Knee Points based Transfer Dynamic Multi-objective Optimization Evolutionary Algorithm

Zhenzhong WANG, Zhongrui Mei, Min JIANG*, *Senior Member, IEEE,* and Gary Yen, *Fellow, IEEE*

***Abstract*—When dynamic multi-objective optimization evolutionary algorithms (DMOEA) are used to solve real world problems, these are not only required to be able to find the Pareto-Optimal Set (POS) quickly, but also the results obtained can be easily used by decision makers. The classic DMOEAs have much room for improvement in both aspects. Recently, the transfer learning based DMOEAs have been proved that these methods can significantly improve the quality of the solution, but there are still too many individuals in the POS obtained by these algorithms. The resulting problems are twofold: this not only consumes a lot of computing resources to those solutions that will not be used, but also makes it more difficult for decision makers to choose. In this paper, we proposed a dynamic multi-objective optimization evolutionary algorithm which combines knee solutions with transfer learning method, and the feature of the proposed method is that it only outputs a very small number of solutions, which can greatly improve the efficiency of decision-making. The proposed algorithm divides the whole decision space into different subspaces, and find a local knee solutions in each subspace, then a transfer learning framework, Tr-DMOEA, is used to predict the knee solutions of the optimization problem at the next moment by using the local knee solutions and a global knee solution. The experimental results show the effectiveness of our design.**

***Index Terms*—Dynamic multi-objective optimization, Knee solutions, Transfer learning, Multi-criteria decision making**

I. INTRODUCTION

Many optimization problems in the real world [1]–[3] involve multiple optimization functions which conflict with each other and change over time. These dynamic optimization problems are called Dynamic Multi-objective Optimization Problems (DMOPs) [4], [5]. In recent years, evolutionary community researches have developed numerous efficient dynamic multi-objective optimization algorithms to find the optimal solutions quickly, and most existing algorithms can be classified into three categories: maintaining-diversity based [6]–[8], prediction based [9]–[15], and the memory-based methods [16]–[19]. Among these categories, the algorithm based on prediction has achieved good performance. For example, Tan *et al.* [20] proposed a prediction model based on the Kalman filter to minimize the influence of noise, and the Kalman filter is used to guide the search to generate a new initial population. Cao *et al.* [21] considered dynamic problems as time series problems, and a support vector regression was employed to construct a predictor. The authors [22]–[24] utilized linear models to predict the population in the next environment. Rong *et al.* [25] proposed a multi-directional prediction strategy to enhance the performance of evolutionary algorithms. Jiang *et al.* [26], [27] noticed the solution distributions are not identical under different environments, and a domain adaptation technique is incorporated to improve prediction accuracy. However, most existing algorithms often need to use output many individuals, which directly reduces the speed of solving, and it is difficult for decision makers (DMs) to use these results directly.

One reason why such DMOEAs are criticized is a large number of solutions are found, however, only one or a small number of solutions can be implemented for the real applications, so picking solutions for implementation among massive candidates is a mentally challenging burden to DMs, especially, DMs should make decision quickly according to the changing environment. For example, in the financial investment, the market environment is changing, and many investment schemes are obtained every moment. Due to the limitation of funds, only one or several schemes can be selected from a large number of schemes for implementation. However, schemes may involve many decision variables and objective variables, and selecting schemes will cost much of energy and time for DMs, and DMs may miss the best investment scheme.

From the above discussions, reducing the computational complexity of prediction based methods and relieving the cognitive burden to DMs remain great challenges in DMOPs. To address these issues, we propose a innovative approach combining both transfer learning technique and knee-based multi-criteria decision making (MCDM) strategy, called KT-MCDM. KT-MCDM identifies the global and local knee solutions to decision making using the Minimum Manhattan distance (MMD) [28], these knee solutions are provided to DMs to relieve the burden of picking solutions. Afterwards, once the environment changes, the knee solutions in the new environment are predicted to guide the population towards the optimums by using a transfer learning technique, called transfer component analysis (TCA) [29].

The contributions of this work are as follows: First, we only predict knee solutions in the new environment by exploiting the knee solutions from the past one environment. That is, instead of predicting a large number of solutions, we only predict a small number of high-quality solutions, knee

Z. Wang, Z. Mei, and M. Jiang are with the school of informatics, Xiamen University, China, Fujian, 361005. M. Jiang is the corresponding author: minjiang@xmu.edu.cn.

G. Yen is with the School of Electrical and Computer Engineering, Oklahoma State University.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

solutions, to guide the population towards the Pareto-optimal front, which can improve the computational efficiency. Second, the knee solutions are mostly preferred by DMs without any explicit preference. The knee-based MCDM approaches have rich geometric interpretations and avoid subjective preference inputs from DMs [30]. From a geometric perspective, the knee solutions benefit from significant improvement in some objectives at the cost of insignificant degradations in the other objectives. In this way, providing knee solutions in DMOPs can bring convenience and benefits to DMs. Third, although there are some research [31], [32] involved using knee points for prediction in DMOPs, but this is the first work that combines transfer learning with MCDM.

The rest of the paper is organized as follows: In Section II, we describes the basic concepts of DMOPs and presents the related transfer learning method used in the KT-MCDM. Section III gives the designed KT-MCDM in detail. In Section IV, experimental results and analysis are shown. Conclusions are drawn in Section V.

II. PRELIMINARY STUDIES AND RELATED RESEARCH

In this section, we briefly describes the definition of DMOPs, the MMD knee selection strategy, and the transfer learning technique used in our proposed algorithm.

A. DMOPs

The mathematical form of DMOPs is as follows:

$$\textbf{minimize } F\left(x,t\right)=\left\langle f_1\left(x,t\right),f_2\left(x,t\right),...,f_M\left(x,t\right)\right\rangle$$

$$s.t.\ x\in\Omega$$

where $x = \langle x_1, x_2, \ldots, x_N\rangle$ is the decision vector and t is the time or environment variable. $f_m\left(x,t\right) : \Omega \rightarrow \mathbb{R}\ (m = 1, \ldots, M)$. The aim of solving DMOPs is to search for the set of solutions at different times or environments, so that all the objectives are as minimal as possible.

Definition 1: **(Dynamic Decision Vector Domination)** At environment t, a decision vector x_1 Pareto-dominates another vector x_2 denoted by $x_1 \succ_t x_2$, if and only if

$$\begin{cases}\forall m=1,...,M, & f_m(x_1,t)\le f_m(x_2,t)\\ \exists m=1,...,M, & f_m(x_1,t)< f_m(x_2,t)\end{cases} \tag{1}$$

Definition 2: **(Dynamic Pareto-Optimal Set, DPOS)** If a decision vector x^* at environment t satisfies

$$DPOS_t=\left\{x^*|\nexists x,x\succ_t x^*\right\}, \tag{2}$$

then x^* is called dynamic Pareto-optimal solution, and the set of dynamic Pareto-optimal solutions is called the dynamic POS (DPOS).

Definition 3: **(Dynamic Pareto-optimal Front)** At environment t, the Dynamic Pareto-optimal Front (DPOF) is the corresponding objective vectors of the DPOS:

$$DPOF_t=\{F(x^*,t)|x^*\in DPOS_t\}.$$

B. Identity Global Knee Solution by MMD

There are some advantages to MMD knee solution selection strategy [30], [33]. Firstly, the MMD approach can be adapted to various types of POF without geometrical limitations, such as linear, non-linear, concave, convex, disconnected or degenerated geometries. This makes MMD selection can combine the advantages of both optimization performance analysis and geometrical information. Secondly, the MMD selection is time-saving which is advantageous for those problems that need to be quickly responded to. Therefore, in this paper, we choose the MMD approach as the baseline method to identity knee solutions.

The primary idea of MMD is to calculate Manhattan Distance for each non-dominated solution under different objective space and select the solution with the minimum distance as the global knee solution. The procedure of the MMD approach is given in Algorithm 1.

Algorithm 1: MMD Knee Selection

Input: Non-dominated Set NDS, the optimization problem $F(x) =< f_1(x), \ldots f_M(x) >$
Output: Global Knee Solution $knee$
$dis(x) = 0, x \in NDS$;
for $m = 1$ *to* M **do**
  Identify minimum value f_m^{min} in objective m from NDS;
  Identify maximum value f_m^{max} in objective m from NDS;
  for $x \in NDS$ **do**
    $dis(x) = dis(x) + \frac{f_m(x)-f_m^{min}}{f_m^{max}-f_m^{min}}$;
  end
end
Select the global knee solution $knee$ with minimum $dis(x)$;
return $knee$;

C. Tr-DMOEA

Jiang *et al.* [26] pointed out that the solution distributions of different environments are not identical, and when the distribution of training samples and prediction samples does not satisfy identical distribution, the prediction methods based on transfer learning tend to be more effective and more promising [34]–[36]. Therefore, a transfer learning based dynamic multi-objective optimization framework is proposed, called Tr-DMOEA. The transfer learning method used in the Tr-DMOEA framework is TCA.

For the domain adaptation problem, TCA can well deal with the situation that the source domain and the target domain meet different data distributions. The main task of TCA is to map the source domain X_{so} and the target domain X_{ta} to a latent space through the mapping function $\phi(\cdot)$, so that the data distribution in the two domains in the space as similar as possible, which is $P(\phi(X_{so})) \approx P(\phi(X_{ta}))$. Then the data in the two domains in the space is further processed.

The Tr-DMOEA algorithm can be divided into the following major steps: Firstly, an initial population is randomly generated at the very beginning, when the environment does not change, it is considered as a static multi-objective optimization problem, and a static multi-objective algorithm (SMOA) used to solve it. Afterwards, when the change in the environment is detected, the initial population at the next environment is predicted using the transfer learning method combined with the optimal solution information obtained at the previous environment, Figure 1 is a schematic diagram. Then, a SMOA is used to solve the new environment of the problem based on the predicted initial population. Repeat the last two steps until the end of the DMOP, and output the obtained optimal solution at each environment.

In the Figure 1, Step I: Construct the latent space by adapting to the solution distributions of two environments. Step II: Map POF_{t-1} into the latent space to get mapped solutions PLS through the mapping function. Step III: Find the solution x in the new environment t such that the mapped solution $\phi(F_t(x))$ in the latent space is close to $l \in PLS$, and the interior point algorithm can be used to solve this problem. The found solution x is added into the initial population to guide the evolutionary process. The detailed steps of predicting the initial population in the Tr-DMOEA are shown in the Algorithm 2.

The Tr-DMOEA combines transfer learning with evolutionary algorithms to improve the SMOA in solving the DMOPs, it retains the advantages of the evolutionary algorithm itself.

Algorithm 2: Tr-IPG [26]

Input: The Dynamic optimization function $F_t(\cdot)$ and $F_{t-1}(\cdot)$, the POF_{t-1} at $F_{t-1}(\cdot)$.
Output: The predicted initial population ip at $F_t(\cdot)$.

$PLS = \varnothing$;
Set the kernel function κ;
For $F_t(\cdot)$ and $F_{t-1}(\cdot)$, randomly generate two sets of the solutions X_{ta} and X_{so};
Calculate the objectives of optimization functions $F_t(X_{ta})$ and $F_{t-1}(X_{so})$;
$W =$ **TCA**$(F_t(X_{ta}), F_{t-1}(X_{so}), \kappa)$;
for *every* $p \in POF_t$ **do**
  $\kappa_p = [\kappa(F_{t-1}(X_{so}(1)), p), \ldots, \kappa(F_t(X_{ta}(|X_{ta}|)), p)]$;
  $\phi(p) = W^T \kappa_p$;
  $PLS = PLS \cup \{\phi(p)\}$;
end
for *every* $l \in PLS$ **do**
  $x = \arg\min_x ||\phi(F_t(x)) - l||$;
  $ip = ip \cup \{x\}$;
end
return ip;

III. Proposed Algorithm

A. Framework

The framework of KT-MCDM is illustrated in Algorithm 3. At the first beginning, MOEA/D [37] solves the initial environmental function F_0 and the knee solutions are identified from the non-dominated set POS_0 by using **KneeSelection**. When environment changes, the **Tr-IPG** utilizes the the objectives o_{t-1} of knee solutions $knee_{t-1}$ at the past one environment $t-1$ to generate a predicted knee points $predictedKnee_t$ for guiding the search towards the true POF, and the knee solutions are identified from POS_t for the next environmental prediction. The details of KT-MCDM are presented in the following section.

Algorithm 3: KT-MCDM

Input: The Dynamic optimization function $F_t(\cdot)$, the parameter of subspaces p.
Output: the knee solutions $knee_t$ and objectives o_t of $knee_t$ of the $F_t(\cdot)$.

Initialization;
Use **MOEA/D** to solve $F_0(\cdot)$ to get POS_0;
$knee_0 =$ **KneeSelection**(POS_0, p);
Compute objectives o_0 of $knee_0$;
while *the environment has changed* **do**
  $t = t + 1$;
  $predictedKnee_t =$ **Tr-IPG**$(F_t(\cdot), F_{t-1}(\cdot), o_{t-1})$;
  $POS_t =$ **MOEA/D**$(F_t(\cdot), predictedKnee_t)$;
  $knee_t =$ **KneeSelection**(POS_t, p);
  Compute objectives o_t of $knee_t$;
  return $knee_t$ and o_t;
end

B. Knee Selection

DMs often desire more knee solutions in addition to the global knee to handle the real-world applications. To address this issue, the objective space in each dimension is divided into several equally sized overlapped subspaces [30]. In each subspace, a local knee solution is identified. The size of each subspace of objective m at environment t is defined as

$$size_{t,m} = \frac{f_{t,m}^{max} - f_{t,m}^{min}}{p}. \quad (3)$$

where $f_{t,m}^{max}$ and $f_{t,m}^{min}$ mean the maximum and minimum value f_t in objective m, respectively. The lower bound of i-th subspace is

$$lowerbound_{t,m,i} = f_{t,m}^{min} + i \times size_{t,m}/2, \quad (4)$$

and the upper bound is calculated according to the following formula:

$$upperbound_{t,m,i} = lowerbound_{t,m,i} + size_{t,m}, \quad (5)$$

where $i = 0, \ldots, 2 \times p - 2$. p determines the number of divisions.

Once we have determined each subspace, the local knee solution of each subspace is identified by MMD approach. For the whole objective space, we also use MMD approach to directly search for the global knee solution. The knee selection process is given in Algorithm 4.

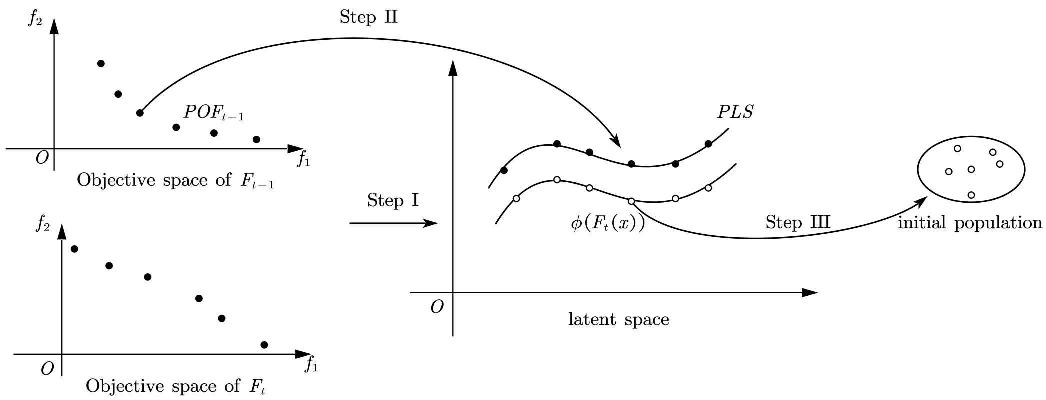

Fig. 1. Predicting the initial population using in Tr-DMOEA: Step I, construct the latent space by using environmental information in F_{t-1} and F_t. Step II, solutions in the F_{t-1} are mapped into the latent space. Step III, generate the initial population.

Algorithm 4: KneeSelection

Input: Non-dominated Set NDS, the parameter of subspaces p.

Output: The knee solutions set $knee$.

```
knee = ∅;
Determine all subspaces according to Formula (4) and (5);
for each subspace do
    Identify minimum value f_{t,m}^{min} in objective m from NDS;
    Identify maximum value f_{t,m}^{max} in objective m from NDS;
    for x ∈ NDS do
        dis(x) = dis(x) + (f_{t,m}(x) − f_{t,m}^{min}) / (f_{t,m}^{max} − f_{t,m}^{min});
    end
    Select the local knee solution with minimum dis(x);
    knee = knee ∪ {x};
end
Identify the global knee x by MMD approach;
knee = knee ∪ {x};
return knee;
```

C. Transfer Knee Solutions

The distribution of the knee solutions may vary at different environments, so it is promising to use the transfer learning technique to predict knee solutions in the new environment. However, from the previous study, using TCA to predict the initial population is very time-consuming, this is a disadvantage to solve applications involved DMOPs. To overcome this disadvantage, in this paper, we consider only to predict a small number of high-quality solutions to reduces the consumption of computing resources.

The transfer method used in the paper is similar to Tr-DMOEA. However, in this paper, we only transfer several high-quality solutions, local knee solutions and the global knee solution for initializing population. Transferring knee solutions rather than the a large number of solutions greatly reduces the computation time in the transfer prediction, but has little negative impact on the population quality. The improvement of transfer prediction speed is significant in solving DMOPs.

In the following, we use Algorithm 2 to briefly describe the process of transferring the knee solutions. Firstly, in Line 3 and Line 4 of Algorithm 2, the solutions at environment $t-1$ and t are sampled. In Line 5, the TCA is used for domain adaptation, so that we can use the matrix W outputted by TCA to construct the latent space. In this latent space, mapped solution distributions of different environment are similar. Then, in Line 6∼10, knee solutions from the $t-1$ environment are mapped in this latent space, the mapped knee solutions set is denoted as PLS. Finally, in Line 11∼14, we find the predicted knee solution x, such that in the latent space, $\phi(F_t(x))$ is closet to $l \in PLS$ in the latent space.

IV. EXPERIMENTS

A. Algorithms Compared, Test Problems, Performance Indicators and Settings

The proposed KT-MCDM algorithm is compared against several popular dynamic MOEAs including MDP [25], MOEA/D-SVR [21], MOEA/D-KF [20], and PPS [24]. For a fair comparison, most parameters of these algorithms are set according to the original references, and the SMOA used in PPS and MDP are replaced by the baseline algorithm MOEA/D, and these two compared algorithms are denoted as MOEA/D-PPS, MOEA/D-MDP, respectively.

All compared algorithms are evaluated based on 14 DMOPs (DF1-DF14) selected from CEC 2018 DMO benchmarks [38]. The dynamics of a DMOP is governed by

$$t = \frac{1}{n_t}\left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \tag{6}$$

where τ, n_t, and τ_t refer to the maximum generation counter, severity of change, and frequency of change, respectively.

In this study, the following metrics are utilized to evaluate the performance of algorithms.

1) Inverted Generational Distance (IGD): The IGD is a metric for quantifying the convergence of the solutions obtained by a multi-objective optimization algorithm. When the IGD

value is small, the convergence of the solution is improved. IGD is defined as

$$IGD(POF^*, POF) = \frac{\sum_{p^* \in POF^*} \min_{p \in POF} \|p^* - p\|^2}{|POF^*|}, \quad (7)$$

where POF^* is the true POF of a multi-objective optimization problem, POF is an approximation set of POF obtained by a multi-objective optimization algorithm.

The MIGD metric is a variant of IGD and is defined as the average of the IGD values in some time steps over a run.

$$MIGD(POF^*, POF) = \frac{\sum_{t \in T} IGD(POF_t^*, POF_t)}{|T|}, \quad (8)$$

where T is a set of discrete time points in a run and $|T|$ is the cardinality of T.

2) KneeDist [30]: The Euclidean distance $KneeDist(knee^*, knee)$ between the obtained knee solutions $knee$ and the true knee solutions $knee^*$ in measure. This metric is similarly modified to the IGD to act as a performance metric for evaluating dynamic MOEAs.

$$MeanKD(knee^*, knee) = \frac{\sum_{t \in T} KneeDist(knee_t^*, knee_t)}{|T|} \quad (9)$$

It should be noted that we chose the solution obtained who has the minimum distance to the true global knee solution as the final obtained knee solution for comparison.

In the experiments, the dimension of decision variables is set to 10, and the population size is set to 100 for bi-objective optimization problems and 150 for tri-objective optimization problems. The severity of the change n_t is set to 5 and 10, while the frequency of change τ_t is kept constant at 10. The τ is set to $20 \times \tau_t$. In our proposed KT-MCDM, the parameter of subspaces p is set to 4. The parameters of TCA are set according to [26]. All the problems are run 10 times independently.

B. Performance on DF Problems

As can be seen from Table I, KT-MCDM achieves 16 out of 28 best results, MOEA/D-MDP has 9 best results, MOEA/D-PPS has 1 best results and MOEA/D-SVR achieves 2 best results. Specifically, KT-MCDM performs better on DF2, DF6, DF7, DF10, DF11, and DF14 under all dynamic test settings. On DF3, DF5, and DF8 problems with setting $(\tau_t = 10, n_t = 5)$, and DF13 problem with setting $(\tau_t = 10, n_t = 10)$, KT-MCDM obtains well-converged solutions. In other cases, KT-MCDM falls behind the corresponding best-performing algorithms.

However, for some benchmark functions, such as DF9 and DF12, which have a time-varying number of POF holes, KT-MCDM leaves some room to be desired, and the reason may be the time-varying POF holes lead to the knee solutions distribution changes dramatically, which lead the Algorithm **Tr-IPG** to the poor prediction performance.

Table II shows the MeanKD values of the five compared algorithms. KT-MCDM performs the best on 14 out of the 28 problems, which is followed by MOEA/D-MDP gaining 8 best results, MOEA/D-SVR and MOEA/D-PPS gaining 3 and 2 best results, respectively, and MOEA/D-KF only gaining 1 best result. To be specific, KT-MCDM performs better on DF2, DF5, DF7, DF8, DF11, and DF14 under all dynamic test settings. On DF6 and DF13 with setting $(\tau_t = 10, n_t = 10)$, KT-MCDM performs better than other compared algorithms.

The feature of the proposed method can output some knee solutions, which can greatly improve the efficiency of decision-making. From the MeanKD values shown in Table II, in most of cases, the KT-MCDM provides the knee solution which is close to the true global knee solution. It means our proposed KT-MCDM is better for DMs to make decisions. The experimental results validate the effectiveness of our design.

V. CONCLUSION

Researchers have essentially focused on searching the well-converged and well-diversified solutions quickly when solving DMOPs. However, these algorithms often require large populations and output POS with large scales. If the size of POS obtained by the algorithm is too large, it may lead to two problems. First, a large amount of computing resources are spent to find those solutions that will not be used at all, and secondly, it also greatly increases the difficulty of decision-making.

In this paper, we propose a dynamic multi-objective optimization evolutionary algoritm, called KT-MCDM. The proposed algorithm combines the knee points strategy and the transfer learning method to address the above issues. In the KT-MCDM, we only predict knee solutions in the new environment by exploiting the knee solutions (local and global knee points) from the past to improve the computational efficiency, and the identified knee solutions are provided to DMs to relieve the difficulty of decision-making. The experiments validate that the proposed KT-MCDM can obtain knee solutions which are close to the true knee solutions on most test functions. In our future work, we will continue to explore how to improve the computational efficiency and relieve the burden of picking solutions in solving DMOPs.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No.61673328).

REFERENCES

[1] C. Raquel and X. Yao, "Dynamic multi-objective optimization: a survey of the state-of-the-art," in *Studies in Computational Intelligence*. Springer Science mathplus Business Media, 2013, pp. 85–106.

[2] M. Jiang, Z. Huang, G. Jiang, M. Shi, and X. Zeng, "Motion generation of multi-legged robot in complex terrains by using estimation of distribution algorithm," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–6.

[3] Min Jiang, Yang Yu, Xiaoli Liu, Fan Zhang, and Qingyang Hong, "Fuzzy neural network based dynamic path planning," in *2012 International Conference on Machine Learning and Cybernetics*, vol. 1, July 2012, pp. 326–330.

[4] Y. Yang, Y. Sun, and Z. Zhu, "Multi-objective memetic algorithm based on request prediction for dynamic pickup-and-delivery problems," in *Evolutionary Computation*, 2017.

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF MIGD METRIC OBTAINED BY COMPETING ALGORITHMS FOR DIFFERENT DYNAMIC TEST FUNCTIONS UNDER VARIOUS TEST SETTINGS

Problems	τ_t, n_t	KT-MCDM	MOEA/D-PPS	MOEA/D-SVR	MOEA/D-MDP	MOEA/D-KF
DF1	10,10	0.1413±7.790e-02	0.1133±1.345e-01	0.1089±1.006e-01	**0.0777±5.079e-02**	0.1749±1.059e-01
	10,5	0.1023±4.562e-02	0.1190±7.716e-02	0.0933±5.695e-02	**0.0890±3.515e-02**	0.1947±9.592e-02
DF2	10,10	**0.0806±8.121e-02**	0.0916±5.994e-02	0.0897±5.537e-02	0.1242±7.592e-02	0.0966±2.430e-02
	10,5	**0.0883±4.726e-02**	0.0933±6.158e-02	0.0964±4.758e-02	0.1369±6.294e-02	0.1066±8.752e-02
DF3	10,10	0.4079±1.823e-01	0.5052±2.217e-01	0.4401±1.699e-01	**0.2372±9.923e-02**	0.3648±1.489e-01
	10,5	**0.3475±2.071e-01**	0.3991±2.494e-01	0.3662±1.733e-01	0.3861±1.041e-01	0.3669±1.406e-01
DF4	10,10	1.5455±4.383e-01	1.5114±2.936e-01	1.5401±5.058e-01	**1.3368±3.015e-01**	2.0044±8.663e-01
	10,5	1.0788±6.346e-01	1.0427±7.003e-01	1.0003±6.831e-01	**0.8965±5.486e-01**	1.3369±9.160e-01
DF5	10,10	0.1463±1.261e-01	**0.0826±1.040e-01**	0.7430±5.357e-01	0.1051±5.300e-02	0.1108±7.386e-02
	10,5	**1.7480±2.754e+00**	1.7532±2.281e+00	6.2743±7.292e+00	1.8374±2.313e+00	1.8212±2.263e+00
DF6	10,10	**1.5125±1.769e+00**	2.2158±3.649e+00	1.7620±2.011e+00	2.8952±3.503e+00	1.6657±1.523e+00
	10,5	**1.6575±1.815e+00**	4.2490±6.414e+00	1.9151±2.492e+00	2.7470±3.417e+00	1.7962±1.831e+00
DF7	10,10	**2.0330±3.750e+00**	2.6881±3.076e+00	2.3395±3.274e+00	2.4637±2.848e+00	2.6119±2.522e+00
	10,5	**1.7954±3.053e+00**	2.6055±4.558e+00	2.1534±2.854e+00	1.8325±2.107e+00	2.5606±3.029e+00
DF8	10,10	0.9319±4.485e-01	1.1585±4.979e-01	**0.9060±3.795e-01**	0.9724±3.799e-01	1.0552±4.172e-01
	10,5	**0.8577±4.402e-01**	1.1705±5.928e-01	0.9389±4.534e-01	0.9311±3.610e-01	1.0896±5.133e-01
DF9	10,10	1.5748±2.434e+00	1.5269±1.550e+00	3.1623±3.229e+00	**1.4790±1.504e+00**	1.8594±1.662e+00
	10,5	2.0647±2.123e+00	1.2717±1.249e+00	2.9173±3.111e+00	**0.9844±9.726e-01**	1.4211±1.200e+00
DF10	10,10	**0.1577±6.581e-02**	0.1822±6.925e-02	11.7191±4.272e+01	0.1724±5.037e-02	0.1840±5.094e-02
	10,5	**0.1853±8.699e-02**	0.2123±8.827e-02	26.9726±8.487e+01	0.2135±4.828e-02	0.2184±8.453e-02
DF11	10,10	**0.1462±2.317e-02**	0.2302±2.090e-01	39.0376±1.109e+02	0.1568±3.818e-02	0.1685±4.055e-02
	10,5	**0.1471±1.713e-02**	0.2964±1.674e-01	22.2910±4.308e+01	0.2167±9.135e-02	0.2530±9.064e-02
DF12	10,10	1.1025±1.214e-01	1.1716±8.978e-02	53.7160±1.544e+02	**0.7030±1.271e-01**	1.1045±1.404e-01
	10,5	1.0410±1.969e-01	1.1588±1.482e-01	19.4573±7.859e+01	**0.6774±1.348e-01**	1.0629±2.016e-01
DF13	10,10	**0.3039±3.506e-02**	0.4235±1.577e-01	0.3165±4.666e-02	0.3464±1.127e-01	0.3370±5.855e-02
	10,5	1.7518±2.209e+00	1.7820±2.032e+00	**1.5097±1.949e+00**	1.7945±2.200e+00	1.7766±2.086e+00
DF14	10,10	**0.0885±1.418e-02**	0.1031±2.893e-02	0.4840±2.445e-01	0.2005±8.881e-02	0.0893±9.118e-03
	10,5	**1.1213±1.598e+00**	1.1617±1.462e+00	4.2882±5.627e+00	1.3100±1.614e+00	1.2192±1.551e+00

[5] T. T. Nguyen, S. Yang, and J. Branke, "Evolutionary dynamic optimization: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 6, pp. 1–24, oct 2012.

[6] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2017.

[7] Y. Jin, C. Yang, J. Ding, and T. Chai, "Reference point based prediction for evolutionary dynamic multiobjective optimization," in *Evolutionary Computation*, 2016.

[8] S. Zeng, R. Jiao, C. Li, X. Li, and J. S. Alkasassbeh, "A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization," *IEEE transactions on cybernetics*, vol. 47, no. 9, pp. 2678–2688, 2017.

[9] G. Ruan, G. Yu, J. Zheng, J. Zou, and S. Yang, "The effect of diversity maintenance on prediction in dynamic multi-objective optimization," *Applied Soft Computing*, vol. 58, pp. 631–647, 2017.

[10] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, 2019.

[11] Z. Wang, M. Jiang, X. Gao, L. Feng, W. Hu, and K. C. Tan, "Evolutionary dynamic multi-objective optimization via regression transfer learning," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 2375–2381.

[12] M. Rong, D. Gong, W. Pedrycz, and L. Wang, "A multi-model prediction method for dynamic multi-objective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019.

[13] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, "A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization," *IEEE transactions on cybernetics*, 2019.

[14] M. Jiang, W. Hu, L. Qiu, M. Shi, and K. C. Tan, "Solving dynamic multi-objective optimization problems via support vector machine," in *2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, March 2018, pp. 819–824.

[15] W. Hu, M. Jiang, X. Gao, K. C. Tan, and Y. Cheung, "Solving dynamic multi-objective optimization problems using incremental support vector machine," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, June 2019, pp. 2794–2799.

[16] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. C. Tan, "A fast dynamic evolutionary multi-objective algorithm via manifold transfer learning," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–12, 2020.

[17] R. Azzouz, S. Bechikh, and L. B. Said, "A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy," *Soft Computing*, vol. 21, no. 4, pp. 885–906, 2017.

[18] C. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, Feb 2009.

[19] K. Deb, B. R. N. Udaya, and S. Karthik, *Dynamic Multi-objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-thermal Power Scheduling*, 2007.

[20] A. Muruganantham, K. Tan, and P. Vadakkepat, "Evolutionary Dynamic Multiobjective Optimization Via Kalman Filter Prediction," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2862–2873, Dec 2016.

[21] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, "Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019.

[22] Y. Ma, R. Liu, and R. Shang, "A hybrid dynamic multi-objective immune optimization algorithm using prediction strategy and improved differential evolution crossover operator," in *Neural Information Processing*, B.-L. Lu, L. Zhang, and J. Kwok, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 435–444.

[23] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based

TABLE II
MEAN AND STANDARD DEVIATION VALUES OF MEANKD METRIC OBTAINED BY COMPETING ALGORITHMS FOR DIFFERENT DYNAMIC TEST FUNCTIONS UNDER VARIOUS TEST SETTINGS

Problems	τ_t, n_t	KT-MCDM	MOEA/D-PPS	MOEA/D-SVR	MOEA/D-MDP	MOEA/D-KF
DF1	10,10	0.1059±8.145e-02	0.0938±1.173e-01	0.0931±1.017e-01	**0.0660±5.048e-02**	0.0780±7.669e-02
	10,5	0.0837±4.374e-02	0.0995±7.371e-02	0.0910±7.309e-02	**0.0602±5.006e-02**	0.1312±9.660e-02
DF2	10,10	**0.1384±7.476e-02**	0.1586±1.259e-01	0.1981±1.382e-01	0.2002±2.218e-01	0.2238±1.558e-01
	10,5	**0.1116±7.117e-02**	0.1655±1.039e-01	0.1907±1.744e-01	0.2458±1.944e-01	0.2189±2.793e-01
DF3	10,10	0.3144±2.482e-01	0.4044±2.926e-01	0.3548±2.151e-01	**0.1324±1.172e-01**	0.2529±2.225e-01
	10,5	0.3897±2.818e-01	0.4241±2.384e-01	0.4059±2.594e-01	**0.2149±1.671e-01**	0.3228±1.920e-01
DF4	10,10	0.3336±2.695e-01	0.4558±4.051e-01	**0.2702±3.281e-01**	0.3179±1.282e-01	1.0209±2.622e+00
	10,5	0.2891±2.243e-01	0.4588±3.573e-01	**0.2521±2.568e-01**	0.4270±3.848e-01	1.4723±2.938e+00
DF5	10,10	**0.1374±1.343e-01**	0.4681±9.090e-02	1.1703±6.307e-01	0.1652±1.572e-01	0.1587±2.414e-01
	10,5	**1.8224±2.726e+00**	1.8331±2.290e+00	7.6944±8.660e+00	1.8419±2.339e+00	1.9249±2.260e+00
DF6	10,10	**1.4431±1.741e+00**	2.1528±3.627e+00	1.8447±1.942e+00	2.8545±3.495e+00	1.8028±1.545e+00
	10,5	1.8655±1.819e+00	4.1562±6.415e+00	2.4231±4.398e+00	2.7109±3.336e+00	**1.8191±1.735e+00**
DF7	10,10	**2.3495±3.797e+00**	2.6336±3.123e+00	2.4053±3.120e+00	2.5737±2.818e+00	2.6194±2.477e+00
	10,5	**2.3834±4.215e+00**	2.4320±4.527e+00	9.6156±4.627e+00	2.7145±2.129e+00	5.9084±1.478e+01
DF8	10,10	**4.7095±2.536e+00**	5.1008±2.718e+00	5.7459±2.359e+00	4.9001±2.429e+00	4.7165±2.473e+00
	10,5	**4.7055±2.713e+00**	4.8778±2.859e+00	5.7869±2.845e+00	4.9213±2.573e+00	4.7324±2.466e+00
DF9	10,10	4.5496±4.035e+00	**2.1128±1.777e+00**	4.0874±3.207e+00	2.8108±3.163e+00	3.1715±3.130e+00
	10,5	4.2154±3.701e+00	**1.4345±1.607e+00**	43.4790±1.655e+02	2.1325±2.563e+00	2.1743±2.292e+00
DF10	10,10	0.1780±2.213e-01	0.2193±3.077e-01	7.6554±3.396e+01	**0.0224±4.442e-02**	0.4474±4.298e-01
	10,5	0.2235±1.714e-01	0.1840±1.553e-01	10.1559±8.707e+01	**0.0491±1.171e-01**	0.4063±4.174e-01
DF11	10,10	**0.1491±9.932e-02**	0.1975±2.713e-01	39.2379±1.106e+02	0.1737±6.692e-02	0.2677±2.920e-01
	10,5	**0.1996±1.318e-01**	0.3501±3.551e-01	22.4750±4.284e+01	0.2422±1.283e-01	0.3293±2.292e-01
DF12	10,10	2.4856±4.308e-01	2.6110±3.662e-01	54.8702±1.541e+02	**1.6192±5.536e-01**	2.5262±4.091e-01
	10,5	2.3031±4.971e-01	2.5711±3.888e-01	20.5929±7.865e+01	**1.6543±5.408e-01**	2.4235±4.890e-01
DF13	10,10	**0.4140±1.855e-01**	0.4536±1.351e-01	0.4353±1.957e-01	0.4917±1.821e-01	0.4705±1.852e-01
	10,5	1.7207±2.146e+00	1.7204±1.954e+00	**1.4885±1.912e+00**	1.7919±2.131e+00	1.8099±2.000e+00
DF14	10,10	**0.1261±5.780e-02**	0.1406±7.507e-02	0.1634±1.489e-01	0.1769±1.827e-01	0.1567±1.209e-01
	10,5	**1.1992±1.643e+00**	1.2388±1.553e+00	1.9629±4.986e+00	1.4268±1.801e+00	1.3428±1.604e+00

population re-initialization for evolutionary dynamic multi-objective optimization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 832–846.

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

[25] M. Rong, D. Gong, Y. Zhang, Y. Jin, and W. Pedrycz, "Multidirectional prediction approach for dynamic multiobjective optimization problems," *IEEE Transactions on Cybernetics*, pp. 1–13, 2018.

[26] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. Yen, "Transfer learning based dynamic multiobjective optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–1, 2017.

[27] M. Jiang, L. Qiu, Z. Huang, and G. G. Yen, "Dynamic multi-objective estimation of distribution algorithm based on domain adaptation and nonparametric estimation," *Information Sciences*, vol. 435, pp. 203 – 223, 2018.

[28] W. Chiu, G. G. Yen, and T. Juan, "Minimum manhattan distance approach to multiple criteria decision making in multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 972–985, Dec 2016.

[29] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[30] K. Zhang, G. G. Yen, and Z. He, "Evolutionary algorithm for knee-based multiple criteria decision making," *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.

[31] Q. Li, J. Zou, S. Yang, J. Zheng, and R. Gan, "A predictive strategy based on special points for evolutionary dynamic multi-objective optimization," *Soft Computing*, no. 23, p. 3723–3739, 2019.

[32] F. Zou, G. G. Yen, and L. Tang, "A knee-guided prediction approach for dynamic multi-objective optimization," *Information Sciences*, vol. 509, pp. 193 – 209, 2020.

[33] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, Dec 2015.

[34] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[35] M. Jiang, W. Huang, Z. Huang, and G. G. Yen, "Integration of global and local metrics for domain adaptation learning via dimensionality reduction," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 38–51, Jan 2017.

[36] G. Chi, M. Jiang, X. Gao, W. Hu, S. Guo, and K. C. Tan, "Online bagging for anytime transfer learning," in *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2019, pp. 941–947.

[37] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[38] S. Jiang, S. Yang, X. Yao, K. TAN, M. Kaiser, and N. Krasnogor, "Benchmark problems for cec2018 competition on dynamic multiobjective optimisation," *CEC2018 Competition*, 2018.