CA15140 Training School November 25, 2019

Benchmarking Iterative Optimization Heuristics with IOHprofiler

1 IOHexperimenter

1. Could you implement the so-called flip_l operator? It is described below:

Algorithm: flip_ℓ chooses ℓ different positions and flips the entries in these positions.

Input: $x \in \{0,1\}^n$, $\ell \in \mathbb{N}$;
Select ℓ pairwise different positions $i_1, \ldots, i_\ell \in [n]$ uniformly at random;
$y \leftarrow x$;
for $j = 1, ..., \ell$ **do** $y_{i_j} \leftarrow 1 - x_{i_j}$;

2. Based on the flip_l operator. Please try to implement the *Randomized Local Search Algorithm* in `R`. The pseudo-code of this algorithm is given as follows:

Algorithm: Randomized local search (RLS)

Initialization: Sample $x \in \{0,1\}^n$ uniformly at random and evaluate $f(x)$;
Optimization: **for** $t = 1, 2, 3, \ldots$ **do**
  create $x^* \leftarrow \text{flip}_1(x)$, and evaluate $f(x^*)$;
  if $f(x^*) \geq f(x)$ **then** $x \leftarrow x^*$;

3. Please try to implement the $(1+\lambda)$ Evolutionary Algorithm in `R`. The pseudo-code of this algorithm is given as follows.

Algorithm: The $(1 + \lambda)$ EA with static mutation rates

Initialization: Sample $x \in \{0,1\}^n$ uniformly at random and evaluate $f(x)$;
Optimization: **for** $t = 1, 2, 3, \ldots$ **do**
  for $i = 1, \ldots, \lambda$ **do**
    Sample $\ell^{(i)} \sim \text{Bin}_{>0}(n, 1/n)$;
    create $y^{(i)} \leftarrow \text{flip}_{\ell^{(i)}}(x)$, and evaluate $f(y^{(i)})$;
  $x^* \leftarrow \arg\max\{f(y^{(1)}), \ldots, f(y^{(\lambda)})\}$ (ties broken by selecting the first max $f(y^{(i)})$);
  if $f(x^*) \geq f(x)$ **then** $x \leftarrow x^*$;

Hint: the $\text{Bin}_{>0}(n, 1/n)$ operator can be implemented using `R` function `l <- rbinom(1, n, 1/n)`

4. Could you benchmark the algorithm you have implemented on the *Pseudo-Boolean Optimization* (PBO) suite from **IOHexperimenter**, with the following setting:
    - $d = 10, 100$
    - the number of instances 20
    - on test functions $F1, F4, F13$

2 IOHanalyzer - Graphical User Interface

This exercise consists of several questions to help you get familiar with **IOHanalyzer**.

- The online version can be accessed through `http://iohprofiler.tutorial.net`
- If you want to install this package and use it locally, please make sure you have a working `R` environment. The installation guide can be found at `https://iohprofiler.github.io/IOHanalyzer/`.

Throughout the exercise, a data set called "2019gecco-ins1-11run.rds" will be used and is already hosted for the online version. For the local usage, it can be downloaded from `https://github.com/IOHprofiler/IOHdata`.

1. Please navigate to the **Data Upload** section and select the required data set in **Load Data from Repository** box. You should observe a list of loaded data set in the as follows:

2. Now please navigate to the **Fixed-Target Results** section and select the subsection **Data Summary**. Please try to figure out the following question using the table provided in this subsection:
    (a) What is the largest observed budget value across all algorithms in 625D on function $f1$?
    (b) For 625D and function $f19$, what is the number of runs that algorithm $(1 + 1)$ $\text{EA}_{>0}$ hits its best function value?
    (c) For 64D and function $f21$, what is the *worst recorded function value* of algorithm gHC (greedy Hill-Climber)? What is the *worst reached function value* then? and what is the difference between those two values?

(d) For 100D and function $f23$, what are the mean, median, standard deviation and *expected running time* (ERT) of algorithm $(1+1)$ fGA (fast GA) at target function value -1? What is the ERT and success rate of this algorithm at its best achieved function value?

(e) For 625D and function $f4$, what is the running time sample of algorithm RLS (Randomized Local Search) at its best achieved function value?

3. Now please select subsection **Expected Runtime**. Please try to figure out the following question using the functionality provided in this subsection:

   (a) For 625D and function $f1$, please draw the ERT curve of three algorithms: (1+1) $EA_{>0}$, (1+10) $EA_{>0}$ and (1+1) fGA. Please set the range of x-axis using the smallest and largest target value observed among all three algorithms.

   (b) Please create a figure showing the ERT-curves of all $625D$ functions for the previously mentioned algorithms.

   (c) Please create a radarplot showing the ERT-based ranking of the $(1+1)$ fGA, RLS and gHC on all $625D$ functions. Please download a png version of this plot.

4. Next, please select the **Probability Mass Function** subsection, and use it to answer the following questions:

   (a) For function $f17$ in $625D$, please create a figure showing histograms of the number of function evaluations needed to reach a target value of 400 for all variants of the $(1+10)$ EA-variants. Please ensure that the bin sizes allow for an easy visual comparison among the different subplots. Please download the resulting figure in svg-format.

   (b) Using the same function and algorithms, please create a plot showing the probability mass functions of the required number of function evaluations needed to reach a target of 450. Please make sure that the individual runtime samples are shown clearly in the figure, scale the y-axis logarithmically and download the resulting plot as a pdf.

5. For the final section of the **Fixed-Budget** perspective, please switch to the **Cumulative Distribution** tab and try to answer the following questions:

   (a) Using function $f3$ in $64D$, please create a plot showing the ECDF-curves for target value 1500 for the RLS and $(30, 30)$ vGA algorithms, and scale the x-axis logarithmically.

(b) Using function $f5$ in $625D$ and the same algorihtms, create an ECDF-curve for the following target-values: 362, 387, 412, 437, 462, 487, 512, 537 and 562.

(c) Using the same algorithms, for all functions in $625D$, which targets are generated in the all-functions aggregated ECDF plot? Change the three first target values for $f1$ to 500, 550 and 600 respectively and refresh the figure.

(d) For function $f10$ in $64D$, create a radarplot showing the area under the ECDF-curve of the variants of the $(1+10)$ EA algorithm and dowlnoad a png version of this plot.

6. Please switch to the **Fixed-Budget Results** section and select the subsection **Data Summary**. Please try to solve the following questions using the functionality available in this tab:

   (a) What is the budget used for the $(30, 30)$ vGA on function $f5$ in $100D$?

   (b) For function $f19$ on $100D$, what is the mean target hit for the RLS algorithm after 100 function evaluations? How does this compare to the other algorithms? Download a csv-table showing this comparison.

   (c) For function $f2$ on $16D$, how many algorithms have hit a mean target value of 16 after 1000 function evaluations? Try to download a table showing only the data for these algorithms in a tex-format.

   (d) For function $f17$ on $100D$, what are the individual target values hit after 1, 15.001, 30.001, 45.001 and 60.001 function evaluations respectively. Please download a table showing this information.

7. Now please change to the **Expected Target Value** subsection and try to answer the following questions:

   (a) For function $f9$ in $100D$, create a plot showing mean target value for the budgets between 100 and 100.000. Show the mean and the standard deviation for the $(30, 30)$ vGA and all $(1 + 10)GA$-variants. Download this plot in png-format.

   (b) Create plots comparing the mean target value for both $(1 + 1)$-based algorithms on all functions in $100D$. Download a pdf showing all of these plots in one figure.

   (c) Create a radarplot showing the ranking of $(30, 30)$ vGA, RLS and gHC on all $625D$ functions. Change the budget-value for which the ranking is generated for $f1$, $f2$ and $f3$ to 100 and refresh the figure. Download the resulting figure in pdf-format.

8. Next, please navigate to the **Probability Density Function** and try to answer the following questions:

   (a) For function $f11$ on $64D$, create a figure showing the distribution of the reached target values after 1024 function evaluations for all algorithms. Ensure that the histograms are using bin-sizes which allow for easy visual comparison.

   (b) For function $f21$ on $625D$, create a figure comparing the empirical probability density functions for all variants of the $(1+10)$ GA at a budget of 10.000 function evaluations. Ensure that the individual runtime samples are shown. Download this plot as a png-file.

9. For the final part of the fixed-budget section, please open the **Cumulative Distribution** section, and try to answer the following questions using this section:

   (a) First, for function $f18$ in $64D$ create a figure showing the ECDF curves for the RLS, gHc and $(1+1)$ EA at a budget of 1000.

   (b) For this same function and the same algorithms, create an ECDF-curve using the following budget-values: 1, 10.001, 20.001, 30.001, 40.001, 50.001, 60.001, 70.001, 80.001, 90.001, 100.001, 110.001, 120.001, 130.001, 140.001 and 150.000.

   (c) Compare the area under the ECDF-curve for function $f20$ in $100D$ for all variants of $(1+10)$ EA. Use a maximum budget of 1.000 and set the step size to 100, such that there are exactly 10 targets generated.

10. The final main section of the IOHanalyzer can be used to investigate recorded parameter values. To use this functionality, please navigate to the **Algorithm Parameters** section. Using this tab, try to answer the following questions:

    (a) For function $f19$ in $100D$, create a figure showing the values for the l and mutation_rate parameters for the $(1+(\lambda,\lambda))$ GA and the $(1+1)$ fGA. Hide the standard deviations and scale the x-axis linearly. Please download the resulting figure as a pdf.

    (b) For the same function, what is the mean value of λ for the $(1+(\lambda,\lambda))$ GA when target value 180 is reached?

(c) For function $f1$ in $625D$, what are the mutation rates during the individual runs of the $(1 + 10)$ EA variants when a target value of 600 is reached? Please download this table in tex-format.