Solving the IEEE CEC 2015 Dynamic Benchmark Problems Using Kalman Filter Based Dynamic Multiobjective Evolutionary Algorithm

Arrchana Muruganantham, Kay Chen Tan and Prahlad Vadakkepat

Abstract Evolutionary algorithms have been extensively used to solve static and dynamic single objective optimization problems, and static multiobjective optimization problems. However, there has only been tepid interest to solve multiobjective optimization problems in dynamic environments. It is only in the past few years that evolutionary algorithms have been used to solve dynamic multiobjective optimization problems and comprehensive benchmark suites have been proposed for testing the performance of algorithms. Prediction based algorithms may be able to provide information about the location of the changed optima and thereby assisting the evolutionary algorithm in the non-trivial task of tracking the changing Pareto Optimal Front or Set. Kalman filter is one of the widely used techniques in prediction scenarios for state estimation. A Dynamic Multi-objective Evolutionary algorithm was proposed in which the Kalman Filter was applied to the whole population to direct the search for Pareto Optimal Solutions in the decision space after a change in the problem has occurred. In this work, the Kalman Filter assisted Evolutionary Algorithm is tested on the IEEE CEC 2015 Benchmark problems set and the results are presented. It is observed that while the proposed algorithm performs well on some problems, more efficient strategies are required to supplement the algorithm in cases of high change severity, isolated and deceptive fronts.

Keywords Dynamic multiobective optimization · Kalman filtered · Evolutionary algorithm

---

A. Muruganantham(✉) · K.C. Tan · P. Vadakkepat
Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117576, Singapore
e-mail: arrchana@u.nus.edu

K. Lavangnananda et al. (eds.), *Intelligent and Evolutionary Systems*,
Proceedings in Adaptation, Learning and Optimization 5,
DOI: 10.1007/978-3-319-27000-5_20

1 Introduction

Optimization problems are aplenty and are found in various fields such as science, engineering, economics, finance, management, scheduling, planning, design, control, etc. The list is ever growing, and scientists and industrialists alike are in the lookout for better and more efficient techniques to solve their problems. Optimization in general refers to the process of finding one or more feasible solutions which correspond to extreme values of one or more objectives. Many researchers have tend to focus on optimization problems which consider a single objective, although most real-world search and optimization problems involve more than one objective. Further, the presence of conflict in the multiple objectives makes these optimization problems (commonly termed as multiobjective optimization problems) more interesting and challenging to solve. Since no single solution can satisfy the multiple conflicting objectives simultaneously, the solution to a multiobjective optimization problem is a set of trade-off optimal solutions. Classical optimization methods such as hill climbing, simulated annealing can at best find one solution in a simulation run, thereby deeming these methods inefficient to solve multiobjective optimization problems.

Evolutionary algorithms are inspired from biological evolution and mimic nature's evolutionary principles to drive the search towards optimal solution(s). These algorithms use a population of solutions in each iteration, consequently making them ideal candidates for solving multiobjective optimization problems. Numerous Evolutionary Algorithms(EAs) have been developed in the past few decades to solve multiobjective optimization problems such as NSGA-II [1], MOEA/D [2], MOEA/D-DE [3], to name a few. The advances of Evolutionary Multiobjective Optimization(EMO) research has been drastic and has resulted in many new paradigms to be developed such as the Estimation of Distribution Algorithms(EDAs), decomposition based algorithms, and so on. Applications of EMO research have been observed in a wide variety of problems [4–9]. However, there has only been lukewarm interest in applying Evolutionary Algorithms to solve dynamic optimization problems, where the optimum(or optima) changes with time. Furthermore, most of the EA researchers in this area have tend to focus on dynamic single-objective optimization problems, while most real-world problems are dynamic multiobjective optimization problems.

Using Evolutionary Algorithms to solve dynamic multiobjective optimization problems has started gaining attention over the past few years. Nevertheless, there is large scope for contribution and improvement in this field. In dynamic multiobjective optimization problems the fitness landscape is changing over time. Preliminary research in solving proposed benchmark problems involved applying Multiobjective Evolutionary Algorithm(MOEA) directly to solve them. However, the inherent characteristic of an MOEA is that it takes significant amount of time to converge to the Pareto Optimal Front(POF). This is an important issue in dynamic multiobjective optimization where the POF and/or the Pareto Optimal Solution(s) (POS) are continuously changing with time. In the current literature, various approaches have been proposed to solve dynamic multiobjective optimization problems. In this paper, the focus is on employing prediction techniques to solve dynamic multiobjective optimization problems. A novel Kalman Filter based dynamic multiobjective optimization algorithm was developed to solve dynamic multiobjective optimization problems [10].

Based on the IEEE-CEC 2015 Dynamic Multi-objective Optimization Benchmark problems, this paper aims to examine and discuss the performance of the Kalman Filter assisted MOEA/D-DE algorithm, MOEA/D-KF in solving the proposed benchmark set. The outline of the paper is as follows: Section 2 provides required background and outlines related work. Section 3 provides the algorithm description including a brief overview of MOEA/D-DE and Kalman Filter prediction method. Section 4 provides the experimental setup, outlines the performance metric used and the results are presented. Section 5 consists of anlaysis of the performance based on the severity and frequency of change in the problems. Section 6 outlines the discussion of the results and Section 7 concludes the work.

2 Background

This section provides the basic definitions used in the evolutionary multiobjective community together with some key concepts which are essential for understanding the work described in a more scientific manner.

2.1 Multiobjective Optimization Problem

A multiobjective problem can be expressed in its general form mathematically as

$$\begin{aligned} \text{Minimize/Maximize } & f_m(x), & m &= 1, 2, \ldots, M; \\ \text{subject to } & g_j(x) \geq 0, & j &= 1, 2, \ldots, J; \\ & h_k(x) = 0, & k &= 1, 2, \ldots, K; \\ & x_i^L \leq x_i \leq x_i^U, & i &= 1, 2, \ldots, n. \end{aligned}$$

where f_i is the i-th objective function and M is the number of objectives.

The vector, $f(x) = [f_1(x)\ f_2(x)\ \ldots\ f_m(x)]^T$ forms the objective vector, $f(x) \in \mathbb{R}^M$. A solution x is a vector of n decision variables: $x = [x_1\ x_2\ \ldots\ x_n]^T$. The above general problem is associated with J inequality and K equality constraints. The last set of constraints are called *variable bounds*, restricting each decision variable x_i to take a value within a lower $x_i^{(L)}$ and an upper $x_i^{(U)}$ bound. These variable bounds constitute the *decision variable space* $\Omega \in \mathbb{R}^n$, or simply the decision space.

In the presence of constraints g_j and h_k, the entire decision variable space Ω may not be feasible. The feasible region S is the set of all feasible solutions in the context of optimization. The feasible search space can be divided into 2 sets of solutions - pareto optimal and non pareto otpimal set. To define pareto optimality, first we need to look into the concept of domination.

Concept of Domination. There are M objective functions in a multiobjective problem. Say, we have 2 solutions, i and j. $i < j$ implies i is better than j *or* i dominates j. A solution x^1 is said to dominate another solution x^2, if both the following conditions are true.

1. The solution x^1 is no worse than x^2 in all objectives, *or* $f_m(x^1)$ is not better than $f_m(x^2)$ for all $m = 1, 2, ..., M$.
2. The solution x^1 is strictly better than x^2 in at least one objective.

Pareto Optimality. Among a set of solutions P, the non-dominated solutions, P^* are those that are not dominated by any member of the set P. When the set P comprises the entire search space, the resulting non-dominated set P^* is the *Pareto Optimal Set*(POS in the decision space). Pareto optimal solutions joined together as a curve form the *Pareto Optimal Front*(POF in the objective space). The front lies in the bottom-left corner of the search space for problems where all objectives are to be minimized.

Goals of an MOEA. The working principle for an ideal multiobjective procedure consists of finding multiple trade-off optimal solutions with a wide range of values for the objectives, and later choosing one of the obtained solutions using higher level information. In such a case it is difficult to prefer one solution over the other without any further information about the problem. If higher level information is satisfactorily available, this can be used to make a biased search. However, in the absence of any such information, all pareto optimal solutions are equally important. Therefore, there are 2 goals:

1. To find a set of solutions as close as possible to the POF, i.e. *Convergence*
2. To find a set of solutions as diverse as possible, i.e. *Diversity*

For each of the M conflicting objectives, there exists one different optimal solution. An objective vector constructed with these individual optimal objective values constitutes the ideal objective vector, z^*, which in general lies in the infeasible space. For more detailed discussion of the concepts on multiobjective optimization , please refer to [11].

2.2 Dynamic Multiobjective Optimization Problem

The various concepts discussed for multiobjective optimization are still essential in dynamic multiobjective optimization together with some additional issues and goal(s). In general, in a dynamic multiobjective optimization problem(DMOOP), the optimum changes with time. Mathematically, a DMOOP can be described as

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

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. Thus, the *POF* and/or *POS* may change over time.

The goals of *convergence* and *diversity* apply to Dynamic Multiobjective Optimization Evolutionary Algorithms (DMOEAs) as well. However, it is not restricted to the above two and there is an additional goal of tracking the changing POF/POS which plays an important role in determining the overall performance.

Classification of DMOOPs. [12] have classified dynamic multiobjective optimization problems based on the possible ways a problem can demonstrate a time varying change.

Table 1 Classification of DMOOPs

Type I	POS changes, but POF does not change
Type II	Both POS and POF change
Type III	POS does not change, POF changes
Type IV	Both POS and POF do not change, although the problem can change

These four cases are summarized in the Table 1. There are other possible ways of classifying DMOOPs as well such as based on severity, predictability and visibility of change, among others [13].

3 Algorithm Description

3.1 MOEA/D-DE

The DMOEA used in this paper is built on the basis of Multiobjective Evolutionary Algorithm with Decomposition based on Differential Evolution (MOEA/D-DE) [3]. MOEA/D-DE decomposes a problem into several sub-problems and simultaneously optimizes them to find the pareto optimal solutions of the Multiobjective optimization problem. Each solution is assigned with a weight vector and neighbourhood relations are defined based on the weight vectors. In the context of dynamic multiobjective optimization, the usage of weight vectors enables the tracking of individual solutions in the decision space which are essential for prediction purposes. Decomposition into sub-problems is performed using the Tchebycheff approach in this paper.

3.2 Change Detection

Sentry particles in the population are used to observe any changes in the system, assuming there is no noise. These change detecter individuals' objective values are recomputed at the beginning of each generation to check if there has been any change since the last objective function evaluation. If there is a change in the objective function values, it is assumed that a change has occured and the Kalman filter based model is used to predict for the optimal values of solutions in the decision space. Otherwise, the optimization process proceeds as in a static MOEA.

3.3 Kalman Filter Prediction Based DMOEA

Kalman Filter is an algorithm that uses a series of measurements observed over time, containing noise and other inaccuracies, and produces statistically optimal estimates of the underlying system state [14, 15]. The algorithm works in a two-step process involving a prediction step and a measurement step. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement is observed, these *a priori* estimates are updated to obtain the *a posteriori* estimates. The 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 DMOOP. Thus, in our study, Kalman filter is applied to the whole population to direct the search for Pareto Optimal Solutions (POS) in the decision space after a change in the problem has occured. Please refer to [10] for more details on the Kalman Filter assisted DMOEA. The 2by2 variant of the Kalman filter based algorithm from [10] is used in this paper.

4 Empirical Study

4.1 Experimental Setup

The Kalman Filter prediction based DMOEA, MOEA/D-KF is tested on the benchmark problems proposed for the IEEE CEC 2015 Competition on Dynamic Multiobjective optimization [16] in this paper. The benchmark set consists of functions from FDA [12], dMOP [17] and HE [18–20] benchmark function suites and were adapted to further test the capabilities of DMOEAs in a more comprehensible manner than currently available in the evolutionary dynamic multiobjective optimization literature. The parameter settings for the experiments are tabulated in Table 2.

4.2 Performance Metric - Modified Inverted Generational Distance

A number of performance metrics are in use for evaluation of static MOEAs which evaluate convergence and diversity quite effectively. These metrics have been modified for usage in evaluation of DMOEAs. The Inverted Generational Distance(IGD) is a unary performance indicator which provides a quantitative measurement for the proximity and diversity goal of multiobjective optimization [21]. It is mathematically given by

Table 2 Experiment Settings

Number of decision variables, n	FDA:12, dMOP and HE:10, HE2:30
Population size	100 for 2 objective problems 200 for 3 objective problems.
Neighborhood	Size: 20. Probability that parents are selected from the neighborhood is 0.9.
Decomposition method	Tchebycheff
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, 0.04). Observation noise: Gaussian of N(0, 0.01).
Number of changes	20

$$IGD(P^{t*}, P^t) = \frac{\sum_{v \in P^{t*}} d(v, P^t)}{|P^{t*}|} \tag{2}$$

where P^{t*} is a set of uniformly distributed Pareto optimal solutions in the POF at time $t(POF^t)$ and P^t is an approximation of the POF obtained by the algorithm in consideration. d is a distance measure between P^t and P^{t*}, given by

$$d(v, P^t) = \min_{u \in P^t} \|F(v) - F(u)\|. \tag{3}$$

A lower value of IGD implies that the algorithm has better optimization performance. To obtain a low value of IGD, it can be seen from the above 2 equations that, P^t must be very close to POF^t and cannot miss any part of POF^t, thus measuring both convergence and diversity.

To adapt the IGD metric for dynamic multiobjective optimization, the average of the IGD values in some time steps over a run is taken as the performance metric, given by

$$MIGD = \frac{1}{|T|} \sum_{t \in T} IGD(P^{t*}, P^t) \tag{4}$$

where T is a set of discrete time points(immediately before the change occurs) in a run and $|T|$ is the cardinality of T. A lower value of the MIGD metric described above would also assist in evaluating the tracking ability, as the approximated pareto front obtained from the algorithm with the changing pareto optimal front.

4.3 Results

The MOEA/D-KF algorithm is compared with a baseline of random immigrants strategy where a percentage of the population is randomly reinitialized when a change

Table 3 n_t and τ_t values for the benchmark functions

n_t	10	10	10	10	1	1	20	20
τ_t	5	10	25	50	10	50	10	50
τ_t	100	200	500	1000	200	1000	200	1000

occurs and this method is indicated by RND. The difficulty of a DMOOP is determined by the parameters n_t and τ_t which denote the severity and frequency of change respectively. The combination of parameter values used in the simulations are given in Table 3. τ_T denotes the maximum number of iterations.

Table 4 MIGD mean and standard deviation statistics for $n_t = 10$

Problem	RND	MOEA/D-KF
FDA4	0.257978 ± 0.202(+)	**0.207295 ± 0.211**
FDA5	0.538704 ± 0.222(-)	**0.383436 ± 0.193**
dMOP1	**0.173554 ± 0.461**	0.248594 ± 0.452(+)
dMOP2	0.882946 ± 0.682(+)	**0.303540 ± 0.449**
dMOP2iso	**0.009556 ± 0.020**	0.009895 ± 0.020(-)
dMOP2dec	2.739828 ± 6.218(-)	**2.671837 ± 6.109**
HE2	**0.057582 ± 0.001**	0.057710 ± 0.001(-)
HE7	**0.223214 ± 0.030**	0.333162 ± 0.073(+)
HE9	**0.403658 ± 0.035**	0.499466 ± 0.068(+)

(a) $n_t = 10, \tau_t = 5$

Problem	RND	MOEA/D-KF
FDA4	0.144859 ± 0.090(+)	**0.122009 ± 0.090**
FDA5	0.347302 ± 0.111(+)	**0.227943 ± 0.094**
dMOP1	**0.037534 ± 0.103**	0.047511 ± 0.101(+)
dMOP2	0.173277 ± 0.119(+)	**0.078387 ± 0.107**
dMOP2iso	**0.004369 ± 0.002**	0.004411 ± 0.002(-)
dMOP2dec	0.901509 ± 2.811(-)	**0.859458 ± 2.824**
HE2	0.057046 ± 0.001(-)	**0.057030 ± 0.001**
HE7	**0.168585 ± 0.018**	0.253887 ± 0.053(+)
HE9	**0.377706 ± 0.039**	0.478676 ± 0.085(+)

(b) $n_t = 10, \tau_t = 10$

Problem	RND	MOEA/D-KF
FDA4	0.085218 ± 0.016(+)	**0.077284 ± 0.014**
FDA5	0.226148 ± 0.083(+)	**0.140106 ± 0.029**
dMOP1	**0.007402 ± 0.013**	0.008307 ± 0.012(+)
dMOP2	0.022815 ± 0.013(+)	**0.013347 ± 0.012**
dMOP2iso	**0.003743 ± 0.000**	0.003747 ± 0.000(-)
dMOP2dec	**0.099968 ± 0.335**	0.106301 ± 0.337(-)
HE2	0.056916 ± 0.001(-)	**0.056910 ± 0.001**
HE7	**0.153983 ± 0.030**	0.206505 ± 0.054(+)
HE9	**0.353986 ± 0.042**	0.445529 ± 0.084(+)

(c) $n_t = 10, \tau_t = 25$

Problem	RND	MOEA/D-KF
FDA4	0.072128 ± 0.003(+)	**0.069282 ± 0.001**
FDA5	0.197262 ± 0.088(+)	**0.130583 ± 0.038**
dMOP1	**0.004237 ± 0.001**	0.004411 ± 0.001(+)
dMOP2	0.006455 ± 0.001(+)	**0.005675 ± 0.001**
dMOP2iso	**0.003730 ± 0.000**	0.003730 ± 0.000(-)
dMOP2dec	**0.029316 ± 0.099**	0.031941 ± 0.099(-)
HE2	0.056915 ± 0.001(+)	**0.056909 ± 0.001**
HE7	**0.147688 ± 0.034**	0.185881 ± 0.053(+)
HE9	**0.330218 ± 0.032**	0.420749 ± 0.083(+)

(d) $n_t = 10, \tau_t = 50$

Tables 4 and 5 provide the MIGD mean and standard deviation statistics for the different combination of parameter values for the various benchmark problems. Statistical t-test was conducted on the results at the 5% significance level and the best value is denoted in bold. (+) (and (−)) indicates that the difference between the marked entry and the best entry is statistically significant (and insignificant, respectively).

FDA4 and FDA5 are 3-objective problems, while the rest of the problems are 2-objective problems. It can be observed from Tables III-V that MOEA/D-KF performs significantly better than RND on FDA4, FDA5 and dMOP2 in all three parameter

Table 5 MIGD mean and standard deviation statistics for $n_t = 1$

Problem	RND	MOEA/D-KF
FDA4	**0.390467 ± 0.089**	0.602495 ± 0.128(+)
FDA5	**1.058697 ± 0.455**	1.193283 ± 0.450(+)
dMOP1	0.112298 ± 0.111(-)	**0.110023 ± 0.110**
dMOP2cec	**7.429464 ± 7.310**	17.893780 ± 15.640(+)
dMOP2iso	**0.084172 ± 0.073**	0.084333 ± 0.073(-)
dMOP2dec	**30.214032 ± 39.352**	31.296647 ± 39.384(-)
HE2	**0.109764 ± 0.061**	0.110784 ± 0.060(+)
HE7	**0.191397 ± 0.026**	0.237358 ± 0.046(+)
HE9	**0.317087 ± 0.110**	0.367213 ± 0.132(+)

(a) $n_t = 1, \tau_t = 10$

Problem	RND	MOEA/D-KF
FDA4	**0.068035 ± 0.001**	0.069631 ± 0.001(+)
FDA5	**0.825898 ± 0.328**	0.902508 ± 0.358(-)
dMOP1	0.083306 ± 0.073(+)	**0.082921 ± 0.072**
dMOP2cec	0.082403 ± 0.069(+)	**0.076870 ± 0.061**
dMOP2iso	**0.083391 ± 0.073**	0.083401 ± 0.073(-)
dMOP2dec	**20.317731 ± 34.716**	20.346623 ± 34.710(-)
HE2	0.107948 ± 0.060(-)	**0.107937 ± 0.060**
HE7	**0.187327 ± 0.055**	0.202417 ± 0.047(-)
HE9	**0.276737 ± 0.086**	0.349261 ± 0.123(+)

(b) $n_t = 1, \tau_t = 50$

settings. dMOP2 is a type II DMOOP and its time-varying POS is sinusoidal in nature. The Kalman filter prediction can track the changing POS better than the random immigrants strategy in this case. A similar explation for FDA5 applies as well. Though FDA4 is a type I DMOOP, wherein its POF does not change with time, its POS also follows a sinusoidal trajectory and MOEA/D-KF's better performance could be attributed to the more efficient POS tracking.

For dMOP1, the optimal values for all decision variables remain the same throughout the iteration. RND performs better than MOEA/D-KF on this problem, as a majority of RND's population is retained without any modification after a change. Once the EA converges on the POS, the RND method does not effectively disrupt the POF/POS attained.

HE7 and HE9 are type III DMOOPs and their POS is not dependent on time. This might result in the better performance of RND compared to the Kalman filter predictions. dMOP2iso and dMOP2dec consist of isolated and deceptive POF respectively. Even though MOEA/D-KF performs better than RND on dMOP2, it does not perform significantly better than RND on the isolated and deceptive POF variants of dMOP2. This maybe a result of trapping into local optima for both the algorithms.

HE2 is a type III DMOOP, similar to HE7 and HE9. However, it has discontinuous POF, with various disconnected continuous sub-regions [16]. This increases the problem complexity significantly and may lead to similar performance on RND and MOEA/D-KF as specific measures have not been taken to handle such scenarios in the Kalman Filter prediction based DMOEA. Both the algorithms seem to give similar performance on all three parameter settings for HE2.

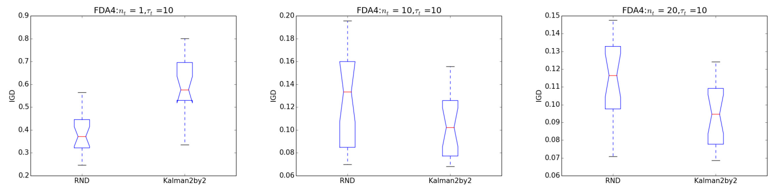
More efficient strategies are required to enhance the performance of MOEA/D-KF on the IEEE CEC 2015 Dynamic benchmark suite, especially in the problems with isolated, deceptive and disconnected POF. Further, the state transition of the model can be modified such that it is able to better model the movement of decision variables to obtain efficient tracking performance.

5 Analysis

5.1 *Effect of Severity of Change*

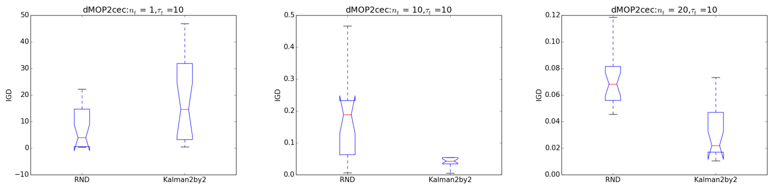
The parameter n_t, denotes the severity of change in the DMOOP. The parameter settings for Table III-(b) and Table IV-(a) are $\tau_t = 10$ and $n_t = 10$, and 1 respectively. It can be seen from the MIGD values on the table that as n_t decreases, the severity of change in the problem increases manyfold. This gets reflected in the MIGD values obtained as high numbers as can be seen in Table IV.

Figures 1 and 2 depict the influence of severity of change on FDA4 and dMOP2 problems. The IGD box plots for RND and MOEA/D-KF are plotted for different values of severity of change. From the range of IGD in the box plots, it can be seen that in the higher n_t setting both the algorithms perform better than in the lower setting. It is also interesting to note that MOEA/D-KF performs better than RND for $n_t = 10, 20$, while it performs worser for the lowest n_t setting.



(a) $n_t = 1$ (b) $n_t = 10$ (c) $n_t = 20$

Fig. 1 Effect of severity of change: IGD box plots for FDA4 with $\tau_t = 10$ and $n_t = 1, 10$ and 20



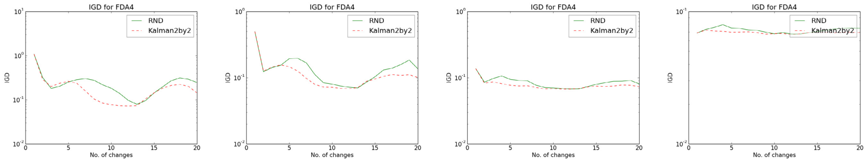
(a) $n_t = 1$ (b) $n_t = 10$ (c) $n_t = 20$

Fig. 2 Effect of severity of change: IGD box plots for dMOP2 with $\tau_t = 10$ and $n_t = 1, 10$ and 20

5.2 *Effect of Frequency of Change*

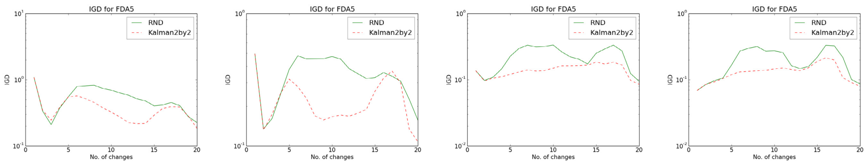
The parameter τ_t, denotes the frequency of change in the DMOOP. It determines the number of generations for which the problem does not change. The smaller the frequency of change the problem changes quickly. Thus, as the frequency of change increases with other parameters kept constant, the difficutly of the DMOOP would be expected to decrease.

Figures 3 and 4 depict the trend of IGD with number of changes in the DMOOP for FDA4, FDA5 and dMOP2 problems. The trends are plotted for a number of values of τ_t : 5, 10, 25 and 50. It can be observed from the figures that the MOEA/D-KF algorithm performs better than RND for the various values of τ_t. However, the problem difficulty reduces for higher values of τ_t and therefore, the distance between the trend curves of MOEA/D-KF and RND reduces as τ_t increases.



(a) $\tau_t = 5$ (b) $\tau_t = 10$ (c) $\tau_t = 25$ (d) $\tau_t = 50$

Fig. 3 Plots of IGD vs Number of Changes for FDA4 problem with $n_t = 10$ and various values of τ_t



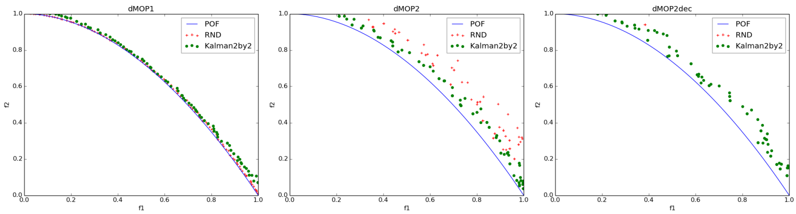
(a) $\tau_t = 5$ (b) $\tau_t = 10$ (c) $\tau_t = 25$ (d) $\tau_t = 50$

Fig. 4 Plots of IGD vs Number of Changes for FDA4 problem with $n_t = 10$ and various values of τ_t

5.3 Initial Populations Obtained Before a Change

Figure 5 shows the initial populations obtained before a change by MOEA/D-KF and RND algorithms for dMOP1, dMOP2 and dMOP2dec problems with $n_t = 10$ and $\tau_t = 10$. All three problems have similar POF. However, their Pareto Optimal Sets are very different from each other. RND is able to obtain solutions close to the POF as the problem's POS does not change with time and the algorithm does not reinitialize a major portion of the population. However, its performance is worser in the dMOP2 and dMOP2dec problems.

In dMOP2, MOEA/D-KF is able to better approximate to the POF than RND. This can be observed in Fig 5(b) as the solutions obtained by MOEA/D-KF are closer to the POF than that of RND. In the case of dMOP2dec, the problem has a deceptive POF. Further, in dMOP2dec, solutions obtained by MOEA/D-KF are within the vicinity of POF whereas only few solutions of RND are visible.



(a) dMOP1 (b) dMOP2 (c) dMOP2dec

Fig. 5 Initial populations obtained before a change by MOEA/D-KF and RND for dMOP1, dMOP2 and dMOP2dec problems with $n_t = 10$ and $\tau_t = 10$

6 Discussion

6.1 Type I DMOOPs

The POF remains static for these problems, while the POS keeps changing with time. In the benchmark problems used in this study, FDA4 falls into this category. MOEA/D-KF seems to perform quite well in these kind of problems in comparison with the RND algorithm.

6.2 Type II DMOOPs

Both the POF and POS change with time in this class of problems. FDA5, dMOP2, dMOP2iso and dMOP2dec come under this categorization. While MOEA/D-KF performs quite well in FDA5 and dMOP2, its performance is much worse in the case of dMOP2iso and dMOP2dec which have isolated POF and deceptive POF respectively. While a generalization cannot be made in this case about the performance of MOEA/D-KF or RND, it is to be noted that in problems with isolated and deceptive POF, DMOEAs may have significant difficulty in tracking the changing POF/POS.

6.3 Type III DMOOPs

The POS remains static in these problems, while the POF keeps changing with time. HE2, HE7 and HE9 problems come under this category. The discontinuous POF characteristic of HE2 leads to similar performance on RND and MOEA/D-KF while RND tends to be perform marginally better than MOEA/D-KF in the other problems as it does not disrupt the converged solutions in the decision space and thereby maintaining the solutions to be close to the POS.

7 Conclusion

In this paper, a brief overview of evolutionary algorithms in the optimization context was provided in general and the issues related to evolutionary dynamic multiobjective optimization were discussed in particular. A Kalman Filter prediction based DMOEA was also outlined. Subsequently, the algorithm was tested on the IEEE CEC 2015 Benchmark suite and its results were compared to the random immigrants strategy also based on MOEA/D-DE. While MOEA/D-KF does perform significantly better than RND in some of the problems, more effective strategies need to be observed to effectively solve them. The effect of severity of change and frequency of change was also analysed. The initial populations obtained by the two algorithms were also visualized to get a better perspective about their optimization performance.

Acknowledgments This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1 under the project R-263-000-A12-112.

References

1. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
2. Zhang, Q., Li, H.: Moea/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evolutionary Computation **11**(6), 712–731 (2007)
3. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. Trans. Evol. Comp **13**(2), 284–302 (2009)
4. Cheong, C., Tan, K., Liu, D., Lin, C.: Multi-objective and prioritized berth allocation incontainer ports. Annals of Operations Research **180**(1), 63–103 (2010)
5. Tan, W., Lu, F., Loh, A., Tan, K.: Modeling and control of a pilot ph plant using genetic algorithm. Engineering Applications of Artificial Intelligence **18**(4), 485–494 (2005)
6. Ang, J., Tan, K., Mamun, A.: An evolutionary memetic algorithm for rule extraction. Expert Systems with Applications **37**(2), 1302–1315 (2010)
7. Tan, K., Tang, H., Ge, S.: On parameter settings of hopfield networks applied to traveling salesman problems. IEEE Transactions on Circuits and Systems I: Regular Papers **52**(5), 994–1002 (2005)
8. Tan, K., Tang, H., Yi, Z.: Global exponential stability of discrete-time neural networks for constrained quadratic optimization. Neurocomputing **56**, 399–406 (2004)
9. Tan, K., Li, Y.: Grey-box model identification via evolutionary computing. Control Engineering Practice **10**(7), 673–684 (2002). Developments in High Precision Servo Systems
10. Muruganantham, A., Zhao, Y., Gee, S.B., Qiu, X., Tan, K.C.: Dynamic multiobjective optimization using evolutionary algorithm with kalman filter. Procedia Computer Science **24**, 66–75 (2013). 17th Asia Pacific Symposium on Intelligent and Evolutionary Systems, (IES2013)

11. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons Inc, New York, NY, USA (2001)
12. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans. Evolutionary Computation **8**(5), 425–442 (2004)
13. Branke, J.: Evolutionary Optimization in Dynamic Environments. Kluwer Academic Publishers, Norwell (2001)
14. Kalman, R.E.: A New Approach to Linear Filtering and Prediction Problems. Transactions of the ASME Journal of Basic Engineering **82**(D), 35–45 (1960)
15. Welch, G., Bishop, G.: An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA (1995)
16. Helbig, M., Engelbrecht, A.: Benchmark functions for cec 2015 special session and competition on dynamic multi-objective optimization. Technical report
17. Goh, C.K.: A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation **13**(1), 103–127 (2009)
18. Helbig, M., Engelbrecht, A.: Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation. In: 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 2047–2054 (June 2011)
19. Helbig, M., Engelbrecht, A.P.: Benchmarks for dynamic multi-objective optimisation. In: 2013 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), pp. 84–91. IEEE (2013)
20. Helbig, M., Engelbrecht, A.P.: Benchmarks for dynamic multi-objective optimisation algorithms. ACM Comput. Surv. **46**(3), 37:1–37:39 (2014)
21. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation **7**(2), 117–132 (2003)