A note on the paper "A multi-population harmony search algorithm with external archive for dynamic optimization problems" by Turky and Abdullah

Amir Ehsan Ranginkaman[1], Javidan Kazemi Kordestani[2,*], Alireza Rezvanian[3], Mohammad Reza Meybodi[3]

[1]*Department of Electrical, Computer and IT Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran*

[2] *Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*

[3]*Soft Computing Laboratory, Computer Engineering and Information Technology Department, Amirkabir University of Technology (Tehran Polytechnic), 424 Hafez Ave., Tehran, Iran*

ranginkaman@qiau.ac.ir, Javidan.kazemi@gmail.com *, a.rezvanian@aut.ac.ir, mmeybodi@aut.ac.ir

Abstract In a very recently presented paper, Turky and Abdullah [5] proposed a novel multi-population harmony search with external archive (MHSA-ExtArchive) for dynamic optimization problems. In the experimental results, the authors claimed that their approach could outperform several state-of-the-art algorithms. They also showed the superiority of their method by means of numerical experiments on Moving Peaks Benchmark (MPB). Despite the interesting idea of applying multi-population scheme on harmony search and using a new type of external archive for dealing with dynamic problems, we believe that there are two very important shortcomings in the result analysis, which we point out in this short note. The main motivation of the present note is to contribute toward preventing the same mistakes from happening by the other researchers.

Keywords dynamic optimization problems, dynamic environments, multi-population harmony search, harmony search with external archive, moving peaks benchmark, MPB

The argument is often made that the "experimental study" is the most important section of a technical or engineering paper where the author(s) describe and justify the outcome(s) of their research. Moreover, in this section the authors compare their proposed method(s) with other state-of-the-art approaches to demonstrate the effectiveness of their methodologies. Also, it is from this section that the readers can distinguish the strength and weakness points of the proposed approach. Therefore, the most critical point in proposing a new algorithm is to make sure that the experimental study section is well-described, easily understandable and on top of those, the simulation results and comparisons are free of any ambiguities and errors.

Recently, we have read the paper by Turky and Abdullah entitled "A multi-population harmony search algorithm with external archive for dynamic optimization problems" [5] with the best intentions. The paper is very remarkable since, to the best of our knowledge, it is the first time that a multi-population harmony search has been applied for solving dynamic optimization problems. Moreover, they have introduced a modified external archive to replace the redundant solutions in the population by good solutions of the archive. In the "experimental study" section, they have evaluated the performance of their algorithm on MPB, which is one of the most widely used synthetic dynamic optimization test suites in the literature. Despite the novelty and promising results of the paper, we observed two very important shortcomings in the result analysis of the paper. The first one, which is more serious, is related to a misunderstanding about the performance measure employed by Turkey and Abdullah. This misunderstanding led Turky and Abdullah to compare their method to state-of-the-art methods using different performance measures. The second one also concerns the performance comparison against the state-of-the-art methods in which they confused the standard deviation with standard error. Therefore, we decided to transfer those points to the readers with the hope to provide a better understanding of the paper.

In the field of dynamic optimization problems, in order to conduct a fair comparison between different algorithms on a specific dynamic environment, the following important points should be taken into account: (*a*) performance measurement methods for evaluating the effectiveness of different algorithms should be the same, and (*b*) in the design of experiments for computational intelligence, suitable statistical testing should be performed on the results to draw an appropriate conclusion on the superiority of a specific method.

In Section 3 of the paper, the authors stated that "In this section, the performance of the proposed algorithm (MHSA) is evaluated using the MPB, which was proposed by Branke [8] and an offline error rate is calculated by Eq. (2) [9]:". From the text, it is understood that offline error was chosen by the authors to measure the efficiency of their proposed algorithms. However, the formulation they have provided in Eq. (2) is not the definition of offline error.

It should be noted that there are two measures in the literature which have been termed as "offline error" in different articles, and this may become a source of confusion when comparing different methods. The first one is the performance measure suggested by Branke and Schmeck [1] which is defined as the average of the smallest error, at every evaluation, found since the last change in the environment over the entire run as follows:

$$E_O = \frac{1}{T}\sum_{t=1}^{T} e_t^* \tag{1}$$

where T is the maximum number of evaluations so far and e_t^* is the minimum error gained by the optimization algorithm since the last change at the t^{th} fitness evaluation. Offline error is useful for measuring the overall performance of an algorithm and comparing the final results of different algorithms [3]. The other measure, first proposed by Trojanowski and Michalewicz [4] as *accuracy* and then named as *best error before change* by Nguyen et al. [3], is calculated as the average of the minimum fitness error achieved by the algorithm at the end of each period right before the moment of change, as follows [6]:

$$E_B = \frac{1}{K}\sum_{k=1}^{K}(h_k - f_k) \tag{2}$$

where f_k is the best solution achieved by the algorithm just before the k^{th} change, h_k is the optimum value of the k^{th} environment and K is the total number of environments. This measure is specifically useful in situations where we are interested in average quality of the best solution achieved by an algorithm right before the change. However, it does not allow studying the performance of an algorithm along the entire search process [3].

Figure 1 illustrates the difference between the calculations of these two measures over 100,000 function evaluations in a visual way. In Figure 1, the solid line shows the current error, which is defined as the smallest error found since the last change in the environment, in which the offline error is derived from. The circles in Figure 1 are the function evaluations which are used to calculate the best error before change. From Figure 1, it is clear that the calculation of offline error and best error before change are totally different.

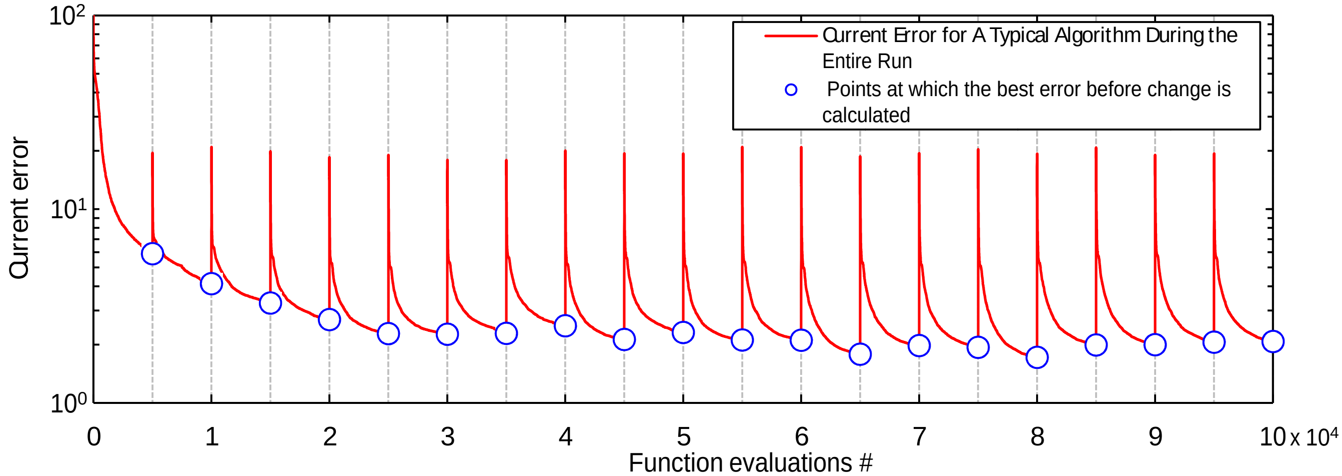

Figure 1. Evolution of the current error versus the number of function evaluations for a method on a typical dynamic environment with change frequency 5000. The vertical dotted lines in the figure indicate the start of a new period.

Figure 2 shows the evolution of the best error achieved by two different algorithms on a typical dynamic environment generated by MPB. From Figure 2, we can observe that although both algorithms can achieve the same error before the moment of changes, i.e. best error before change, the way they can achieve this error is totally different. In another word, the best error before change for both algorithms is equal but the offline error of algorithm 2 is smaller. Hence, the most important point when comparing different algorithms is to make sure that they are evaluated with the same criterion.

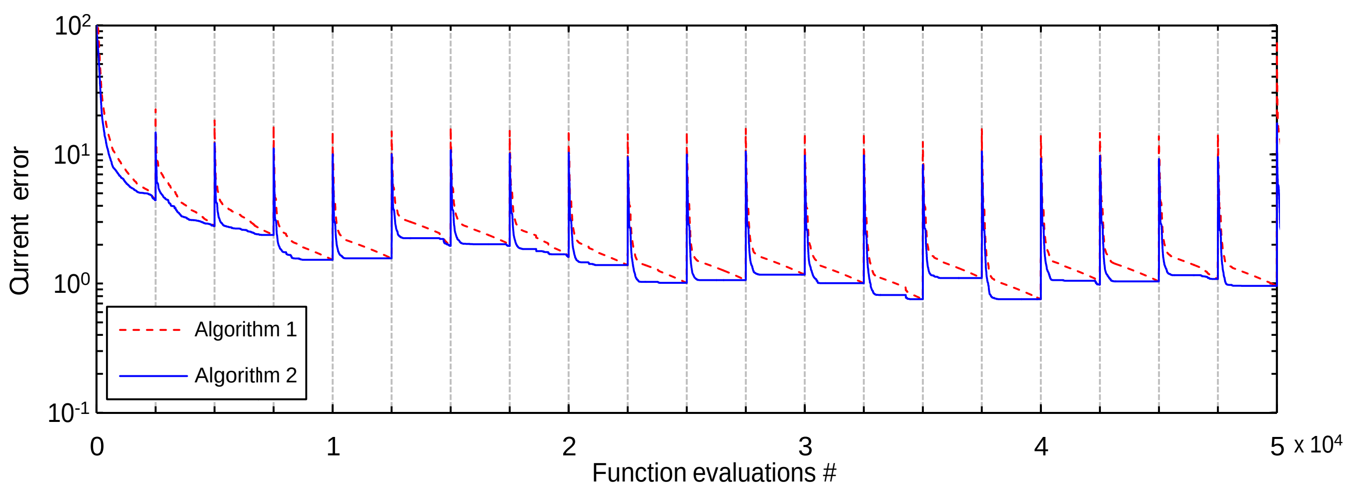

Figure 2. Evolution of the current error versus the number of function evaluations for two different methods on a typical

dynamic environment with change frequency 5000. The vertical dotted lines in the figure indicate the start of a new period.

In the paper [5], the performance comparison between MHSA-ExtArchive and other state-of-the-art algorithms is provided in Table 10. In Table 10, the results for the competing algorithms have been directly reported from their respective papers. Considering Table 10, by checking the performance measure used by the contestant methods, it can be seen that some of the reported results are based on offline error and some others are based on best error before change. For example, the results for CPSO by Yang and Li [6] are based on best error before change and the results for EO+HJ by Moser and Chiong [2] are based on offline error. Therefore, conclusions cannot be drawn about superiority of the MHSA-ExtArchive over the other tested algorithms.

Moreover, the authors stated in the title of Table 10 that they have reported the results in the form of offline error ± standard deviation, but the results for some of the peer algorithms in their original papers were in the form of offline error ± standard error, where standard error is calculated as standard deviation divided by the squared root of the number of runs, that makes the comparisons shown in Table 10 to be unfair.

From the above discussions, the researchers in this domain are strongly recommended to be more careful when comparing their methods with other contestant optimization algorithms taken from the literature.

References

[1] J. Branke, H. Schmeck, Designing Evolutionary Algorithms for Dynamic Optimization Problems, in: A. Ghosh, S. Tsutsui (Eds.), Advances in Evolutionary Computing, Springer Berlin Heidelberg, 2003: pp. 239–262.

[2] I. Moser, R. Chiong, Dynamic function optimisation with hybridised extremal dynamics, Memetic Comp. 2 (2010) 137–148.

[3] T.T. Nguyen, S. Yang, J. Branke, Evolutionary dynamic optimization: A survey of the state of the art, Swarm and Evolutionary Computation. 6 (2012) 1–24.

[4] K. Trojanowski, Z. Michalewicz, Searching for optima in non-stationary environments, in: Proceedings of the 1999 Congress on Evolutionary Computation, 1999: pp. 1843–1850.

[5] A.M. Turky, S. Abdullah, A multi-population harmony search algorithm with external archive for dynamic optimization problems, Information Sciences. 272 (2014) 84–95.

[6] S. Yang, C. Li, A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments, IEEE Transactions on Evolutionary Computation. 14 (2010) 959–974.