KaBOOM

Sensitivity analysis

Posted by Rinat Sergeev on Oct 16

Hi folks,

Here I am starting to post the results of the data-analysis I run over the ***top-5*** solutions from Asteroid Tracker Marathon. These results are preliminary, as they are going to be officially presented next week, after all the pieces get assembled together, but I am welcoming everyone who is curious to leave their questions, thoughts, and suggestions.

I. Learning from the scoring data, collected during the contest
http://community.topcoder.com/longcontest/stats/?module=ViewOverview&rd=16051.

Previously, I have already posted in a different thread a plot of the provision scoring dynamics ("*Submission Score Dynamics for KaBoom MM*" attached), what is a good illustration of the competition patterns observed in this contest. Two strong competitors (***psyho*** and ***wleite***) had been fighting each other in a tough race to the end, squeezing maximum out of their approaches, and reaching some visually observed saturation in scores. These two were keeping a significant leading advantage over the rest of contestants till the last few days, when this gap was nearly closed by the other three contestant (***zaq1xsw2tktk***, ***jonathanps***, and ***yowa***), who occupied ***3nd*** to ***5th*** places. It seems, that all three of those have started the competition very late and, despite the high results, their scoring curves are far from saturation. In regard to the solution characteristics, this pattern suggests that we have two very strong, well-polished, and similar quality algorithms that took ***1st*** and ***2nd*** place, and we have 3 other approaches that are very promising, may have good ideas behind, but may contain suboptimal components.

The final scoring results (*see Leaderboard screenshot* attached) are in a strong agreement with the provisional ones:
The measured difference between the top-2 solutions (***1st***, ***psyho***, *736.23* score, and ***2nd***, ***wleite***, *734.69*) is of the order of *0.2%* of their scoring value, whereas the gap between ***2nd*** and ***3rd*** places is ***22 times*** larger.

One of the simplest tests to illustrate the significance of the obtained scoring values is to treat the Provisional and Final scoring tests as two independent measures of algorithm's quality and measure the inaccuracy of the metrics by comparing those tests with each other (see "*Robustness test: Comparison between the final and provisional scores*" attached).
The scores are in strong correlation with each other and fit well a close-to-diagonal linear curve:
FinalScore = 1.023 Provisional Score
(Solid Line on the plot, as compared to dashed one, showing the diagonal)
The coefficient in this expression shows that the Scoring dataset, on average, allowed to earn a 2% higher scores than the Provisional one, and reflects a sensitivity of the absolute value of the score to random differences between the datasets.
The absolute majority of the dots, representing the contestants' final submissions, are located very close to the line, with no visual outliers below the curve, what is a sign that there is no noticeable overfitting of the solutions to Provisional dataset. Under this assumption, the mean deviation of the dots from the line is a good measure of the scoring accuracy.
It can be visually noticed that the precision of the scoring is very high, and allows to confidently distinguish ***5th***, ***4th***, and ***3rd*** places from each other, from the places below and above. At the same time, the scoring difference between the ***1st*** and ***2nd*** places is on the edge of the scoring precision, and these two solutions should be treated as rather of equal strength, than as one being confidently stronger than the other.
To conclude the scoring part, the contest has produced ***two equal strength and unbiased solutions*** that show a significant gap over all others.

II. Looking on what is hidden behind the scores.
The same score can result from quite different solution behaviors, so there is a challenge of how to illustrate different aspects of the solutions' behavior and sensitivity of those strategies to the assumptions we made. I want to note, that the current contest was quite unique in the amount of external parameters, assumptions, and strategic features that can potentially impact the optimal strategies, and if we want to use these solutions in a real world, the question of how generalizable the solutions we developed are has to be explored.

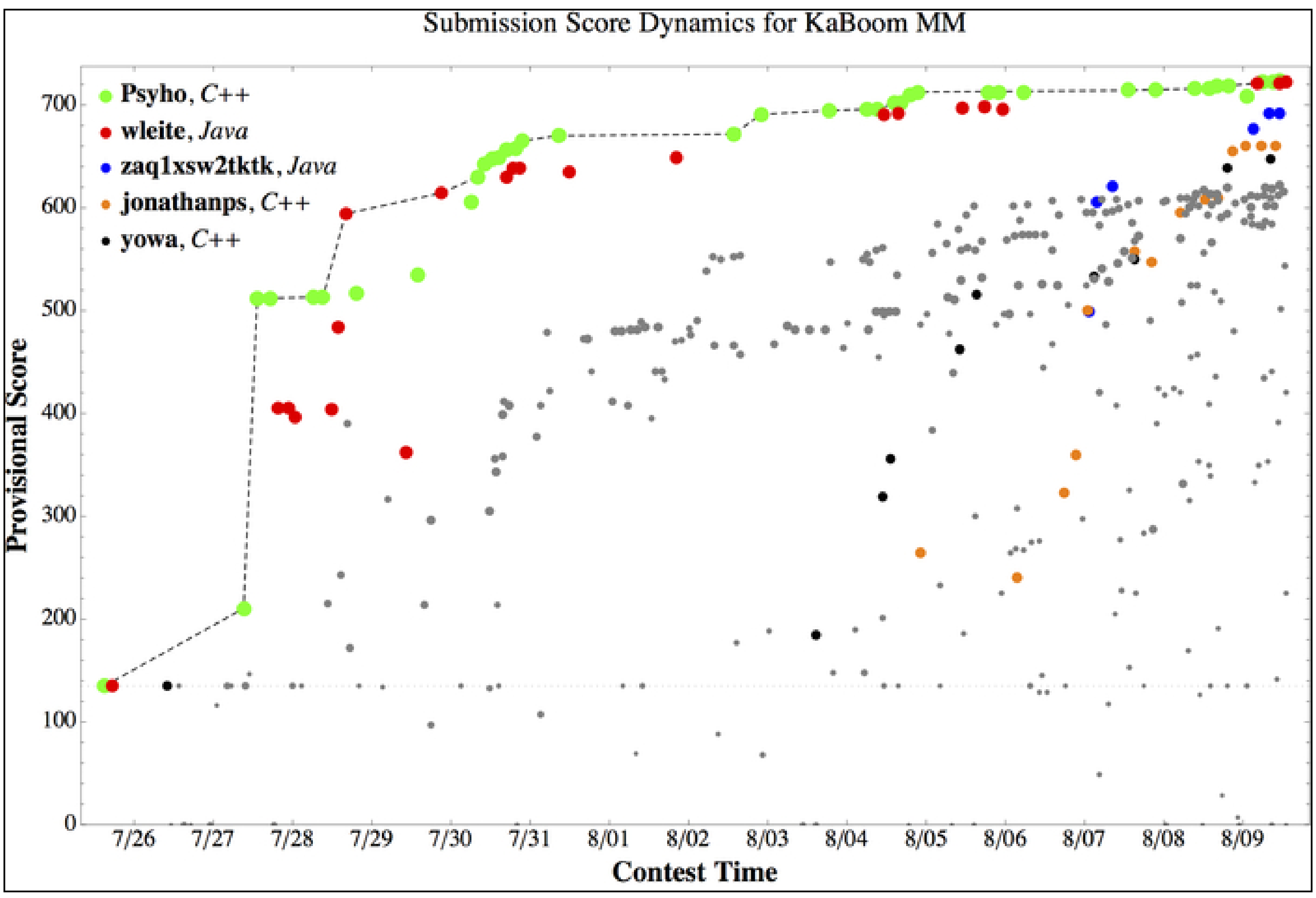
To address that, I have requested and analyzed the samples of the solution performance data that I believed cover the most of the dimensions, present in the contest. I have split my conclusions into three parts, according to the dimension types I have explored:

Part 1. ***The sensitivity of the algorithms to build-in parameters.***
In this part I analyzed whether the variation of the build-in parameters chances the algorithm's strategy and how it affects the algorithm's scoring on the tests.

Part 2. ***Performance of the algorithms as a function of asteroid parameters***
In this part I looked for whether there are any distinquishable subclasses of asteroids with certain set of parameters the algorithms show preferences for.

Part 3. ***Performance of the algorithms as a function of test parameters***
In this part I analyzed whether there are any distinguishable preferences of the algorithms to certain tests, than can differ from each by the numbers of antennas or numbers of asteroid in test.

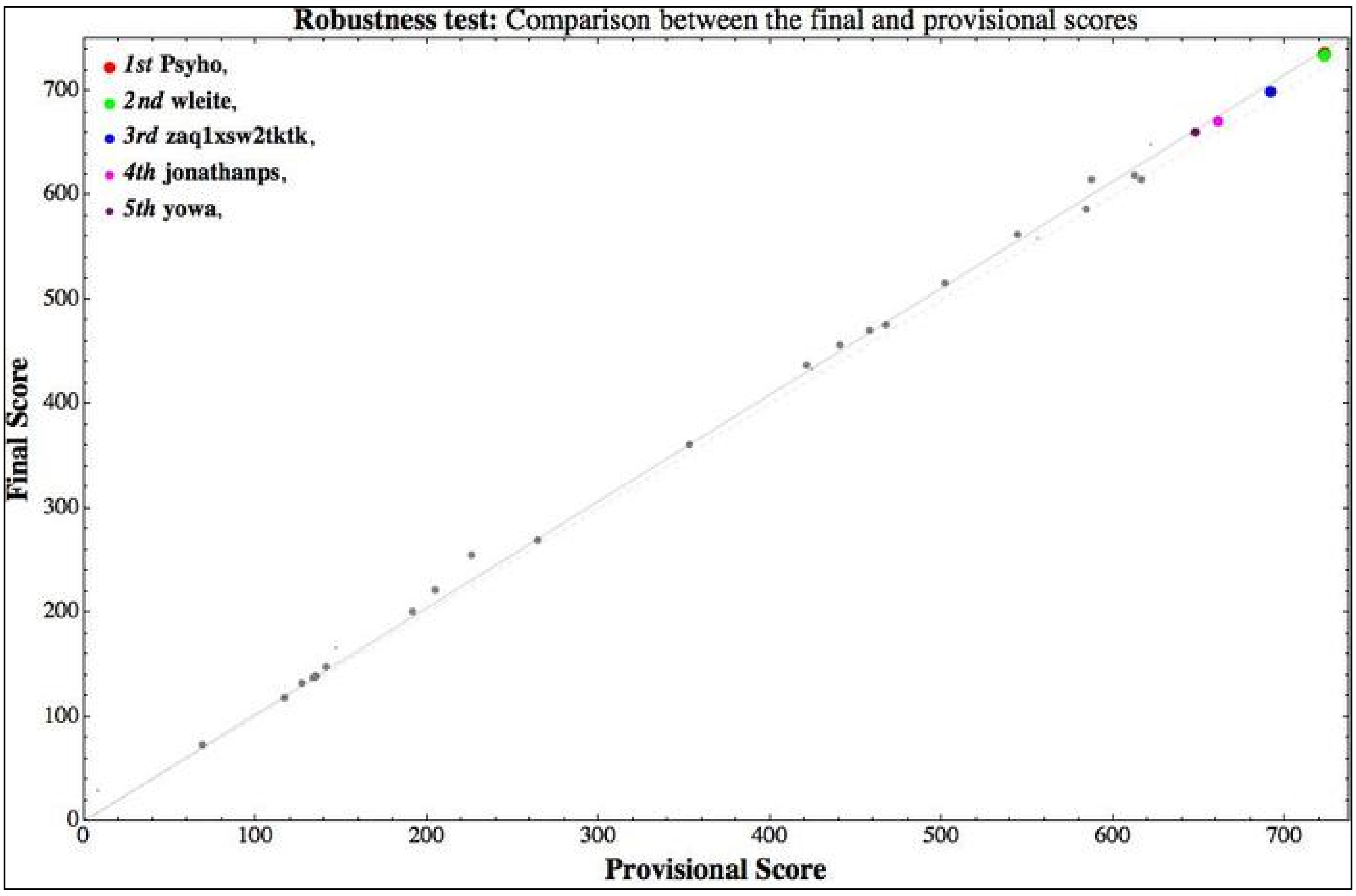
I am posting the **Part 1** and **Part 2** of the analysis below, and I will post the **Part 3** after I resolve some minor data question with Ambi.



Score_Dynamics.jpg

Rank	Handle	Provisional Rank	Provisional Score	Final Score	Language
1	Psyho	1	723.17	736.23	C++
2	wleite	2	722.45	734.69	Java
3	zaq1xsw2tktk	3	691.41	699.80	Java
4	jonathanps	4	660.59	671.08	C++
5	yowa	5	647.20	660.23	C++

LeaderBoard.jpg



Final_vs_Provisional.jpg

View all of these files at once

Discuss this message

Rinat Sergeev
Part 1. ***The sensitivity of the algorithms to build-in parameters.***

There are 12 build-in parameters in the contest specs http://community.topcoder.com/longcontest/?module=ViewProblemStatement&rd=16051&pm=13156 that decided to include into the sensitivity analysis. The list of the parameters with their names (as in specs) and their initial values is *attached*.

Methodology:

1. I selected two scoring test cases (N1 and N2) and requested to re-train and re-run the five algorithms over those tests while the input parameters have been modified (a single-parameter modification at a time).

The following parameter modifications have been applied
(multipliers to the parameter are shown):

Baseline - 1x (*no modifications*)
Critical Angle for aligned objects - 0x; 10x; (*special* 10800x);
Image Score coefficient - 0x;
Trajectory Score coefficient - 0x;
Noise Level - 1/10x; 2x
Image measurement threshold, Q_Image - 1/2x; 2x;
Lost of object threshold, Q_lost - 1/2x; 2x;
Tragectory measurement threshold, Q_tragectory - 1/2x; 2x;
Antenna Radius - 0x; 2x;
Relocation Speed - 0x;
Relocation Power - 0x;
Shileding Effect - 1/2x; 2x;
Initial Knowledge, T_min - 2x;

What combines into *2 testcases * 5 solutions * 21 cases = 210* test outputs.

2. I requested to modify the test output to retrieve more details of the algorithm's performance. I collected the following components:

- *Total score* (3 parameters):
Image Score
Trajectory Score
Energy Score

- *Antenna usage parameters* (5 parameters):
Mean and deviation of the number of the active antennas
Mean and deviation of the largest subset of antennas, tracking the same target
Number of relocations.

3. I applied highest level interpolation (linear or parabolic) to normalized scores in corresponding tests in order to compute normalized derivatives (the normalized score per normalized parameter value).

4. I compared two test cases (N1 and N2) to estimate the mean squared deviation of the normalized derivative. I computed the significance based on ration of Mean normalized derivative to it's deviation

5. I compared the cumulative values of the antenna usage between the tests in order to measure whether the algorithm modifies it's output in response to different input parameters or keeps it the same. The change in the antenna usage parameters was considered as a sign that the algorithm adopts the strategy to the parameter change.
In the case of Critical_Angle there is no difference found for small angle variations, so I used very high value of the angle (*pi*) to distinguish between the algorithms that start failing on some asteroids and those who processes it correctly.

6. I used the difference in normalized derivatives between the algorithms and their total deviations to compute whether there are parameters that could predictably change the placement of the solutions, what percent those parameters have to be changed by, and estimated the significance of the effect.

Results.

The following *five tables* (one for each solution) are showing the normalized effect of the each of the 12 parameter variations on the score of the solution, the significance of the effect, and the binary classifier of whether the algorithm's output adopts to the parameter change.

With a few exceptions, the absolute values of the majority of normalized derivatives are much less than 1, meaning that even a strong variation in the parameter value causes only small effect on the algorithm's score.

At the same time, if some algorithms differ significantly in response to the chance of one of the parameters, it may reflect either different strategies used, or potentially suboptimal response of one of the algorithms to the parameter change.

Interesting Fact -1.
Only the 4th place algorithm does the processing of the Critical_Angle correctly. Apparently, the top-submitters have not found this to impact the score significant enough to implement it. We need to keep it in mind when we move forward. May need to be added to the list of the bugs to be corrected in future.

Interesting Fact -2.
None of the algorithms have shown any noticeable effect of the "Relocation Power" on the strategy. Apparently, it is too small on the scoring scale.

I have compared the effects of the parameter variation in the next table, "*Sensitivity of scoring delta of Psyho Algorithm*", where I compared the 1st place algorithm with the other ones in respect to whether the differential effect of the parameter variation can produce a predictable effect on the scoring difference between the algorithms, comparable to the scoring difference value.

The table shows that the top-2 algorithms are very balanced over each other, as despite a very small scoring difference, there are no strong effects that can change the placements of those algorithms in predictable manner. The lower 3 algorithms are more imbalanced and show strong differential effects to the changes in the scoring structure (Image vs Trajectory) or, as in the case of 3rd place, to the changes in the measurement thresholds.

Posted on Oct 16, updated less than a minute later

Parameter_List.jpg

Psyho, 1st place, scored 1. of maximum	Initial value	Normalized effect on score	Deviation	Significance	Adopting Strategy?
Critical Angle for aligned objects	2.90888×10^{-4}	0.	0.		N
Image Score coefficient	30.	0.2452	0.0683	+++	Y
Trajectory Score coefficient	9.9206×10^{-5}	0.8116	0.0634	+++	Y
Noise Level	$1. \times 10^{-29}$	−0.1724	0.0539	---	Y
Image measurement threshold, Q_Image	$1. \times 10^{8}$	−0.1517	0.0229	---	Y
Lost of object threshold, Q_lost	2.87015×10^{5}	0.0819	0.0448	+	Y
Tragectory measurement threshold, Q_tragectory	$6. \times 10^{3}$	0.0046	0.0092		Y
Antenna Radius	11.	0.0039	0.0006	+++	Y
Relocation Speed	5.23599×10^{-3}	0.0073	0.0189		Y
Relocation Power	$5. \times 10^{3}$	−0.0013	0.0002	---	N
Shileding Effect	$1. \times 10^{-26}$	0.011	0.0171		Y
Initial Knowledge, T_min	0.1	0.0028	0.0085		Y

1_Psyho_sensitivity.jpg

wleite, 2nd place, scored 0.998 of maximum	Initial value	Normalized effect on score	Deviation	Significance	Adopting Strategy?
Critical Angle for aligned objects	2.90888×10^{-4}	0.	0.		N
Image Score coefficient	30.	0.2237	0.0493	+++	Y
Trajectory Score coefficient	9.9206×10^{-5}	0.7988	0.08	+++	Y
Noise Level	$1. \times 10^{-29}$	−0.1051	0.0406	--	Y
Image measurement threshold, Q_Image	$1. \times 10^{8}$	−0.1309	0.0144	---	Y
Lost of object threshold, Q_lost	2.87015×10^{5}	0.099	0.0718	+	Y
Tragectory measurement threshold, Q_tragectory	$6. \times 10^{3}$	−0.0113	0.0075	-	Y
Antenna Radius	11.	0.0048	0.009		Y
Relocation Speed	5.23599×10^{-3}	−0.0033	0.0006	---	Y
Relocation Power	$5. \times 10^{3}$	−0.0011	0.	---	N
Shileding Effect	$1. \times 10^{-26}$	−0.013	0.0064	--	Y
Initial Knowledge, T_min	0.1	−0.0016	0.0017		Y

2_wleite_sensitivity.jpg

zaq1xsw2tktk, 3rd place, scored 0.951 of maximum	Initial value	Normalized effect on score	Deviation	Significance	Adopting Strategy?
Critical Angle for aligned objects	2.90888×10^{-4}	0.	0.		N
Image Score coefficient	30.	0.3997	0.0724	+++	Y
Trajectory Score coefficient	9.9206×10^{-5}	0.7678	0.0454	+++	Y
Noise Level	$1. \times 10^{-29}$	−0.1206	0.0189	---	Y
Image measurement threshold, Q_Image	$1. \times 10^{8}$	−0.2095	0.042	---	Y
Lost of object threshold, Q_lost	2.87015×10^{5}	0.5592	0.0654	+++	Y
Tragectory measurement threshold, Q_tragectory	$6. \times 10^{3}$	−0.0011	0.0006	-	Y
Antenna Radius	11.	−0.0006	0.0026		Y
Relocation Speed	5.23599×10^{-3}	−0.0017	0.0014	-	Y
Relocation Power	$5. \times 10^{3}$	−0.0014	0.0001	---	N
Shileding Effect	$1. \times 10^{-26}$	−0.0034	0.0017	-	N
Initial Knowledge, T_min	0.1	−0.0001	0.0014		Y

3_zaq1xsw2tktk_sensitivity.jpg

jonathanps, 4th place, scored 0.912 of maximum	Initial value	Normalized effect on score	Deviation	Significance	Adopting Strategy?
Critical Angle for aligned objects	2.90888×10^{-4}	0.	0.		Y
Image Score coefficient	30.	1.0137	0.1899	+++	Y
Trajectory Score coefficient	9.9206×10^{-5}	0.8006	0.0276	+++	Y
Noise Level	$1. \times 10^{-29}$	-0.1149	0.0428	--	Y
Image measurement threshold, Q_Image	$1. \times 10^{6}$	-0.1748	0.0188	---	Y
Lost of object threshold, Q_lost	2.87015×10^{5}	0.1065	0.075	+	Y
Tragectory measurement threshold, Q_tragectory	$6. \times 10^{3}$	-0.0067	0.0031	--	Y
Antenna Radius	11.	0.0001	0.	+++	Y
Relocation Speed	5.23599×10^{-3}	-0.0062	0.0006	---	Y
Relocation Power	$5. \times 10^{3}$	-0.0024	0.0001	---	N
Shileding Effect	$1. \times 10^{-26}$	-0.0015	0.0062		Y
Initial Knowledge, T_min	0.1	0.	0.		N

4_jonathanps_sensitivity.jpg

yowa, 5th place, scored 0.897 of maximum	Initial value	Normalized effect on score	Deviation	Significance	Adopting Strategy?
Critical Angle for aligned objects	2.90888×10^{-4}	0.	0.		N
Image Score coefficient	30.	0.5894	0.1544	+++	Y
Trajectory Score coefficient	9.9206×10^{-5}	0.8359	0.0559	+++	N
Noise Level	$1. \times 10^{-29}$	-0.1423	0.0419	---	Y
Image measurement threshold, Q_Image	$1. \times 10^{6}$	-0.1362	0.0041	---	Y
Lost of object threshold, Q_lost	2.87015×10^{5}	0.1138	0.1046	+	Y
Tragectory measurement threshold, Q_tragectory	$6. \times 10^{3}$	-0.0028	0.0007	---	N
Antenna Radius	11.	-0.0226	0.0206	-	Y
Relocation Speed	5.23599×10^{-3}	-0.0013	0.	---	N
Relocation Power	$5. \times 10^{3}$	-0.0013	0.	---	N
Shileding Effect	$1. \times 10^{-26}$	-0.0023	0.0008	--	N
Initial Knowledge, T_min	0.1	0.0019	0.0026		N

5_yowa_sensitivity.jpg

Sensitivity of scoring delta of Psyho algorithm over (in percent to the initial parameter value)	wleite		saqlxsw2tktk		jonathanps		yowa	
	Variation	Significance	Variation	Significance	Variation	Significance	Variation	Significance
Critical Angle for aligned objects								
Image Score coefficient			28	+	10	+++	25	++
Trajectory Score coefficient							86	+
Noise Level								
Image measurement threshold, Q_Image			-71	+				
Lost of object threshold, Q_lost			10	+++				
Tragectory measurement threshold, Q_tragectory	-13	+						
Antenna Radius								
Relocation Speed								
Relocation Power								
Shileding Effect	-8	+						
Initial Knowledge, T_min								

Psyho_scoring_delta_sensitivity.jpg

View all of these files at once

Rinat Sergeev
Part 2. ***Performance of the algorithms as a function of asteroid parameters***

In this part I will demonstrate the difference between the algorithm strategies in using the antennas and preferences/performance of the algorithms on the asteroid level.

I. Antenna strategies.

Approach:
1. I have combined the tests N1 and N2 (to increase the accuracy) and collected the following information, related to the antenna use:

- Mean of the number of the active antennas
- Deviation of the number of the active antennas
- Mean of the largest subset of antennas, tracking the same target
- Deviation of the largest subset of antennas, tracking the same target
- Number of relocations

for all the 5 algorithms

2. I normalized the first 4 values on the total number of antennas, and the 5th - on the maximal number of relocations among all the algorithms and rounded that to percent.

Results:
The results are shown in the "*Antenna usage*" table below.

What is interesting, the values change quite a lot between the algorithms, suggesting different strategies. Also, the difference between 1-st and 3-rd columns (Means of Active Antennas and those, concentrated on a Single Target) is a strong measure of how often the algorithm tracks simultaneously more than one object, instead on concentrating on just one at a time. this difference also varies a lot between the algorithms.

Thus, the **1st** solution prefers to use more antennas on average than the **2nd** one, and it also simultaneously tracks more than one target more often than the others. Also, this solution uses the

least number of antenna relocations.
The **2nd** solution keeps the least number of antennas active, but varies this number a lot. Sometimes, but rare, it traces more than one object at a time. It's pattern resembles a bit the one of 4th one, but it suggests very reasonable number of relocations for such strategy.
The **3rd** solution keeps nearly all antennas active nearly all the time.
The **4th** solution never traces more than one object at a time, whereas all the other solutions try it at least sometimes. it prefers to check the targets one by one with the majority of the antennas involved. No surprise, that this solution requires to relocate all the antennas all the time!
The **5th** solution resembles the pattern of the **1st** one, but rarely allows for a second object. May be, that is the reason it lost some score?

II. Target-level analysis

Approach:
1. I requested a detailed per-asteroid output for the same tests N1 and N2 (discussed in previous post).

2. For each asteroid I collected:

- Asteroid number in a test
- Reflectivity Multiplier
- Science Score Multiplier
- Visible Time
- Distance to the asteroid, averaged with the weight 1/Distance^4
- Score per asteroid (Image, Trajectory)

3. The outputs of the two tests were united

4. I run the visual correlation search for the most representative parameter combination that can illustrate the differences between the algorithm's preferences.

Results:
Among all the combinations, the most illustrative one I found was a correlation between "Total Score", earned on the object, and a combined variable "Expected Science Return" that was calculated by formula:

"Expected_Science_Return" = "Reflectivity_Multiplier" x "Science_Score_Multiplier" x "Visible_Time" / (Average Distance)^4

The expression above has a transparent meaning of expected average of returned signal multiplied by it's scientific value, or, simply, it is proportional to a scientific value we expect to collect from a given object over the time it is visible.

On the figure "*Performance per target*" below I am comparing the scoring return vs Expected_Science_Return for the 5 algorithms.
What is interesting, there is a very strong correlation between the algorithms for the objects with low Expected_Science_Return, in fact, for the majority of the asteroids there all the algorithms return the same scoring (all 5 dots coincide), means that for those targets there is a singe optimal strategy, shared by all algorithms. As the Science_Return grows, the difference between algorithms appear.

To illustrate the differences for asteroid with high "Science_Return", I added the cleaned version of the same figure "*Performance per target cropped*": it is rescaled and shown for the top-2 algorithms only.

In this representation the difference between the top-2 algorithms become clearly seen: the **1st** place algorithms concentrates on the targets with higher Scientific_Return, while the **2nd** place accumulates more score from the less valuable targets.

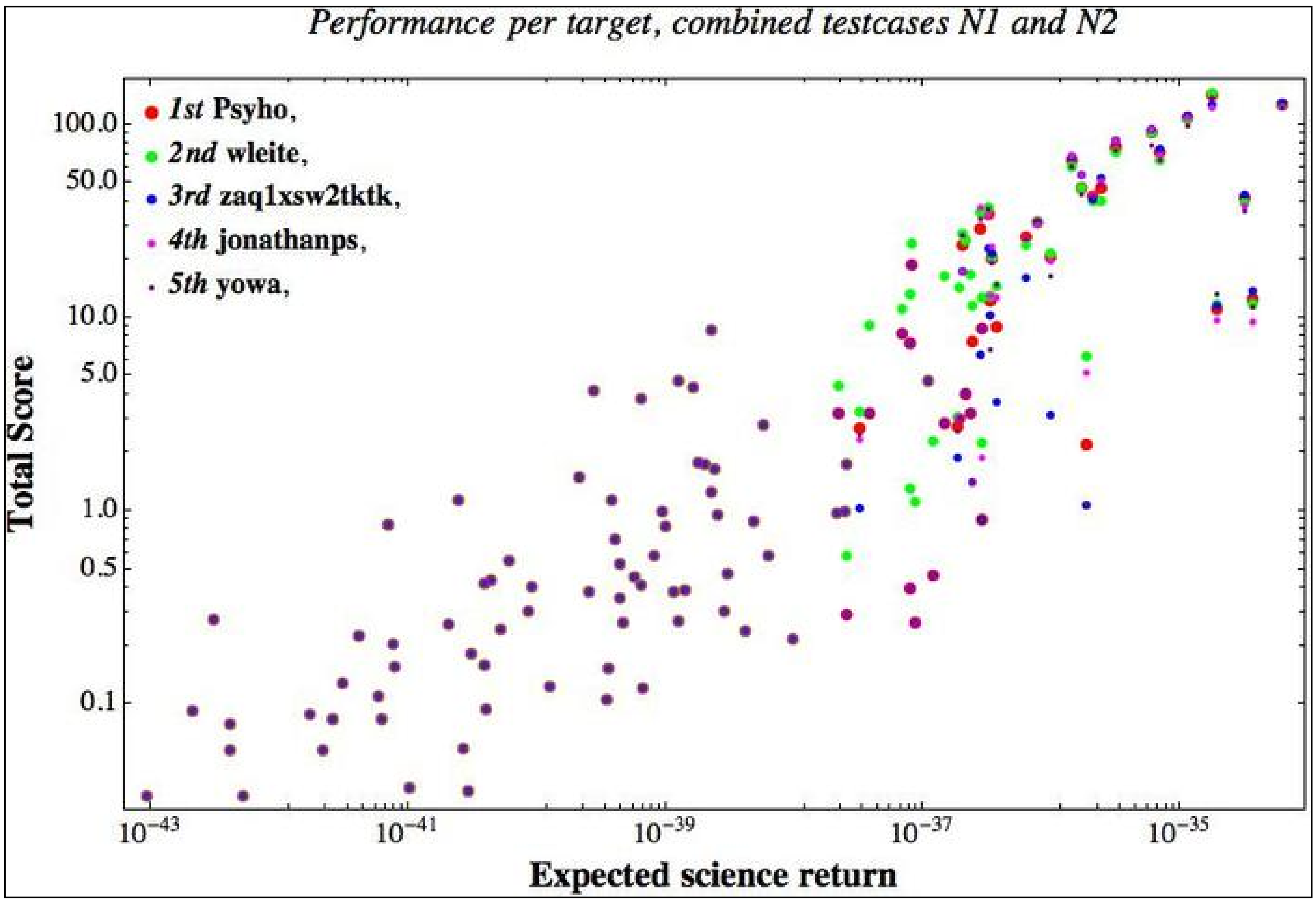
The same difference can be shown by just direct target-based comparison between the top-2 solutions (see the next figure - "*Comparison of the scores, per target*"). The plot clearly distinguishes the preferences of the Psyho's and wleite's algorithms - while the first one wins on the objects with highest scores, the second one collects more score from the low-scored ones.

If we link this conclusion with the data from antenna usage section at the beginning of this post, we can safely guess that the **1st** place solution tries leave some antennas to track the most valuable objects even if there are other objects present, whereas the **2nd** place solution tries to switch between the objects more often.
Posted on Oct 16

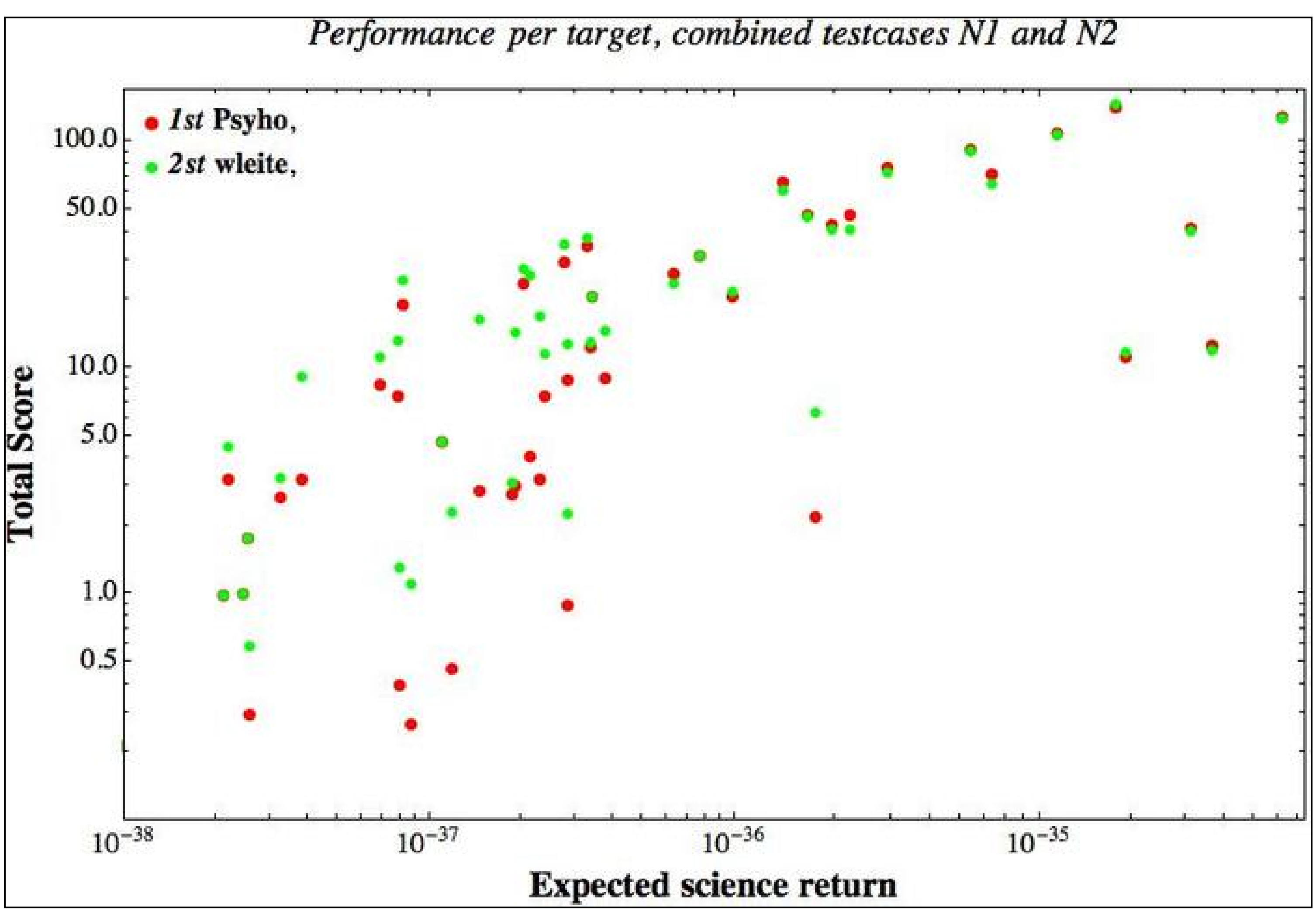
Posted on Oct 16

Antenna usage, percent of the total	Active Antennas, % Mean	Active Antennas, % Variance	On Single Target, % Mean	On Single Target, % Variance	Relocations % of the max
Psyho	89	17	69	21	48
wleite	76	27	71	28	60
zaq1xsw2tktk	95	12	75	20	68
jonathanps	82	27	82	27	100
yowa	86	17	83	20	54

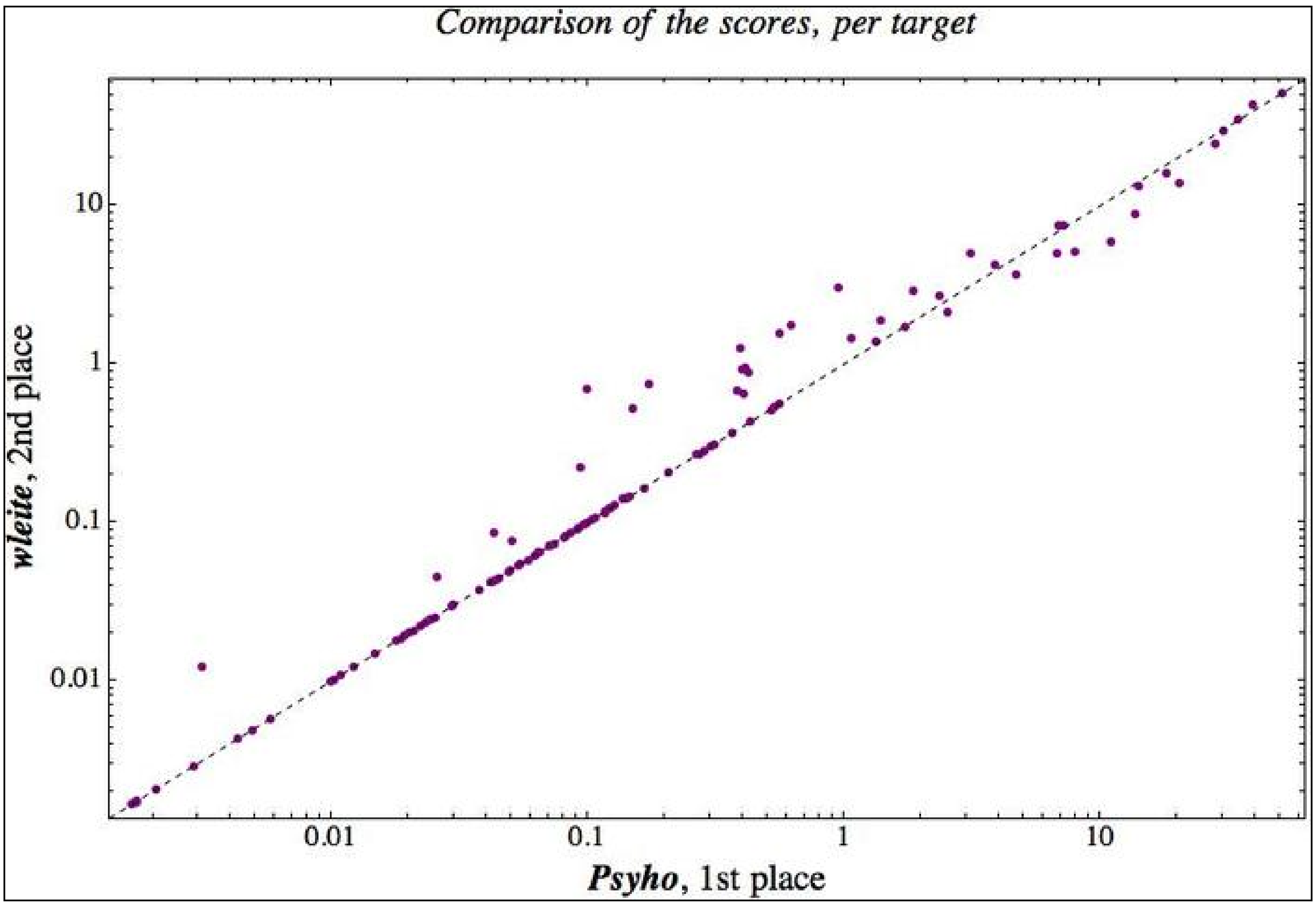
Antenna_Use.jpg



Score_vs_Science_Return.jpg



Score_vs_Science_Return_cropped.jpg



Psyho_vs_wleite.jpg

View all of these files at once

Rinat Sergeev
Hello again! Here is the missing part 3 of the sensitivity analysis.
Part 3. *Performance of the algorithms as a function of test parameters*

In this part I explore a high-level performance of the solutions across the different tests.

Approach:
For each of the 5 algorithms and 1000 system tests, I have measured
A. Three parameters characterizing the test:
1. Number of antennas in radar
2. Maximal emission power of antenna
3. Number of asteroids in test
Below attached are ***three histograms***, illustrating the distribution of each of these three parameters in the system dataset.

B. Two parameters of the algorithm performance:
1. Score on a test
2. Computational time

Then I use the results of statistical analysis of **B** as a function of **A** to
illustrate algorithms' performance on and preferences to the tests of different conditions.

I. Scoring (quality) of the algorithms on different tests

The ***first table*** after the histograms shows the coefficients of a linear regression model, fitted to the algorithms' scores:

Score = ***ScoreMean*** + **x** ***DAntNum*** + **y** ***DAntPower*** + **z** ***DAstNum***
DAntNum = ***AntennaNumber*** / ***MeanAntennaNumber*** - 1
DAntPower = ***AntennaPower*** / ***MeanAntennaPower*** - 1
DAstNum = ***AsteroidNumber*** / ***MeanAsteroidNumber*** - 1

The first numerical column shows an extrapolated score (***ScoreMean***) on an average test with values of the Antenna Number (***MeanAntennaNumber*** = **14.065**), Antenna Power (***MeanAntennaPower*** = **1.115 10^6**), and Asteroid Number (***MeanAsteroidNumber*** = **50.087**) equal to the mean values among all the tests. The next three columns show the values for corresponding regression coefficients **x**, **y**, and **z**. The simplest way to read the table is that the value of each regression coefficient estimates the change in the score that would occur if the value of the corresponding parameter increased by 100%.

Additionally, the data on the test scores is illustrated by the ***next three plots***, showing the scores of the algorithm as a function of each of the three parameters, adjusted by the values of the other two.

The table shows, that neither the number of antennas in radar not the peak antenna power have a high impact on a score. Indeed, doubling each of these parameters would only add ~5% to the scoring output, what is, to remind, a direct measure of the value of retrieved information. On the other hand, increasing the asteroid load twice nearly doubles the score.
This is a strong sign that the asteroid load, used in the tests, was not actually high for the suggested radar parameters, and the peak tracking capability of the radar is at least by order of magnitude (my estimate is ~20 times) higher.

The other conclusion from the table is that the solutions respond differently to the variation of the test parameters. Thus, the solution **N2** (***wleite***) tends to be more powerful than the solution **N1** (***Psyho***) when the complexity of the test increases (larger radar, higher energies, more asteroids).

Thus, the ***next figure*** (*Comparison of the scores, per test*) shows a direct comparison of the **N2** vs **N1** test scores across all tests. The plot conforms a general trend of the solution **N2** outperforming the solution **N1** on the tests with higher scores. In fact, it even shows the solution **N2** outperforms the solution **N1** on the unsignificant majority of the tests (*509* out of *1000*), but allows for some strong "outliers" - a few tests when **N2** receives much less score than **N1**.

II. Speed of the algorithms on different tests

In a similar way, the ***second table*** shows the coefficients of a linear regression model, fitted to the algorithms' computational time on a test:

CompTime = ***CompTimeMean*** + **x** ***DAntNum*** + **y** ***DAntPower*** + **z** ***DAstNum***
DAntNum = ***AntennaNumber*** / ***MeanAntennaNumber*** - 1
DAntPower = ***AntennaPower*** / ***MeanAntennaPower*** - 1
DAstNum = ***AsteroidNumber*** / ***MeanAsteroidNumber*** - 1

Several observations here:

1. Each of the top (**N1** or **N2**) algorithms is much slower than the algorithms **N3-5**, what indirectly correlates with the complexity of the model implemented. In fact, the difference in computational times may be as high as 30 times (**N5** vs **N2**, see the first numerical column of the table).

2. The **N2** algorithm is, in fact, the slowest among all, being ~1.6 times slower than **N1**. This is illustrated by the **last figure** (*Comparison of the adjusted algorithms' speeds*) that shows the distributions of the algorithm's computational times speeds relative to the one of **N1**, adjusted on all there test parameters.

3. The computational time for the majority of the algorithms scales up approximately as (*AsteroidNumber*)*(*AntennaNumber*), however with some differences between the algorithms. Thus, the computational time for algorithm **N1** is a sublinear function of the antenna number, whereas the one for **N4** - super linear.
On the other hand, the computational time for **N2** is less sensitive to the number of asteroids than the one for **N1**.

THE CONCLUSION

The solutions **N1** and **N2** should be recognized as the two of equal strength and completely different approaches. Both solutions are applicable and optimized to a broad range of parameters and conditions (with the exemption that none of them has implemented a proximity constraint, however, easy introducible and being implemented in **N4**).

Also, the presented performance/sensitivity analysis, while being entirely data-driven, correlates well with the descriptions of the algorithms, provided by the contestants:

N1 has implemented a complex, hierarchically structured, solution that prioritizes the objects in a non-trivial way, but with a strong emphasis on their value, making sure that the most valuable objects are not missed. This correlates with higher scores of this solutions on more valuable targets, high stability of this solution (it always ensures good output from the test), and low dependence of the solution computational time on the number of the antennas.

N2 has implemented a truly generic approach when the best tracking strategy is picked through adjusted Monte-Carlo approach in the space of all possible strategies. Potentially, it allows for any type of strategy, including a simultaneous tracking of multiple asteroids. At the same time,

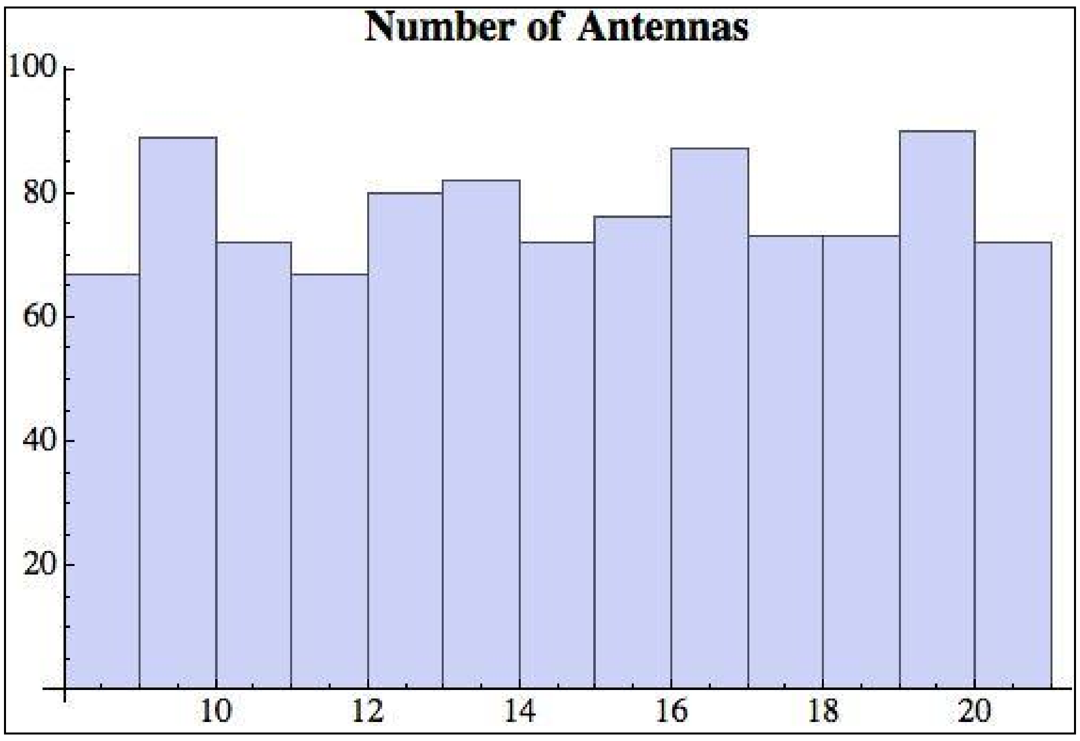
surprisingly, a multiple-object tracking is suggested rather rare, than often, by this solution (see the antenna usage statistics from previous post).
Thus solution is more demanding in the sense of computational time, it also allows random outliers (suboptimal choices) due to limitations of randomized search. At the same time, it becomes more powerful when scored on low-output objects, or on the complex tests. In fact, if we increase the computational time allowance, the solution **N2** is expected to outscore the solution **N1**. Moreover, given that we used too low frequency of the objects as compared to the tracking capacity of the radar in the test (see the first part of this post), setting the former to higher values would shift the balance further from **N1** to **N2**.
Also, there is a potential possibility to improve the solution **N2** by using the output strategy of the solution **N1** as a starting seed for optimization.

N3 optimizes the strategy over targeting up to 2 asteroids per time. It uses simplified, hard-coded expressions for the estimate of the scoring gain, what may explain a suboptimal response of this solution to rebalancing of the scoring formula (see the last paragraph of the Part-1 post). The solution also makes plenty of assumptions and simplifications (induced noise, speed of light, relocation speed, etc..) what makes it reasonably fast to compute, but somewhat unstable to parameter change.

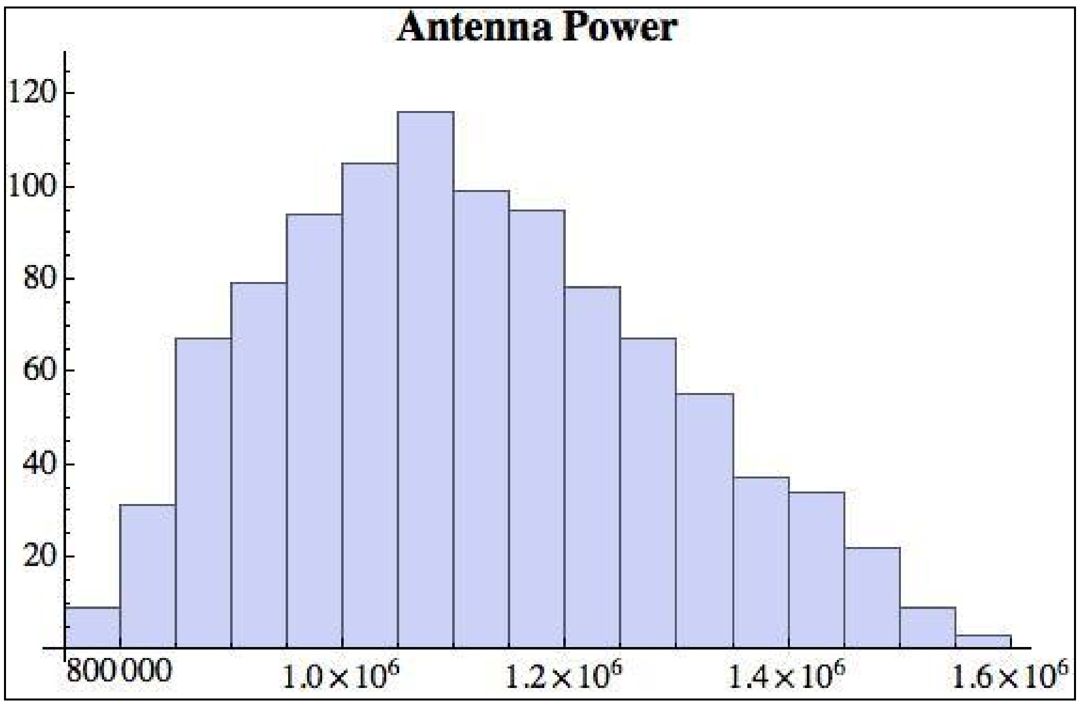
N4 implements a complex and time-consuming approach for tracking one object at a time. The solution implements a comprehensive forward-tracing scheme (modeling the known future in advance, somewhat, similar to **N1**), and uses a lot of analytic simplifications (approximation formulae for optimal signal power level or for returned signal value). It lacks a good multi-dimensional optimization scheme, what results in suboptimal scoring, but it is the only solution that implemented a target proximity constraint. The other solutions have either forgotten about it, or purposefully missed it due to absence of such cases in the example tests.

N5 Implemented a very fast submission that uses large time intervals to reduce the computational time, but sub-divides them if needed. It primarily targets one object at a time, but allows those antennas that cannot emit towards the main object, find a second-main one. A computational speed is definitely a strong side of this simple, but elegant, approach.

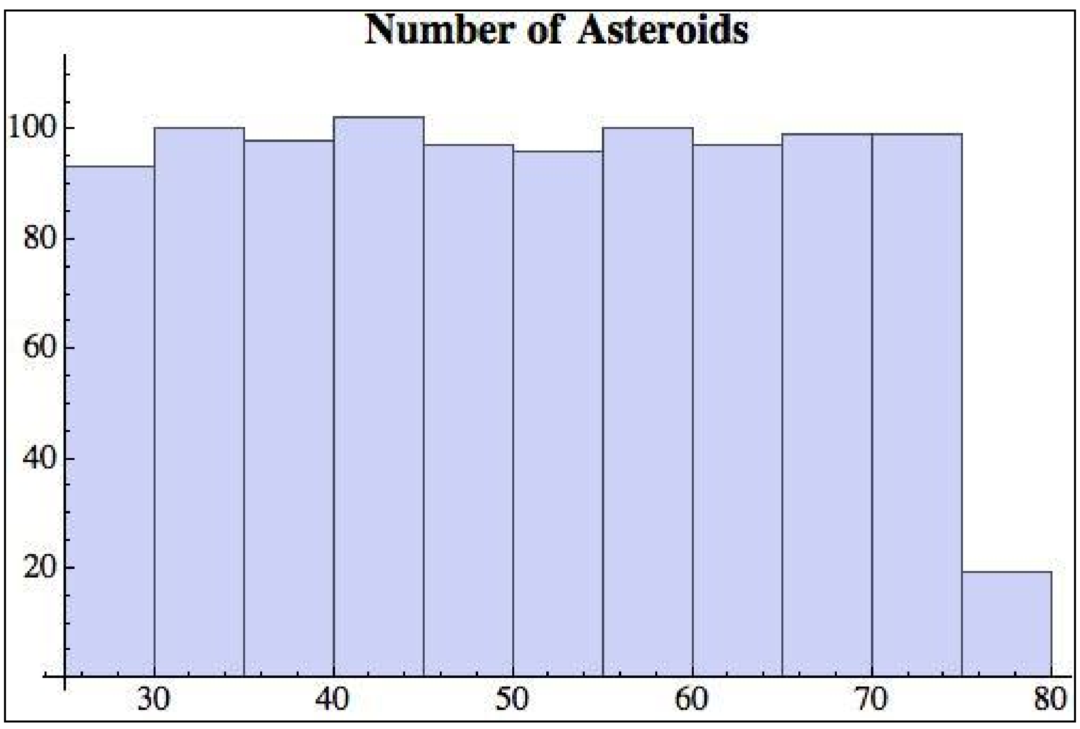
Posted on Nov 3



AntennaNum_Histogram.jpg



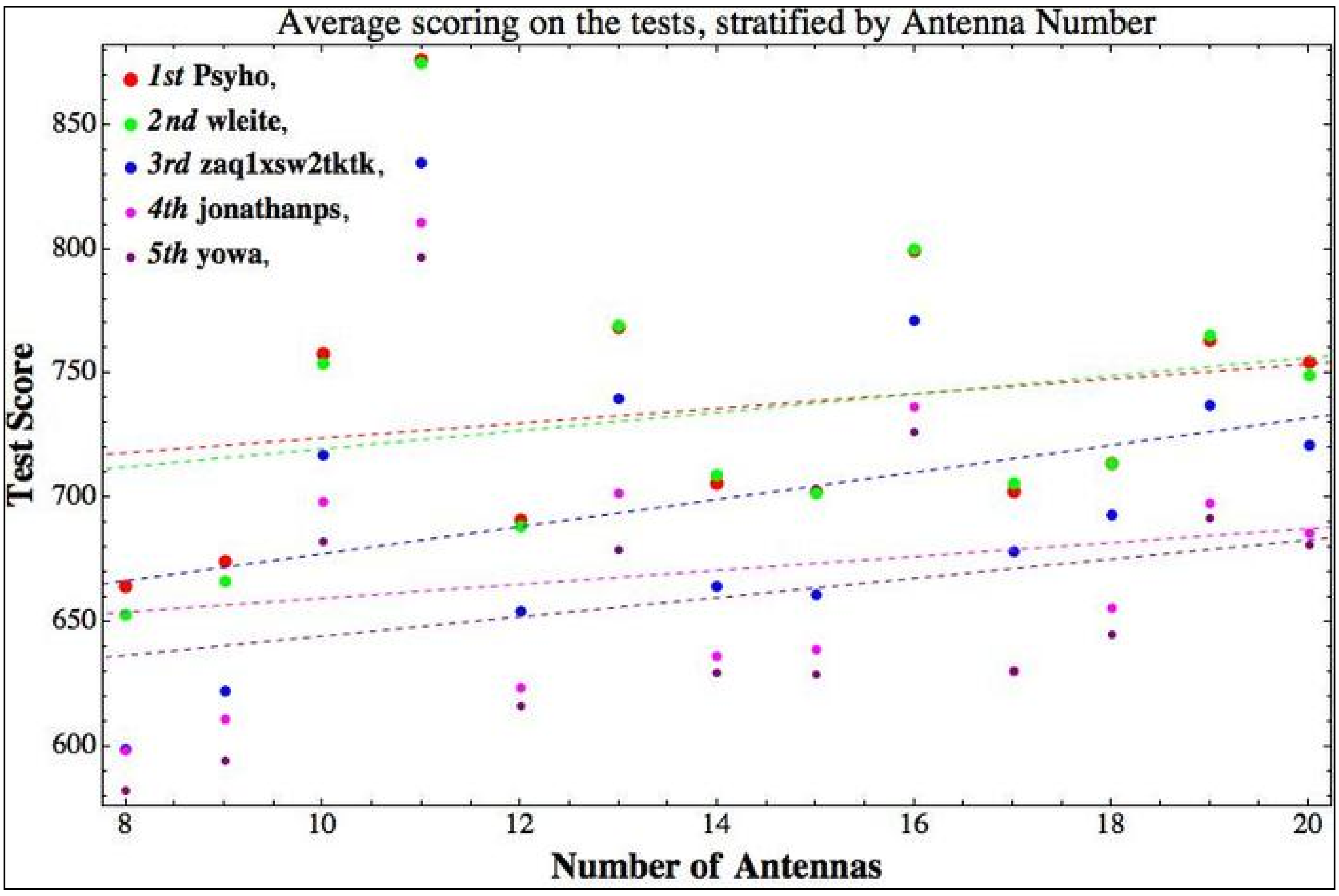
AntennaPower_Histogram.jpg



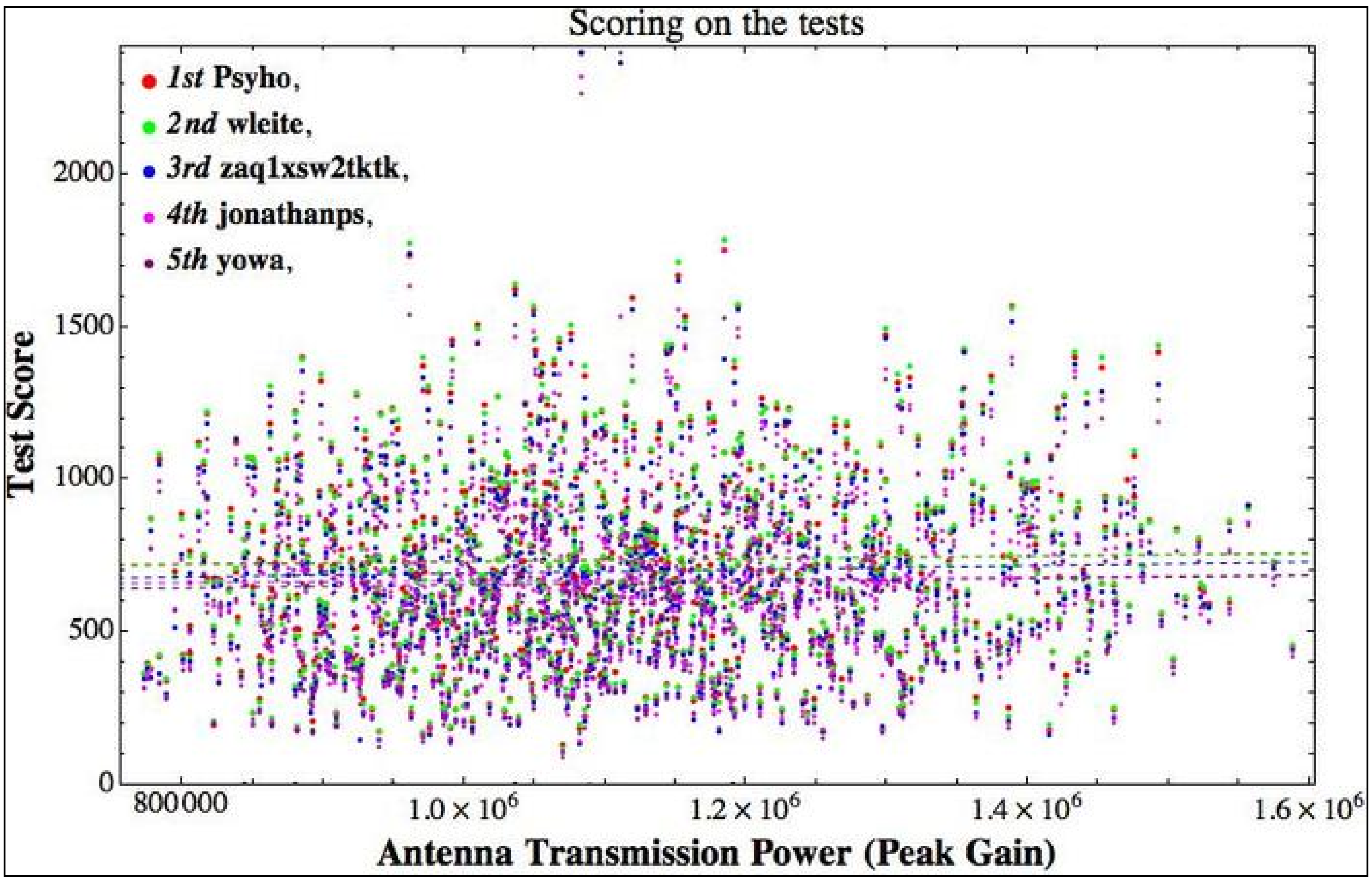
AsteroidNum_Histogram.jpg

Solution	Score in averaged test	Normalized Effect of Antenna Number	Normalized Effect of Antenna Power	Normalized Effect of Asteroid Number
Psyho	736.232	38.4558	29.0714	711.243
wleite	734.692	48.0599	39.3988	721.586
zaqlxsw2tktk	699.798	73.2484	49.8363	689.388
jonathanps	671.075	36.3011	19.746	642.182
yowa	660.228	51.2838	42.7947	636.159

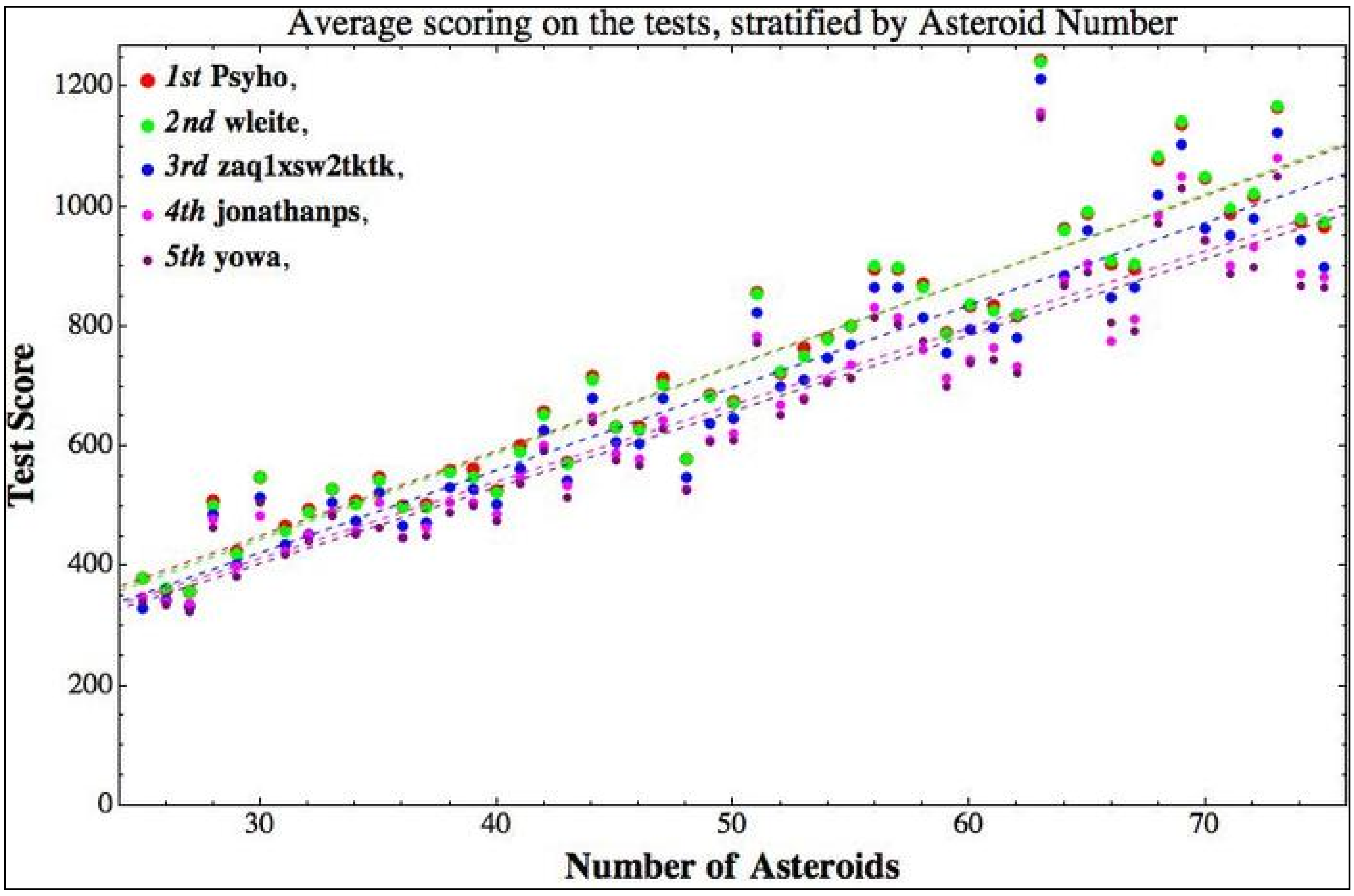
ScoreRegression.jpg



TestScore_AntennaNumber_Adjusted.jpg



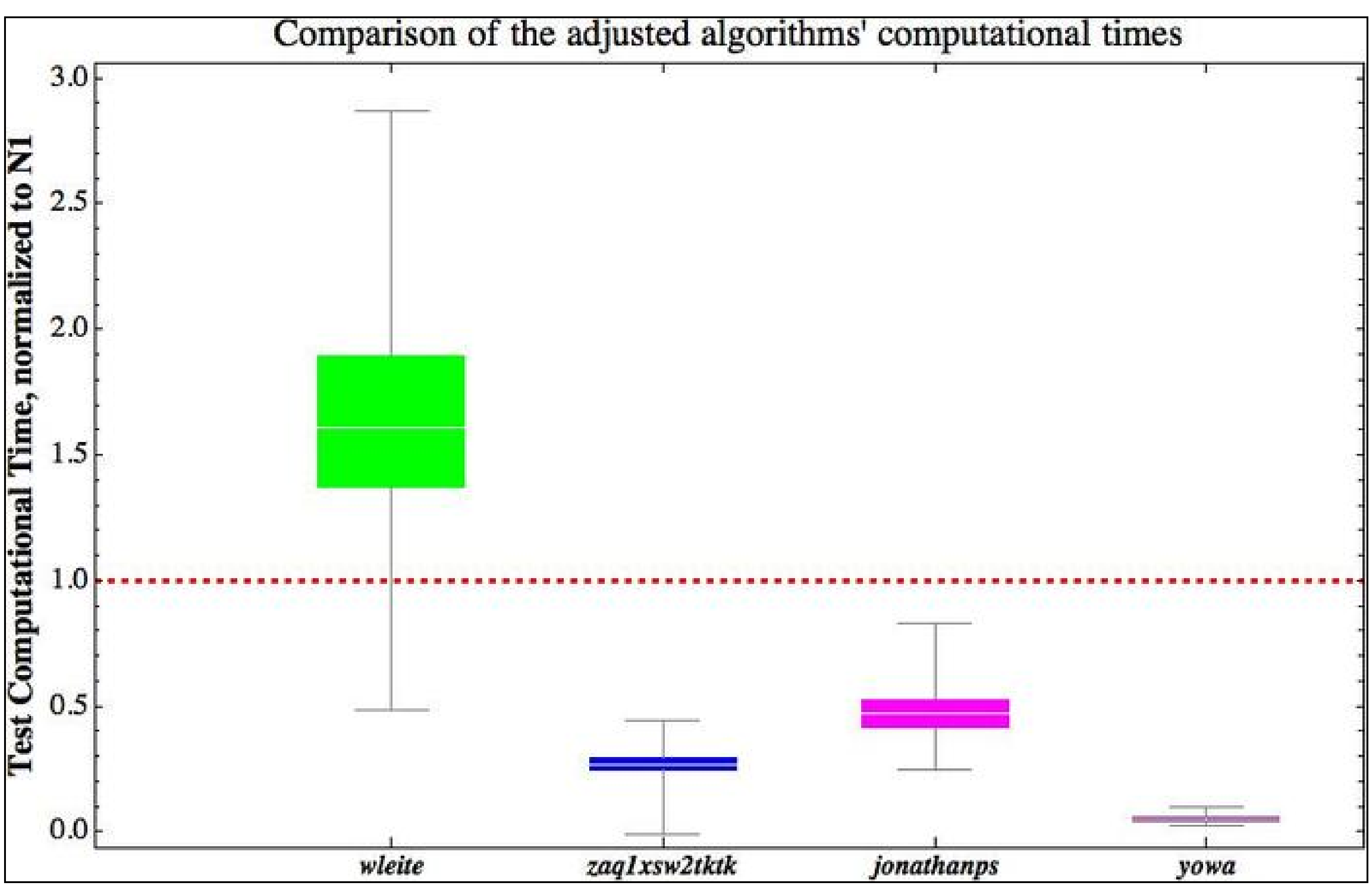
TestScore_AntennaPower_Adjusted.jpg



TestScore_AsteroidNumber_Adjusted.jpg

Solution	Computational time in averaged test	Normalized Effect of Antenna Number	Normalized Effect of Antenna Power	Normalized Effect of Asteroid Number
Psyho	10 309.1	2262.97	423.514	14 740.6
wleite	16 728.2	15 607.2	2882.74	12 843.1
zaq1xsw2tktk	2775.18	1604.5	-37.8537	2366.57
jonathanps	4809.66	6785.68	-205.311	4648.78
yowa	528.316	656.507	-13.3818	376.808

CompTimeRegression.jpg

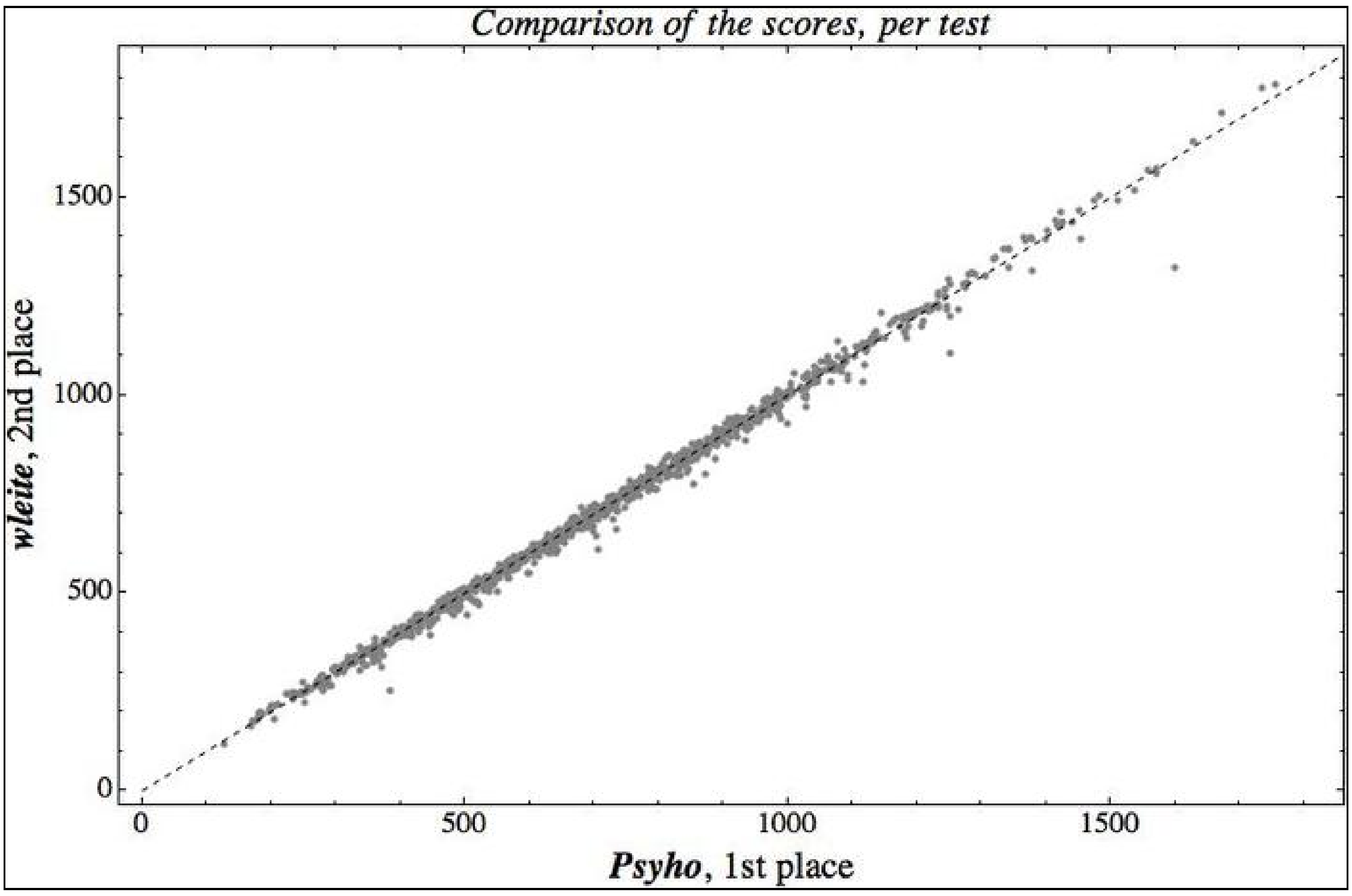


AlgorithmsSpeedsBar.jpg

View all of these files at once

Rinat Sergeev

Posted on Nov 6



wleite_vs_Psyho_tests.jpg

By-the-minute history for this message...