Fitness-Dependent Hybridization of Clonal Selection Algorithm and Random Local Search

Nina Bulanova
ITMO University
49 Kronverkskiy Pr.
Saint Petersburg, Russia
ninasbulanova@gmail.com

Arina Buzdalova
ITMO University
49 Kronverkskiy Pr.
Saint Petersburg, Russia
abuzdalova@gmail.com

Maxim Buzdalov
ITMO University
49 Kronverkskiy Pr.
Saint Petersburg, Russia
mbuzdalov@gmail.com

ABSTRACT

Artificial immune systems (AIS) and local search algorithms have remarkable differences in the structure of mutation operators. Thus AIS algorithms may be more efficient at the beginning of optimization, while local search algorithms are more efficient in the end, when we need to do small improvements. Our goal is to combine several mutation operators in one algorithm so that the new algorithm will be efficient on fixed budget and will reach optimum within reasonable time bounds.

We propose to select mutation operators used in AIS and local search according to a specific exponential probability function which depends on the fitness of the current individual. During the experimental study, we constructed hybrids from AIS mutation operator CLONALG (Clonal Selection Algorithm) and RLS mutation operator (Random Local Search) and used them to solve OneMax problem. We compared the proposed method with a simple hybrid algorithm and empirically confirmed the hypothesis that hybrids are efficient on fixed budget and need only a slightly higher number of iterations to reach the optimum.

Keywords

hybrid algorithms; RLS; AIS; artificial immune systems.

1. INTRODUCTION

Design and analysis of methods that combine mutation operators in one algorithm is an actively researched area. For example, in memetic algorithms, the combination of evolutionary algorithms and local search is used [4]. Local search operators in memetic algorithms may be selected adaptively from the predetermined set of operators [5, 6]. Hyper-heuristics are worth mentioning as well [1]. A hyper-heuristic may be described as a new heuristic created by combining and adapting several simpler heuristics.

Advantages of different algorithms may also be combined by *hybridization*. In paper [2], hybridization was used to combine mutation operators of AIS and local search in one algorithm. In this approach, selection between mutation operators is performed with constant probability, therefore, this approach does not make adjustments during optimization. We propose a method of fitness-dependent hybridization of AIS and local search, where probability of selection of mutation operator depends on fitness of a current individual.

Algorithm 1 (1+1) Hybrid Algorithm

$x \leftarrow$ random bit string of length n
$v_0 \leftarrow f(x)/\max(f)$ normalized fitness of the initial individual, $max(f)$ — upper bound of fitness value for the considered problem
while (optimum is not found or limit of iterations is not reached) **do**
  $v \leftarrow f(x)/\max(f)$ normalized fitness of the current individual
  With probability of $n^{(-v+v_0)}$
    $x' \leftarrow Mutation1(x)$ **or**
  With probability of $1 - n^{(-v+v_0)}$
    $x' \leftarrow Mutation2(x)$
  if $f(x') \geq f(x)$ **then**
    $x \leftarrow x'$

2. FITNESS-DEPENDENT HYBRID ALGORITHM

The general scheme of the algorithms considered in this paper is (1+1) Algorithm [2]. (1+1) Hybrid Algorithm (see Algorithm 1) combines two mutation operators (Mutation1 — CLONALG, Mutation2 — RLS) and selects one of them according to probability function which depends exponentially on the current fitness. This approach helps to shift preference between mutation operators from one operator to another during the optimization process.

3. RESULTS AND DISCUSSION

The average number of fitness evaluations needed to reach the optimum is presented in Table 1. The results are shown for RLS, CLONALG and the algorithms created by hybridizing the considered algorithms using the exponential probability function (CLONALG+RLS (exp)) and the constant probability function (CLONALG+RLS (const)) with probability $p = 0.5$. The lower the values are, the better the corresponding method is. Standard deviation is given in the rightmost column for each considered algorithm. In order

GECCO '16 July 20-24, 2016, Denver, CO, USA

ACM ISBN 978-1-4503-4323-7/16/07.
DOI: http://dx.doi.org/10.1145/2908961.2908996

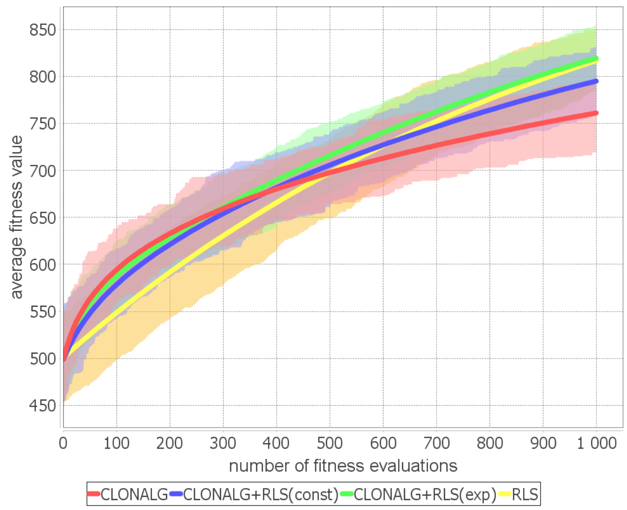

Figure 1: CLONALG, RLS and hybrids solving the OneMax problem

Table 1: Number of fitness evaluations needed to optimize OneMax by the hybrids and the conventional algorithms

Algorithm	Evaluations	Deviation
RLS	6.78×10^3	1.30×10^3
CLONALG	1.69×10^4	3.42×10^3
CLONALG+RLS (const)	9.50×10^3	1.88×10^3
CLONALG+RLS (exp)	6.91×10^3	1.28×10^3

to investigate the results for statistical significance, the unpaired Wilcoxon test was used with the level of significance of $\alpha = 0.05$, then the Holm correction was applied. The obtained p-values were less than 2.5×10^{-7} for every possible pair of algorithms.

Overall, the best performing algorithm is RLS, while the hybrid algorithm with the exponential probability function shows slightly worse but still efficient enough results. The fact that the hybrid algorithms are slightly less efficient than RLS may be explained as follows. In the end of optimization, the probability of selection of the inefficient algorithm is slightly higher than zero and therefore has some effect on the hybrid performance.

The plot shown in Fig. 1 illustrates the optimization of OneMax by RLS, CLONALG and hybrids with up to 1000 function evaluations. As we can see, on the interval from 300 to 900 CLONALG+RLS (exp) is the best algorithm and it is always better than the constant probability hybrid.

Artificial immune systems (AIS) are efficient at the early stages of optimization, but are outperformed by local search as optimization progresses [3]. To illustrate that the proposed method demonstrates beneficial properties of both approaches, we use the drift plot (Fig. 2). In this plot the average increase of the current fitness value in CLONALG, RLS and the fitness-dependent hybrid algorithm CLONALG+RLS (exp) is presented.

4. CONCLUSION

We proposed a method that selects mutation operators from different algorithm classes according to the probability

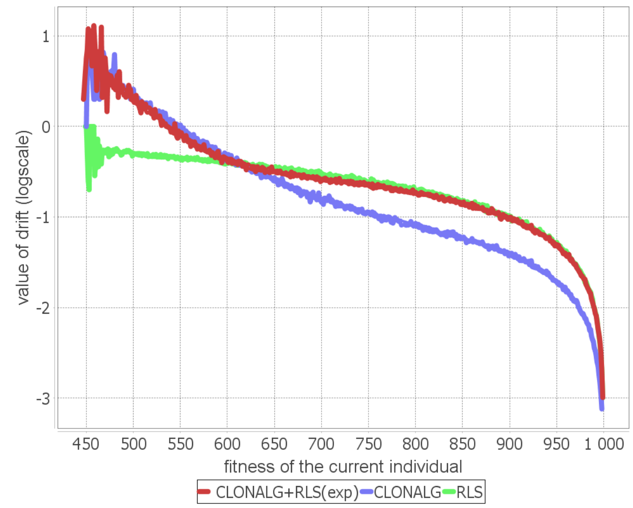

Figure 2: The drift plot for RLS, CLONALG and CLONALG+RLS (exp) algorithms on the OneMax problem

function which depends exponentially on the current fitness. It seems that using AIS mutations is efficient only during a small period at the beginning of optimization, and further on the probability of choosing an AIS mutation should be reduced very quickly. Therefore, the exponential probability function demonstrates higher efficiency than the constant one. The proposed method is efficient enough on the fixed budget runs and reaches optimum within reasonable time bounds. Thus, fitness-dependent hybrids can be used in situations when it is needed to stop the optimization process at any time and have a sufficiently good solution.

This work was partially financially supported by the Government of Russian Federation, Grant 074-U01.

5. REFERENCES

[1] E. K. Burke, M. Gendreau, M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu. Hyper-heuristics: a survey of the state of the art. *JORS*, 64(12):1695–1724, 2013.

[2] D. Corus, J. He, T. Jansen, P. S. Oliveto, D. Sudholt, and C. Zarges. On easiest functions for somatic contiguous hypermutations and standard bit mutations. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 1399–1406, 2015.

[3] T. Jansen and C. Zarges. Reevaluating immune-inspired hypermutations using the fixed budget perspective. *IEEE Trans. Evolutionary Computation*, 18(5):674–688, 2014.

[4] F. Neri, C. Cotta, and P. Moscato, editors. *Handbook of Memetic Algorithms*, volume 379 of *Studies in Computational Intelligence*. Springer, 2012.

[5] Y. Ong, M. Lim, N. Zhu, and K. W. Wong. Classification of adaptive memetic algorithms: a comparative study. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 36(1):141–152, 2006.

[6] J. E. Smith. Self-adaptative and coevolving memetic algorithms. In *Handbook of Memetic Algorithms*, pages 167–188. 2012.