Supplementary Document of "Dynamic Multi-Objectives Optimization with a Changing Number of Objectives"*

Ke Li#1, Renzhi Chen#2 and Xin Yao[2]

[1]Department of Computer Science, University of Exeter

[2]CERCIA, School of Computer Science, University of Birmingham

*Email: k.li@exeter.ac.uk, {rxc332, x.yao}@cs.bham.ac.uk

#The first two authors make equal contributions to this paper.

1 Proof of Theorem 1

Proof. Let us consider the scenario of increasing the number of objectives at first, where we prove the theorem by contradiction. At time step t_1, we assume that $\mathbf{x}^1$ and $\mathbf{x}^2$ are in PS_{t_1}. Accordingly, $\mathbf{F}(\mathbf{x}^1, t_1)$ and $\mathbf{F}(\mathbf{x}^2, t_1)$ are in PF_{t_1}. At time step t_2, we increase the number of objectives by one, i.e., $m(t_2) = m(t_1)+1$. Assume that $\mathbf{x}^1$ is still in PS_{t_2} whereas $\mathbf{x}^2$ is not, thus we have $\mathbf{x}^1 \preceq_{t_2} \mathbf{x}^2$. In other words, $\forall i \in \{1, \cdots, m(t_1), m(t_2)\}$ (i.e., $\forall i \in \{1, \cdots, m(t_1), m(t_1)+1\}$), $f_i(\mathbf{x}^1, t_2) \leq f_i(\mathbf{x}^2, t_2)$; and $\exists j \in \{1, \cdots, m(t_1), m(t_1)+1\}$, $f_j(\mathbf{x}^1, t_2) < f_j(\mathbf{x}^2, t_2)$. This contradicts the assumption that $\mathbf{x}^1$ and $\mathbf{x}^2$ are non-dominated from each other at time step t_1. Then, we conclude that PF_{t_1} is a subset of PF_{t_2} when increasing the number of objectives.

Now let us consider the scenario of decreasing the number of objectives. At time step t_1, we assume that $\mathbf{x}^1$ and $\mathbf{x}^2$ are in PS_{t_1}. Accordingly, $\mathbf{F}(\mathbf{x}^1, t_1)$ and $\mathbf{F}(\mathbf{x}^2, t_1)$ are in PF_{t_1}. Furthermore, we assume that $\forall i \in \{1, \cdots, m(t_1)-1\}$, we have $f_i(\mathbf{x}^1, t_1) \leq f_i(\mathbf{x}^2, t_1)$ and $f_{m(t_1)}(\mathbf{x}^1, t_1) > f_{m(t_1)}(\mathbf{x}^2, t_1)$. At time step t_2, we decrease the number of objectives by one, i.e., $m(t_2) = m(t_1)-1$. If $f_{m(t_1)}$ is removed at time step t_2, we can derive that $\mathbf{x}^1 \preceq_{t_2} \mathbf{x}^2$. That is to say $\mathbf{x}^2$ is not in PF_{t_2}. On the other hand, if f_i, where $i \in \{1, \cdots, m(t_1)-1\}$, is removed at time step t_2, we can derive that $\mathbf{x}^1$ and $\mathbf{x}^2$ are still non-dominated from each other. In other words, $\mathbf{x}^1$ and $\mathbf{x}^2$ are still in PF_{t_2}. All in all, we conclude that PF_{t_1} is a superset of PF_{t_2}. ■

*This is the supplementary document for the corresponding paper, which is published in IEEE Trans. on Evolutionary Computation, Vol 21, Issue 1, pp: 157-171, 2018.

2 Mathematical Definitions of Benchmark Problems

This section provides the mathematical definitions of the dynamic multi-objective benchmark problems used in our empirical studies. Note that these benchmark problems are developed from the classic DTLZ benchmark suite [1]. Furthermore, in addition to the changing number of objectives, F5 and F6 are also accompanied by a time-dependent change of the shape or position of the PF or PS.

Table 1: Mathematical Definitions of Dynamic Multi-Objective Benchmark Problems

Problem Instance	Definition	Domain
F1	$f_1 = (1+g)0.5\prod_{i=1}^{m(t)-1} x_i$ $f_{j=2:m(t)-1} = (1+g)0.5(\prod_{i=1}^{m(t)-j} x_i)(1-x_{m(t)-j+1})$ $f_{m(t)} = (1+g)0.5(1-x_1)$ $g = 100[n-m(t)+1+\sum_{i=m(t)}^{n}((x_i-0.5)^2-\cos(20\pi(x_i-0.5)))]$	$[0,1]$
F2	$f_1 = (1+g)0.5\prod_{i=1}^{m(t)-1} \cos(x_i\pi/2)$ $f_{j=2:m(t)-1} = (1+g)0.5(\prod_{i=1}^{m(t)-j} \cos(x_i\pi/2))(\sin(x_{m(t)-j+1}\pi/2))$ $f_{m(t)} = (1+g)\sin(x_1\pi/2)$ $g = \sum_{i=m(t)}^{n}(x_i-0.5)^2$	$[0,1]$
F3	as F2, except g is replaced by the one from F1	$[0,1]$
F4	as F2, except x_i is replaced by x_i^{α}, where $i \in \{1,\cdots,m(t)-1\}, \alpha > 0$	$[0,1]$
F5	as F2, except $g = \sum_{i=m(t)}^{n}(x_i - G(\bar{t}))^2$ where $G(\bar{t}) = \lvert\sin(0.5\pi\bar{t})\rvert$, $\bar{t} = \frac{1}{n_{\bar{t}}}\lfloor\frac{\tau}{\tau_{\bar{t}}}\rfloor$	$[0,1]$
F6	as F2, except $g = G(\bar{t}) + \sum_{i=m(t)}^{n}(x_i - G(\bar{t}))^2$ where $G(\bar{t}) = \lvert\sin(0.5\pi\bar{t})\rvert$, $\bar{t} = \frac{1}{n_{\bar{t}}}\lfloor\frac{\tau}{\tau_{\bar{t}}}\rfloor$ and x_i is replaced by $x_i^{F(\bar{t})}$, where $i \in \{1,\cdots,m(t)-1\}$ and $F(\bar{t}) = 1 + 100\sin^4(0.5\pi\bar{t})$	$[0,1]$

3 Descriptions of Different Compared Algorithms

In our empirical studies, four state-of-the-art EMO algorithms are used for comparisons: the dynamic version of the elitist non-dominated sorting genetic algorithm (DNSGA-II) [2] and MOEA/D with Kalman Filter prediction (MOEA/D-KF) [3]; and their corresponding stationary baseline NSGA-II [4] and MOEA/D [5]. They were chosen because of their popularity and good performance in both dynamic and static environments. Note that there is a significant amount of developments in the EMO field, e.g. [6–28]. Comparisons with the baseline algorithms are important. Because we want to check whether the dynamic algorithms outperform their static counterparts or not when handling the DMOP with a changing number of objectives. The following paragraphs provide some brief descriptions of these compared algorithms.

- *DNSGA-II*: To make the classic NSGA-II suitable for handling dynamic optimization problems, [2] suggested to replace some population members with either randomly generated solutions or mutated solutions upon existing ones once a change occurs. As reported in [2], the prior one performs better on DMOPs with severely changing environments while the latter one may work well on DMOPs with moderate changes. In our experiments, we adopt the prior DNSGA-II version in view of its slightly better performance reported in [2].

- *MOEA/D-KF*: This is a recently proposed prediction-based strategy that employs a linear discrete time Kalman Filter to model the movements of the PS in the dynamic environment. Thereafter, this model is used to predict the new location of the PS when a change occurs. Empirical results in [3] has shown that MOEA/D-KF is very competitive for the dynamic optimization and it outperforms the other state-of-the-art predictive strategies, e.g., [29] and [30].

- *NSGA-II*: It at first uses non-dominated sorting to divide the population into several non-domination levels. Solutions in the first several levels have a high priority to be selected as the next parents. The exceeded solutions are trimmed according to the density information.

- *MOEA/D*: This is a representative of the decomposition-based EMO methods. Its basic idea is to decompose the original MOP into several subproblems, either single-objective scalar functions or simplified MOPs. Thereafter, it employs some population-based techniques to solve these subproblems in a collaborative manner.

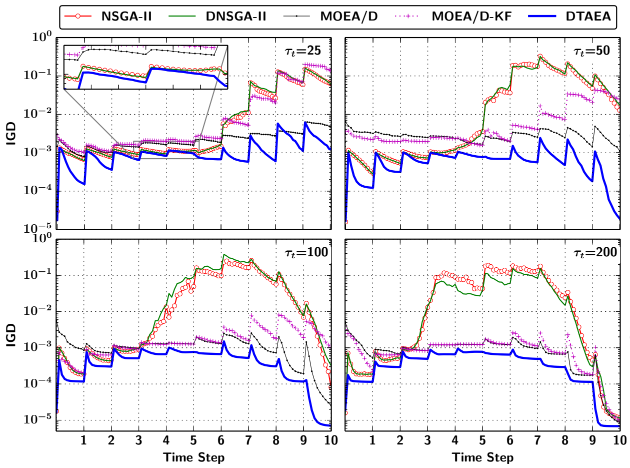
4 Settings of the Weight Vectors

This section provides the settings of the number of weight vectors used in our empirical studies. Note that we use the method developed in [16] to generate weight vectors when the number of objectives is larger than 4.

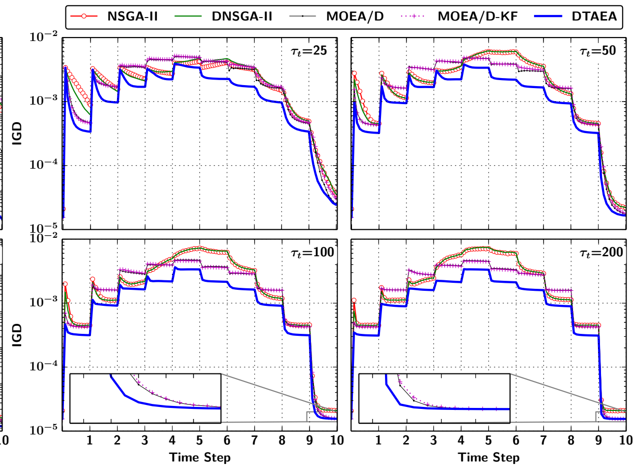
Table 2: Number of Weight Vectors

m	# of weight vectors
2	300 ($H = 299$)
3	300 ($H = 23$)
4	286 ($H = 10$)
5	280 ($H_1 = 6$, $H_2 = 4$)
6	273 ($H_1 = 5$, $H_2 = 2$)
7	294 ($H_1 = 4$, $H_2 = 3$)

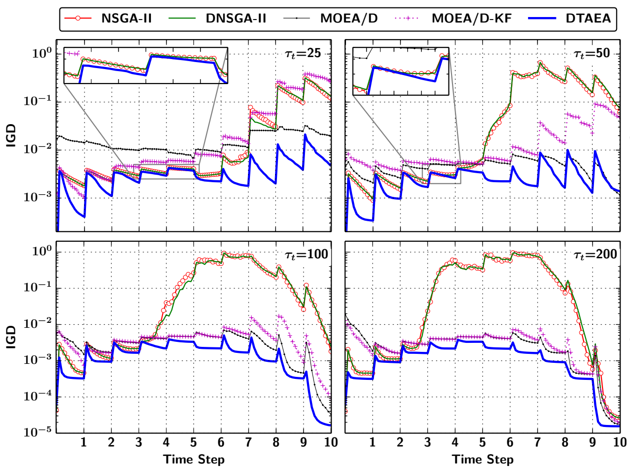
H is the number of divisions on each coordinate. Two-layer weight vector generation method is applied for 5- to 7-objective cases. H_1 and H_2 is the number of divisions for the boundary and inside layer, respectively.



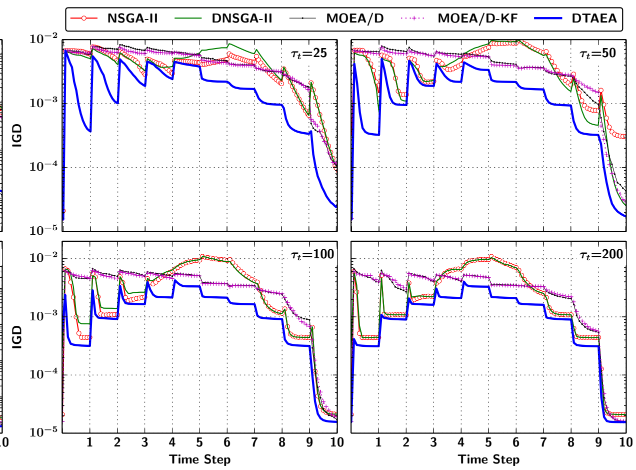
(a) F1



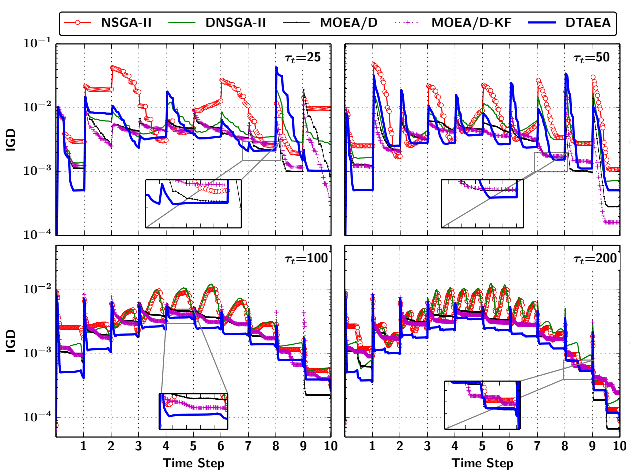
(b) F2



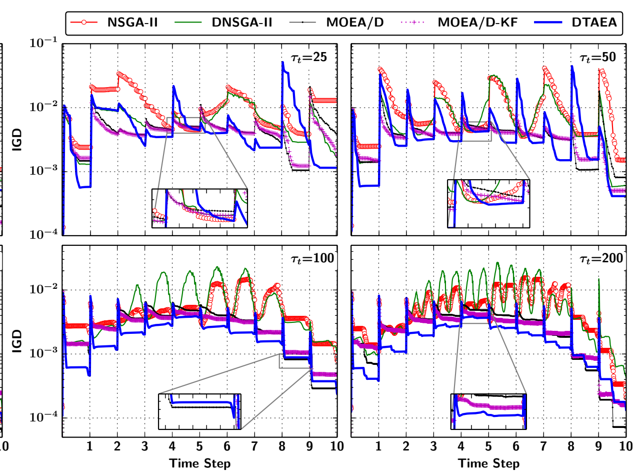
(c) F3



(d) F4

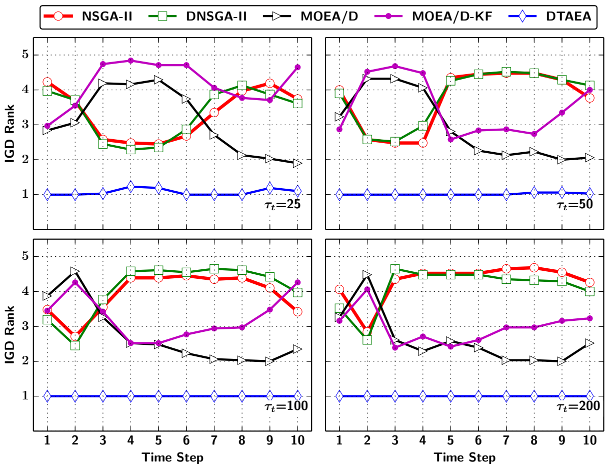


(e) F5

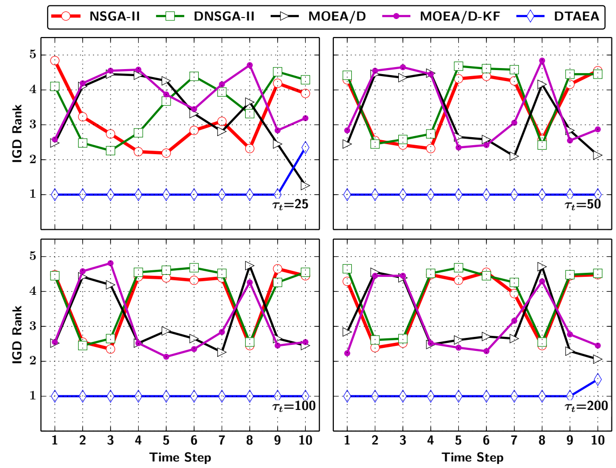


(f) F6

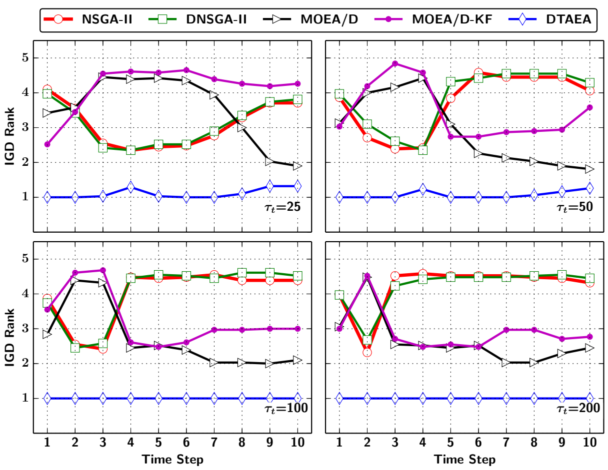
Figure 1: IGD trajectories across the whole evolution process.



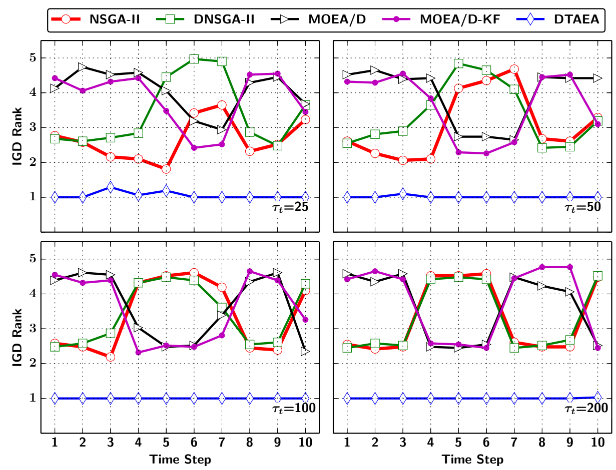
(a) F1



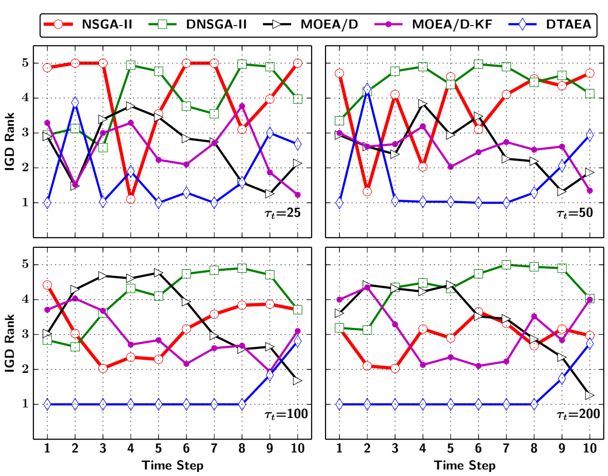
(b) F2



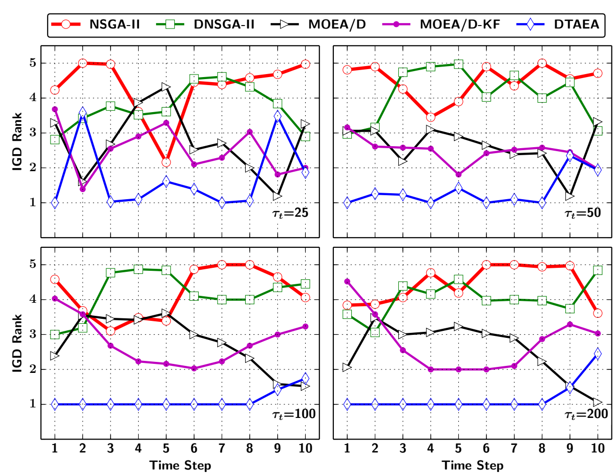
(c) F3



(d) F4

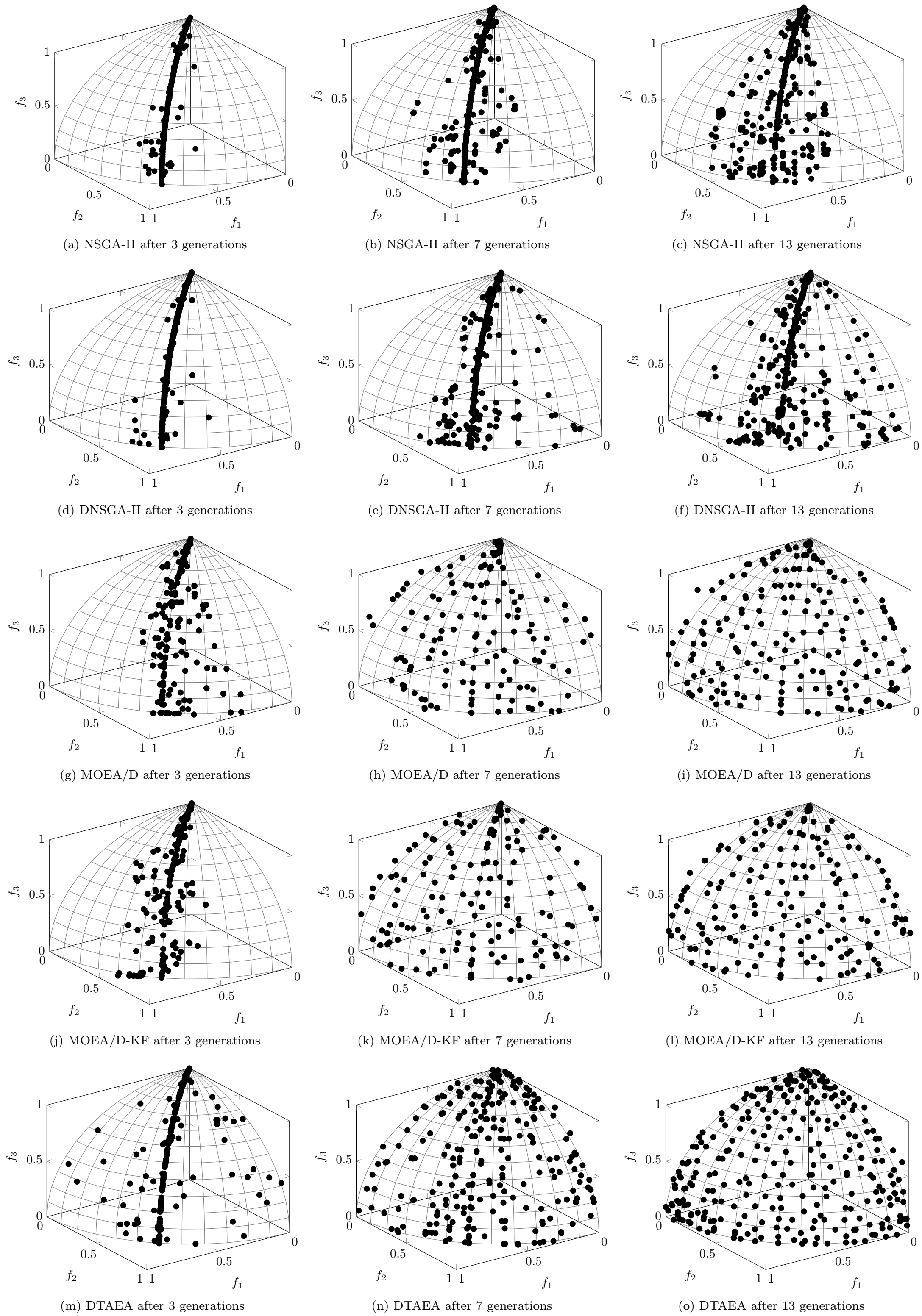


(e) F5



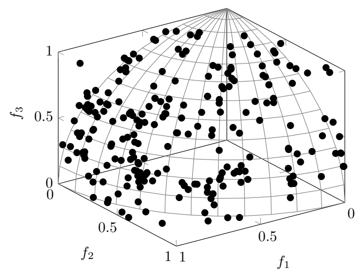
(f) F6

Figure 2: The rank of IGD obtained by different algorithms at each time step.

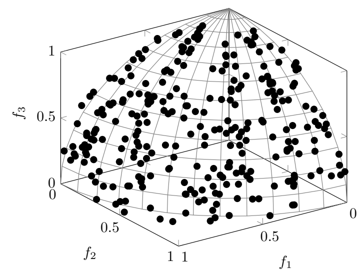


(a) NSGA-II after 3 generations
(b) NSGA-II after 7 generations
(c) NSGA-II after 13 generations
(d) DNSGA-II after 3 generations
(e) DNSGA-II after 7 generations
(f) DNSGA-II after 13 generations
(g) MOEA/D after 3 generations
(h) MOEA/D after 7 generations
(i) MOEA/D after 13 generations
(j) MOEA/D-KF after 3 generations
(k) MOEA/D-KF after 7 generations
(l) MOEA/D-KF after 13 generations
(m) DTAEA after 3 generations
(n) DTAEA after 7 generations
(o) DTAEA after 13 generations

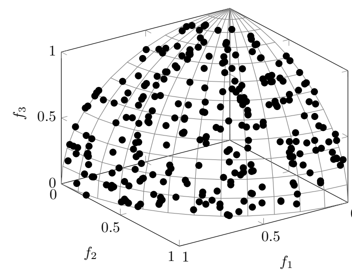
Figure 3: Variation of the population distribution when increasing the number of objectives from 2 to 3.



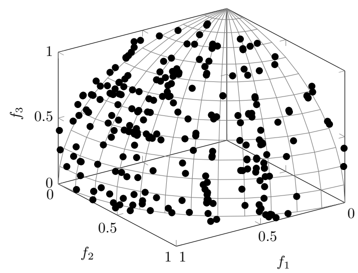
(a) NSGA-II after 3 generations



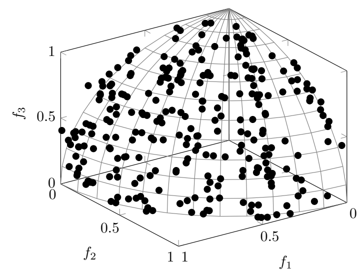
(b) NSGA-II after 7 generations



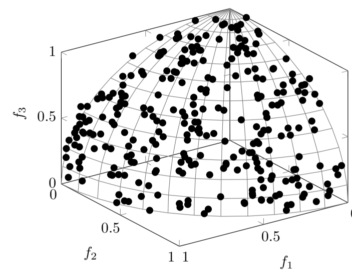
(c) NSGA-II after 13 generations



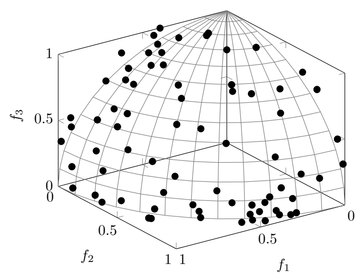
(d) DNSGA-II after 3 generations



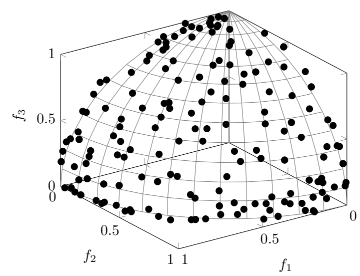
(e) DNSGA-II after 7 generations



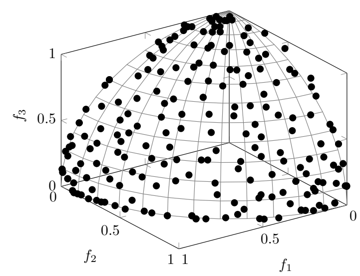
(f) DNSGA-II after 13 generations



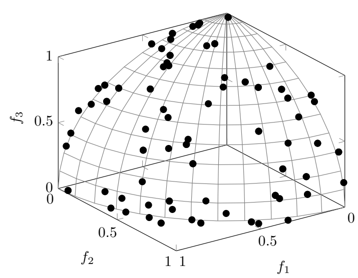
(g) MOEA/D after 3 generations



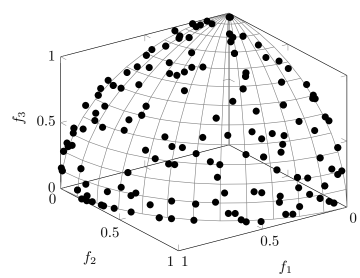
(h) MOEA/D after 7 generations



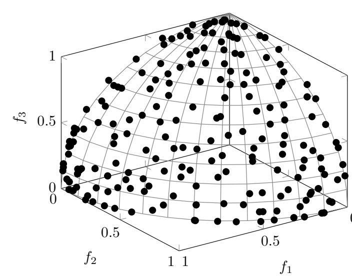
(i) MOEA/D after 13 generations



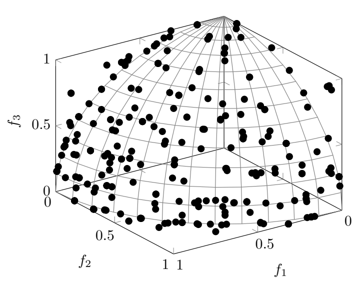
(j) MOEA/D-KF after 3 generations



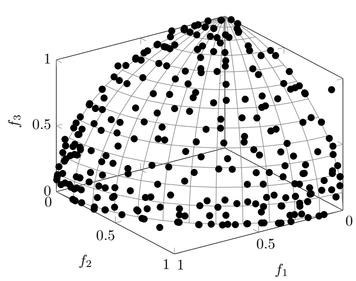
(k) MOEA/D-KF after 7 generations



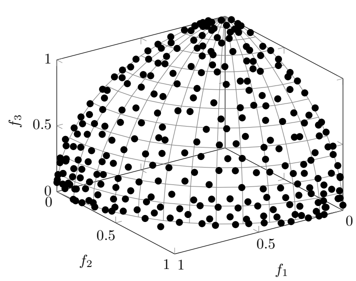
(l) MOEA/D-KF after 13 generations



(m) DTAEA after 3 generations



(n) DTAEA after 7 generations



(o) DTAEA after 13 generations

Figure 4: Variation of the population distribution when decreasing the number of objectives from 4 to 3.

5 Effects of the Update Mechanisms

As discussed in Section III-B of our paper, the update mechanisms are used to maintain the complementary effects between the CA and the DA. The CA keeps a continuously strong selection pressure for the population convergence; while the DA maintains a set of well diversified solutions. In order to validate the importance of the three different components of DTAEA, we developed three DTAEA variants as follows:

- DTAEA-$v1$: this variant modifies DTAEA by removing the restricted mating selection mechanism proposed in Section III-C of our paper. In particular, now the mating parents are respectively selected from the CA and the DA without considering the population distribution.
- DTAEA-$v2$: this variant modifies DTAEA by removing the reconstruction mechanism proposed in Section III-A of our paper. In other words, it does not make any response to the changing environment.
- DTAEA-$v3$: this variant merely uses the update mechanisms to maintain the CA and the DA whereas it does not make any response to the changing environment. In addition, it does not use the restricted mating selection mechanism as DTAEA-$v1$.

We conduct the experiments on F1 to F6 according to the same experimental settings introduced in Section IV of our paper. Table III and Table IV of the supplementary file give the median and IQR values according to the MIGD and MHV metrics. From these results we can see that the original DTAEA, consisted of all three components, has shown clearly better performance than the other three variants. More specifically, as shown in Table III and Table IV, the performance of DTAEA-$v3$ is the worst among all three variants. This observation is reasonable as DTAEA-$v3$ neither responds to the changing environment nor takes advantages of the complementary effect of the CA and the DA for offspring generation. The performance of DTAEA-$v2$ is slightly better than DTAEA-$v3$. Therefore, we can see that even without any response to the changing environment, the restricted mating selection mechanism can also provide some guidance to the search process. As for DTAEA-$v1$, we can see that the performance can be significantly improved by using the reconstruction mechanism proposed in Section III-A of our paper to respond to the changing environment. This superiority is more obvious when the frequency of change is high. By comparing the results between DTAEA-$v1$ and the original DTAEA, we can clearly see the importance of taking advantages of the complementary effect of the CA and the DA for offspring generation. All in all, we can conclude that all three components of DTAEA are of significant importance for handling the DMOP with a changing number of objectives.

Table 3: Performance Comparisons of DTAEA and its Three Variants on MIGD Metric

	τ_t	DTAEA-$v1$		DTAEA-$v2$		DTAEA-$v3$		DTAEA	
		MIGD	R	MIGD	R	MIGD	R	MIGD	R
F1	25	**5.05E-4(1.85E-4)**	**1.9**	7.14E-4(1.72E-4)†	2.8	7.29E-4(2.40E-4)†	2.9	5.50E-4(1.99E-4)	2.3
	50	3.95E-4(2.49E-5)	2.5	4.02E-4(1.54E-5)†	2.6	4.04E-4(1.19E-5)†	2.8	**3.88E-4(1.03E-5)**	**2.1**
	100	3.72E-4(3.97E-6)†	3.5	3.64E-4(3.09E-6)	2	3.65E-4(2.43E-6)	2.7	**3.64E-4(2.49E-6)**	**1.8**
	200	3.59E-4(1.07E-6)†	3.4	3.56E-4(1.20E-6)	1.8	3.59E-4(2.46E-6)†	3.4	**3.55E-4(1.41E-6)**	**1.5**
F2	25	1.28E-3(5.20E-6)†	2.6	1.39E-3(1.97E-5)†	2.8	1.40E-3(4.08E-5)†	3.1	**1.26E-3(4.58E-6)**	**1.5**
	50	1.26E-3(1.59E-6)†	3.4	1.25E-3(1.76E-5)	2	1.25E-3(2.77E-5)†	2.8	**1.25E-3(3.26E-6)**	**1.9**
	100	1.23E-3(1.88E-6)†	3.4	1.22E-3(2.24E-6)†	1.9	1.23E-3(2.32E-6)†	3	**1.22E-3(1.30E-6)**	**1.7**
	200	1.21E-3(1.86E-6)†	3.3	1.20E-3(1.31E-6)†	1.8	1.21E-3(1.73E-6)†	3.2	**1.20E-3(2.20E-6)**	**1.7**
F3	25	**1.92E-3(9.48E-4)**	**1.9**	2.63E-3(1.29E-3)†	3	2.32E-3(8.94E-4)	2.9	2.06E-3(1.34E-3)	2.2
	50	**1.32E-3(4.29E-5)**‡	**2.1**	1.42E-3(5.20E-5)	2.8	1.44E-3(5.95E-5)	2.9	1.39E-3(1.06E-4)	2.3
	100	1.26E-3(5.32E-6)†	3.1	1.25E-3(1.38E-5)	2.1	1.25E-3(1.87E-5)†	2.7	**1.24E-3(9.04E-6)**	**2**
	200	1.23E-3(3.09E-6)†	3	1.22E-3(2.43E-6)†	2.1	1.23E-3(5.26E-6)†	3.2	**1.22E-3(4.17E-6)**	**1.8**
F4	25	**1.29E-3(3.40E-5)**‡	**1.7**	6.82E-3(2.18E-7)†	3.3	6.82E-3(1.87E-7)†	3.3	1.32E-3(7.80E-5)	1.7
	50	1.25E-3(2.17E-6)†	2	6.81E-3(8.96E-8)†	3.3	6.81E-3(7.24E-8)†	3.3	**1.24E-3(4.05E-6)**	**1.4**
	100	1.22E-3(1.58E-6)†	2	6.81E-3(1.93E-7)†	3.3	6.81E-3(1.05E-7)†	3.4	**1.22E-3(1.50E-6)**	**1.4**
	200	1.21E-3(1.85E-6)†	2.1	6.81E-3(1.58E-4)†	3.3	6.81E-3(3.71E-4)†	3.4	**1.20E-3(1.35E-6)**	**1.2**
F5	25	**2.52E-3(9.09E-5)**‡	1.8	7.69E-3(5.69E-4)†	3.3	7.66E-3(3.22E-4)†	3.2	2.61E-3(9.71E-5)	**1.6**
	50	2.02E-3(6.94E-5)†	1.8	1.35E-2(3.26E-4)†	3.5	1.34E-2(2.17E-4)†	3.5	**1.91E-3(6.69E-5)**	**1.2**
	100	1.49E-3(1.72E-5)†	1.7	1.80E-2(4.43E-4)†	3.5	1.79E-2(9.68E-4)†	3.5	**1.45E-3(2.98E-5)**	**1.3**
	200	1.38E-3(1.17E-5)†	1.7	1.95E-2(5.39E-4)†	3.4	1.97E-2(3.73E-4)†	3.6	**1.36E-3(6.77E-6)**	**1.3**
F6	25	**2.90E-3(1.68E-4)**	1.6	1.11E-2(7.24E-4)†	3.5	1.12E-2(5.90E-4)†	3.5	2.98E-3(1.76E-4)	**1.4**
	50	2.22E-3(7.36E-5)†	1.7	1.61E-2(6.16E-4)†	3.5	1.59E-2(7.97E-4)†	3.5	**2.08E-3(5.08E-5)**	**1.3**
	100	1.57E-3(2.31E-5)	1.6	2.10E-2(9.39E-4)†	3.6	2.10E-2(9.48E-4)†	3.4	**1.56E-3(2.25E-5)**	**1.4**
	200	1.46E-3(1.07E-5)	1.6	2.23E-2(7.09E-4)†	3.6	2.19E-2(6.50E-4)†	3.4	**1.46E-3(2.09E-5)**	**1.4**

R denotes the global rank assigned to each algorithm by averaging the ranks obtained at all time steps. Wilcoxon's rank sum test at a 0.05 significance level is performed between DTAEA and each of DTAEA-$v1$, DTAEA-$v2$ and DTAEA-$v3$. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of DTAEA, respectively. The best median value is highlighted in boldface with gray background.

Table 4: Performance Comparisons of DTAEA and its Three Variants on MHV Metric

	τ_t	DTAEA-$v1$		DTAEA-$v2$		DTAEA-$v3$		DTAEA	
		MHV	R	MHV	R	MHV	R	MHV	R
F1	25	**99.85%(4.60E-3)**	**1.7**	99.57%(3.05E-3)	3	99.45%(6.72E-3)†	3.3	99.7%(6.03E-3)	2
	50	100.00%(1.40E-4)†	2.1	99.98%(1.86E-4)†	2.7	99.98%(2.71E-4)†	3.3	**100.0%(8.39E-5)**	**1.8**
	100	100.00%(1.82E-5)†	2.5	100.00%(2.95E-5)†	2.4	100.00%(2.75E-5)†	3.4	**100.0%(1.84E-5)**	**1.7**
	200	100.00%(1.66E-5)†	2.5	100.00%(2.19E-5)†	2.2	100.00%(2.65E-5)†	3.1	**100.0%(2.69E-5)**	**2.1**
F2	25	94.37%(7.52E-5)†	2.6	93.94%(1.08E-3)†	2.7	93.86%(1.94E-3)†	3.5	**94.4%(6.26E-5)**	**1.2**
	50	94.42%(2.46E-5)†	2.9	94.42%(2.21E-4)†	2.4	94.38%(2.65E-4)†	3.5	**94.4%(3.09E-5)**	**1.2**
	100	94.45%(8.64E-6)†	3.3	94.46%(7.38E-6)†	1.9	94.45%(1.34E-5)†	3.4	**94.5%(8.88E-6)**	**1.5**
	200	94.46%(4.63E-6)†	3.4	94.46%(5.29E-6)†	1.8	94.46%(8.98E-6)†	3.2	**94.5%(4.03E-6)**	**1.7**
F3	25	**91.08%(6.51E-2)**	**1.7**	89.86%(7.13E-2)	3	89.84%(4.89E-2)	3.3	90.6%(9.47E-2)	1.9
	50	**94.24%(2.74E-3)**	**1.9**	93.90%(1.66E-3)†	2.8	93.83%(2.62E-3)†	3.2	94.0%(3.66E-3)	2.1
	100	94.39%(3.78E-4)†	2.5	94.40%(2.96E-4)†	2.4	94.38%(5.41E-4)†	3.3	**94.4%(3.32E-4)**	**1.8**
	200	94.43%(1.60E-4)†	2.9	94.44%(1.52E-4)†	2.1	94.44%(2.72E-4)†	3.2	**94.4%(1.26E-4)**	**1.8**
F4	25	94.39%(1.29E-4)	1.9	79.48%(5.82E-5)†	3.3	79.47%(6.49E-5)†	3.3	**94.4%(1.60E-4)**	**1.5**
	50	94.44%(1.76E-5)†	2.1	79.47%(1.25E-4)†	3.3	79.47%(1.24E-4)†	3.3	**94.4%(1.16E-5)**	**1.3**
	100	94.46%(8.42E-6)†	2.1	79.46%(2.01E-4)†	3.3	79.47%(1.21E-4)†	3.3	**94.5%(3.26E-6)**	**1.2**
	200	94.46%(8.77E-6)†	2	79.47%(1.22E-2)†	3.3	79.47%(3.27E-2)†	3.4	**94.5%(4.19E-6)**	**1.2**
F5	25	**90.37%(5.17E-3)**‡	1.8	65.36%(2.23E-2)†	3.4	64.82%(8.67E-3)†	3.4	89.4%(6.64E-3)	**1.5**
	50	93.53%(1.34E-3)†	1.8	46.58%(1.64E-2)†	3.5	46.18%(2.98E-2)†	3.5	**93.7%(1.25E-3)**	**1.2**
	100	94.53%(4.58E-4)†	1.8	15.51%(3.19E-2)†	3.5	16.54%(3.35E-2)†	3.5	**94.6%(3.62E-4)**	**1.2**
	200	94.73%(3.03E-4)†	1.8	11.56%(1.35E-2)†	3.4	11.09%(6.06E-3)†	3.6	**94.8%(1.84E-4)**	**1.2**
F6	25	**89.31%(1.18E-2)**	1.7	51.61%(4.21E-2)†	3.5	51.54%(2.37E-2)†	3.5	89.1%(9.72E-3)	**1.3**
	50	92.93%(1.19E-3)†	1.8	33.26%(1.84E-2)†	3.5	33.80%(2.49E-2)†	3.5	**93.7%(1.45E-3)**	**1.2**
	100	93.92%(5.32E-4)†	1.8	7.99%(8.41E-3)†	3.5	8.22%(7.57E-3)†	3.5	**94.5%(3.50E-4)**	**1.2**
	200	94.14%(3.41E-4)†	1.8	7.92%(7.58E-3)†	3.5	8.07%(9.59E-3)†	3.5	**94.7%(3.39E-4)**	**1.2**

R denotes the global rank assigned to each algorithm by averaging the ranks obtained at all time steps. Wilcoxon's rank sum test at a 0.05 significance level is performed between DTAEA and each of DTAEA-$v1$, DTAEA-$v2$ and DTAEA-$v3$. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of DTAEA, respectively. The best median value is highlighted in boldface with gray background.

6 Performance Comparisons on a Different Changing Sequence

In Section V-A and Section V-B of our paper, the experiments only consider the scenarios where the number of objectives increases or decreases by one at each time step. A natural question is: how is the performance of our proposed algorithm under the circumstance where the number of objectives changes in a different sequence? Without loss of generality, this further experiment considers the time varying number of objectives $m(t)$ as follows:

$$m(t) = \begin{cases} 3, & t = 1 \\ m(t-1) + 2, & t \in [2, 3] \\ m(t-1) - 2, & t \in [4, 5] \\ m(t-1) - 1, & t = 6 \end{cases} \tag{1}$$

where $t \in \{1, \cdots, 6\}$ is a discrete time. Here we also consider four different frequencies of change, i.e., τ_t is as 25, 50, 100 and 200, respectively. From the empirical results shown in Table 5 and Table 6, we can clearly see that our proposed DTAEA is the best optimizer on almost all comparisons (92 out of 96 for MIGD and 88 out 96 for MHV). Similar to the observations from the previous sections, the performance of DTAEA might not be stable under a high frequency of change; while its performance becomes constantly competitive with the increase of τ_t.

Table 5: Performance Comparisons on MIGD Metric with a Different Changing Sequence.

	τ_t	NSGA-II		DNSGA-II		MOEA/D		MOEA/D-KF		DTAEA	
		MIGD	R	MIGD	R	MIGD	R	MIGD	R	MIGD	R
F1	25	2.97E-3(3.40E-3)†	2.7	2.00E-3(3.56E-3)†	2.8	2.40E-3(8.80E-3)†	3.8	1.67E-2(1.81E-2)†	4.3	**1.10E-3(6.03E-4)**	**1.4**
	50	1.31E-2(1.26E-2)†	3.8	1.33E-2(1.21E-2)†	3.8	7.78E-4(2.92E-4)†	3	1.80E-3(2.10E-3)†	3.5	**3.38E-4(3.38E-5)**	**1**
	100	2.54E-2(3.50E-2)†	4	3.23E-2(3.94E-2)†	4.1	5.73E-4(4.71E-5)†	2.8	7.27E-4(4.18E-4)†	3.2	**2.92E-4(1.75E-6)**	**1**
	200	3.11E-3(4.08E-2)†	4.3	1.25E-2(3.68E-2)†	4.3	5.02E-4(2.16E-4)†	2.4	6.39E-4(1.89E-3)†	3	**2.90E-4(2.10E-6)**	**1**
F2	25	1.41E-3(5.72E-5)†	3.2	1.48E-3(1.11E-4)†	3.4	1.81E-3(8.28E-5)†	3.5	1.82E-3(4.91E-5)†	3.9	**1.07E-3(5.22E-6)**	**1**
	50	1.54E-3(1.01E-4)†	3.8	1.63E-3(1.28E-4)†	4	1.70E-3(6.46E-5)†	2.9	1.71E-3(6.81E-5)†	3.3	**1.06E-3(3.42E-6)**	**1**
	100	1.82E-3(1.32E-4)†	4.1	1.90E-3(9.04E-5)†	4.2	1.62E-3(2.88E-5)†	2.7	1.64E-3(2.74E-5)†	3.1	**1.04E-3(1.83E-6)**	**1**
	200	1.98E-3(1.03E-4)†	4.1	1.96E-3(1.38E-4)†	4.2	1.58E-3(3.77E-5)†	2.8	1.56E-3(2.12E-5)†	2.8	**1.03E-3(1.34E-6)**	**1.1**
F3	25	8.26E-3(9.99E-3)†	2.8	8.09E-3(5.22E-3)†	2.8	4.88E-3(4.31E-3)†	3.8	2.11E-2(4.73E-2)†	4.2	**2.66E-3(3.29E-3)**	**1.4**
	50	1.33E-2(1.64E-2)†	3.4	1.09E-2(1.84E-2)†	3.4	2.79E-3(6.58E-4)†	3.2	5.05E-3(6.62E-3)†	3.8	**1.20E-3(1.04E-4)**	**1.1**
	100	9.02E-2(5.18E-2)†	3.9	9.16E-2(4.43E-2)†	4	2.42E-3(2.19E-3)†	2.8	2.41E-3(1.63E-3)†	3.2	**1.06E-3(8.73E-6)**	**1**
	200	1.51E-1(1.28E-1)†	4.4	1.76E-1(9.13E-2)†	4.4	1.69E-3(5.03E-5)†	2.4	2.01E-3(2.33E-4)†	2.9	**1.05E-3(7.16E-6)**	**1**
F4	25	2.64E-3(4.22E-4)†	2.5	2.79E-3(3.27E-4)†	3	4.08E-3(7.27E-4)†	4.2	3.91E-3(3.17E-4)†	4.3	**1.10E-3(2.29E-5)**	**1**
	50	1.77E-3(3.40E-4)†	2.8	2.26E-3(3.41E-4)†	3.5	3.44E-3(5.08E-4)†	4	3.03E-3(3.74E-4)†	3.8	**1.06E-3(1.24E-3)**	**1**
	100	2.19E-3(2.30E-4)†	3.2	2.29E-3(3.31E-4)†	3.4	3.04E-3(3.42E-4)†	3.8	3.14E-3(3.88E-4)†	3.7	**1.04E-3(2.78E-6)**	**1**
	200	2.26E-3(1.92E-4)†	3.3	2.24E-3(1.88E-4)†	3.1	2.71E-3(1.56E-4)†	3.8	2.68E-3(8.39E-4)†	3.8	**1.03E-3(2.42E-6)**	**1**
F5	25	7.14E-3(3.50E-3)†	3.9	3.87E-3(5.66E-4)†	3.5	2.53E-3(2.74E-4)‡	2.5	**2.30E-3(2.67E-4)**‡	**2.5**	2.94E-3(3.79E-4)	2.7
	50	2.83E-3(2.91E-4)†	3.6	3.27E-3(3.15E-4)†	4.4	1.99E-3(9.74E-5)†	2.6	1.94E-3(1.17E-4)†	2.4	**1.84E-3(8.27E-5)**	**2**
	100	1.72E-3(2.23E-4)†	3	1.84E-3(1.50E-4)†	3.8	1.80E-3(7.46E-5)†	3.3	1.74E-3(1.32E-4)†	3.3	**1.29E-3(1.26E-5)**	**1.6**
	200	1.57E-3(1.12E-4)†	2.8	1.83E-3(1.01E-4)†	4.1	1.68E-3(3.56E-5)†	3.2	1.64E-3(7.30E-5)†	3	**1.19E-3(1.29E-5)**	**1.9**
F6	25	8.61E-3(1.74E-3)†	4.4	5.03E-3(5.70E-4)†	3.5	2.58E-3(3.54E-4)‡	2.3	**2.57E-3(3.77E-4)**‡	2.6	3.04E-3(1.71E-4)	**2.2**
	50	2.92E-3(2.43E-4)†	4	3.72E-3(6.24E-4)†	3.8	2.24E-3(2.14E-4)†	2.5	2.38E-3(1.57E-4)†	2.7	**2.05E-3(1.03E-4)**	**1.9**
	100	2.69E-3(3.24E-4)†	3.9	2.62E-3(1.58E-4)†	4.3	1.89E-3(9.26E-5)†	2.5	1.98E-3(1.20E-4)†	3	**1.35E-3(1.34E-5)**	**1.4**
	200	2.89E-3(3.07E-4)†	4.3	2.53E-3(1.01E-4)†	4	1.87E-3(7.80E-5)†	2.4	1.85E-3(7.47E-5)†	2.9	**1.27E-3(2.42E-5)**	**1.4**

R denotes the global rank assigned to each algorithm by averaging the ranks obtained at all time steps. Wilcoxon's rank sum test at a 0.05 significance level is performed between DTAEA and each of NSGA-II, DNSGA-II, MOEA/D and MOEA/D-KF. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of DTAEA, respectively. The best median value is highlighted in boldface with gray background.

References

[1] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization*, ser. Advanced In-

Table 6: Performance Comparisons on MHV Metric with a Different Changing Sequence.

	τ_t	NSGA-II		DNSGA-II		MOEA/D		MOEA/D-KF		DTAEA	
		MHV	R	MHV	R	MHV	R	MHV	R	MHV	R
F1	25	87.4%(1.13E-1)†	3.1	92.5%(1.33E-1)†	3.1	91.6%(3.74E-1)†	3.5	70.2%(1.77E-1)†	3.9	**98.2%(2.31E-2)**	**1.3**
	50	66.4%(1.49E-1)†	4.2	63.6%(2.28E-1)†	4.2	99.4%(7.77E-3)†	2.5	92.7%(1.24E-1)†	3	**100.0%(1.26E-4)**	**1**
	100	71.8%(1.12E-1)†	4.1	71.4%(8.87E-2)†	4	99.9%(1.26E-3)†	2.7	99.5%(1.13E-2)†	3.2	**100.0%(5.61E-5)**	**1**
	200	84.8%(1.97E-1)†	4	82.1%(2.41E-1)†	4	100.0%(1.08E-2)†	2.6	99.7%(9.86E-2)†	3.3	**100.0%(1.73E-5)**	**1**
F2	25	91.1%(3.29E-3)†	3.8	91.5%(2.75E-3)†	3.8	92.1%(1.93E-3)†	2.9	92.0%(1.49E-3)†	3.5	**92.9%(1.06E-4)**	**1**
	50	91.6%(3.16E-3)†	4.1	91.6%(3.23E-3)†	4	92.3%(1.01E-3)†	2.8	92.3%(1.28E-3)†	3.1	**92.9%(5.67E-5)**	**1**
	100	90.6%(8.76E-3)†	4.3	90.3%(8.01E-3)†	4.5	92.5%(6.31E-4)†	2.4	92.4%(6.47E-4)†	2.7	**92.9%(1.65E-5)**	**1**
	200	89.7%(7.17E-3)†	4.4	89.8%(9.25E-3)†	4.5	92.5%(5.50E-4)†	2.5	92.6%(4.09E-4)†	2.3	**93.0%(5.51E-6)**	**1.1**
F3	25	76.0%(5.96E-2)	3.1	75.8%(1.80E-2)	3.3	78.4%(3.14E-1)†	3.3	62.2%(6.96E-2)†	3.7	**85.5%(1.92E-1)**	**1.6**
	50	63.2%(1.23E-1)†	3.8	62.9%(2.14E-1)†	3.7	86.8%(3.79E-2)†	2.8	74.3%(2.87E-1)†	3.6	**92.5%(3.81E-3)**	**1.1**
	100	52.5%(9.26E-2)†	4.2	54.2%(5.61E-2)†	4.1	87.6%(1.24E-1)†	2.7	88.0%(1.04E-1)†	3	**92.9%(4.27E-4)**	**1**
	200	60.7%(6.73E-2)†	4.4	60.3%(7.66E-2)†	4.3	92.2%(3.02E-3)†	2.3	90.7%(2.05E-2)†	2.9	**92.9%(3.01E-4)**	**1**
F4	25	90.3%(5.61E-3)†	3	89.5%(5.88E-3)†	3.6	87.5%(4.24E-2)†	3.8	88.7%(1.84E-2)†	3.7	**92.9%(1.65E-4)**	**1**
	50	91.3%(1.32E-2)†	3.3	89.2%(1.91E-2)†	3.8	89.9%(2.30E-2)†	3.5	90.6%(5.72E-3)†	3.5	**92.9%(4.48E-2)**	**1**
	100	88.9%(1.26E-2)†	3.5	88.2%(2.45E-2)†	3.6	90.5%(3.51E-3)†	3.4	90.4%(9.23E-3)†	3.5	**93.0%(9.81E-6)**	**1**
	200	87.2%(1.65E-2)†	4.3	87.7%(1.92E-2)†	4	90.7%(5.10E-4)†	2.9	90.7%(1.84E-2)†	2.8	**93.0%(4.10E-6)**	**1**
F5	25	55.6%(5.86E-2)†	3.9	77.3%(3.98E-2)†	3.7	90.3%(1.23E-2)‡	**2.2**	**91.0%(4.75E-3)**‡	2.4	84.7%(2.59E-2)	2.8
	50	83.9%(1.80E-2)†	4.1	83.3%(2.76E-2)†	4.3	91.8%(3.26E-3)	2.2	**91.9%(4.21E-3)**	**1.8**	91.2%(2.63E-3)	2.6
	100	91.4%(8.99E-3)†	3.3	91.7%(3.53E-3)†	4	92.2%(2.68E-3)†	2.9	92.1%(5.46E-3)†	2.7	**92.3%(4.12E-4)**	**2.1**
	200	92.1%(5.78E-3)†	3.3	91.6%(3.04E-3)†	4.2	92.4%(5.56E-4)†	3	92.3%(3.23E-3)†	2.5	**92.7%(2.17E-4)**	**2.1**
F6	25	54.3%(5.32E-2)†	4.6	68.8%(3.85E-2)†	3.5	90.8%(1.37E-2)‡	**2**	**91.1%(7.15E-3)**‡	2.3	85.2%(1.40E-2)	2.6
	50	86.6%(1.04E-2)†	4.5	80.1%(4.06E-2)†	3.9	**91.8%(6.93E-3)**‡	**2**	91.7%(4.34E-3)‡	2	90.7%(3.66E-3)	2.7
	100	88.4%(1.29E-2)†	4.7	90.5%(3.61E-3)†	4	92.2%(1.36E-3)†	2.3	92.1%(4.34E-3)†	2.5	**92.3%(4.21E-4)**	**1.6**
	200	88.3%(1.27E-2)†	4.7	90.9%(3.60E-3)†	3.9	92.4%(8.28E-4)†	2.3	92.3%(2.97E-3)†	2.4	**92.6%(4.68E-4)**	**1.6**

R denotes the global rank assigned to each algorithm by averaging the ranks obtained at all time steps. Wilcoxon's rank sum test at a 0.05 significance level is performed between DTAEA and each of NSGA-II, DNSGA-II, MOEA/D and MOEA/D-KF. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of DTAEA, respectively. The best median value is highlighted in boldface with gray background.

formation and Knowledge Processing, A. Abraham, L. Jain, and R. Goldberg, Eds. Springer London, 2005, pp. 105–145.

[2] K. Deb, U. B. R. N., and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *EMO'07: Proc. of the 4th International Conference on Evolutionary Multi-Criterion Optimization*, 2007, pp. 803–817.

[3] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via kalman filter prediction," *IEEE Trans. Cybernetics*, 2015, accepted for publication.

[4] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[6] K. Li, J. Zheng, M. Li, C. Zhou, and H. Lv, "A novel algorithm for non-dominated hypervolume-based multiobjective optimization," in *SMC'09: Proc. of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, 2009, pp. 5220–5226.

[7] K. Li, S. Kwong, R. Wang, J. Cao, and I. J. Rudas, "Multi-objective differential evolution with self-navigation," in *SMC'12: Proc. of the 2012 IEEE International Conference on Systems, Man, and Cybernetics*, 2012, pp. 508–513.

[8] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Inf. Sci.*, vol. 182, no. 1, pp. 220–242, 2012.

[9] K. Li, S. Kwong, R. Wang, K. Tang, and K. Man, "Learning paradigm based on jumping genes: A general framework for enhancing exploration in evolutionary multiobjective optimization," *Inf. Sci.*, vol. 226, pp. 1–22, 2013.

[10] K. Li, R. Wang, S. Kwong, and J. Cao, "Evolving extreme learning machine paradigm with adaptive operator selection and parameter control," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 21, pp. 143–154, 2013.

[11] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

[12] K. Li, Á. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014.

[13] K. Li and S. Kwong, "A general framework for evolutionary multiobjective optimization via manifold learning," *Neurocomputing*, vol. 146, pp. 65–74, 2014.

[14] M. Wu, S. Kwong, Q. Zhang, K. Li, R. Wang, and B. Liu, "Two-level stable matching-based selection in MOEA/D," in *SMC'15: Proc. of the 2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 1720–1725.

[15] K. Li, K. Deb, and Q. Zhang, "Evolutionary multiobjective optimization with hybrid selection principles," in *CEC'15: Proc. of 2015 IEEE Congress on Evolutionary Computation*, 2015, pp. 900–907.

[16] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[17] K. Li, S. Kwong, and K. Deb, "A dual-population paradigm for evolutionary multiobjective optimization," *Inf. Sci.*, vol. 309, pp. 50–72, 2015.

[18] K. Li, M. N. Omidvar, K. Deb, and X. Yao, "Variable interaction in multi-objective optimization problems," in *PPSN'16: Proc. of 14th International Conference on Parallel Problem Solving from Nature*, 2016, pp. 399–409.

[19] K. Li and K. Deb, "Performance assessment for preference-based evolutionary multi-objective optimization using reference points," COIN Report, Tech. Rep., 2016.

[20] M. Wu, S. Kwong, Y. Jia, K. Li, and Q. Zhang, "Adaptive weights generation for decomposition-based multi-objective optimization using gaussian process regression," in *GECCO'17: Proc. of the 2017 Genetic and Evolutionary Computation Conference*, 2017, pp. 641–648.

[21] K. Li, K. Deb, O. T. Altinöz, and X. Yao, "Empirical investigations of reference point based methods when facing a massively large number of objectives: First results," in *EMO'17: Proc. of the 9th International Conference on Evolutionary Multi-Criterion Optimization*, 2017, pp. 390–405.

[22] M. Wu, K. Li, S. Kwong, Y. Zhou, and Q. Zhang, "Matching-based selection with incomplete lists for decomposition multiobjective optimization," *IEEE Trans. Evolutionary Computation*, vol. 21, no. 4, pp. 554–568, 2017.

[23] K. Li, K. Deb, Q. Zhang, and Q. Zhang, "Efficient nondomination level update method for steady-state evolutionary multiobjective optimization," *IEEE Trans. Cybernetics*, vol. 47, no. 9, pp. 2838–2849, 2017.

[24] R. Chen, K. Li, and X. Yao, "Dynamic multiobjectives optimization with a changing number of objectives," *IEEE Trans. Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, 2018.

[25] K. Li, K. Deb, and X. Yao, "R-metric: Evaluating the performance of preference-based evolutionary multi-objective optimization using reference points," *IEEE Trans. Evol. Comput.*, 2018, accepted for publication.

[26] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multi-objective optimization," *IEEE Trans. Evol. Comput.*, 2018, accepted for publication.

[27] M. Wu, K. Li, S. Kwong, Q. Zhang, and J. Zhang, "Learning to decompose: A paradigm for decomposition-based multi-objective optimization," *IEEE Trans. Evol. Comput.*, 2018, accepted for publication.

[28] K. Li, R. Chen, G. Min, and X. Yao, "Integration of preferences in decomposition-based evolutionary multi-objective optimization," *IEEE Trans. Cybernetics*, 2018, accepted for publication.

[29] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybernetics*, vol. 44, no. 1, pp. 40–53, 2014.

[30] Y. Wu, Y. Jin, and X. Liu, "A directed search strategy for evolutionary dynamic multiobjective optimization," *Soft Comput.*, vol. 19, no. 11, pp. 3221–3235, 2015.