A LONG-TERM MEMORY APPROACH FOR DYNAMIC MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

Alan Díaz-Manríquez, Gregorio Toscano-Pulido and Ricardo Landa-Becerra
Information Technology Laboratory, CINVESTAV-Tamaulipas, Parque Científico y Tecnológico TECNOTAM
Km. 5.5 carretera Cd. Victoria-Soto La Marina, Cd. Victoria, Tamaulipas 87130, Mexico

Keywords: Evolutionary algorithms, Dynamic multiobjective optimization.

Abstract: A dynamic optimization problem (DOP) may involve two or more functions to be optimized simultaneously, as well as constraints and parameters which can be changed over time, it is essential to have a response approach to react when a change is detected. In the past, several memory-based approaches have been proposed in order to solve single-objective dynamic problems. Such approaches use a long-term memory to store the best known solution found so far before a change in the environment occurs, such that the solutions stored can be used as seeds in subsequent environments. However, when we deal with a Dynamic Multiobjective Problems with a Pareto-based evolutionary approach, it is natural to expect several traded-off solutions at each environment. Hence, it would be prohibitive to incorporate a memory-based methodology into it. In this paper, we propose a viable algorithm to incorporate a long-term memory into evolutionary multiobjective optimization approaches. Results indicate that the proposed approach is competitive with respect to two previously proposed dynamic multiobjective evolutionary approaches.

1 INTRODUCTION

Since life is dynamic, it is only natural to expect that the problems from daily life are dynamics. A dynamic optimization problem may involve two or more functions to be optimized simultaneously (also known as dynamic multiobjective optimization problems, or DMOPs for short), as well as constraints and parameters which can be changed over time. Although the study of this type of problems is not new, most of the proposed approaches transform the original dynamic problem into many static optimization problems. The evolutionary computation community has focused their efforts on designing approaches to solve these problems without performing any transformation.

This work proposes to incorporate a change response methodology into Dynamic Multiobjectve Evolutionary Algorithms (DMOEAs). Such methodology uses a long-term memory which minimizes the information to be stored in order to replicate a specific state of the search if it is needed in subsequent environments.

The remain of this paper is organized as follows: In Section 2, we present the dynamic multiobjective state-of-the-art. Section 3 describes the proposed algorithm and its conformed components. Section 4 presents the experiments and comparison of results. Finally, Section 5 provides our conclusions as well as some possible directions for future research.

2 RELATED WORK

Evolutionary algorithms have been successfully applied to solve DMOPs. Their success rate might be directed for their population-based nature, since this allows them to use the most of the previous discovered knowledge in order to follow a change in the environment. Bingul (Bingul, 2007) solved a dynamic multiobjective optimization problem (DMOP) using an aggregating function approach with a Genetic Algorithm (GA). Hatzakis and Wallace (Hatzakis and Wallace, 2006) proposed a forward-looking approach which combines a forecasting technique with an evolutionary algorithm. Deb *et al.* proposed two modifications to the NSGA-II in order to able it to handle DMOPs. In the first modification, the population is reinitialized while in the second, the population is mutated depending on the type of change in the environment (Deb et al., 2006). Talukder and Kirley (Talukder and Kirley, 2008) used a new variation operator to follow the Pareto front in DMOPs.

Díaz Manríquez A., Toscano-Pulido G. and Landa-Becerra R..
A LONG-TERM MEMORY APPROACH FOR DYNAMIC MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS.
DOI: 10.5220/0003675403330337
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2011), pages 333-337
ISBN: 978-989-8425-83-6

3 PROPOSED APPROACH

Since the natural behavior of a DMOP is to be changing, it is essential to perform a response action when a change is detected. Several authors (Mori et al., 1996; Branke, 1999; Branke et al., 2000) have proposed memory-based approaches in order to solve single-objective dynamic problems. Such approaches store the best known solution found so far before a change in the environment occurs and use such stored solutions to be seeds in subsequent environments. However, when we deal with a DMOP using a Pareto-based approach, it is natural to expect several traded-off solutions for each environment. Hence, it would be prohibitive to incorporate a memory-based methodology into it. However, if we had a special interpolation operator which could restore the previously known $\mathcal{PF}$ using only few points, then it would be possible to have a memory-based multiobjective optimization approach.

The basis of our proposal lays on the construction of a special interpolation operator that will predict several non-dominated solutions in order to connect two points located on the extremes of the $\mathcal{PF}$. We adopt a methodology which uses a long-term memory that minimizes the information to be store in order to reproduce a specific state of the search. Thereby, the methodology uses such information as knowledge in subsequent environments. However, it is necessary to generate new knowledge from few points. To achieve this goal we propose a method to generate non-dominated solutions between a pair of non-dominated solutions. In this manner, if we choose the extremes of the original $\mathcal{PF}$, we will be able to generate solutions which presumable will belong to the $\mathcal{PF}$. The proposed operator to achieve works as follows:

3.1 Operator to Create Solutions from the Extremes of a Bi-objective $\mathcal{PF}$

Given two non-dominated points $\vec{a} = [a_1, \ldots, a_n]^T$ and $\vec{b} = [b_1, \ldots, b_n]^T$, such that $f_1(\vec{a}) < f_1(\vec{b}) \wedge f_2(\vec{a}) > f_2(\vec{b})$.

We want to find the point $\vec{z} \in \mathcal{P}^t$ whose evaluation in the objective space $\vec{f}(\vec{z})$ is located between $\vec{f}(\vec{a})$ and $\vec{f}(\vec{b})$ (i.e., $f_1(\vec{a}) < f_1(\vec{z}) < f_1(\vec{b}) \wedge f_2(\vec{a}) > f_2(\vec{z}) > f_2(\vec{b})$. The complete procedure to find $\vec{z}$ is shown below:

Step 1. Construct a system of equations of the form $\mathbf{AX} = \mathbf{C}$:

$$\mathbf{A} = \begin{bmatrix} f_1(\vec{a}) & f_2(\vec{a}) \\ f_1(\vec{b}) & f_2(\vec{b}) \end{bmatrix} \mathbf{X} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & \ldots & X_{1n} \\ X_{21} & X_{22} & X_{23} & \ldots & X_{2n} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} a_1 & a_2 & a_3 & \ldots & a_n \\ b_1 & b_2 & b_3 & \ldots & b_n \end{bmatrix}$$

Step 2. Solve the system of equations in order to obtain $\mathbf{X}$.

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{C} \tag{1}$$

Step 3. Construct $\vec{f}'(\vec{z})$ with desired objectives. This is performed increasing in Δ the first objective and decreasing in Δ the second one (using Equation 2).

$$\vec{f}'(\vec{z}) = \begin{bmatrix} f_1(\vec{a}) + \Delta & f_2(\vec{a}) - \Delta \end{bmatrix} \tag{2}$$

Step 4. Multiply $\mathbf{X}$ by $\vec{f}'(\vec{z})$ in order to obtain the $\vec{z}$ values (using Equation 3).

$$\vec{z} = \vec{f}'(\vec{z})\mathbf{X} \tag{3}$$

Step 5. Once the $\vec{z}$ value is computed, it is necessary to evaluate it in order to calculate its true objective values, i.e. to compute $\vec{f}(\vec{z})$. The new computed point will serve as seed for the next point to be generated.

This mechanism will be used until $f_1(\vec{z}) > f_1(\vec{b})$, (i.e., when the current approximated point reach the final position).

With the aim to validate the current operator in static environments, we selected 30 pairs of points from several test functions (WFG1 to WFG9, ZDT1 to ZDT3 and Kursawe. Due to space limitations, and also because of the scope of this paper, the details are not provided here. However, the main conclusions of such experiment are:

- The proposed operator was able to connect two points in the decision variables space of the test functions whose solutions evaluated in the objective function space became non-dominated.
- The proposed approach showed satisfactory results for: ZDT1 to ZDT3, WFG2 to WFG7 and WFG9.
- The operator malfunctioned when optimizing the Kursawe test functions due to the disconnection of the problem in the parameter space. However, when the extreme points of any connected region in the parameter space were provided, then the operator was able to approximate the portion of the front belonged to such space. On the other hand, when the operator was trying to approximate WFG1 and WFG8 did not work as we expected, since they are disconnected in the parameter space, so it was impossible to find a path between two disconnected points in $\mathcal{SP}$.

3.2 Methodology for Environmental Change Response

Above we proposed an operator to generate solutions between two points belonging to a connected region in the parameter space. Given such operator, we can develop an algorithm to reduce the information needed to produce a Pareto by storing only two extreme solutions by change in the enviroment. Below, we present such methodology:

Once that the change in the environment is detected, we can use the solutions previously stored in the repository. First, the repository and the current population should be re-evaluated such that we can integrate them in order to obtain the overall extreme points of the current Pareto front. From this two solutions, we should create a $\zeta\%$ new solutions using the proposed operator (shown in Section 3.1). The remain $100-\zeta\%$ population will be randomly generated in order to incorporate diversity to the population. This procedure is shown in Algorithm 1. Once the response to the change is achieve. Then, we can use the produced population with any MOEA. In this case, we used the NSGA-II (Algorithm 2).

Algorithm 1: Response to an environmental change.

```
F = Non-dominated solutions of P^t
Report F
Extreme = extreme solutions from F
M = M ∪ Extreme
Reevaluate M and F
Obtain the extreme non-dominated solutions from M ∪ F
Replace a ζ% of F with new individuals generated with the
    procedure shown in Section 3.1).
Generate 100 − ζ% randomly individuals
```

Algorithm 2: DNSGA-II-LTM.

```
t = 0
M = ∅
Initialize the population P^t
𝓕 = nondominatedsorting(P^t)
For each F_i ∈ 𝓕 do
    crowdingdistance(F_i)
end for
repeat
    if environmental change is detected then
        Response to the change (Algorithm 1)
    end if
    P^{t+1} = Elitism
    t = t + 1
until Termination criteria is fulfilled
```

4 EXPERIMENTS AND COMPARISON OF RESULTS

In order to know how competitive is our approach, we decided to compare our results with respect to those obtained by the DNSGA II-A and the DNSGA II-B (Deb et al., 2006). In order to have a fair comparison, we hand-tuned each change to be activated every 500 evaluations of the objective function. For such sake, we setup the following empirical tuning: Simulated binary crossover ($P_c = 0.9$, $\eta_c = 15$), parameter-based mutation ($P_m = 1/nvars$,$\eta_c = 20$), $\mathcal{P} = 100$, $\zeta = 30$. Two test functions were taken from the specialized literature to compare our approaches and other two are proposed by us. In order to allow a quantitative assessment of the performance of a multiobjective optimization algorithm two standard performance measures were adopted: the Inverted Generational Distance (IGD) and the Hypervolume ratio (HVR).

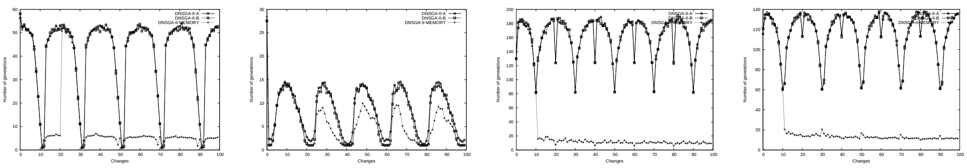
The tests functions used for the algorithms are FDA1 and FDA2, both were proposed by Farina *et al.* (Farina et al., 2003), and the DZDT2 and DZDT3, are proposed here. They are modifications to the ZDT2 and ZDT3, respectively and are shown in Table 1.

Table 1: Tests functions.

Dynamic Multiobjective Problems
DZDT2
$f_1(\vec{x}_I) = x_1$, $f_2(\vec{x}) = g(\vec{x}_{II})\left(1-\left(\frac{x_1}{g(\vec{x}_{II})}\right)^2\right)$, $G(t) = \sin(0.5\pi t)$, $t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_T} \right\rfloor$ $g(\vec{x}_{II}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n}(x_i - abs(G(t)))$ τ is the generation counter and τ_T is the number of generations that t is fixed. $x_1,\ldots,x_n \in [0,1]$ and $n = 30$
DZDT3
$f_1(\vec{x}_I) = x_1$, $f_2(\vec{x}) = g(\vec{x}_{II})\left(1-\sqrt{\frac{x_1}{g(\vec{x}_{II})}} - \frac{x_1}{g(\vec{x}_{II})}\sin(10\pi x_1)\right)$ $G(t) = \sin(0.5\pi t)$, $t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_T} \right\rfloor$ $g(\vec{x}_{II}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} abs(x_i - abs(G(t)))$ τ is the generation counter and τ_T is the number of generations that t is fixed. $x_1,\ldots,x_n \in [0,1]$ and $n = 30$

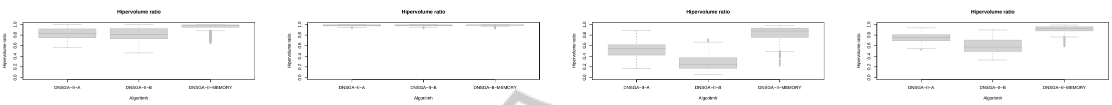
We measured the number of generations that requires our approach, the DNSGA-II-A and the DNSGA-II-B to approximate the true $\mathcal{PF}$ at a IGD distance of 0.01 during 100 changes. This procedure was executed 100 times and the solutions obtained were averaged. The we plotted the mean of such solutions during the 100 changes. This experiment was made in order to check if the use of memory can help to the algorithm to decrease the number of generations needed to approximate to the true $\mathcal{PF}$.

In Figure 1(a) we can see as the DNSGA-II-LTM could stabilize its behavior over the time, in such a way that it could reduce the number of generations necessary to follow the $\mathcal{PF}$ of FDA1 (to a value of IGD=0.01). Figures 1(c) and 1(d) show how the two proposed problems were more difficult that the two problems taken from the literature, since the number of generations needed to reach the IGD=0.01 was considerably higher, in these problems it is easy to observe a similar behavior that the one observed in



(a) FDA1 (b) FDA2 (c) DZDT2 (d) DZDT3

Figure 1: Number of generations required for approximate at the $\mathcal{PF}$ at a IGD of 0.01. Problem FDA1.



(a) FDA1 (b) FDA2 (c) DZDT2 (d) DZDT3

Figure 2: Boxplots of 100 executions for FDA1, FDA2D, DZDT2 and DZDT3 during 100 changes.

FDA1 (the algorithm DNSGA-II-LTM could stabilize its behavior over the time).

In FDA2 the three algorithm presented a good behavior, but again, the DNSGA-II-LTM reduced the number of generations needed to follow the $\mathcal{PF}$ at a IGD of 0.01 (see Figure 1(b)).

Since results from IGD indicate that the proposed modification could outperform NSGA-II-A and NSGA-II-B, we decided to perform a second experiment. For this new experiment, we decided to measure the hypervolume ratio during 100 changes using the three approaches (ours, the DNSGA-II-A and the DNSGA-II-B). In order to present such results in a friendly-comparison way, we decided to present them as box-plot graphics which are shown in Figure 2.

From box-plots, we can see that the DNSGA-II-LTM could reach to a hypervolume value close to 1.0 in most of the problems. The anomalous results shown also in boxplots are due to the start of the optimization process the algorithm had not sufficient knowledge in memory and therefore, it was more difficult to follow the movement of the optimum. But when the algorithm gained enough knowledge, the algorithm could reach the $\mathcal{PF}$ most of the time.

5 CONCLUSIONS

According at the obtained result we can conclude that the use of a long-term memory in dynamic multiobjective evolutionary algorithms reduces the number of fitness function evaluations needed to optimize a DMOP. It should be clear that in order to use a long-term memory-based approach is necessary to use a method to reduce the amount of information of each environment to be stored in order to avoid saturation of the memory. The proposed approach to generate solutions from two points (in a connected region in the parameter space) can be used to store only two solutions at each change of the environment. It should be noted that this method was only tested for problems with two objective functions. However, we are planning to explore more dimensions in the objective space in the future.

ACKNOWLEDGEMENTS

The first author gratefully acknowledges support from CONACyT through project 105060. Also, this research was partially funded by project number 51623 from “Fondo Mixto Conacyt-Gobierno del Estado de Tamaulipas”. We would like to thank to Fondo Mixto de Fomento a la Investigación científica y Tecnológica CONACyT - Gobierno del Estado de Tamaulipas for their support to publish this paper.

REFERENCES

Bingul, Z. (2007). Adaptive Genetic Algorithms Applied to Dynamic Multi-Objective Problems. *Appl. Soft Comput.*, 7(3):791–799.

Branke, J. (1999). Memory enhanced evolutionary algorithms for changing optimization problems. In *Congress on Evolutionary Computation CEC99*, pages 1875–1882. IEEE.

Branke, J., Kaussler, T., Schmidt, C., and Schmeck., H. (2000). A multi-population approach to dynamic optimization problems. In *Adaptive Computing in Design and Manufacturing*, pages 299–307.

Deb, K., N., U. B. R., and Karthik, S. (2006). Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. In *EMO*, pages 803–817.

Farina, M., Deb, K., and Amato, P. (2003). Dynamic Multiobjective Optimization Problems: Test Cases, Approximation, and Applications. In Fonseca, C. M., Fleming, P. J., Zitzler, E., Deb, K., and Thiele, L., editors, *Evolutionary Multi-Criterion Optimization. Sec-*

ond International Conference, EMO 2003, pages 311–326, Faro, Portugal. Springer. Lecture Notes in Computer Science. Volume 2632.

Hatzakis, I. and Wallace, D. (2006). Dynamic Multi-Objective Optimization with Evolutionary Algorithms: A Forward-Looking Approach. In et al., M. K., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 2, pages 1201–1208, Seattle, Washington, USA. ACM Press. ISBN 1-59593-186-4.

Mori, N., Imanishi, S., Kita, H., and Nishikawa, Y. (1996). Adaptation to a changing environment by means of the thermodynamical genetic algorithm. In Voigt, H., editor, *Parallel Problem Solving from Nature*, volume 1141 of *LNCS*, pages 513–522. Springer Verlag, Berlin.

Talukder, A. K. A. and Kirley, M. (2008). A pareto following variation operator for evolutionary dynamic multi-objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation 2008 (CEC 2008)*, Hong Kong, China. IEEE Press, Piscataway, NJ.