A Classification-Based Selection for Evolutionary Optimization

Jinyuan Zhang, Jimmy Xiangji Huang[1], Qinmin Vivian Hu[2]
Information Retrieval and Knowledge Management Research Lab
[1] School of Information Technology, York University, Canada
[2]Department of Computer Science, Ryerson University, Canada
{jyzh,jhuang}@yorku.ca,vivian@ryerson.ca

ABSTRACT

For evolutionary algorithms (EAs), selection is one of the main components which decides solutions for the new population. Most selection strategies are fitness-based and prodigal in fitness evaluations, since many evaluated solutions are discarded immediately due to their worse values. It is desirable to predict the quality of new solutions without the evaluations before selection, thus the efficiency of EAs can be improved. Naturally, selection can be considered as a classification problem: selected solutions belong to the 'good' class and the discarded ones belong to the 'bad' class. This paper demonstrates this idea by introducing a *classification-based selection (CBS)* strategy for EAs. The CBS is integrated into a state-of-the-art algorithm and studied on a test suite. The experimental results evidence the efficiency of CBS on saving the number of fitness evaluations when compared with the original algorithm.

CCS CONCEPTS

• **Computing methodologies** → **Search methodologies**;

KEYWORDS

Classification, selection, evolutionary algorithms

ACM Reference Format:
Jinyuan Zhang, Jimmy Xiangji Huang[1], Qinmin Vivian Hu[2]. 2019. A Classification-Based Selection for Evolutionary Optimization. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3322077

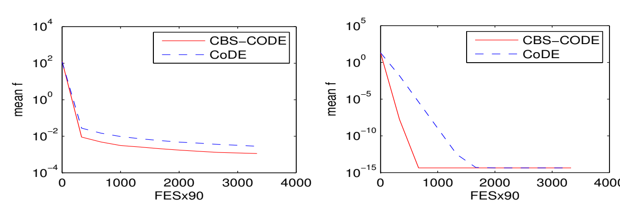
1 MOTIVATION, PROBLEM AND OUR IDEA

In evolutionary algorithms (EAs) [9], selection strategy is one of the main components which aims to decide the new population. Since most selection strategies are fitness-based, there exists a waste on many fitness evaluations, where only evaluated solutions with better fitness values can enter the following optimization procedures and the worse ones are discarded directly. Since the number of fitness evaluations is one of the main measurements for the efficiency of EAs, saving the number of fitness evaluations has attracted much attention, and many methods in this category have been proposed. Among them, the surrogate (meta) model-based approaches [5] are the promising ones, where the surrogate models are built to evaluate the solutions instead of the real optimization problem. In these cases, the number of fitness evaluations can be significantly reduced. However, selection in EAs can naturally be regarded as a classification problem, where the chosen solutions belong to the 'good' class, and the discarded ones belong to the 'bad' class. Based on this idea, classification model-based approaches, which are special cases of surrogate-based approaches are of interest [3, 6, 7, 14, 15].

In most of EAs, mostly the 'good' solutions are attracted much attention, and the current population contains relative 'good' solutions found so far. It is beneficial to construct a classifier to assist selection based on the current population.

Following the above ideas, we propose a *classification-based selection (CBS)* strategy for EAs. In this proposed strategy, at first, a classification model is built based on the current population. Next, the model is employed to predict the quality of the generated solutions. Then, the newly generated 'bad' solutions are discarded directly without being evaluated according to the classification model, where only the 'good' ones will be evaluated and enter the following optimization procedures. Therefore, the number of fitness evaluations can be saved by only evaluating a part of the generated population. The framework of the proposed CBS-EA is as follows:

Step 1: Initialize the population P.
Step 2: Assign each individual $x \in P$ a label.
Step 3: Build a classification model by the data set $\{< x, l_x >\}$.
Step 4: For each $x \in P$
  Step 4-1: Generate an offspring solution y, set the fitness value as $f(y) = \infty$.
  Step 4-2: Predict the label l_y of y by classification model.
  Step 4-3: Evaluate the real fitness value of y if $l_y == 1$.
  Step 4-4: If $f(y) < f(x)$, set $x = y, f(x) = f(y)$.
Step 5: If termination condition is not satisfied, go back to **Step 2**, otherwise, stop.



(a) f7 (b) f10

Figure 1: The mean fitness values versus the number of fitness evaluations obtained by CBS-CoDE and CoDE.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07...$15.00
https://doi.org/10.1145/3319619.3322077

Table 1: The number of fitness evaluations required by CBS-CoDE vs. CoDE to achieve the same fitness values on $f1-f13$.

	mean	value	mean	value
	CBS-CoDE		CoDE	
$f1$	**1.04e+05**(+)	8.65e-61	2.74e+05	9.78e-61
$f2$	**9.98e+04**(+)	9.96e-31	2.63e+05	9.82e-31
$f3$	**2.80e+05**(+)	9.89e-16	3.00e+05	1.52e-15
$f4$	**1.27e+05**(+)	9.49e-16	3.00e+05	1.01e-15
$f5$	2.67e+05(-)	1.50e+01	**1.09e+05**	1.50e+01
$f6$	**1.04e+04**(+)	0.00e+00	2.10e+04	0.00e+00
$f7$	**2.64e+04**(+)	9.97e-03	8.77e+04	9.94e-03
$f8$	7.51e+04(-)	1.97e+01	**4.82e+04**	9.77e+00
$f9$	**5.76e+04**(+)	0.00e+00	1.80e+05	0.00e+00
$f10$	**3.72e+04**(+)	9.68e-11	9.85e+04	9.69e-11
$f11$	**1.38e+04**(+)	9.82e-04	3.98e+04	9.28e-04
$f12$	**7.85e+03**(+)	9.11e-03	2.26e+04	9.68e-03
$f13$	**5.47e+04**(+)	8.70e-31	1.44e+05	9.78e-31
+/-/∼	11/2/0			

2 EXPERIMENTAL RESULTS

The proposed approach $CBS-EA$ is applied to CoDE [10] and studied on the first 13 benchmark functions from the YLL test suite [12]. The classification model employed in experiments is implemented in *libsvm* [1], which is a *support vector machine (SVM)* based *one-class classification* model [8]. The kernel of the model is *radial basis function (RBF)*. The control parameters of the model are set as the default values in *libsvm*. The variable dimensions are $n = 30$ for all instances. The population size is $N = 30$ for CoDE and its variants. The stop condition is $FEs = 300,000$ for all algorithms. Each algorithm is executed on each test instance for 30 independent runs. The Wilcoxon rank sum test [11] is applied to compare the experimental results. The "+", "−", or "∼" in the following table indicate the number of fitness evaluations obtained by the CBS based algorithm is smaller than, bigger than, or similar to the original algorithm at 95% significance level.

Results in Table 1 show that on 11 test instances, CBS-CoDE achieves the same fitness value with fewer number of fitness evaluations than CoDE. But on two instances $f5$, $f8$, CoDE takes fewer number of fitness evaluations than CBS-CoDE. For other instances, CBS-CoDE almost takes the half number of evaluations of CoDE. The above situations may occur depending on the classification model, thus for some instances, the classification model is not really suitable. As well as for CoDE, $f5$, $f8$ are hard to be optimized. Figs. 1 plots the statistical results obtained by CBS-CoDE and CoDE in terms of the mean fitness values versus the number of fitness evaluations on $f7$, $f10$. The curves suggest CBS-CoDE converges faster than CoDE and obtains better optimal results.

3 CONCLUSION AND FUTURE WORK

This paper proposes a *classification-based selection (CBS)* strategy to improve the efficiency of *evolutionary algorithms (EAs)*. The strategy uses the current population to build a classifier, and then predicts the quality of the newly generated offspring solutions by the model. Since the 'bad' solutions are discarded directly by the model without evaluation, the number of fitness evaluations can be reduced. The CoDE is chosen as the optimizer. The SVM based one-class classification model is chosen as the classifier. The CBS based algorithm and the original algorithm are studied on the YLL test suite [12]. The statistical results indicate that CBS can improve the efficiency of the algorithm.

There are some future works that could be done for CBS-EA. These topics include: (a) the CBS is only applied to a differential evolution algorithm in this paper, as a general strategy, we will extend the CBS with other kinds of EAs; (b) this paper only uses CBS for single-objective optimization problems, we will apply the CBS to multi-objective optimization problems, especially, the computationally expensive optimization problems; (c) this paper only employs a SVM based one-class classification model for experiments, we will combine CBS with other kinds of classifiers and improve the accuracy of the classification models; and (d) the CBS is only tested in one test suite, we will apply the CBS assisted EAs to more test instances and real-world applications (e.g. [2, 4, 13]).

ACKNOWLEDGEMENT

This research is supported by the Natural Sciences and Engineering Research Council of Canada, the York Research Chairs program and an ORF-RE (Ontario Research Fund-Research Excellence) award in BRAIN Alliance. All the work is done when the first author is a postdoctoral fellow sponsored by an NSERC discovery grant.

REFERENCES

[1] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems & Technology* 2, 3 (2011), 389–396.

[2] Wenying Feng, Qinglei Zhang, Gongzhu Hu, and Jimmy Xiangji Huang. 2014. Mining network data for intrusion detection through combining SVMs with ant colony networks. *Future Generation Computer Systems* 37 (2014), 127–140.

[3] Stephanus Daniel Handoko, Kwoh Chee Keong, and Ong Yew Soon. 2008. Using classification for constrained memetic algorithm: A new paradigm. In *IEEE International Conference on Systems, Man and Cybernetics*. 547–552.

[4] Xiangji Huang and Qinmin Hu. 2009. A bayesian learning approach to promoting diversity in ranking for biomedical information retrieval. In *Proceedings of the 32nd ACM SIGIR*. 307–314.

[5] Yaochu Jin. 2011. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation* 1, 2 (2011), 61–70.

[6] Xiaofen Lu, Ke Tang, and Xin Yao. 2011. Classification-assisted differential evolution for computationally expensive problems. In *2011 IEEE Congress on Evolutionary Computation (CEC)*. 1986 – 1993.

[7] Teresa Miquelez, Endika Bengoetxea, and Pedro Larranaga. 2004. Evolutionary computations based on bayesian classifiers. *International Journal of Applied Mathematics and Computer Science* 14, 3 (2004), 335–349.

[8] Bernhard Scholkopf, John C. Platt, John C. Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. 2001. Estimating the support of a high dimensional distribution. *Neural Computation* (2001).

[9] Back Thomas, David B. Fogel, and Zbigniew Michalewicz. 1997. *Handbook of evolutionary computation*. IOP Publishing Ltd.

[10] Yong Wang, Zixing Cai, and Zhang Qingfu. 2011. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Transactions on Evolutionary Computation* 15 (2011), 55–66.

[11] Frank Wilcoxon, SK Katti, and Roberta A Wilcox. 1970. Critical values and probability levels for the Wilcoxon rank sum test and the Wilcoxon signed rank test. *Selected tables in mathematical statistics* 1 (1970), 171–259.

[12] Xin Yao, Yong Liu, and Guangming Lin. 1999. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation* 3, 2 (1999), 82–102.

[13] Xiaohui Yu, Yang Liu, Xiangji Huang, and Aijun An. 2012. Mining online reviews for predicting sales performance: A case study in the movie domain. *IEEE Transactions on Knowledge and Data engineering* 24, 4 (2012), 720–734.

[14] Jinyuan Zhang, Aimin Zhou, and Guixu Zhang. 2018. Preselection via classification: A case study on global optimization. *International Journal of Bio-Inspired Computation* 11, 4 (2018), 267–281.

[15] Jinyuan Zhang, Aimin Zhou, and Guixu Zhang. 2018. Preselection via one-class classification for evolutionary optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion 2018*. 259.