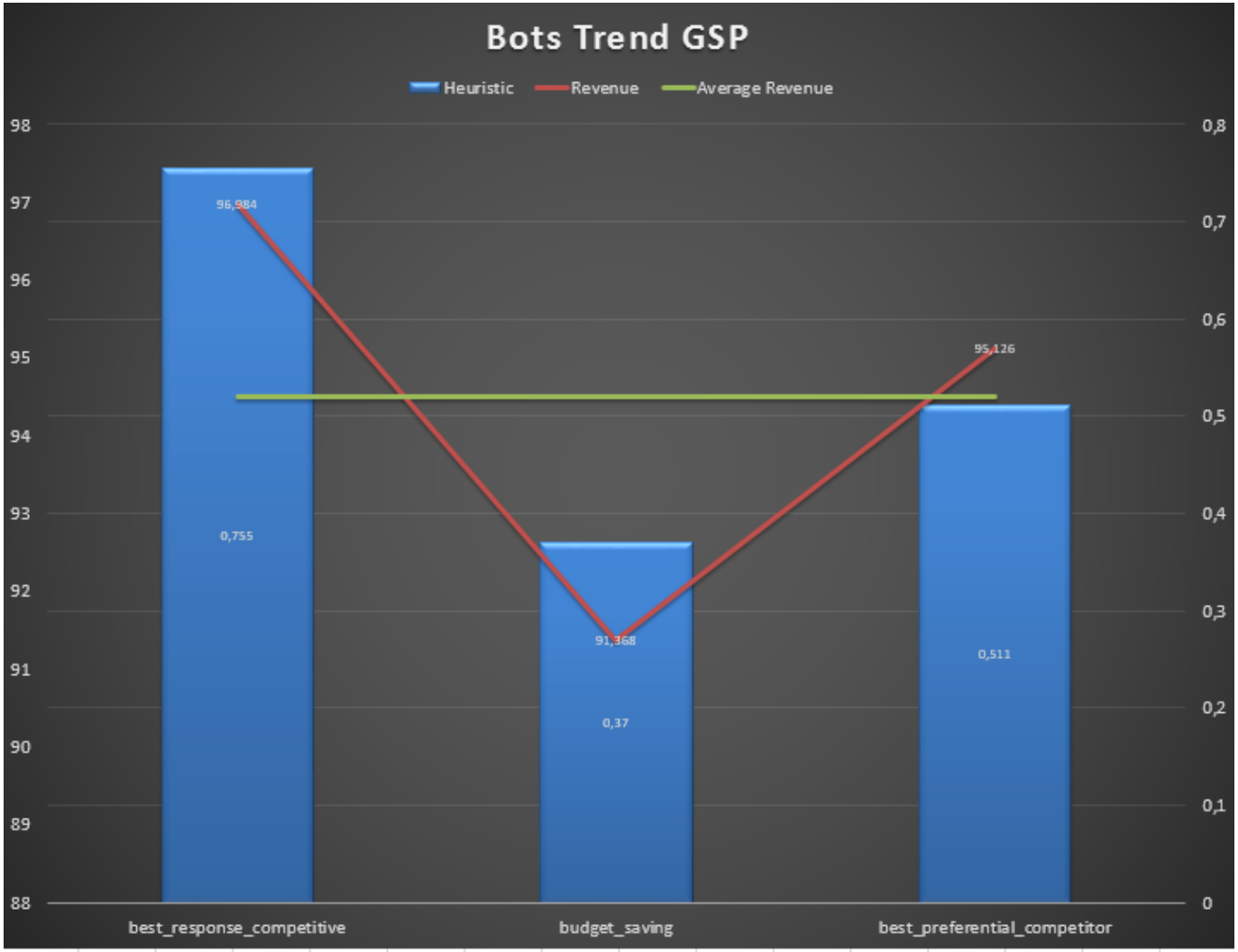
As expected the utility shows the same behaviour as in the fixed auction and difference on the revenue are almost irrelevant.

We notice worsening on the best_preferential_competitor due the fact that his behaviour is based on the threshold, that can change on different runs, this leads to more variability.

Utility GSP


Bots Trend GSP
Utility
Revenue
Average Revenue
98
97
96
95
94
93
92
91
90
89
88
50,4
50,2
50
49,8
49,6
49,4
49,2
49
48,8
48,6
96,984
50,148
91,368
49,161
95,126
49,854
best_response_competitive
budget_saving
best_preferential_competitor

Heuristic GSP



Consideration on GSP

In regards to utility and revenue, we notice that GPS shows the same behaviour, as said in FPA best_prefential_competitor shows a worsening for the same reason.

Instead budget_saving improve his behaviour, still remaining the worst, because in each run the budget is not fixed.

FPA vs GSP

As seen in the fixed example, we can observe that:

- FPA lead to a higher revenue compared to GPS
- GPS hold a higher utility and heuristic in relation to FPA

Thank you for the attention.