Runtime Analysis of a Population-based Evolutionary Algorithm with Auxiliary Objectives Selected by Reinforcement Learning

Supplementary Materials: Experiments

Denis Antipov
ITMO University
Saint-Petersburg, Russia
antipovden@yandex.ru

Arina Buzdalova
ITMO University
Saint-Petersburg, Russia
abuzdalova@gmail.com

Andrew Stankevich
ITMO University
Saint-Petersburg, Russia
stankev@gmail.com

ABSTRACT

This document contains the description of the experiments and the discussion of their results that supplement the main theoretical results on the runtime of the $(2 + 2\lambda)$-EA+RL.

ACM Reference Format:
Denis Antipov, Arina Buzdalova, and Andrew Stankevich. 2018. Runtime Analysis of a Population-based Evolutionary Algorithm with Auxiliary Objectives Selected by Reinforcement Learning: Supplementary Materials: Experiments. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3205651.3208231

1 EXPERIMENTS

For our empirical research we ran the algorithm on both considered problems with parameters $k \in [2..6]$ and $n \in [20..100]$ with step 10. For each set of parameters we performed 800 runs that were executed in parallel in 8 threads (100 runs per each thread). We used the parameter of the forgetting mechanism $C = 450$.

The plots in Fig. 1, Fig. 2 and Fig. 3 illustrate the results of the experiments. Note that all the plots in Fig. 2 and Fig. 3 have a logarithmic scale over the y-axis and linear scale over the x-axis.

In Fig. 2 we see that on the (XdivK, OneMax) problem the $(2 + 2n)$-EA+RL is not far from the best algorithms for $k \leq 4$. For $k = 5$ and $k = 6$ it outscores other algorithms, when $n \geq 30$. On the (XdivK, OneMax, ZeroMax) problem the $(2 + 2n)$-EA+RL is also not far from other algorithms for $k \leq 4$. However, it is the best algorithm only when $n \geq 90$ for $k = 5$ and when $n \geq 70$ for $k = 6$. It is notably that for $k = 5$ and $k = 6$ the $(2 + 2n)$-EA+RL concedes only to the $(2 + 2)$-EA+RL that is also the proposed $(2 + 2\lambda)$-EA+RL for $\lambda = 1$. In this general sense the $(2 + 2\lambda)$-EA+RL outperformed the initial EA+RL and conventional EAs for $k \geq 3$. However, these results make us assume that parameter tuning may significantly increase the efficiency of the proposed method.

Fig. 2 also shows that the theoretical upper bound for the expected runtime of the $(2+2n)$-EA+RL is rather pessimistic. However, it seems to have the right asymptotics, as it differs from the mean runtime of the $(2 + 2n)$-EA+RL only by a constant factor. Note that since we use the logarithmic scale on the plots, the difference by a constant factor equals to the gap of the constant height between plots. We support these statements with Fig. 1, which illustrates the ratio of the theoretical upper bound on the expected runtime of the $(2 + 2n)$-EA+RL to its mean runtime. Note that for each k this ratio is almost constant, if $n \geq 40$. For lower values of n the ratio may be different, that may be explained by the specific behaviour of the $(2 + 2n)$-EA+RL in low-dimensional search spaces.

Note that the ratio in Fig. 1 is significantly different for different values of k. We can see that our upper bound is very pessimistic for $k = 2$, and at the same time it is not so pessimistic for $k \geq 4$. We assume that this behaviour is explained by the fact that the probability of the worst-case scenario (when the forgetting mechanism is triggered) for each plateau is lower for greater k.

From Fig. 3 we see that the $(2 + 2n)$-EA+RL has a very small variation of the runtime compared with (1+1)-EA+RL and the (2+2)-EA+RL. Moreover, runtimes of the $(2 + 2)$-EA+RL are concentrated around several values. Precise analysis of the experimental results revealed that the $(2+2n)$-EA+RL extremely rarely enters the plateau with only one individual, while the $(2 + 2)$-EA+RL has a lot of runs when it pulls up after choosing XdivK and spends C iterations to forget it. However, the $(2 + 2)$-EA+RL often pulls up after choosing OneMax, and this fortunate scenario may be the reason why the $(2 + 2)$-EA+RL outperforms $(2 + 2n)$-EA+RL for small values of k and n.

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208231

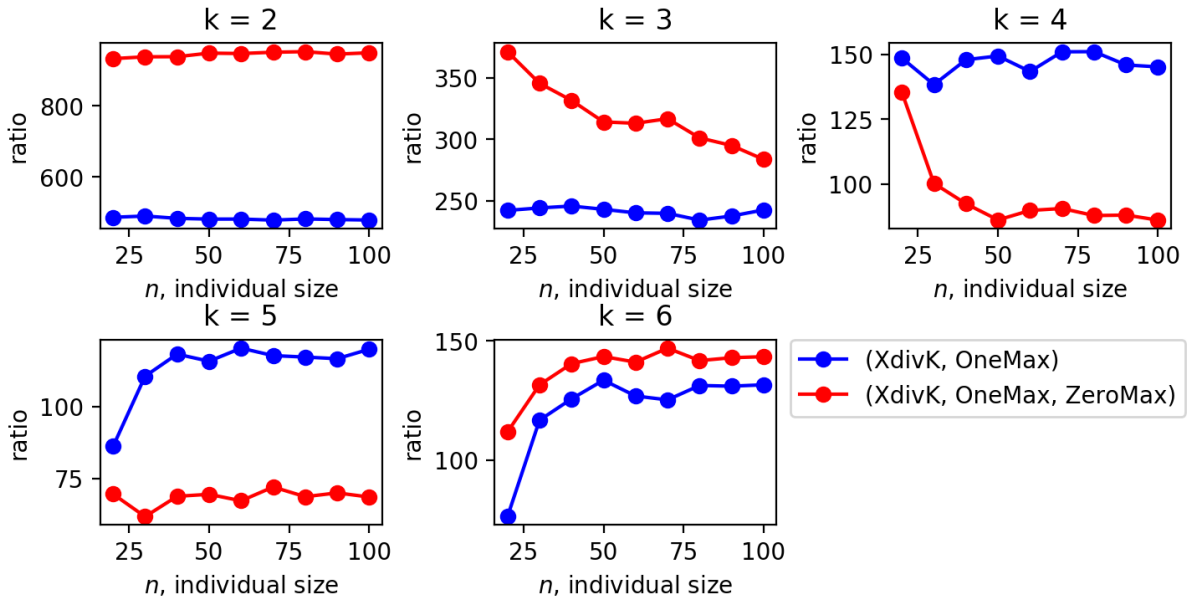

Figure 1: The ratio of the theoretical upper bound to the mean runtime of the $(2+2n)$-EA+RL for problems (XdivK, OneMax) and (XdivK, OneMax, ZeroMax) for all $k \in [2..6]$ and for all $n \in [20..100]$ with step 10.

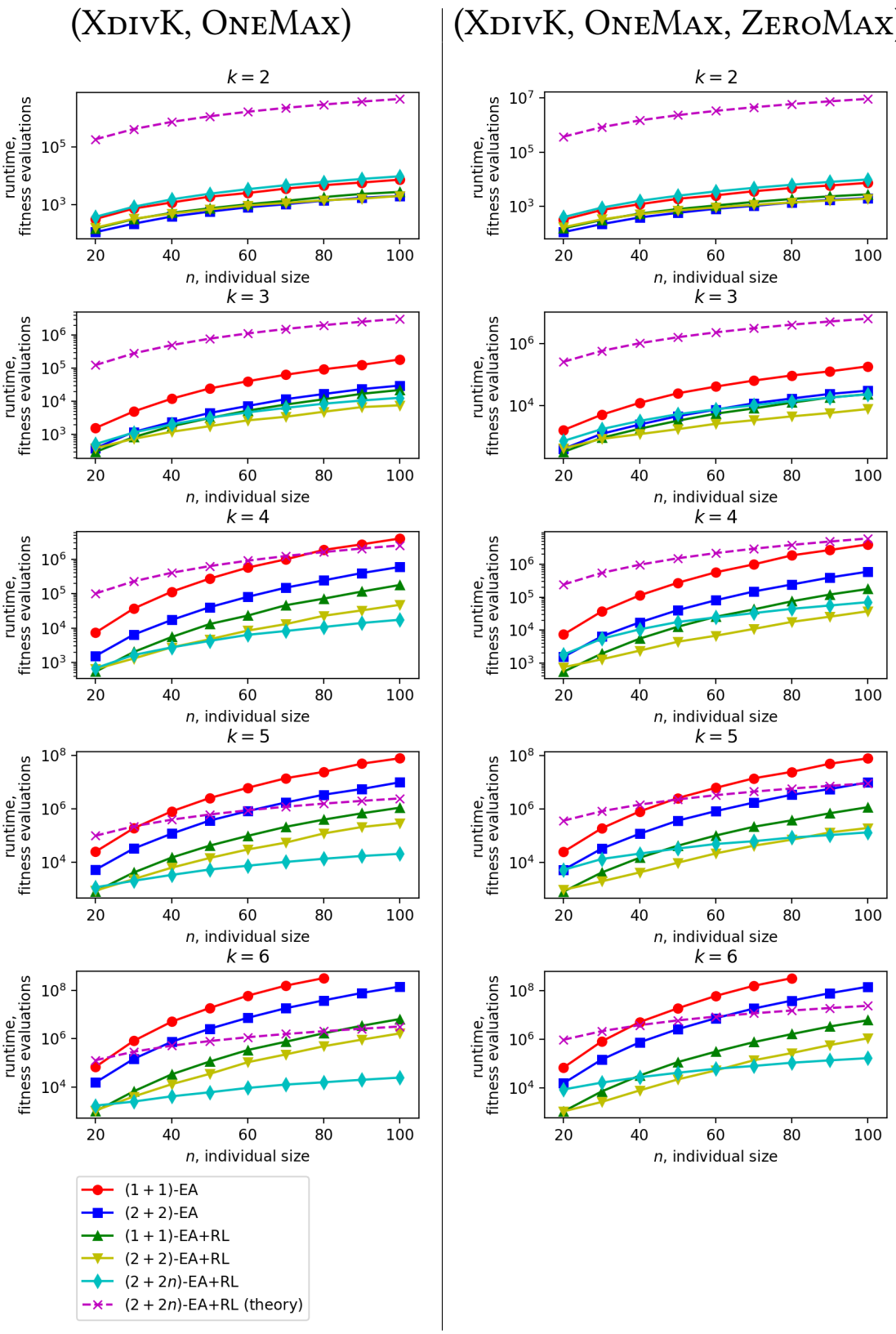

Figure 2: Mean value of the runtime of the $(2 + 2\lambda)$-EA+RL compared with the mean runtime of the $(1 + 1)$-EA+RL with preservation of the best individual, the $(1 + 1)$-EA, the $(2 + 2)$-EA and with the theoretical upper bound for the (XdivK, OneMax) and (XdivK, OneMax, ZeroMax) problems.

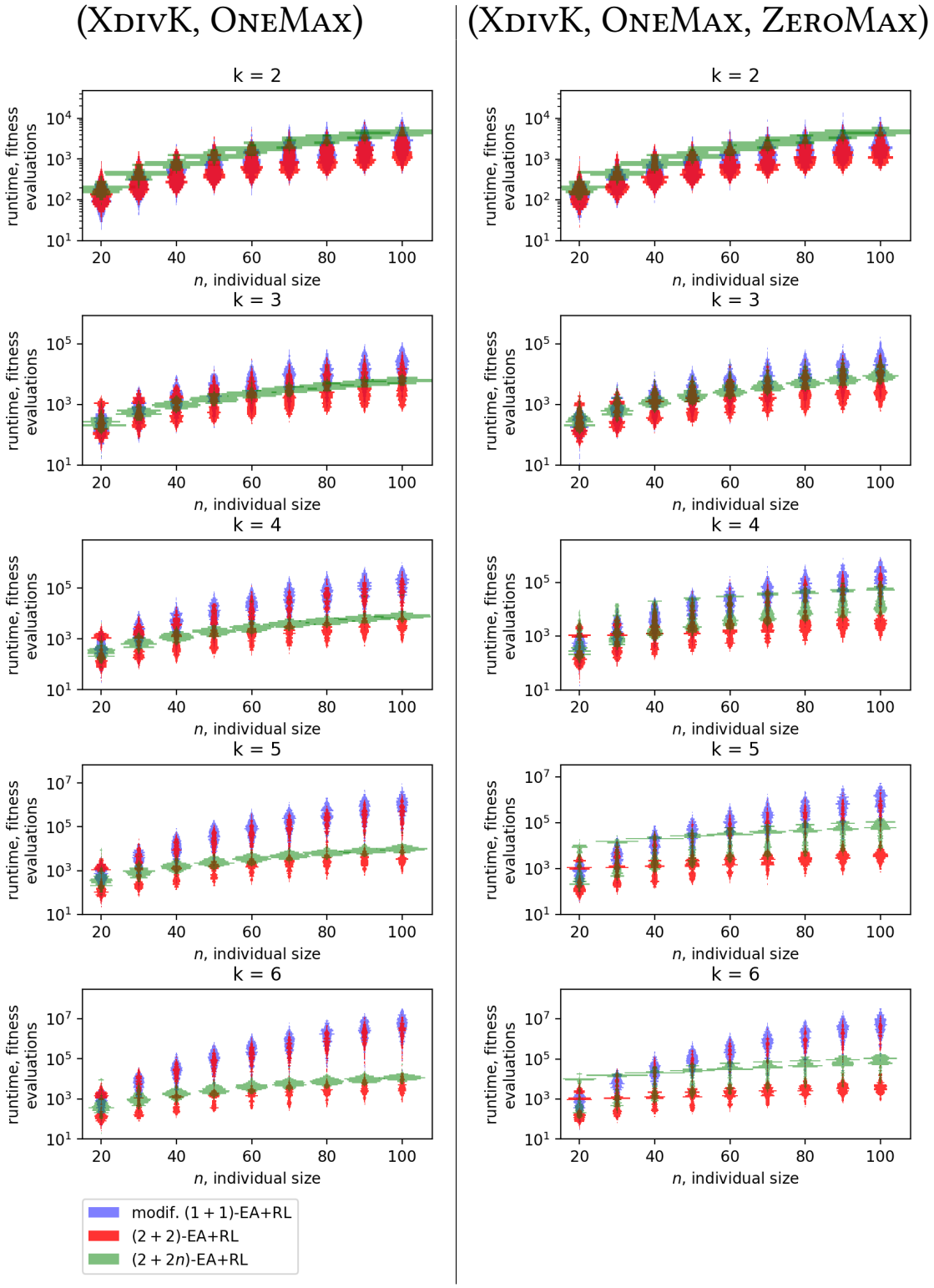

Figure 3: Vertical histograms that illustrate the variation of the number of fitness evaluations for the $(1+1)$, $(2+2)$ and $(2+2n)$-EA+RL.