

A Brief Review of Solving Dynamic Optimization Problems

Nazanin Fouladgar^a, Shahriar Lotfi^b

^a *Computer Engineering Department, College of Nabi Akram, Tabriz, Iran.*

^b *Computer Science Department, University of Tabriz, Tabriz, Iran.*

Abstract

Dynamic environments are those which possess the potentiality of changes over time. These changes can appear on parameters and different environmental agents. In this regard, some problems, called Dynamic Optimization Problems (DOPs), are defined both theoretically and practically in the real world. The most important challenge in solving these kinds of problems is related to how to become compatible with the newly changed environment as much as possible. Therefore, it is necessary to track and follow up the new optimum (optima) of the search space in optimization process. Moreover, finding the optimum (optima) as accurately as possible is another necessity of this compatibility. In order to meet the challenges, researchers intended to benefit from applying swarm intelligence, evolutionary algorithms inspired by the principle of evolving process and a set of special mechanisms. This paper reviews the methods of solving DOPs with some of their previous works. Diversity Scheme, Memory and Diversity Scheme, Multi-Population Scheme and Other Solutions are emphasized on these methods.

Keywords: Dynamic Environments, Dynamic Optimization Problems, Optimization and Evolutionary Algorithms

Introduction:

Optimization is a process of finding the most suitable value of certain activities regarding the available resources. Thus, it is considered among one of the major domains for researchers. There are different categories of this area's problems depending on the scope of applications. One category can be based on unchangeable or changeable environments. If the environment encounters without any changes, the problems are defined as Static Optimization Problems and if the environment encounters with changes, the problems are defined as Dynamic Optimization Problems. The goal of first type is to find and estimate the optimum (optima) as accurate as possible. However, in the second one, which is the purpose of this paper, not only the goal of first type should be satisfied but also the optimum should be tracked as quickly as possible. This arises from the fact that when the environment changes, the optimum (optima) may move to another area of search space.

Researchers benefited from swarm intelligence, evolutionary algorithms inspired by the principle of evolving process and some adaptive mechanisms to make them compatible with environmental changes. According to the previous works up to now, we can divide the solutions of solving DOPs into four categories: (1) Diversity Scheme (2) Memory and Diversity Scheme (3) Multi-Population Scheme (4) Other Solutions. In this domain, a lot of algorithms consist of genetic, memetic, Particle Swarm Optimization (PSO), differential evolution, ant colony optimization, artificial immune, artificial fish swarm, firefly, harmony search and population-based incremental learning have been employed with a set of mechanisms. Also, in one case, a kind of adaptive evolutionary strategy has been applied to meet the requirements of environmental changes.

Basic Concepts

In this section, dynamic environments and dynamic optimization problems are studied. In addition, the performance measure and dynamic benchmarks are determined to compare algorithms with each other.

Dynamic Environments and Dynamic Optimization Problems

Dynamic Environments are the environments consist of continuous or discrete changes over time. Accordingly, the changes cause the shift of optimum (optima) location in the environment. A set of problems, called Dynamic Optimization Problems, are specified in this area. Precisely, they are defined as bellows [1]:

$$DOP = \left\{ \begin{array}{l} \text{optimize } f(x, t) \\ \text{s.t. } x \in F(t) \subseteq S, t \in T \end{array} \right. \quad (1)$$

In which $S \in R^n$ is the search space, t is time, $f: S \times T \rightarrow R$ is fitness function that associates the values of $f(x, t) \in R$ to each feasible solution $x \in S$ in time t and $F(t)$ is a set of feasible solutions ($x \in F(t) \subseteq S$) in time t . As DOPs are considered in a comprehensive domain of different sciences such as computer, management, economics, mathematics and so on, it's worth study about them.

Problem Performance

Offline error is used to measure the performance of algorithms in the problem space. It's equal to the average of deviance between the best found position by the algorithm in time t and the position of the global optimum in the same environment [2]. The mathematical form of offline error can be seen as bellows:

$$offline_error = \frac{1}{FEs} \sum_{t=1}^{FEs} (fitness(gbest(t)) - fitness(globalOptimum(t))) \quad (2)$$

in which FEs is the maximum function evaluations, $gbest(t)$ and $globalOptimum(t)$ are respectively the best found position by the algorithm and global optimum at t th fitness evaluation.

Dynamic Benchmarks

Over years, some dynamic benchmarks have been introduced to compare the performance of algorithms in dynamic environments. One of widely used ones, called Moving Peak Benchmark (MPB), was presented by Branke. This benchmark is created by shape changes in fitness function. It consists of several peaks in a multi-dimension space in which the height, width and position of peaks can change in the time of environmental changes [3]. Moreover, in [4], Li and Yang presented Generalized Dynamic Benchmark Generator (GDBC) by changes in system control parameter(s). It includes 7 change types: small step, large step, random, chaotic, recurrent, recurrent with noise and dimensional change.

Previous Works

As mentioned before, researchers employed some algorithms and mechanisms to solve DOPs. To do so, the solutions are classified into four different groups in this paper: (1) Diversity Scheme (2) Memory and Diversity Scheme (3) Multi-Population Scheme (4) Other solutions. In this section, some previous works up to now in each class are studied.

Diversity Scheme

In static optimization, convergence of population is one of the essentialities of finding optimum (optima) in promising areas. However, in dynamic optimization, it may have negative effects because changes may occur in the area that there aren't any existing members of population. Consequentially, the changes aren't detected. Diversity enhancement in population is one of the solutions to meet this problem.

In 1993, Cobb and Grenfestet [5] studied three strategies of increasing diversity including a high level of mutation rate, the Random Immigrants and the Triggered Hypermutation on genetic algorithm. The first mechanism was like what there had been in the standard genetic algorithm, the second one specified replacement of the fraction of population with the random generated members and the third one used a high value of mutation rate whenever the average best performance of the population decreases. The results showed that it was useful to apply the first mechanism for linearly and continuously changing environments and the second one for abruptly changing environments. The last strategy, Triggered Hypermutation, performed well for certain classes of environmental changes. However, it sometimes didn't leave suitable results in abruptly changing environments.

In another work, in 2009 [6], the memetic algorithm based on genetic algorithm was presented to perform a better local search with a learning mechanism along with hill-climbing. Also, to increase diversity, two methods of Adaptive Dual Mapping (ADM) and Triggered Random Immigrants were applied. Dual of each individual was defined as its complement and Triggered Random Immigrants were such as what there were in [5] but the immigrants were introduced into the algorithm only when the convergence degree of population was below a threshold. Finally, this algorithm showed better results than its comparable algorithms in many dynamic environments.

In 2010 [7], Rezvanian and Meybodi employed a combination of learning automata and artificial immune algorithm to increase diversity in dynamic environments. They used the adaptive setting of mutation rate to select one action according to the probabilistic vector on immune cells. After detecting environmental changes, the parameters were reset. So, this algorithm didn't cause a perfect performance in small changes environments.

In [8], the authors proposed a self-adaptive mechanism of replacement rate (the number of random generated immigrants) on evolutionary algorithms. It should be mention that in all previous related works, the value of this parameter was considered fix. As an advantage of this method, it was much less sensitive to its new presented parameter than to the replacement rate.

Evolutionary Strategy with self-adaptive mutation distribution was the other solution of this section [9]. Tinos and Yang attempted to change the shape of distribution either to converge faster to optimum (optima) or to escape from local optimum (optima). The algorithm was compared with two other ones on four dynamic environments which were generated by mobile robots and showed better results.

Memory and Diversity Scheme

If the changes in dynamic environments are recurrent and iterative, the new optimum (optima) may be located in the area near to the previous optimum (optima) position(s). In this case, reusing the information of previous optimum (optima) consumes less computational cost. To meet this requirement, a separate memory space is employed to save the useful information. Mostly, the best individual of the memory is used in the time of need.

Yang proposed a genetic algorithm in which a memory space was applied in 2005 [10]. The best individual of memory generated some Random Immigrants and then they were replaced with the worst individual of the population. Moreover, the memory was updated in random period times related to the last memory update. In this case, the best individual of the population was replaced with even any existing random points in memory or the most similar individual of memory to the best individual of population in the case of betterment.

In 2007 [11], a promoted version of genetic algorithm existed in [10] was presented. In this work, the immigrants were generated based on the best individual of previous generations. Therefore, it increased the performance of this algorithm better than the algorithm of [10].

In another work, Mavrovouniotis and Yang proposed an ant colony optimization algorithm with a short-term memory [12]. To increase diversity, they applied three mechanisms of Random Immigrants, Elitism-Based Immigrants and the combination of two previous ones. In experimental results, this algorithm showed convenient performance.

In 2014 [13] Random Immigrants, Memory Scheme and Memory-Based Random Immigrants were employed in harmony search algorithm. The results showed that Memory-Based Random Immigrants had better performance than two others on MPB.

Multi-Population Scheme

Another approach mostly used to solve DOPs, is Multi-Population Scheme. Each population can take over a part of search space and become in charge of separate tasks to find the optimum (optima). As an example, some populations may concentrate on global search while others on local search. Also some of them may search for finding promising areas and some others may track the environmental changes. However, populations can interact information with each other to get better results. The most considerable point, regarded in most of presented works, is that populations shouldn't overlap with each other in the space they are searching. It makes them avoid searching the same space and have a convenient diversity.

In 2003, Yang and Yao [14] proposed another version of population-based incremental learning algorithm, in which two population and two probability dual vectors were applied. It is worth point out that the usage of dual mechanism made the algorithm achieves high performance in subject to significant environmental changes since it increases diversity in the search space.

In another work, in 2006 [15], a Multi-Population genetic algorithm was introduced. The population of this algorithm was divided into several sub-populations among which only the best one was applied and generated two child sub-populations to perform the algorithm procedure. This method showed

convenient results in recurrent and discrete environmental changes. It was because of transferring useful previous information to next generations. As a disadvantage, it should be mentioned that there were high amounts of function evaluations especially when the frequency of changes increased.

Hashemi and Meybodi, in 2009 [16] employed PSO with Cellular Automata to solve DOPs. Cellular Automata split the population into different subpopulations across the cells. Also, the density of each cell's particles was defined to avoid the convergence of population. Using Cellular Automata (instead of calculating Euclidean distances of particles), it led to less computational cost for checking exclusions of sub-populations in the search space.

In another work, a self-adaptive version of differential evolution algorithm with a Multi-Population Scheme was proposed in 2009 by Brest et al. [17]. They also applied an age mechanism to solve DOPs. This algorithm was tested on 6 benchmarks in [4] and achieved the appropriate results.

In 2010, Kamosi, Hashemi and Meybodi [18] presented a multi-swarm local PSO algorithm. They implemented a parent swarm to explore and several child swarms to exploit the search space. After environmental changes, all parent particles were re-evaluated but the child particles changed their role to quantum particles [19] to perform local search randomly around their best previous positions of their own child swarm. Thus, it made them more useful to track optimum (optima) in small environmental changes.

In [20], again the multi-swarm scheme PSO was proposed in 2011 but with the difference of using weight particles to accelerate the algorithm's convergence after environmental changes. Also, a fuzzy clustering method was applied to check the exclusion of swarms. This method let the number of swarms converge to the number of peaks, which is a beneficial point.

Upon another algorithm, Yazdani et al. applied Multi-Population Scheme with a modified artificial fish swarm algorithm in 2012 [21]. In this work, the number of swarms was always one more than the number of peaks, since a new swarm was generated in the time of swarm convergence. After the environmental changes, the fishes of each swarm were distributed randomly in a d-dimensional ball around the best artificial fish of that swarm with a radius of shift-length peaks. Then all the fishes were re-evaluated to update the previous memory. To increase the ability of search, the Pray behavior in artificial fish swarm algorithm was performed several times in the best swarm of population. Parameters setting and diversity enhancement of this method led to track optimum (optima) perfectly after the environmental changes.

In 2012 [22], firefly algorithm with the mechanism of Multi-Population was introduced to solve DOPs. Using two populations, it was utilized discoverer swarm for exploring and fireflies swarm for exploiting the search space. The algorithm started with one population of fireflies but in the following of execution, the population was divided into several species to move through local optimum (optima). To study about modifications, we should point out the usage of best position of each firefly in each execution procedure which was inspired by PSO algorithm. Also, the random parameter of firefly algorithm became adaptive in the modification area. To increase the search ability, the idea of quantum particles [19] was applied in this work. This algorithm was attempted to satisfy the requirements of facing with dynamic environments and showed acceptable results on MPB.

Again in [23], Yazdani et al. employed Multi-Population Scheme in PSO algorithm. In this study, the scheme usage was the same as what was proposed in [21]. However, the number of particles in created swarms was always less than those in the swarm which was responsible for exploration of search space. In addition, they presented a novel mechanism, called sleep-awake to devote more function evaluations to the best swarm of population. To increase the performance of algorithm, a local search procedure was applied around the best particle of swarm and to increase diversity of algorithm, some modifications were applied in particle's velocity and position. This algorithm showed better results in comparison with other prior works.

Unlike previous methods, in 2014 [24] Kordestani et al. utilized two populations, on each of which different algorithms were performed. The first population executed Crowding-based Differential

Evolution [25] to create suitable diversity. Nevertheless, the second population either restarts around the best position of first population or performs a modified PSO algorithm based on the best position of first population. Moreover, in some cases such as trapping in optimum, a hill-climbing was employed in each genome of first population. As an advantage of this algorithm we can point out to the usage of a few numbers of genomes in first population considered to search the problem space.

Other Solutions

In other solutions to solve dynamic optimization problems, some combinatorial methods were proposed by researchers.

In 2011 [26] Junior et al. combined two algorithms of PSO and Fish School Search, called Volitive-PSO, by using feeding operator and collective-volitive movement operator. This method made particles expand or contract in an appropriate time. In the result section, it showed sensitivity to the parameters. Thus, it may decrease the algorithm performance. One year later, Volitive-PSO was developed to several species in each of which the best one was regarded as a leader [27]. All procedures of the algorithm was the same as what there was in [26] for each species, except applying local or global PSO on the leaders of species. Experiments showed better performance of this developed algorithm to solve DOPs when the leaders performed local PSO.

In another combination scheme, hill-climbing with evolutionary algorithm was employed by Turkey et al. [28]. In this work, it was utilized Perturbation mechanism in evolutionary algorithm. Then a solution was selected and over several iterations of hill-climbing, the best one was added to the population in case of betterment. Finally, the experiments were achieved competitive results with the comparable methods on the dynamic optimization functions.

Again, Sharifi et al. applied PSO algorithm but this time with local search methods in 2015 [29]. In this case they benefited of a fuzzy social-only model for exploring the promising areas and some local search methods for exploiting them. In addition, to increase the performance of algorithm, a density control mechanism was proposed. The role of this mechanism was the allocation of more local search to the best agent and deactivation of useless local search agents. The results showed the suitable performance of proposed algorithm in solving DOPs.

Conclusion and Future Works

In this paper, we categorized the solutions of solving dynamic optimization problems into four classes and reviewed some previous works up to 2015.

Diversity Scheme is one of the solutions in dynamic environments. When the optimum changed its position after the environmental changes, the optimizer should have been able to cover all promising areas in the search space. In this regard, increasing diversity was proposed to meet the requirements. This solution was performed on genetic, memetic and artificial immune algorithms.

In another class, employment of Memory with Diversity Scheme was presented. This solution was convenient for small environmental changes in which we needed to reuse the previous information. Among algorithms, genetic, ant colony optimization and harmony search were applied on this class.

One of the most widely used solutions was Multi-Population Scheme. This solution was utilized several populations in the search space each of which undertook similar or different responsibility. Some of them explored and some others exploited the search space. It is worth mention that the most usage of Multi-Population Scheme was on PSO algorithm. However, it was employed in population-based incremental learning, genetic, artificial fish swarm, firefly and differential evolutionary algorithms too.

Finally, some combinatorial methods were presented in the class of other solutions to solve dynamic optimization problems. The used algorithms were the combination of PSO and fish school search, hill-climbing and evolutionary algorithm and also PSO and local search.

Among the future works, we can apply from the potentiality of other algorithms and their combinations with some learning mechanisms to track and find optimum (optima) in dynamic environments. Moreover, according to variability and adaptation of human behavior in different solutions, it is worth investigating their modelization and employments to follow the environmental changes.

References:

- Brest J., Zamuda A., Boskovic B., Maucec M.S., Žumer V. (2009). Dynamic Optimization using Self-Adaptive Differential Evolution, *Evolutionary Computation*, 2009. CEC '09. IEEE Congress on, 415-422.
- Cavalcanti-Júnior G.M., Bastos-Filho C.J.A., Lima-Neto F.B.D. (2012). Volitive Clan PSO - An Approach for Dynamic Optimization Combining Particle Swarm Optimization and Fish School Search. In: Parpinelli R: *Theory and New Applications of Swarm Intelligence*, ISBN 978-953-51-0364-6. Croatia, 69-86.
- Cavalcanti-Júnior G.M., Bastos-Filho C.J.A., Lima-Neto F.B.D., Castro R.M.C.S. (2010). A Hybrid Algorithm Based on Fish School Search and Particle Swarm Optimization for Dynamic Problems. *Advances in swarm intelligence*, 6729, 543-552.
- Cobb H.G., Grefenstette J.J. (1993). Genetic Algorithms for Tracking Changing Environments. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 523-530.
- Cruz C., Gonzalez J.R., Pelta D.A. (2011). Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing*, 15, 1427-1448.
- Hashemi A.B., Meybodi M.R. (2009). Cellular PSO: A PSO for Dynamic Environments. *ISICA '09 Proceedings of the 4th International Symposium on Advances in Computation and Intelligence*, 422-433.
- Kamosi M., Hashemi A.B., Meybodi M.R. (2010). A New Particle Swarm Optimization Algorithm for Dynamic Environments. *Swarm, Evolutionary, and Memetic Computing*, 6466, 129-138.
- Kordestani J.K., Rezvanian A.R., Meybodi M.R. (2014). CDEPSO: a bi-population hybrid approach for dynamic optimization algorithms. *Applied Intelligence*, 40, 682-694.
- Li C., Yang S., Nguyen T.T., Yu E.L., Yao X., Jin Y., Beyer H.-G., Suganthan P.N. (2008). Benchmark Generator for CEC'2009 Competition on Dynamic Optimization. University of Leicester, University of Birmingham, Nanyang Technological University.
- Li, C. (2010). Particle Swarm Optimization in Stationary and Dynamic Environments, Doctor of Philosophy Thesis, University of Leicester.
- Mavrovouniotis M., Yang S. (2010). Ant Colony Optimization with Immigrants Schemes in Dynamic Environments. *PPSN'10 Proceedings of the 11th international conference on Parallel problem solving from nature: Part II*, 371-380.
- Nasiri B., Meybodi M.R. (2012). Speciation based firefly algorithm for optimization in dynamic environments. *International Journal of Artificial Intelligence*, 8, 118-132.
- NGUYEN T.T. (2010). Continuous Dynamic Optimisation Using Evolutionary Algorithm. Doctor of Philosophy Thesis, University of Birmingham.

- Pant M., Thangaraj R., Abraham A. (2008). A New Quantum Behaved Particle Swarm Optimization. GECCO '08 Proceedings of the 10th annual conference on Genetic and evolutionary computation, 87-94.
- Rezazadeh I., Meybodi M.R., Naebi A. (2011). Adaptive Particle Swarm Optimization Algorithm for Dynamic Environments. *Advances in swarm intelligence*, 6728, 120-129.
- Rezvanian A., Meybodi M.R. (2010). Tracking Extrema in Dynamic Environments Using a Learning Automata-Based Immune Algorithm. *Grid and Distributed Computing, Control and Automation*, 121, 216-225.
- Sharifi A., Kordestani J.K., Mahdavian M., Meybodi M.R. (2015). A Novel Hybrid Adaptive Collaborative Approach Based on Particle Swarm Optimization and Local Search for Dynamic Optimization Problems. *Applied Soft Computing*, 32, 432-448.
- Thomsen R. (2004). Multimodal optimization using crowding-based differential evolution. *Congress on Evolutionary Computation*, 1382-1389.
- Tinós R., Yang S. (2010). Evolution Strategies with q-Gaussian Mutation for Dynamic Optimization Problems. *Neural Networks (SBRN), 2010 Eleventh Brazilian Symposium on*, 223-228.
- Turky A.M., Abdullah S., McCollum B., Sabar N.R. (2013). An Evolutionary Hill Climbing Algorithm for Dynamic Optimization Problems. *The 6th Multidisciplinary Int. conf. on Scheduling: Theory and Applications (MISTA 2013)*, 27-30.
- Turky A.M., Abdullah S., Sabar N.R. (2014). A Hybrid Harmony Search Algorithm for Solving Dynamic Optimization Problems. *International Conference on Computational Science*, 29, 1926-1936.
- Wang H., Wang D., Yang S. (2009). A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Computing*, 13, 763-780.
- Xin Y., Ke T., Xin Y. (2011). Immigrant schemes for evolutionary algorithms in dynamic environments: Adapting the replacement rate. *Science China Information Sciences*, 54, 1352-1364.
- Yang S. (2005). Memory-Based Immigrants for Genetic Algorithms in Dynamic Environments. *Genetic and Evolutionary Computation Conference*, 1115-1122.
- Yang S. (2007). Genetic Algorithms with Elitism-Based Immigrants for Changing Optimization Problems. *Applications of Evolutionary Computing*, 4448, 627-636.
- Yang S., Yao X. (2003). Dual Population-Based Incremental Learning for Problem Optimization in Dynamic Environments. In *Proceeding of the 7th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, 49-56.
- Yazdani D., Akbarzadeh-Totonchi M.R., Nasiri B., Meybodi M.R. (2012). A New Artificial Fish Swarm Algorithm for Dynamic Optimization Problems. *Evolutionary Computation (CEC), 2012 IEEE Congress on*, 1-8.
- Yazdani D., Nasiri B., Sepas-Moghaddam A., Meybodi M.R. (2013). A Novel Multi-Swarm Algorithm For Optimization In Dynamic Environments Based On Particle Swarm Optimization. *Applied Soft Computing*, 13, 2144- 2158.
- Zwanepol Klinkmeijer L., de Jong E.D., Wiering M.A. (2006). A Serial Population Genetic Algorithm for Dynamic Optimization Problems. *Proceedings of the 15th Belgian-Dutch Conference on Machine Learning*, 41-48.