Towards High Speed Multiobjective Evolutionary Optimizers

A.K.M. Khaled Ahsan Talukder
Department of Computer Science and Software Engineering
The University of Melbourne
Victoria, Australia
akmkat@csse.unimelb.edu.au

ABSTRACT

One of the major difficulties when applying Multiobjective Evolutionary Algorithms (MOEA) to real world problems is the large number of objective function evaluations. Approximate (or surrogate) methods offer the possibility of reducing the number of evaluations, without reducing solution quality. Artificial Neural Network (ANN) based models are one approach that have been utilized to approximate the future front from the current available fronts with acceptable accuracy levels. However, the associated computational costs limit their effectiveness. In this research project, we have developed a simple approximation technique with comparatively smaller computational cost. Our model, has been developed as a variation operator that can be utilized in any kind of multiobjective optimizer. Initial simulation experiments have produced encouraging results in comparison to other existing sequential algorithms (i.e. NSGA-II, SPEA-II). In the next phase of the project, this model will be integrated into other existing parallel MOEA's to solve more complex and time intensive bench mark problems.

Categories and Subject Descriptors: I.2.m [Artificial Intelligence]: Miscellaneous—Evolutionary computing and genetic algorithms ; I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—Heuristic methods

General Terms: Algorithms

Keywords: Evolutionary Multiobjective Optimization, Variation Operator, Dynamic System Identification, Function Evaluation

1. INTRODUCTION

A major computational bottleneck in many MOEA applications (as well as in other numerical or real-world design/optimization problems) is the evaluation of complex non-linear multiobjective functions, implying algorithmic parallelization may improve computational efficiency. Just as in single-objective optimization, "expensive" objective function evaluations (in terms of CPU time) are often completed in less wall clock time by decomposing the computational load across two or more processors. Such models typically require a large number of networked computers (scaling in size from local clusters to full Grid deployment) and an adequate parallelization of the numerical code [6]. However, a parallel approach *per se* does not necessarily reduce the number of function evaluations. Experimentations and analysis of some possible Grid deployment models for parallel MOEA can be found in [10].

The development of techniques enabling a reduction in the number of function evaluations, without reducing solution quality (without parallelization) is an important goal of MOEA research. An on-going challenge, therefore, is to develop good approximate methods that can be used to solve multi-objective problems while considering the number of objectives and the possible interaction between them.

In the MOEA domain, there have been relatively few papers reporting the use of surrogate models. For example, [2] and [7] have incorporated a Gaussian Random Field Metamodel into the algorithm. Artificial Neural Networks (ANN) based approximate models have also been used with some success [3, 1]. Typically, the ANN model approximates the design variables from the current front of the objective space by treating the objective vectors as inputs to the ANN and the design variables as output. Determining the best network structure (specifically, the number of hidden layers) and the total learning costs must be factored into the computational costs of the model.

In our study, we propose a novel approximation model for MOEA, which has a comparatively smaller computational cost than other surrogate models. We have developed this model as a variation operator that can be used in any multiobjective optimizer to speed up the search process by reducing the number of function evaluations and thus encourage the evolving population to follow the right trajectory towards a range of good trade-off solutions.

2. PROJECT OVERVIEW

This research project is made up of three phases: (1) the design and development of a fast converging MOEA; (2) the integration of the this Pareto-following variation operator into other parallel MOEA to solve more time intensive problems; and (3) investigating the efficacy of the model using real-world applications (i.e. Grid Workflow Scheduling, Dynamic Filter Coefficient Optimization, Circuit Sizing etc). Due to space constraints, we limit the discussion to a brief overview of the proposed model and preliminary simulation analysis. More detailed explanations and experimental results of our model can be found in [12] and [13].

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

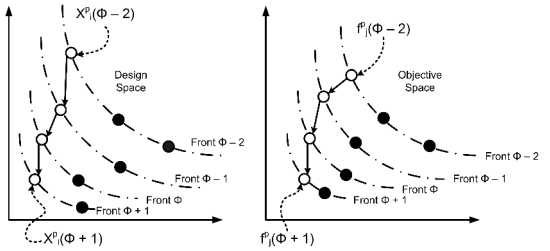

Figure 1: After nondominated sorting, individuals in each front are sorted with respect to one objective. Individual (∘) is the same individual moving from front $\phi - 2$ to front ϕ

2.1 The Pareto-Following Variation Operator

When designing and analyzing the Pareto-following variation operator (PFVO), we have considered the whole search procedure as a dynamic system [8], which takes the available objective values in the current nondominated front as inputs and generates approximated design variables for the next front as the output. As such, this approach could be thought of as a form of "Response Surface Approximation" [5]. The ANN techniques described in section 1 are representative examples of this technique. However, in our approach we have replaced the ANN with a simple Linear Time Invariant system (LTI) [9, 8].

In the initial design, we have attempted to model the dynamic system using frequency domain analysis. In a subsequent design, the linear least square method was used. Such approaches are widely used in dynamic system identification. We use the corresponding model to approximate the next front from the current front and create the so called "Mirage Solutions" [1]. The Mirage Solutions are then used as input to the dynamic system to approximate the corresponding design variables. The resulting solutions (individuals) are then added to the current population of the hosting optimizer. Although this sort of naive technique does not guarantee a 100% correct mapping from objective space to design variable space, this technique is capable of approximating the solutions that resides in close vicinity of the Pareto-front relatively quickly.

2.2 Modeling Using Frequency Domain Analysis

During our initial model development, we have applied Frequency domain analysis for the dynamic system identification. In this case, data from the design variable space $\{X(\phi), X(\phi-1), \ldots\}$ and objective space $\{F(\phi), F(\phi-1), \ldots\}$ are transformed to spatial domain (objective values and design variables in X-Y coordinate) to Fourier domain. Here $X(\phi)$ and $F(\phi)$ denote design variables and objective values in front ϕ respectively (refer to Figure 1). After that, we apply Fast Fourier Transform to find the transfer function $\mathcal{H}(f)$ of the system as indicated below -

$$\{F(\phi), F(\phi-1), F(\phi-2)\ldots\} \rightarrow \mathcal{F}(f) \quad (1)$$

$$\{X(\phi), X(\phi-1), X(\phi-2)\ldots\} \rightarrow \mathcal{X}(f) \quad (2)$$

$$\mathcal{H}(f) = \mathcal{F}(f)/\mathcal{X}(f) \quad (3)$$

Once after calculating $\mathcal{H}(f)$ of the dynamic system, the inverse function $\mathcal{H}(f)^{-1}$ can easily be calculated. Next we approximate the objective values of the next front $\{F(\phi+1)\}$ from the existing values $\{F(\phi), F(\phi-1), \ldots\}$ as -

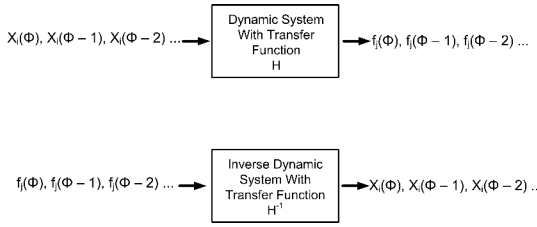

Figure 2: The forward and inverse dynamic system

$$F(\phi+1) = \{F(\phi) \pm \Delta f, F(\phi-1) \pm \Delta f, \ldots\} \quad (4)$$

and next, we inverse map the design variables of the next front -

$$\{F(\phi+1), F(\phi), F(\phi-1), \ldots\} \rightarrow \mathcal{F}_{approx}(f) \quad (5)$$

$$\mathcal{X}_{approx}(f) = \mathcal{H}(f) \cdot \mathcal{F}_{approx}(f) \quad (6)$$

and applying inverse Fourier transform,

$$\{X(\phi+1), X(\phi), X(\phi-1)\ldots\} \leftarrow \mathcal{X}_{approx}(f) \quad (7)$$

Here, $X(\phi+1)$ is the approximated design variables of the next front. The value Δf is different for different problem we have decided its value using empirical experiments. From the on going discussion, the formulated model is now capable of approximating the future Pareto-front from the existing fronts and thus can be used as a surrogate model. Figure 2 represents the idea behind this approach. The detailed implementation and analysis of this approach can be found in [12].

2.2.1 Experimental Results

To test the efficacy of this approach, we applied our model to solve benchmark dynamic Multiobjective Optimization (MOP) problems, such as FDA2, FDA3 and FDA5. All of them are converted into type-III problems. The detailed definitions of these bench mark dynamic MOP's can be found in [4]. Since in the case of dynamic MOP's, the Pareto-front changes with time and hence, if an algorithm is capable to converge to the Pareto-front within smaller function evaluations, we can infer that the algorithm will also be capable of following a changing front more efficiently. Our model proves the claim in the same sense. As a proof we illustrate the experimental results with FDA2 and FDA5 in Figure 3 and 4 respectively.

2.3 Modeling Using Linear Least Square

The second formulation is similar to the previous one. However, we have used the linear least square method in the time domain (front domain) to find the transfer function of the dynamic system. Although the page constraints do not allow us to elaborate description of this approach, the details can be found in [13]. However, Figure 5 may be helpful to understand the model.

First, we construct a simple dynamic system with i^{th} design variable and j^{th} objective:

$$x_i(\phi) + a_0 x_i(\phi-1) = b_0 f_j(\phi) + b_1 f_j(\phi-1) + \epsilon(\phi) \quad (8)$$

Here, f_j are input and x_i are considered as output.

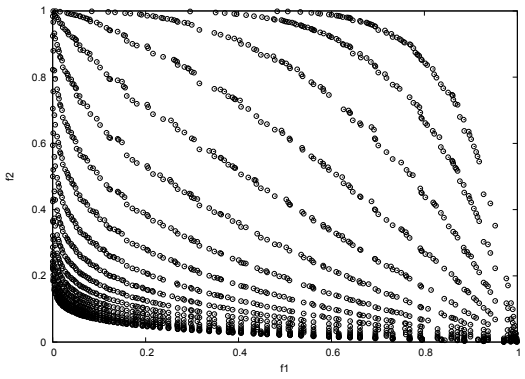

Figure 3: Tracking Time-Varying Pareto-Front in Problem FDA2

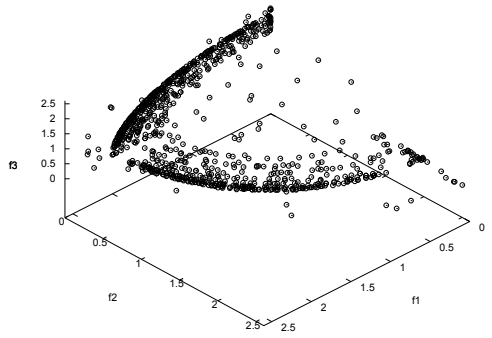

Figure 4: Tracking Time-Varying Pareto-Front in Problem FDA5(Type III)

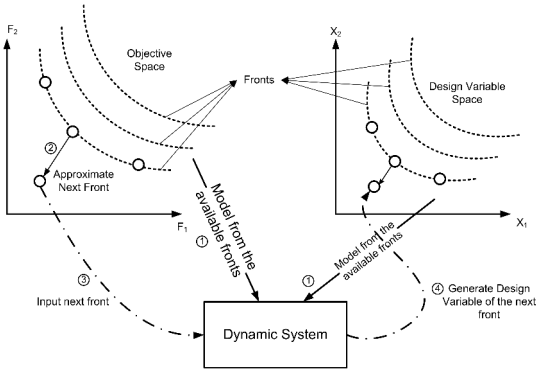

Figure 5: Working steps of the Pareto following variation operator

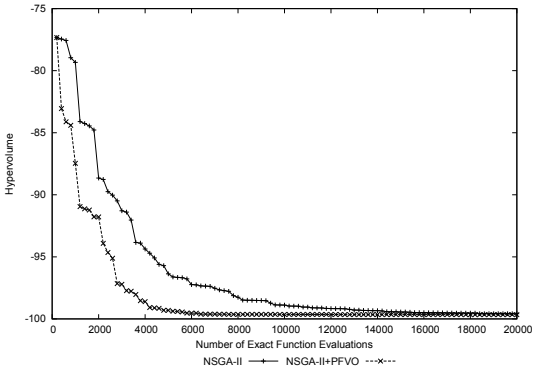

Figure 6: ZDT1: Hypervolume vs. Exact Function Evaluations by NSGA-II and NSGA-II with PFVO

Therefore,

$$x_i(\phi) = \begin{bmatrix} -x_i(\phi-1) & f_j(\phi) & f_j(\phi-1) \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \\ b_1 \end{bmatrix} + \epsilon(\phi) \tag{9}$$

Now, we can construct a matrix formation as:

$$\underbrace{\begin{bmatrix} x_i(\phi) \\ x_i(\phi-1) \\ x_i(\phi-2) \\ \vdots \\ x_i(2) \end{bmatrix}}_{\mathrm{y}} = \underbrace{\begin{bmatrix} x_i(\phi-1) & f_j(\phi) & f_j(\phi-1) \\ x_i(\phi-2) & f_j(\phi-1) & f_j(\phi-2) \\ x_i(\phi-3) & f_j(\phi-2) & f_j(\phi-3) \\ \vdots & \vdots & \vdots \\ x_i(1) & f_j(2) & f_j(1) \end{bmatrix}}_{\Phi} \cdot \underbrace{\begin{bmatrix} a_0 \\ b_0 \\ b_1 \end{bmatrix}}_{\beta_{ij}}$$

$$+ \underbrace{\begin{bmatrix} \epsilon(\phi) \\ \epsilon(\phi-1) \\ \epsilon(\phi-2) \\ \vdots \\ \epsilon(2) \end{bmatrix}}_{\epsilon} \tag{10}$$

Or we can rewrite

$$y = \Phi \cdot \beta_{ij} + \epsilon \tag{11}$$

Here, β_{ij} denotes parameter of the dynamic system for i^{th} design variable and j^{th} objective. β_{ij} can be approximated using least squares. After calculating β_{ij}, we will approximate objective values of the next fronts from the existing ones (similar as the previous case) and then apply β_{ij} to inverse map these approximated objective values to design variable space.

2.3.1 *Experimental Results*

To test the efficiency of the second approach (Dynamic System Identification using Linear Least Square), we integrated our model into NSGA-II and SPEA-II to solve some benchmark static MOP's, such as ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6. Since the space constraints does not allow us to include all experimental results, we are providing the results of ZDT1 by both algorithms in Figure 6 and 7.

From the figures, we can see that, when our Pareto-following Variation Operator (PFVO) is applied to NSGA-II and SPEA-II, the convergence speed is drastically changed. NSGA-II converges within 15000 function evaluations approximately, on the other hand, our model can reach the same Hypervolume within only 6000 function evaluations. The details of running time analysis can be found in [13].

3. FUTURE WORK: THE NEXT PHASE OF THE PROJECT

In recent years, there have been several investigations into the parallelization of MOEA's [14]. Our next target, is to extend our model and introduce parallel versions based on models such as in The Island Model, Master-Slave model and Diffusion Model [11]. In recent study [10], an enumerative

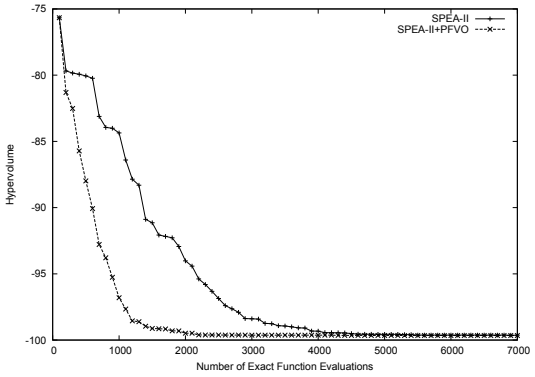

Figure 7: ZDT1: Hypervolume vs. Exact Function Evaluations by SPEA-II and SPEA-II with PFVO

MOEA has been implemented in Global Grid environment where Osyczka2 and Golinski problems were solved within 90.2 and 775.4 days respectively. Our motivation is to reduce the convergence time to solve similar problems.

4. CONCLUSION

An important contribution of this research, was the design and evaluation of a novel Pareto following variation operator for MOEA. A key component of our first approach was to approximate the next expected Pareto-front and from this expected front, generate an approximated set of design (or decision) variables. Here, we have adopted the concept of integral transformation as the approximation and mapping method. Experimental results show that by using an "inverse mapping" of the Pareto-set from the Pareto-front, the algorithm was able to track/follow the search trajectory for a given MOP.

In the second approach, we have designed and evaluated the efficacy of another novel approximation model for MOEA, which has a comparatively smaller computational cost than other surrogate models. An important contribution of this work was that our Pareto following variation operator can be used in conjunction with any nondominated sorting MOEA. There is obviously scope to investigate the relative worth of a non-linear dynamic system model such as the nonlinear ARX and Hammerstein-Wiener models [8]. This particular line of research is currently underway. Recursive System Identification may also be a useful approach to investigate.

5. REFERENCES

[1] S. F. Adra, I. Griffin, and P. J. Fleming. An Informed Convergence Accelerator for Evolutionary Multiobjective Optimiser. In *The 9th Annual Conference on Genetic and Evolutionary Computation (GECCO-2007)*, pages 734–740, London, England, 2007. ACM Press, NY, USA.

[2] M. Emmerich, K. Giannakoglou, and B. Naujoks. Single- and Multiobjective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels. *IEEE Transactions on Evolutionary Computation*, 10(4):421–439, August 2006.

[3] M. Farina. A Neural Network Based Generalized Response Surface Multiobjective Evolutionary Algorithm. In *IEEE Congress on Evolutionary Computation (CEC-2002)*, volume 1, pages 956–961, Honolulu, HI, May 2002. IEEE Press.

[4] M. Farina, K. Deb, and P. Amato. Dynamic multiobjective optimization problems: Test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation*, 8(5):425–442, October 2004.

[5] A. Gaspar-Cunha and A. Vieira. A Multiobjective Evolutionary Algorithm Using Neural Networks to Approximate Fitness Evaluations. *International Journal of Computers, Systems and Signals*, 6(1):18–36, 2005.

[6] M. Kirley and R. Stewart. An Analysis of The Effects of Population Structure on Scalable Multiobjective Optimization Problems. In *The 9th Annual Conference on Genetic and Evolutionary Computation (GECCO-2007)*, pages 845–852, London, England, 2007. ACM Press.

[7] W. Liu, Q. Zhang, E. P. K. Tsang, C. Liu, and B. Virginas. On the Performance of Metamodel Assisted MOEA/D. In L. Kang, Y. Liu, and S. Y. Zeng, editors, *ISICA-2007*, volume 4683 of *LNCS*, pages 547–557, Wuhan, China, August 2007. Springer.

[8] L. Ljung. *System Identification: Theory for The User*. Prentice-Hall Inc., NJ, USA, 1999.

[9] L. Ljung and T. Glad. *Modelling of Dynamic Systems*. Prentice-Hall Inc., NJ, USA, 1994.

[10] A. J. Nebro, E. Alba, and F. Luna. Multi-Objective Optimization using Grid Computing. *Soft Computing*, 11(6):531–540, 2007.

[11] F. Streichert, H. Ulmer, and A. Zell. Parallelization of multi-objective evolutionary algorithms using clustering algorithms. In C. A. C. Coello, A. H. Aguirre, and E. Zitzler, editors, *EMO*, volume 3410 of *Lecture Notes in Computer Science*, pages 92–107. Springer, 2005.

[12] A. K. A. Talukder and M. Kirley. 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, June 1-6 2008. IEEE Press, Piscataway, NJ. Accepted, To be published.

[13] A. K. A. Talukder, M. Kirley, and R. Buyya. A Pareto Following Variation Operator for Fast-Converging Multiobjective Evolutionary Algorithms. In *Proceedings of the* 10^{th} *Annual Conference on Genetic and Evolutionary Computation 2008 (GECCO 2008)*, Georgia, Atlanta, USA, July 12-16 2008. ACM Press, NY, USA. Accepted, To be published.

[14] D. Van Veldhuizen, J. Zydallis, and G. Lamont. Considerations in engineering parallel multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):144–173, April 2003.