Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 24 (2013) 66 – 75
Procedia Computer Science

17th Asia Pacific Symposium on Intelligent and Evolutionary Systems, IES2013

Dynamic Multiobjective Optimization Using Evolutionary Algorithm with Kalman Filter

Arrchana Muruganantham[a], Yang Zhao[a], Sen Bong Gee[a], Xin Qiu[b], Kay Chen Tan[a]

[a]Department of Electrical and Computing Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore 117576.
[b]NUS Graduate School for Integrative Sciences and Engineering, Centre for Life Sciences, #05-01, 28 Medical Drive, Singapore 117456

Abstract

Multiobjective optimization is a challenging task, especially in a changing environment. The study on dynamic multiobjective optimization is so far very limited. Benchmark problems, appropriate performance metrics, as well as efficient algorithms are required to further the research in this field. In this paper, a Kalman Filter prediction-based evolutionary algorithm is proposed to solve dynamic multiobjective optimization problems. This prediction model uses historical information to predict for future generations and thus, direct the search towards the Pareto optimal solutions. A scoring scheme is then devised to further enhance the performance by hybridizing the Kalman Filter prediction model with the random re-initialization method. The proposed models are tested and analysis of the experiment results are presented. It is shown that the proposed models are capable of improving the performances, as compared to using random re-initialization method alone. The study also suggests that additional features could be added to the proposed models for improvements and much more research in this field is still needed.

Selection and peer-review under responsibility of the Program Committee of IES2013

Keywords: Evolutionary Multiobjective (EMO), dynamic multiobjective optimization, Kalman Filter, scoring scheme, evolutionary algorithm

1. Introduction

In recent years, there has been a growing interest in the field of multiobjective optimization. Multiobjective optimization is concerned with the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints. Evolutionary algorithms have been widely applied to solve MOP. Evolutionary algorithm (EA) is a population-based stochastic method and comprises genetic steps such as selection, mutation and recombination [1-2]. Multiobjective evolutionary algorithm (MOEA) works very well in terms of optimization performance [3].

* Corresponding author : Arrchana Muruganantham. Tel.: +65-92996186
E-mail address: arrchana@nus.edu.sg

Selection and peer-review under responsibility of the Program Committee of IES2013
doi:10.1016/j.procs.2013.10.028

Despite the widespread victory of the research on multiobjective optimization, there is a growing need to apply evolutionary algorithm on Dynamic Multiobjective Problems (DMOP). This can be used to optimize objectives in dynamic environment such as movement tracking [4]. The study on dynamic multiobjective optimization is very limited. Benchmark problems, appropriate performance metrics, as well as efficient algorithms are needed to further the research in this field. Some preliminary research includes applying MOEA directly to dynamic problems [5]. However, in a dynamic multiobjective optimization problem, the fitness landscape is changing over time. Due to the inherent characteristics of evolutionary algorithm, MOEA generally takes a significant amount of time to converge to Pareto optimal front (POF). To reduce the number of fitness evaluations, prediction is a promising method as reported in the literature [6]. To direct the search efficiently, historical information should be utilized to predict for future generations.

One simple way to use the historical information is the random re-initialization (RND) model modified from the method proposed in [6]. This method randomly re-initializes part of the whole population, which increases the diversity and the chance of exploration. Since it is assumed that the change in the problem definitions could not be enormous, which is the case in many real world applications, it makes sense for the RND model to retain some of the solutions while exploring in the decision space.

Many other methods are proposed and investigated as well. [7] and [8] survey various existing techniques to solve DMOP. They classify and discuss the uncertainties in general. The main approaches suggested to deal with these uncertainties are: generating diversity after a change, maintaining diversity throughout the run, memory-based approach, and multi-population approach. In particular, Kalman Filter is used to track moving optima in [4]. Some prediction methods have been proposed to solve various problems, which could be transformed into a method that makes good use of the historical information to solve DMOP. Among them, Kalman Filter prediction has successful been applied in many real world problems, such as oil price prediction, movement tracking, among others [9-10]. Kalman Filter operates recursively in time series analysis. The fact that Kalman Filter can run in real time makes it a good candidate for the prediction model in solving DMOP. In our study, Kalman Filter is applied to the whole population to direct the search for Pareto optimal solutions in the decision space, which has not been investigated elsewhere to our best knowledge.

To facilitate understanding of the subject, this paper will then briefly explain MOEA/D- DE (MOEA based on Decomposition with Differential Evolution) and how the Kalman Filter prediction is introduced into this algorithm. Next, a scoring scheme which combines Kalman Filter prediction and random re-initialization is presented. Experiment results on the modified FDA Problem Set [11] and ZJZ Problem [6] are presented. It is shown that the proposed methods outperforms random re-initialization method in terms of optimization performance. We interpret and analyze the results from different perspectives and further possible research direction is discussed.

2. Background

A Multiobjective Optimization Problem (MOP) involves optimizing two or more conflicting objectives subject to certain constraints. In general in a dynamic multiobjective optimization problem (DMOP), the optimum changes with time. Mathematically, a DMOP can be described as

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{f}(\mathbf{x}, t) = [f_1(\mathbf{x}, t)\ f_2(\mathbf{x}, t)\ \dots\ f_m(\mathbf{x}, t)]^T \\ \text{subject to} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where t represents time index, $x \in \mathbb{R}^n$ represents the decision vector, n is the number of decision variables and $\Omega \subset \mathbb{R}^n$ represents the decision space. m is the number of objectives, $\mathbb{R}^m$ is the objective space and $f(\mathbf{x}, t)$ consists of m real-valued objective functions, each of which is continuous with respect to x over Ω.

The Pareto Optimal Front (POF) in the objective space and the Pareto Optimal Set (POS) in the decision space may change over time. The task of a dynamic multiobjective optimization algorithm is to trace the movement of the POF and POS with reasonable computational costs. Thus, a correct guess of the new location of the changed optimum is of great interest.

3. Prediction Model

3.1. Multiobjective Evolutionary Algorithm with Decomposition based on Differential Evolution

The prediction model proposed in this paper is built on the structure of Multiobjective Evolutionary Algorithm with Decomposition based on Differential Evolution (MOEA/D-DE) algorithm [3]. Recently, MOEA/D-DE has received significant attention due to its decent optimization performance in continuous multiobjective optimization problem. It is capable of solving continuous multiobjective optimization problems with relatively fast convergence and diverse spread. As the name suggests, the algorithm decomposes a problem into several sub-problems and simultaneously optimizes these sub-problems using neighborhood relations, which are defined on the distances between their weight vectors. The decomposition is performed using some classical approaches, such as the Tchebycheff approach and the weighted sum approach, among others. In this paper, Tchebycheff approach is used due to its simplicity and decent optimization performance. Detailed description and analysis of MOEA/D-DE algorithm could be found in [3].

3.2. Change Detection Function

The relationship of MOEA/D-DE with the Kalman Filter (KF) prediction model is shown in Figure 1.

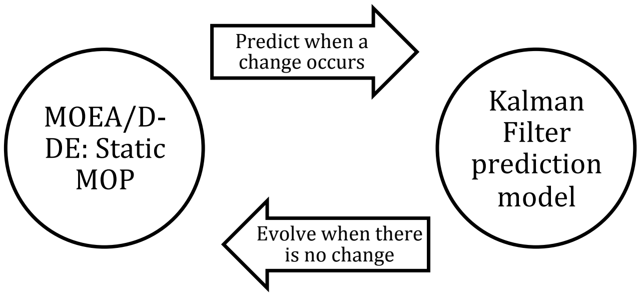


Figure 1: Relationship diagram for Kalman Filter model.

When there is no change detected, MOEA/D-DE takes control and the population will evolve accordingly. Otherwise, the Kalman Filter prediction model will direct the search for Pareto optimal solutions in the decision space.

As expected, a change detection function is needed to combine the prediction model with the MOEA/D-DE algorithm. Assuming that there is no noise in the system, some individuals are randomly selected as detectors and their objective values are stored in the system. At the beginning of each generation, the detectors' objective values are calculated again and compared with the previously stored values. A mismatch in the objective values suggests that a change of the problem has occurred caused by moving POS or POF landscape. Even though either POS or POF of the problem could remain constant over time, the mapping is done differently for the multiobjective problem. Therefore, this detection function should be able to deal with all four types of DMOP problems as introduced in [11].

3.3. Kalman Filter Prediction Model

In 1960, R. E. Kalman published a paper describing a method which can process a time series of measurements and predict unknown variables more precisely than that based on a single measurement alone [12]. This is referred to as the Kalman Filter. Kalman Filter maintains state vectors, which describe the system state, along with its uncertainties. The equations for the Kalman filter fall into two groups, time update and measurement update equations, which are performed recursively for the Kalman Filter to make prediction. Here, the Kalman Filter is used to directly predict for future generations in the decision space and the two major steps are described below:

1. Measurement Update.
   The measurement update equations are responsible for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The individual solutions just before the change occurs are

taken as the actual measurements of the previous predictions. This information is used to update the Kalman Filter prediction model.

2. Time Update.
The time update equations are responsible for projecting forward the current state and error estimate covariance estimates to obtain the a priori estimates for the next step. New solutions are predicted based on the corrected Kalman Filter associated with each individual in the decision space. These are a priori estimates of the future Pareto optimal solutions, which will then be used to update the reference points and subproblems.

The specific equations[15] for the two steps are presented below. (2) gives the equations for the time update step.

$$\begin{aligned} \hat{x}_k^- &= A\hat{x}_{k-1} + Bu_{k-1} \\ P_k^- &= AP_{k-1}A^T + Q \end{aligned} \tag{2}$$

(3) gives the equations for the measurement update step.

$$\begin{aligned} K_k &= P_k^- H^T (HP_k^- H^T + R)^{-1} \\ \hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\ P_k &= (I - K_k H)P_k^- \end{aligned} \tag{3}$$

where x is the state vector to be estimated by the Kalman Filter, A denotes the state transition model, u is the optional control input to the state x, B is the control input matrix, P is the error covariance estimate. z denotes the measurement of the state vector, H is the observation matrix and the process and measurement noise covariance matrices are Q and R respectively. K is the Kalman filter gain.

As shown, the current estimates are made using only the previous predictions and the current observation. There are two variants of Kalman Filters designed for prediction, a two-dimensional Kalman Filter (2by2KF) and a three-dimensional Kalman filter (3by3KF). In both the variants, the observation model is identity matrix, since the decision variables can be directly measured. Further, there are no control inputs in the system. The process and observation noise are Gaussian noise of N(0,σ) and the corresponding covariance Q_k and R_k can be calculated.

(a) 2by2 Kalman Filter (2by2KF).
The state vector is $X = \begin{bmatrix} x \\ v \end{bmatrix}$, where x is the vector for the decision variables and v is the vector of the first order change in the decision variables. The state transition model used is $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$, and the covariance of state vectors is initialized as $P_0 = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. In this case, the Kalman Filter is a first order linear model perturbed by Gaussian noise. The initial covariance P_0 suggests some uncertainty in the initial state vectors, which is adaptive and will be updated as time proceeds. Noise cannot be modelled exactly in this context and assumed to be a constant Gaussian noise.

(b) 3by3 Kalman Filter (3by3KF).
The state vector in this case is $X = \begin{bmatrix} x \\ v \\ a \end{bmatrix}$, where x, v are the same as in 2by2KF, and A is the vector of the second order change in the decision variables. The state transition model used is $A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$, and the covariance of state vectors is initialized as $P_0 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$. In this case, the Kalman Filter is a second order linear model perturbed by Gaussian noise. According to the state transition model, the decision variables are updated by previous x and v only. The second order vector a, is controlled by the Gaussian noise and is only used to estimate the first order

change term of the state vector, as can be deducted from the state transition model. It is designed in this way to include more historical information but not greatly rely on it.

Both the variants are discrete and linear Kalman Filters. Higher order change is ignored in trade-off for speed and computation resources. The various steps of the proposed model to solve DMOP are shown in Figure 2 for clarity.

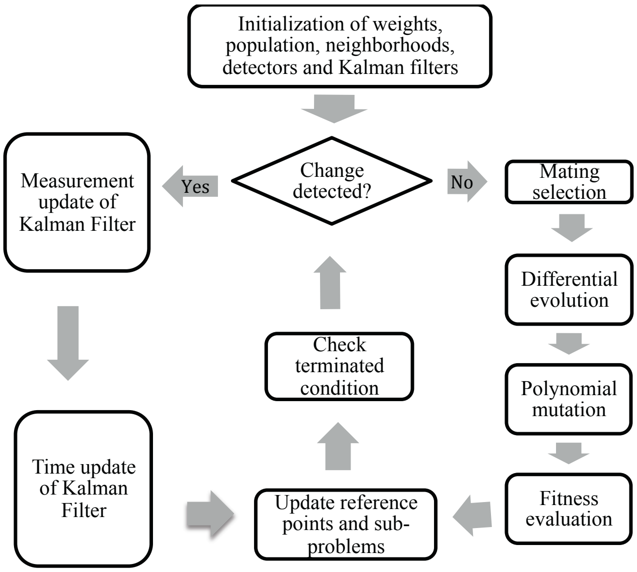


Figure 2: Flow chart of MOEA/D-DE with Kalman Filter prediction model.

3.4. Scoring Scheme

Since Kalman Filter prediction makes the assumption of linear dynamic model of the system, it may cause some problems when the system violates the assumption. To circumvent such a situation, random re-initialization method is introduced into the algorithm. A scoring scheme (SC) is proposed to hybridize Kalman Filter prediction and random re-initialization method. The diagram illustrating the relationship of the scoring scheme with the proposed model is shown in Figure 3.

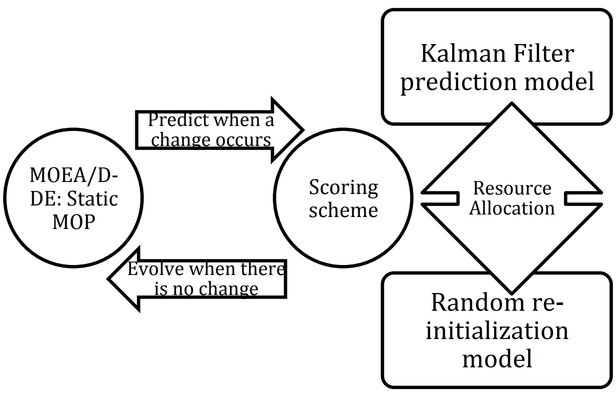


Figure 3: Relationship diagram for scoring scheme.

In order to allocate the resources efficiently, the scoring scheme tries to compute a RND Score, which is the percentage of choosing RND model to produce future generation. A random number then can be generated and if it is smaller than this percentage, random re-initialization is used; otherwise, Kalman Filter prediction is used.

To start with, the chance of producing one solution for the next generation is 50-50 between random re-initialization method and Kalman Filter prediction. The method used to produce each child solution is stored as an attribute of the individual. After a change is detected, the Euclidean distances from the solutions just before the change to the solutions after the previous change are computed and the average is taken. A smaller-than-average distance implies that the improvement made over generations by MOEA/D-DE is not very much. Therefore, the corresponding method in use is likely to produce solutions closer to the true Pareto-optimal set in the current setting. The scores of both methods are recorded and normalized by the total number of solutions produced by each method. The overall score of random re-initialization to Kalman Filter prediction is then calculated by dividing the re-initialization score to the sum of both scores.

This scoring scheme gives a higher chance of using the method that performs better in the previous prediction. It includes other features to be able to respond faster to dynamic changes and to prevent from sticking to any one of the methods fairly well. Note that even if the random re-initialization model is used, the measurement update step of the Kalman Filter model is still performed. This is to keep track of the changes in the system and update the model, in case that the Kalman Filter prediction model is used later for the individual.

4. Experiment Results

4.1. Expriment Settings

The parameter settings for the experiments on FDA problem set and ZJZ problem are tabulated in Table 1.

Table 1: Experiment Settings.

Number of decision variables	FDA1: 10; FDA2: 11; FDA3: 12; FDA4: 12; FDA5: 12; and ZJZ: 10.
Population size	FDA1, FDA2, FDA3, ZJZ: 100; FDA4, FDA5: 105.
Number of generations	600
Neighborhood	Size: 20. Probability that parents are selected from the neighborhood is 0.9.
Differential Evolution	CR = 1.0 and F = 0.5.
Polynomial Mutation	$\eta = 20$, $p_m = 1/n$. The number of solutions replaced by any child solution is at most 2.
Number of detectors	10
Percentage for RND model	20%
KF model	Process noise: Gaussian of N(0, 0.1). Observation noise: Gaussian of N(0, 0.1).
Dynamic Setting	Frequency of change t: 5 or 10. Severity of change n: 5 or 10. Four settings for each test case.

4.2. Performance Metric

The performance metric used in the experiment is the inverted generational distance (IGD) as discussed in [13]. The smaller the IGD value, the better the performance is, in terms of both convergence to the Pareto optimal fronts as well as maintaining the diversity.

Table 2: Statistics of optimization performance on FDA and ZJZ test problems obtained by 30 independent runs.

Dynamic Setting	5n5t	5n10t	10n5t	10n10t	Dynamic Setting	5n5t	5n10t	10n5t	10n10t
FDA1					FDA4				
RND	0.3957186108 ±0.24358602	0.1807282423 ±0.15803272	0.1999779921 ±0.1442224	0.0571861747 ±0.05631281	RND	0.4078904677 ±0.22099426	0.2734695157 ±0.20629929	0.2616666534 ±0.15769446	0.1824732645 ±0.14365608
2by2	0.0999431063 ±0.09446388	0.0399362111 ±0.05659355	**0.0301333438 ±0.05123471**	**0.0193255301 ±0.04862983**	2by2	0.3316914395 ±0.25039969	0.2124250789 ±0.18971535	0.1761358173 ±0.1373597	0.1307702463 ±0.13175923
3by3	**0.0625796984 ±0.06162243**	**0.0328063934 ±0.0527785**	0.030475321 ±0.05064817	0.0198392531 ±0.04917729	3by3	0.2571115573 ±0.2331025	**0.1810409051 ±0.16950467**	**0.1565837004 ±0.11821896**	**0.1252597058 ±0.1239513**
2by2SC	0.1146944239 ±0.10564765	0.0461133773 ±0.06194388	0.0354694138 ±0.05106015	0.0207978562 ±0.0436107	2by2SC	0.3149996619 ±0.19475594	0.2074875202 ±0.16628707	0.1873222215 ±0.14278954	0.1352904697 ±0.1327661
3by3SC	0.0667187002 ±0.06972619	0.0358470328 ±0.05838962	0.0320286731 ±0.04974936	0.0205318364 ±0.04290637	3by3SC	**0.2546430468 ±0.1641074**	0.181121705 ±0.15004972	0.1613553388 ±0.12669278	0.1268281636 ±0.1254651
FDA2					FDA5				
RND	0.0845503711 ±0.07292062	0.0852087983 ±0.07563083	0.0156723533 ±0.04430382	**0.0123144999 ±0.04260599**	RND	0.9068696141 ±0.34555204	0.6489636985 ±0.22163548	0.5914477759 ±0.20389524	0.3995333719 ±0.15106912
2by2	0.0756756401 ±0.06643912	0.085860257 ±0.07558412	0.0308662655 ±0.05108116	0.0191989058 ±0.05014009	2by2	0.6022824513 ±0.2583613	0.5081063808 ±0.21540758	**0.2894995275 ±0.13455761**	**0.2242221287 ±0.11998053**
3by3	0.0715416768 ±0.05912759	0.0843604598 ±0.07311961	0.0417280891 ±0.04861258	0.0213351345 ±0.52361669	3by3	**0.5428163885 ±0.2030498**	0.4879245442 ±0.19666386	0.3076475022 ±0.12255066	0.2427485285 ±0.68721749
2by2SC	**0.0200147569 ±0.04479928**	**0.0146910911 ±0.04268311**	0.0168749022 ±0.04606006	0.0125063568 ±0.04280279	2by2SC	0.6398313004 ±0.24638747	**0.4370987242 ±0.20654777**	0.3403698653 ±0.16358903	0.2603123547 ±0.13279212
3by3SC	0.0828749312 ±0.0732742	0.0846386325 ±0.07578067	**0.015669241 ±0.04326286**	0.0126213937 ±0.04245717	3by3SC	0.6274832716 ±0.2339522	0.5233069827 ±0.19465615	0.3121563032 ±0.14331677	0.2473197122 ±0.13221083
FDA3					ZJZ				
RND	1.0312009321 ±0.51856466	0.6281072211 ±0.24840448	0.5831695672 ±0.18467301	0.3166717897 ±0.10506932	RND	0.4581084025 ±0.3290063	0.1688643781 ±0.19969031	0.1397849854 ±0.1663089	0.0722648536 ±0.14938632
2by2	1.0570728837 ±0.69917088	0.6302308008 ±0.35369504	0.4634431697 ±0.20234969	0.2866251422 ±0.12093344	2by2	0.1631257846 ±0.17961146	0.118332199 ±0.16457478	0.0837517522 ±0.17647397	0.0478670765 ±0.16134715
3by3	0.8149117669 ±0.47410913	0.5453656682 ±0.2604122	0.468618152 ±0.18039348	0.2941936973 ±0.63290749	3by3	**0.1305484971 ±0.16858983**	0.1164116216 ±0.16477289	0.0881774482 ±0.17446476	0.0497774083 ±0.56346948
2by2SC	0.8290392085 ±0.31099362	0.4719401829 ±0.17673935	0.4154382249 ±0.14803308	0.2708415548 ±0.09216788	2by2SC	0.1524415883 ±0.18015592	**0.0721737202 ±0.15954314**	0.0639722706 ±0.15445636	0.0420147027 ±0.14725622
3by3SC	**0.7523403127 ±0.32612085**	**0.4654581407 ±0.18627094**	**0.3926398823 ±0.14811428**	**0.25962894 ±0.09191797**	3by3SC	0.1322288545 ±0.16376607	0.1098072568 ±0.15998015	**0.0621972105 ±0.14932354**	**0.0416952597 ±0.14356927**

4.3. Experiment Results

For each dynamic setting (described in section 5.1) of every test problem, the algorithm is run for 30 independent times. The inverted generational distances are averaged among the 600 generations and then the 30 runs. RND denotes the method wherein a percentage of the population is randomly reinitialized. 2by2 and 3by3 are the two variants of the Kalman Filter prediction model proposed in this paper. The Kalman Filter variants (2by2 and 3by3) when hybridized with random reinitialization using the Scoring Scheme (SC) lead to 2by2SC and 3by3SC. The final result of the average values and standard deviations of IGD is shown in Table 2, with the best performance highlighted in bold.

5. Analysis and Discussion

To have a better understanding of the experiment results, we perform the following analysis on the results from different perspectives.

5.1. Dynamic Settings

The dynamic settings of the problems have an impact on the performance. The severity of change, n denotes the number of steps to complete one period. The frequency of change, t denotes the number of generations before a change occurs. Therefore, a smaller value of n means larger change, whereas a smaller value of t means more frequent occurrence of changes. n and t can take a value of 5 or 10. The four possible combinations, leads to the dynamic settings of 5n5t, 5n10t, 10n5t and 10n10t. Looking through one row of Table 2, it is observed that the smaller the change and the less frequent the change occurs, the better the performance is. This meets our expectation as the population are nearer to the true solution set and has more time to evolve in this case.

5.2. Evolution over Generations

To show clearly how the IGD values change over the generations for different problems, the case of n = 5, t = 10 is carefully examined. For each generation, the IGD values are averaged among the 30 independent runs. The evolutions of IGD values are shown in Figure 4 for comparison.

5.3. Further Discussion

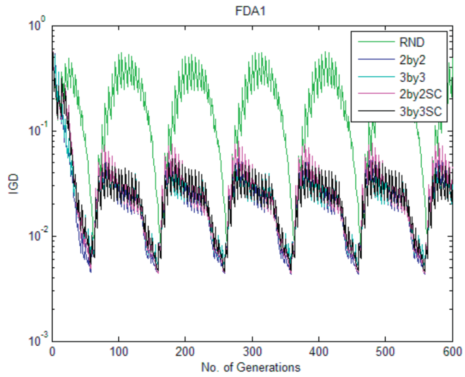
The performance of the scoring scheme proposed is analyzed. During the research, other simple combination methods are also implemented and tested. One way is to use Kalman Filter to predict for half of the whole population. The other is to fix the RND score as 0.5 for every generation. They are tested and the results are compared with that for the scoring scheme. It is shown that these simple combination methods are able to introduce diversity and flexibility to the Kalman Filter model and improve the performance. The performances achieved are comparable to the scoring scheme on most of the test cases. This implies that better scoring scheme could be devised to allocate the resources faster and more accurately.

6. Conclusion

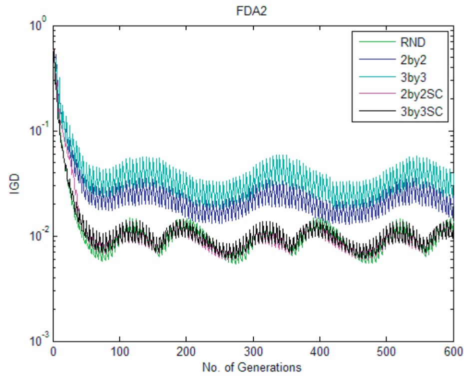
This paper proposes to solve dynamic multiobjective optimization problem by using the Kalman Filter prediction model with MOEA/D-DE and the scoring scheme. The experiment results show that the proposed methods are able to improve the performance significantly as compared to the random re-initialization method. Detailed analysis and discussion are performed from different perspectives. The study suggests that Kalman Filter prediction is a very promising tool for dynamic multiobjective optimization problems.

Some future research directions are pointed out by the study, which include building more complex Kalman Filter and finding more efficient resource allocation methods to reduce the response time. In addition, a forward-looking approach [14] could be introduced into the models proposed. The Kalman Filter prediction model and the scoring scheme could very likely respond to the change better, if a change could be anticipated beforehand.

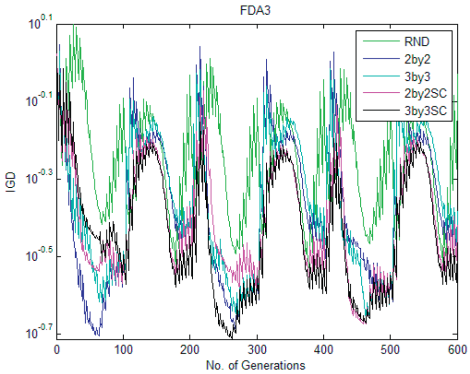
Dynamic multiobjective optimization problems are difficult to solve and this field has not been extensively studied so far. Dynamic multiobjective optimization evolutionary algorithms should have more efficient ways to direct the search for Pareto optimal solutions when changes are detected. Other prediction methods using the historical information could be introduced into MOEAs.



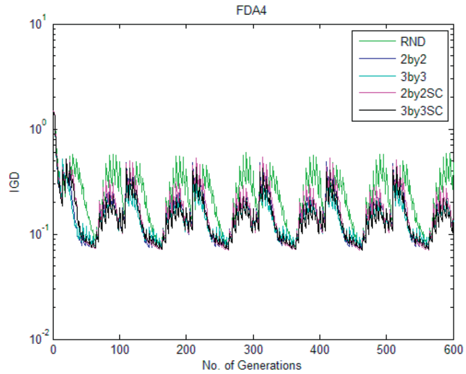
(a) FDA1



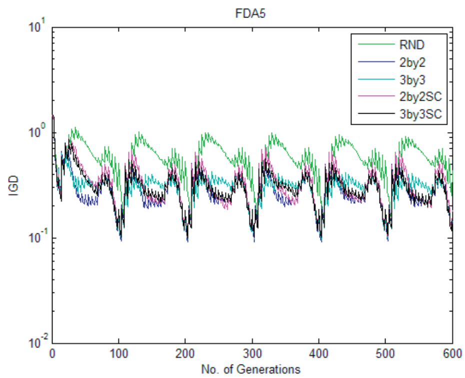
(b) FDA2



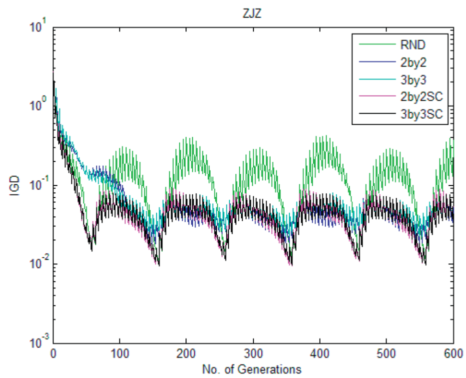
(c) FDA3



(d) FDA4



(e) FDA5



(f) ZJZ

Figure 4: Evolution of IGD over generations.

References

1. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. *A Fast and Elitist Multiobjective Genetic Al- gorithm: NSGA-II*. In: *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, April 2002.
2. Q. Zhang and H. Li. *MOEA/D: A multi-objective evolutionary algorithm based on decom- position*. In: *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712 - 731, December 2007.

3. H. Li and Q. Zhang. *Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II*. In: *IEEE Transactions on Evolutionary Computation*, Vol. 13, No. 2, April 2009.
4. C. Rossi, M. Abderrahim, and J. C. Daz. *Tracking moving optima using kalman-based predictions*. In: *Evolutionary Computation*, Vol. 16, No. 1, pp.1-30, Spring 2008.
5. J. Mehnen, T. Wagner, and G. Rudolph. *Evolutionary Optimization of Dynamic Multi-objective Test Functions*. In: *Technical Report CI 204/06*, University of Dortmund, 2006.
6. A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. *Prediction-Based Population Re- initialization for Evolutionary Dynamic Multi-objective Optimization*. In: *EMO07 Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization*, pages 832 - 846, Springer-Verlag Berlin, Heidelberg, 2007
7. J. Branke. *Evolutionary approaches to dynamic optimization problems*. In: J. Branke, and T. Baeck (Eds.), *Evolutionary Algorithms for Dynamic Optimization Problems*, pp. 2730, 2001.
8. Y. Jin and J. Branke. *Evolutionary optimization in uncertain environments a survey*. In: *IEEE Transactions on Evolutionary Computation*, 9(3):303317, 2005.
9. E. Schwartz and J. E. Smith. *Short-Term Variations and Long-Term Dynamics in Commodity Prices*. In: *Management Science*, vol. 46, no. 7, pp. 893-911. July 2000.
10. A. Kiruluta, M. Eizenman, and S. Pasupathy. *Predictive Head Movement Tracking Using a Kalman Filter*. In: *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, Vol. 27, No. 2, April 1997.
11. M. Farina, K. Deb, and P. Amato. *Dynamic Multiobjective Optimization Problems: Test Cases, Approximations, and Applications*. In: *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 5, October 2004.
12. R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*. In: *Transactions of the ASME Journal of Basic Engineering, 82 (Series D)*, pp. 35 45, 1960.
13. E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. *Performance Assessment of Multiobjective Optimizers: An Analysis and Review*. In: *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 2, pp. 117 132, April 2003.
14. I. Hatzakis and D. Wallace. *Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach*. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 1201- 1208, 2006.
15. K. Deb, N. Rao, and S. Karthik. *Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-thermal Power Scheduling*. In: *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, Vol. 4403, pp. 803-817, 2007.