SELECTION OF AUXILIARY OBJECTIVES IN ARTIFICIAL IMMUNE SYSTEMS: INITIAL EXPLORATIONS

Arina Buzdalova and Nina Bulanova

ITMO University
Computer Technologies Laboratory
49 Kronverkskiy prosp., Saint-Petersburg, Russia
abuzdalova@gmail.com, ninokfox@gmail.com

Abstract: *Multi-objectivization of artificial immune systems (AIS) is considered for several discrete optimization problems with certain properties. It is performed by adding auxiliary objectives which are selected online using reinforcement learning. The results suggest that using auxiliary objectives may help to improve performance of AIS. It is also shown that reinforcement learning is able to ignore inefficient auxiliary objective in AIS, as it is in EA. To the best of our knowledge, this is the first attempt to study effects of multi-objectivization on AIS.*

Keywords: *multi-objectivization, helper-objectives, reinforcement learning, BCA*

1 Introduction

Auxiliary objectives may be introduced in order to enhance single-objective optimization. The corresponding approach is called multi-objectivization [9,10,12]. Multi-objectivization may help to escape from local optima of the target objective and maintain diversity among candidate solutions. Efficiency of using multi-objectivization in evolutionary algorithms was widely studied in recent years [14]. There are also works concerning multi-objectivization in reinforcement learning [1].

Artificial immune systems (AIS) may be used for solving of optimization tasks, as evolutionary algorithms do [8]. However, to the best of our knowledge it has not been studied yet whether multi-objectivization is efficient for AIS. In the current study we investigate whether a specific multi-objectivization approach help solving some artificially developed model problems.

We consider so called EA+RL approach, where EA stands for evolutionary algorithm and RL is reinforcement learning. In EA+RL, auxiliary objectives are dynamically selected with reinforcement learning during one run of an evolutionary algorithm. Thus EA+RL may be useful in the case when different auxiliary objectives are efficient at different stages of optimization and we may want to switch between them. Inspiration for such problem formulation comes from a real world application of EA+RL to generation of test cases for solutions of programming contest problems [2].

The rest of the paper is organized as follows. Firstly, we describe the previously proposed EA+RL approach, which is going to be used together with AIS in the current paper. Secondly, we consider methodology used in the study, i. e. we give motivation for the considered optimization problems and describe the algorithms being compared. We also describe how a statistical test was used to decide whether the results are significant. Then the results of running AIS with auxiliary objectives are presented and compared to the results obtained by the other considered algorithms. Finally, some conclusions about using auxiliary objectives in AIS are made.

2 Related Work

We consider single-objective optimization of a *target* objective. The optimization is performed using a single-objective EA. There is a set of predefined auxiliary objectives. We do not know properties of auxiliary objectives in advance, although it is implied that optimizing some of them may decrease the number of EA generations needed to find the optimum of the target objective. An objective to be optimized at the current EA generation is selected from both the set of auxiliary objectives and the target objective. Selection is based on a reinforcement learning algorithm.

In reinforcement learning, an agent selects an action and applies it to an environment. The environment returns a numerical reward and some representation of its state. The agent updates the quality estimation of the actions according to the reward, makes a new selection and so on.

The general scheme of using reinforcement for selection of auxiliary objectives was proposed in the EA+RL method [5]. In this method the environment corresponds to EA and the actions correspond to the auxiliary objectives. To apply an action means to select an objective to be used as a fitness function in the current generation of EA.

The goal of reinforcement learning is to maximize the total reward [15]. In EA+RL, the reward function usually reflects how the target objective value has changed after applying some objective. In maximization problems, the higher the difference between the current and the previous value of target objective is, the higher the reward is. So the reinforcement agent selects the objectives that lead to the faster growth of the target objective.

We consider optimization problems where all objectives, including the target objective, are calculated during one processing of an individual [4, 5]. Therefore, evaluation of reward does not increase the number of objective evaluations.

3 Methodology

The aim of the current paper was to investigate whether EA+RL may help to enhance efficiency of AIS, as it does for EAs. So we plugged AIS in EA+RL instead of EA and thus we got AIS+RL algorithm.

3.1 Considered Algorithms

For EA, we used random local search (RLS) with the mutation operator that flipped one random bit in an individual. For AIS, we used a simplified version of BCA algorithm. The only difference of this simplified BCA from RLS is the mutation operator, which was Contiguous Hypermutation Operator [8]. However, even with this one difference AIS behaves differently from EA. For example, the considered simplified BCA is able to efficiently optimize highly multimodal H-IFF function, whereas RLS is not [8].

3.2 Parameters of Algorithms

Each algorithm was run 100 times, then the results were averaged. The parameters of RL used in EA were the learning rate $\alpha = 0.5$ and the discount factor $\gamma = 0.5$. For AIS, the parameters of RL were $\alpha = 0.7$ and $\gamma = 0.7$. In the both cases, the greedy exploration strategy was used [15]. There was always one individual in a generation, so the algorithms could be compared by the number of generations needed to reach the optimum of a target objective. The limit of fitness functions evaluations was set to 10×6.

In order to investigate the results for statistical significance, Wilcoxon test was applied with the level of significance of $\alpha = 0.05$. The detailed descriptions of how the statistical test was performed are given in the corresponding sections.

3.3 Considered Optimization Problems

We considered three problems. The aim of the first problem called XdivK was to investigate whether the running time of AIS may be decreased using an efficient auxiliary objective. The second problem called Switching problem was used to demonstrate behaviour of AIS+RL in the case when different auxiliary objectives are efficient at different stages of optimization. The third OneMax with ZeroMax problem includes an inefficient objectives. AIS+RL should be able to ignore such objectives. In addition, results for two more problems are considered in Section 5 as well. In all problems, individuals were encoded as bitstrings of length n.

4 Results of Experiments

4.1 RL Increases Efficiency of AIS: XdivK Problem

Consider two objectives: the target one is XdivK and the auxiliary one is OneMax. OneMax is number of one bits in an individual. XdivK is evaluated as follows: OneMax value is divided by some parameter k and the integer part of the result is taken. It was shown by theoretical analysis that OneMax is an efficient auxiliary objective for XdivK [3].

Experimental results are shown in Table 1. Average number of generations needed to reach the target optimum is presented for each problem size n and for the each considered approach, so the lower values are, the better the corresponding method is. The AIS and RLS columns refer to XdivK optimization by the corresponding algorithm when no auxiliary objective is used.

For the cases when the optimum of XdivK was not always reached within the fixed number of generations, a percent of successful runs is given with the average number of generations in a successful run. For the rest of the cases, we applied the unpaired Wilcoxon test to the results of AIS and AIS+RL. The difference between AIS and AIS+RL turned to be statistically significant for all the considered problem instances.

It can be seen from the Table 1 that using of auxiliary objectives enhances both AIS and EA for all considered problem instances. However, AIS itself is outperformed by EA while solving this problem. Note that for the biggest problem instance (see the last line of the Table 1), AIS+RL is even able to outperform EA. So using of auxiliary objectives may help AIS to outperform EA.

Table 1: Number of generations needed to optimize XdivK with OneMax

n	k	AIS	AIS+RL	EA	EA+RL
12	2	598.4	387.7	123.8	66.4
16	2	1451.9	959.8	192.3	111.8
20	2	3142.1	1567.2	310.7	202.9
18	3	5372.1	3498.3	1077.3	637.5
24	3	21639.0	10156.9	3142.9	1336.1
30	3	48915.6	20166.5	5239.7	3058.1
24	4	79280.7	44855.4	11820.7	5783.6
32	4	93%(337810.6)	99%(123298.3)	48746.2	25662.7
40	4	72%(455946.1)	75%(154427.3)	116844.0	46239.4
30	5	61%(494223.6)	85%(186701.6)	168184.7	82513.2
40	5	14%(539217.6)	58%(52390.5)	77%(420403.1)	87%(127970.8)
50	5	2%(592953.5)	58%(38212.2)	36%(444729.2)	61%(144981.4)

4.2 RL Selects Proper Objectives in AIS: The Switching Problem

In the Switching problem the target objective is also XdivK as in Section 4.1. There are two auxiliary objectives: $\min(OneMax, d)$ and $\max(OneMax, d)$, where d is a *switching point*. The first auxiliary objective is efficient for the individuals with less than d bits set to one, while the second one is efficient for the individuals with more than d one bits. Thus, different objectives are efficient at different stages of optimization.

4.2.1 Efficiency of Selection of Auxiliary Objectives

In the Table 2 the average target fitness after 10^6 function evaluations is presented. So the higher are values, the better is performance of a corresponding algorithm. In parentheses, standard deviation is given.

Table 2: Switching problem: average target fitness after a fixed number of generations

n	k	d	AIS	AIS+RL	EA	EA+RL
200	5	165	35.3(0.6)	35.4(0.8)	37.6(0.6)	37.8(0.6)
300	10	245	22.5(0.6)	23.2(0.8)	23.6(0.6)	24.3(1.5)
400	10	266	29.6(0.6)	29.8(0.6)	31.2(0.6)	32.1(2.6)
400	10	333	29.7(0.7)	31.2(1.7)	31.4(0.6)	32.1(1.3)
500	10	415	36.7(0.7)	38.7(2.4)	39.0(0.7)	40.8(2.7)
600	15	460	27.0(0.5)	28.6(1.7)	27.8(0.6)	29.8(3.5)
600	10	495	43.8(0.8)	45.8(2.5)	46.6(0.8)	48.0(2.6)

According to the Table 2, AIS+RL with auxiliary objectives performs better than AIS for all the considered problem instances. Thus AIS+RL is able to switch between the objectives and take advange of the objective which is efficient at the current stage of optimization.

In order to perform statistical test, random n and k were generated 20 times and the corresponding problem instances were optimized with AIS and AIS+RL. Parameter d was set at about 80% of n, as in the Table 2. The results for each problem instance were averaged and used as an input in the paired Wilcoxon test. Finally, the obtained p-values were less than 1.9×10^{-6}, so using of auxiliary objectives in AIS for the considered problem is significantly different from optimizing only the target objective.

4.2.2 Dynamics of Selection of Auxiliary Objectives

In the Figure 1, behaviour of RL agent is shown in dynamics for a particular instance of Switching problem ($n = 500, k = 10, d = 415$). Horizontal axis corresponds to the number of bits set to one in the current individual. Each color corresponds to a certain objective, as shown in the legend at the top of the Figure 1.

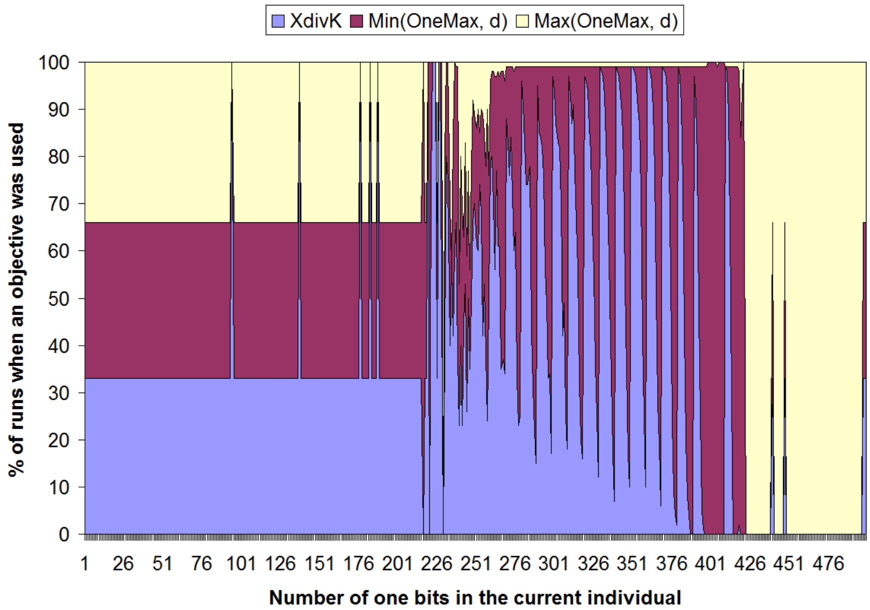

Figure 1: Percentage of choices of different objectives in the Switching problem

If we fix number of one bits, the height of a colored line will correspond to the percentage of runs when the corresponding objective was chosen for the individual with the corresponding number of one bits.

Let us analyse the Figure 1. At first, the agent is not trained enough and selects each objective nearly equiprobably. It should be noticed that at early stages of optimization, a plenty of possibilities exists to enhance an individual, since many bits are still set to zero. Thus selection of a proper objective is not so important.

Roughly after a half of bits were set to one, agent finally learns that the $\max(OneMax, d)$ is an obstructive objective for the current stage of optimization, so nearly no selection of this objective is performed. This observation is reflected in the Figure 1: very little amount of the corresponding color presents between 250 and 415 one bits.

After the switch point $d = 415$ is reached, the agent learns rather quickly that $\max(OneMax, d)$ is the proper objective for this new stage of optimization. From point 415 to point 426 the agent is still selecting $\min(OneMax, d)$ objective, but the corresponding reward is not high, so after the point 426 the agent starts to select the proper objective that leads to higher reward. Correspondingly, the color of the proper objective $\max(OneMax, d)$ prevails from the point 426 to the end of the plot.

To sum up, both the efficiency measures and the analysis of dynamics confirm that the RL agent selects proper objectives at the corresponding stages of optimization.

4.3 RL Ignores Inefficient Objective in AIS: OneMax with ZeroMax Problem

OneMax is a well known benchmark function which is easily optimized by various EAs [13]. At the same time, OneMax is not so easy for AIS.

We used an obstructive objective ZeroMax in this problem in order to show that RL is able to ignore such objectives. Since OneMax counts the number of ones in a bit string and ZeroMax is just the opposite and counts the number of zeros, this is an extreme example of a problem with an obstructive objective.

It was theoretically shown for randomized local search that EA+RL is able to solve the OneMax with ZeroMax problem asymptotically as fast as a conventional EA without ZeroMax. So it is of interest whether AIS+RL would be comparable with AIS in this problem.

In the Table 3 the number of generations needed by various algorithms to reach the optimum of OneMax is presented. In parentheses, standard deviation is given. According to the unpaired Wilcoxon test, AIS+RL is indistinguishable from AIS, as well as EA+RL is indistinguishable from EA. So the obstructive ZeroMax objective is successfully ignored both in AIS and EA. The test was performed the same way as described in Section 4.1.

Table 3: Number of generations needed to optimize OneMax with ZeroMax auxiliary objective

n	AIS	AIS+RL	EA	EA+RL
20	931.1(552.2)	874.3(493.2)	60.6(25.1)	59.0(23.3)
30	2089.7(896.8)	2162.3(1069.6)	103.8(39.4)	97.0(35.9)
40	4235.6(1730.2)	4490.4(2052.7)	138.3(38.3)	152.1(55.7)
50	7520.6(3323.1)	7348.5(3826.3)	181.9(53.4)	187.5(53.3)
60	11602.1(4104.2)	11376.8(5443.4)	231.6(74.5)	236.1(69.7)
70	14802.1(4539.8)	15473.6(5679.2)	286.4(91.9)	293.5(88.8)
80	20810.1(8079.7)	21302.1(8377.3)	320.8(89.1)	362.1(109.7)
90	25990.4(9361.7)	28543.3(10554.5)	400.6(106.6)	399.1(100.7)

5 Discussion

5.1 Auxiliary Objectives in AIS: Positive Examples

Three different problems were considered. In the first two cases using of efficient auxiliary objectives helped to enhance performance of AIS. In the last case, an inefficient auxiliary objective was successfully ignored. So in all the considered problems multi-objectivization of AIS had the same effect on the performance as multi-objectivization of EA.

What is more, we saw an example when using of auxiliary objectives even helped AIS perform better than EA, although EA normally outperforms AIS on the corresponding problem. It is known that AIS may outperform EA at the early stages of optimization, while it would usually be worse than EA on long runs [8]. So using auxiliary objectives may potentially combine the ability of AIS to perform well at the beginning of optimization and a good overall performance.

An interesting research direction may be to apply AIS with auxiliary objectives for solving of some practical problems where multi-objectivization were used. The examples of such problems are job-shop scheduling [11], protein structure prediction [6, 7] and test case generation [2].

5.2 Auxiliary Objectives in AIS: Negative Examples

In the previous section, using auxiliary objectives was efficient for AIS, as well as for EA. However, results of using auxiliary objectives in AIS may not always be the same as results for EA. For the sake of brevity, we present only the general observations of the corresponding cases without giving a detailed description of results.

Consider H-IFF problem [10]. In this problem, AIS performs asymptotically better than EA (more precisely, RLS) and there is no space for improvement using auxiliary objectives. Indeed, according to our experimental results, AIS+RL with auxiliary objectives performed just as well as AIS without auxiliary objectives.

One more interesting example was obtained while solving the LeadingOnes problem with the OneMax auxiliary objective. OneMax is solved by EA asymptotically faster than LeadingOnes, so using of auxiliary objectives helps to improved performance of EA. However, OneMax is not so easy for AIS, so it could not act as an efficient auxiliary objective for AIS.

The examples above show that while using of auxiliary objectives may help to improve efficiency of AIS, it is important to use proper objectives. This proper objectives may not be the same as for EA, since AIS performs differently from EA on certain problems.

6 Conclusion

In order to investigate the impact of using auxiliary objectives in AIS, several problems were solved. The auxiliary objectives were selected with reinforcement learning. The results suggest that using of auxiliary objectives may help to increase performance of AIS. This results were shown to be statistically significant. However, in certain problems different auxiliary objectives may be needed for AIS comparing to EA, since AIS may behave differently from EA on these problems.

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

References

[1] Brys, T., Harutyunyan, A., Vrancx, P., Taylor, M.E., Kudenko, D., Nowé, A.: Multi-objectivization of reinforcement learning problems by reward shaping. In: 2014 International Joint Conference on Neural Networks, pp. 2315–2322 (2014)

[2] Buzdalov, M., Buzdalova, A.: Adaptive Selection of Helper-Objectives for Test Case Generation. In: 2013 IEEE Congress on Evolutionary Computation, vol. 1, pp. 2245–2250 (2013)

[3] Buzdalov, M., Buzdalova, A.: Onemax Helps Optimizing XdivK: Theoretical Runtime Analysis for RLS and EA+RL. In: Proceedings of Genetic and Evolutionary Computation Conference Companion, pp. 201–202. ACM (2014)

[4] Buzdalov, M., Buzdalova, A., Petrova, I.: Generation of Tests for Programming Challenge Tasks Using Multi-Objective Optimization. In: Proceedings of Genetic and Evolutionary Computation Conference Companion, pp. 1655–1658. ACM (2013)

[5] Buzdalova, A., Buzdalov, M.: Increasing Efficiency of Evolutionary Algorithms by Choosing between Auxiliary Fitness Functions with Reinforcement Learning. In: Proceedings of the International Conference on Machine Learning and Applications, vol. 1, pp. 150–155 (2012)

[6] Garza-Fabre, M., Toscano-Pulido, G., Rodriguez-Tello, E.: Locality-based multiobjectivization for the hp model of protein structure prediction. In: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, GECCO '12, pp. 473–480. ACM, New York, NY, USA (2012)

[7] Handl, J., Lovell, S.C., Knowles, J.: Investigations into the effect of multiobjectivization in protein structure prediction. In: G. Rudolph, T. Jansen, N. Beume, S. Lucas, C. Poloni (eds.) Parallel Problem Solving from Nature PPSN X, *Lecture Notes in Computer Science*, vol. 5199, pp. 702–711. Springer Berlin Heidelberg (2008)

[8] Jansen, T., Zarges, C.: Understanding randomised search heuristics. lessons from the evolution of theory: A case study. In: Proceedings of the 20th International Conference on Soft Computing MENDEL, pp. 293–298 (2014)

[9] Jensen, M.T.: Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation: Evolutionary Computation Combinatorial Optimization. Journal of Mathematical Modelling and Algorithms **3**(4), 323–347 (2004)

[10] Knowles, J.D., Watson, R.A., Corne, D.: Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In: Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, pp. 269–283. Springer-Verlag (2001)

[11] Lochtefeld, D.F., Ciarallo, F.W.: Helper-Objective Optimization Strategies for the Job-Shop Scheduling Problem. Applied Soft Computing **11**(6), 4161–4174 (2011)

[12] Neumann, F., Wegener, I.: Can Single-Objective Optimization Profit from Multiobjective Optimization? In: Multiobjective Problem Solving from Nature, Natural Computing Series, pp. 115–130. Springer Berlin Heidelberg (2008)

[13] Oliveto, P.S., He, J., Yao, X.: Time Complexity of Evolutionary Algorithms for Combinatorial Optimization: A Decade of Results. International Journal of Automation and Computing **4**(3), 281–293 (2007)

[14] Segura, C., Coello, C.A.C., Miranda, G., Léon, C.: Using multi-objective evolutionary algorithms for single-objective optimization. 4OR **3**(11), 201–228 (2013)

[15] Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, USA (1998)