Surrogate Assisted Evolutionary Algorithm Based on Transfer Learning for Dynamic Expensive Multi-Objective Optimisation Problems

Xuezhou Fan†, Ke Li‡, Kay Chen Tan†
†*Department of Computer Science, City University of Hong Kong*
Tat Chee Avenue, Hong Kong SAR, China
xuezhofan2-c@my.cityu.edu.hk, kaytan@cityu.edu.hk
‡*Department of Computer Science, University of Exeter*
EX4 4QF, Exeter, UK
k.li@exeter.ac.uk

***Abstract*—Dynamic multi-objective optimisation has attracted increasing attention in the evolutionary multi-objective optimisation community in recent years. Comparing to its static counterpart, which has been studied for more than half a century, the involvement of dynamic and uncertain features, including but not limited to the changing Pareto-optimal set, Pareto-optimal front and problem formulation, pose significant more challenges to evolutionary algorithms. This will become even more complicated when the underlying problem involves computationally expensive objective functions which are not rare in many real-world application scenarios. In this paper, we pave an initial step towards the study of dynamic multi-objective optimisation with computationally expensive objective functions. More specifically, we use a surrogate assisted evolutionary algorithm, MOEA/D-EGO in particular, as the baseline in order to carry out evolutionary optimisation with a limited amount of function evaluations. Furthermore, instead of restart the MOEA/D-EGO from scratch after each change, we use transfer learning to map the previously archived training data to the current landscape in order to jump start the surrogate model building process. By doing so, we can expect a better adaptation to the new environment. Proof-of-concept experiments fully demonstrate the effectiveness of our proposed method.**

***Index Terms*—Dynamic multi-objective optimization, evolutionary algorithm, surrogate modeling, Gaussian process, transfer learning**

I. Introduction

Multi-objective optimisation problems (MOPs) are ubiquitous in many real-world applications, such as engineering [1], biology [2] and economics [3]. Comparing to predefined and static MOPs, it is not uncommon and unarguably more challenging to involve time varying features, including but not limited to decision variables, objective functions and constraints, termed as dynamic multi-objective optimisation problems (DMOPs). Without loss of generality, the DMOP considered in this paper is formulated as:

$$\begin{aligned} &\text{minimise} \quad \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x},t),\cdots,f_m(\mathbf{x},t))^T \\ &\text{subject to} \quad \mathbf{x}\in\Omega, t\in\Omega^t \end{aligned}, \tag{1}$$

where t is a discrete time step defined as $t=\lfloor(\tau/\tau_t)\rfloor$, τ and τ_t represent the iteration counter and the frequency of change, respectively, and $\Omega^t\in\mathbb{N}$ is the time space. $\Omega=\Pi_{i=1}^n[a_i,b_i]\subseteq\mathbb{R}^n$ is the decision (variable) space, $\mathbf{x}=(x_1,\cdots,x_n)^T\in\Omega$ is a candidate solution. $\mathbf{F}:\Omega\rightarrow\mathbb{R}^m$ consists of m conflicting objective functions and $\mathbb{R}^m$ is the objective space. Given the dynamic characteristics, the objective functions $f_i(\mathbf{x},t)$, $i\in\{1,\cdots,m\}$, are changing with time step t.

Given two candidate solutions $\mathbf{x}^1$ and $\mathbf{x}^2$ of the DMOP defined in (1), $\mathbf{x}^1$ is said to dominate $\mathbf{x}^2$ at the time step t (denoted as $\mathbf{x}^1\preceq_t\mathbf{x}^2$) if and only if $f_i(\mathbf{x}^1,t)\leq f_i(\mathbf{x}^2,t)$ for all $i\in\{1,\cdots,m\}$ and there exists at least one objective $j\in\{1,\cdots,m\}$ such that $f_j(\mathbf{x}^1,t)<f_j(\mathbf{x}^2,t)$. A solution $\mathbf{x}^*\in\Omega$ is called Pareto-optimal at the time step t in case there does not exist another solution $\mathbf{x}\in\Omega$ such that $\mathbf{x}\preceq_t\mathbf{x}^*$. All Pareto-optimal solutions at the time step t constitute the t-th Pareto-optimal set (PS_t) whilst its image in the objective space is the t-th Pareto-optimal front (PF_t). In [4], Farina and Deb for the first time formalised the benchmark test problems for dynamic multi-objective optimisation and classified DMOPs into four categorises according to the dynamics of PS and PF: 1) the PS changes over time but the PF is fixed; 2) both the PS and PF change over time; 3) the PS is fixed whereas the PF changes over time; and 4) both the PS and PF are fixed, but the problem formulation changes over time.

Due to the population-based characteristics, evolutionary algorithms (EAs) have been widely accepted as a major approach for multi-objective optimisation. However, with the presence of aforementioned dynamic features, DMOPs bring significantly more challenges to traditional evolutionary multi-objective optimisation (EMO) algorithms for tracking moving PSs and/or PFs. In addition, one of the major obstacles that prevents the widespread use of EAs in the real world is their intensive

This work is partially supported by the National Natural Science Foundation of China (NSFC) under grant No. 61876162, by the Shenzhen Scientific Research and Development Funding Program under grant JCYJ20180307123637294, and by the Research Grants Council of the Hong Kong SAR under grant No. CityU11202418 and CityU11209219. K. Li was supported by UKRI Future Leaders Fellowship (Grant No. MR/S017062/1) and Royal Society (Grant No. IEC/NSFC/170243).

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

requirement of function evaluations which were assumed to be non-trivial. On the contrary, evaluating a objective function in many real-world scenarios is the most *expensive* part as it either involves computationally expensive numerical simulations or costly physical experiments. For example, computational fluid dynamic (CFD) simulations usually take from minutes to hours to run one function evaluation [5]. In this sense, dynamic multi-objective optimisation under an expensive objective setting is even more challenging because an EMO algorithm is required to track the time varying PSs and/or PFs within a limited number of function evaluations. In particular, the computational budget between two time steps can even be much less than the classic expensive optimisation settings.

To enable EAs to solve expensive optimisation problems, surrogate models have been widely used to simulate the originally computationally demanding objective function, as known as surrogate-based EAs (SAEAs). Many machine learning algorithms, such as Gaussian processes, neural networks and radial basis function networks, have been applied to build surrogate models in SAEAs. Interested readers are referred to a recent survey [6] on data-driven optimisation about the current developments along this line of research. It is worth noting that surrogate modelling is not a panacea, given the intrinsic approximation errors of surrogate models brought by limited training data. Such imperfect surrogate model may mislead the evolutionary search. Under a dynamic and uncertain environment, surrogate modelling becomes even more challenging since the training data in the previous time step highly likely to be useless for modelling the current landscape. Given the short turnaround time between two consecutive changes, it is suspicious to collect enough training data to build a reliable surrogate model even at the end of a time step.

Bearing the above considerations in mind, this paper proposes a SAEA based on transfer learning for solving DMOPs with computationally expensive objective functions. In particular, for proof-of-concept purpose, MOEA/D-EGO [7] is chosen as the baseline surrogate-assisted optimiser and our algorithm is thus dubbed as TL-MOEA/D-EGO. Instead of building a surrogate model from scratch at each time step, TL-MOEA/D-EGO applies transfer learning to map the training data collected from previous time step(s) to enrich the data used to build the surrogate model at the current time step. Proof-of-concept experiments on several dynamic benchmark test problems demonstrate the effectiveness and competitiveness of our proposed TL-MOEA/D-EGO comparing to four state-of-the-art peer competitors.

The rest of this paper is organised as follows. Section II provides a pragmatic overview of some related works on dynamic EMO algorithms. Section III describes the algorithmic implementation of our proposed algorithm. Experimental results are presented and analysed in Section IV. Section V concludes this paper and threads some potential future directions.

II. Related Works

In the section, we will give a pragmatic overview of some important works on both surrogate assisted EAs for solving computationally expensive MOPs and techniques for solving dynamic MOPs.

A. Surrogate Assisted EMO Methods

Most surrogate assisted EMO algorithms are derived from those SAEAs designed for expensive global optimisation. For example, in [8], Goel et al. proposed a function approximation method that uses the polynomial functions to approximate the originally expensive objective functions. It is worth noting that this method works well for problems with two and three objectives in a convex space. In [9], Ponweiser et al. adapted the Kriging model and the efficient global optimisation (EGO) [10] framework to the SMS-EMOA [11] and proposed a SMS-EGO. In [7], Zhang et al. proposed to incorporate the EGO framework into the MOEA/D. Its basic idea is to decompose the original MOP into several subproblems. By this means, the surrogate model of the objective function is transformed into a surrogate of a subproblem. During the optimisation process, the expected improvement of each subproblem is used as the selection criterion. In [12], Chugh et al. proposed a surrogate-assisted reference vector guided EA for expensive MOP with more than three objectives. In managing the surrogate models, the algorithm focuses on the balance between convergence and diversity by making use of the uncertainty information derived from the surrogate model.

B. Techniques for DOMPs

According to the way about how to adapt to the changing environment, the existing techniques for DMOPs can be categorised into four classes.

1) Diversity-Based Techniques: The basic idea of this type of techniques is try to maintain the population diversity as much as possible so that to preserve a better exploration in the search space. By doing so, the evolutionary population is expected to keep a record of the representative spots of the search space to quickly adapt to the changing environment. For example, Deb et al. [13] proposed two algorithms based on the fast non-dominated sorting genetic algorithm (NSGA-II) [14]. Their basic idea is to introduce diversified solutions by replacing $\zeta\%$ population with randomly generated solutions or mutated ones. It is worth noting that both algorithms are sensitive the control parameter ζ. In [15], Jiang et al. proposed to keep some of the solution in the previous population untouched and use some prediction method to generate new solutions according to the movement of the changing PF. Unfortunately, it is far from trivial to keep track of the moving trajectory of the changing PF's manifold.

2) Memory-Based Techniques: Given the observation of the oscillation of DMOPs, the basic idea of this sort of method is to keep a record of some promising/representative solutions in the previous population. By doing so, it is expected to adapt to the those memorised solutions once the environment is changed back to the previous steps. In [16], an adaptive hybrid strategy is proposed to combine the memory-based strategy and the random strategy. However, when the changing severity is lower which means the difference between two population

is large, this algorithm may lose diversity and even worse than randomly reinitialization. In [17], a multiple reference point-based algorithm is proposed to use a set of predefined search directions to navigate the search process.

3) Multi-Population-Based Techniques: The rigor of this type of techniques is similar to the diversity-based techniques. It aims to use several co-evolving populations to maintain an acceptable diversity of the search space. For example, in [18], Branke et al. proposed to divide the evolutionary population into two parts, i.e., scout and base populations. During the search process, the base population searches for the optimal solutions in the current environment whilst the scout population keeps on tracking the change of the optimal solutions.

4) Prediction-Based Techniques: This type of techniques aims to use model-based method to keep track of the dynamics of the changing environment. In [19], a linear discrete-time Kalman filter (KF) [20] is used along with the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [21] for solving DMOPs. Since the KF is able keep track of the dynamics of the system and smooth the landscape by removing unnecessary noises, it is ideal tool for DMOPs. In particular, the KF is used to predict the potential position of the best initial population in the next time step. Unfortunately, the linear KF used in this work does not work for non-linear environment.

According to the above overview, we can find that it is important to leverage the historical information in the previous time steps to help keep track of the dynamic features of the problem landscape. This motivates us to take advantage to exploit the knowledge embedded in those historical information by using transfer learning.

III. Proposed Method

Since MOEA/D-EGO is used as the baseline optimiser of TL-MOEA/D-EGO, we first provide a briefing of its working principle. Afterwards, the basic idea and the mathematical underpinnings of transfer component analysis (TCA) [22], i.e., the transfer learning techniques used in TL-MOEA/D-EGO, are described. At the end, the additional steps used to incorporate TCA into MOEA/D-EGO to constitute TL-MOEA/D-EGO is explained step by step.

A. Working Principle of MOEA/D-EGO

MOEA/D-EGO is a MOEA/D [21] variant with the Gaussian process (GP) model for solving expensive multi-objective optimisation. Generally speaking, MOEA/D-EGO consists of the following six steps:

- **Step 1. Initialisation:** Use an experimental design method to sample N_I solutions from Ω and evaluate their objective values. Set $P_e = \{\mathbf{x}^i\}_{i=1}^{N_I}$
- **Step 2. Termination Criterion:** Terminate MOEA/D-EGO and output all non-dominated solutions in P_e.
- **Step 3. Model Building:** Using P_e to build a GP model for each objective and thus the acquisition function.
- **Step 4. Locating Candidate Solutions:** Using the vanilla MOEA/D to search a population of solutions $P_s = \{\tilde{\mathbf{x}}^i\}_{i=1}^{N}$ that optimise the acquisition function.
- **Step 5. Selecting Solutions for Function Evaluation:** Select N_E solutions from P_s according to a selection criterion.
- **Step 6. Update Database:** Evaluate the objective values of all N_E solutions from **Step 5** and add them all to P_e and go to **Step 2**.

In the following paragraphs, we will delineate some important steps directly related to TL-MOEA/D-EGO.

1) GP Modelling: In the SAEA literature, GP model [23] has been widely used as a relatively cheap to evaluate surrogate of the original expensive objective function. Given a set of training data $\mathcal{D} = \{(\mathbf{z}_I^i, z_O^i)|i = 1, \cdots, \tilde{N}\}$, GP regression aims to learn a latent function $g(\mathbf{z}_I)$ by assuming $z_O^i = g(\mathbf{z}_I^i) + \varepsilon$, where $\varepsilon \sim \mathcal{N}(0, \sigma_n^2)$ is an independently and identically distributed Gaussian noise. For a new and unknown data $\mathbf{z}_I^* \in [0,1]^{m-1}$, the mean and variance of the target $g(\mathbf{z}_I^*)$ are predicted as:

$$\begin{aligned} \overline{g}(\mathbf{z}_I^*) &= m(\mathbf{z}_I^*) + \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}(\mathbf{z}_O - \mathbf{m}(Z_I)) \\ \mathbb{V}[g(\mathbf{z}_I^*)] &= k(\mathbf{z}_I^*, \mathbf{z}_I^*) - \mathbf{k}^{*T}(K + \sigma_n^2 I)^{-1}\mathbf{k}^* \end{aligned}, \quad (2)$$

where $Z_I = (\mathbf{z}_I^1, \cdots, \mathbf{z}_I^{\tilde{N}})^T$ and $\mathbf{z}_O = (z_O^1, \cdots, z_O^{\tilde{N}})^T$. $\mathbf{m}(Z_I)$ is the mean vector of Z_I, $\mathbf{k}^*$ is the covariance vector between Z_I and $\mathbf{z}_I^*$, and K is the covariance matrix of Z_I. The predicted mean $\overline{g}(\mathbf{z}_I^*)$ is directly used as the prediction of z_O^*, and the prediction variance $\mathbb{V}[g^*]$ quantifies the uncertainty. Since there are multiple conflicting objective functions in a MOP, it is required to use GP to build a model for each objective function in Step 3 of MOEA/D-EGO.

The basic idea of MOEA/D is to decompose the original MOP into a population of subproblems and solve them in a collaborative manner. Without loss of generality, here we only consider the widely used Tchebycheff function as the subproblem formulation:

$$\begin{aligned} &\text{minimise} \quad g^t(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = \max_{1 \le i \le m} \{|f_i(\mathbf{x}) - z_i^*|/w_i\} \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned}, \quad (3)$$

where $\mathbf{w} = (w_1, \cdots, w_m)^T$ is a weight vector used to characterise a subproblem and $\mathbf{z}^*$ is the ideal point. In particular, MOEA/D uses Das and Dennis's method [24] to generate a population of weight vectors $W = \{\mathbf{w}^1, \cdots, \mathbf{w}^N\}$. In MOEA/D-EGO, instead of working with the surrogates of objective functions, it uses model composition method to derive a predictive distribution of $g^t(\mathbf{x}|\mathbf{w}, \mathbf{z}^*)$ as:

$$g^t(\mathbf{x}|\mathbf{w}, \mathbf{z}^*) = \max\{\max\{|f_i(\mathbf{x}) - z_i^*|/w_i\}\}_{i=1}^m, \quad (4)$$

where we only consider the cases of $m = 2$ and $m = 3$.

2) Acquisition Function: Under the efficient global optimisation (EGO) framework [10], instead of optimising the surrogate function directly, it optimises the acquisition function to guide the sampling and infill criterion in the next step. As for MOEA/D-EGO, each subproblem $g^t(\mathbf{x}|\mathbf{w}^i, \mathbf{z}^*)$,

$i \in \{1, \cdots, N\}$, can have an acquisition function $\xi^i(\mathbf{x})$ where MOEA/D-EGO uses the expected improvement (EI) in particular which is formulated as:

$$\xi^i(\mathbf{x}) = E[\max\{g^t_{\min}(*|\mathbf{w}^i, \mathbf{z}^*) - g^t((\mathbf{x}|\mathbf{w}^i, \mathbf{z}^*))\}] \quad (5)$$

where $g^t_{\min}(*|\mathbf{w}^i, \mathbf{z}^*)$ is the current minimum aggregation function value of the i-th subproblem. In Step 4, different from the vanilla MOEA/D which optimises the aggregation function of each subproblem, MOEA/D-EGO uses the acquisition function of each subproblem instead.

3) Infill Criterion: Different from the classic EGO framework which only update the training dataset one solution at a time, MOEA/D-EGO aims to evaluate $N_E > 1$ solutions simultaneously in a batch manner. Specifically, in Step 5, it first trims the evolutionary population P_s according to their similarity to the solutions in P_e. In other words, only different solutions can be considered to carry out function evaluations. Thereafter, it uses k-means algorithm to identify N_E clusters. At the end, the solution having the best acquisition function value at each cluster is chosen to have a function evaluation.

B. Domain Adaptation by Transfer Component Analysis

Transfer Component Analysis (TCA) has been widely recognised as one of the most successful domain adaptation methods for transfer learning purpose. In the context of DSAEA/TL, TCA is applied to adapt the evaluated training data to the current problem landscape, in order to overcome the shortage of data in the fast moving dynamic environment.

More specifically, we assume that the training data collected before the current time step can be formulated as $\mathcal{D}_s = \{(\mathbf{x}^i, \mathbf{F}(\mathbf{x}^i))\}_{i=1}^{n_s}$, which is called the source domain under the transfer learning rigor. As for the data collected at the current time step, they can be formulated as $\mathcal{D}_t = \{(\hat{\mathbf{x}}^i, \mathbf{F}(\hat{\mathbf{x}}^i))\}_{i=1}^{n_t}$ which is termed as the target domain. It is not uncommon that the marginal probability distribution of the data in the source domain is different from that of the target domain, i.e., $P(\mathcal{D}_s) \neq P(\mathcal{D}_t)$. In order to implement a domain adaptation, the basic idea of TCA is to use a feature mapping ϕ over the source and the target domain, thus leading to $P(\mathbf{F}(\mathbf{x})|\phi(\mathbf{x})) \approx P(\mathbf{F}(\hat{\mathbf{x}})|\phi(\hat{\mathbf{x}}))$. It is non-trivial to align two multi-variable probability distribution. Instead, Pan et al [22] proposed to reformulate this as an optimisation problem that aims to minimise the maximum mean discrepancy (MMD) [25] of two marginal probability distributions. Specifically, given two sets of observations $X = \{x_1, \cdots, x_{n_s}\}$ and $Y = \{y_1, \cdots, y_{n_t}\}$ drawn independently and identically distributed (i.i.d.) from two probability distributions p and q respectively. Then MMD and its empirical estimate is defined as:

$$\begin{aligned} MMD(\mathcal{F}, p, q) &= \sup_{f \in \mathcal{F}} (\mathbb{E}_{x \sim p}[f(x)] - \mathbb{E}_{y \sim q}[f(y)]) \\ MMD(\mathcal{F}, X, Y) &= \sup_{f \in \mathcal{F}} (\frac{1}{n_s} \sum_{i=1}^{n_s} f(\mathbf{x}^i) - \frac{1}{n_t} \sum_{i=1}^{n_t} f(\hat{\mathbf{x}}^i)) \end{aligned}, \quad (6)$$

where $\mathcal{F}$ is a class of functions $f : \mathcal{X} \to \mathbb{R}$. By using kernel theory, f can be written as $f(x) = \langle \phi(x), f \langle$ in a reproducing kernel Hilbert space (RKHS), where $\phi(x) : \mathcal{X} \to \mathcal{H}$. In this sense, the empirical estimate of MMD can be written as:

$$MMD(\mathcal{F}, X, Y) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\mathbf{x}_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(\mathbf{x}_j) \right\|^2_{\mathcal{H}}. \quad (7)$$

By using *kernel tricks*, equation (7) can be written as a quadratic product of kernel matrices K

$$\begin{aligned} &\left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\mathbf{x}_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(\mathbf{x}_j) \right\|^2 \\ &= \mathrm{tr}\left(\begin{bmatrix} \phi(\mathbf{x}_s) & \phi(\mathbf{x}_t) \end{bmatrix} \begin{bmatrix} \frac{1}{n_s^2}\mathbf{1}\mathbf{1}^{\mathrm{T}} & \frac{-1}{n_s n_t}\mathbf{1}\mathbf{1}^{\mathrm{T}} \\ \frac{-1}{n_s n_t}\mathbf{1}\mathbf{1}^{\mathrm{T}} & \frac{1}{n_t^2}\mathbf{1}\mathbf{1}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \phi(\mathbf{x}_s)^{\mathrm{T}} \\ \phi(\mathbf{x}_t)^{\mathrm{T}} \end{bmatrix} \right) \\ &= \mathrm{tr}\left(\begin{bmatrix} \phi(\mathbf{x}_s)^{\mathrm{T}} \\ \phi(\mathbf{x}_t)^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \phi(\mathbf{x}_s) & \phi(\mathbf{x}_t) \end{bmatrix} \begin{bmatrix} \frac{1}{n_s^2}\mathbf{1}\mathbf{1}^{\mathrm{T}} & \frac{-1}{n_s n_t}\mathbf{1}\mathbf{1}^{\mathrm{T}} \\ \frac{-1}{n_s n_t}\mathbf{1}\mathbf{1}^{\mathrm{T}} & \frac{1}{n_t^2}\mathbf{1}\mathbf{1}^{\mathrm{T}} \end{bmatrix} \right), \\ &= \mathrm{tr}\left(\begin{bmatrix} < \phi(\mathbf{x}_s), \phi(\mathbf{x}_s) > & < \phi(\mathbf{x}_s), \phi(\mathbf{x}_t) > \\ < \phi(\mathbf{x}_t), \phi(\mathbf{x}_s) > & < \phi(\mathbf{x}_t), \phi(\mathbf{x}_t) > \end{bmatrix} \mathbf{L} \right) \\ &= \mathrm{tr}\left(\begin{bmatrix} K_{s,s} & K_{s,t} \\ K_{t,s} & K_{t,t} \end{bmatrix} \mathbf{L} \right) \end{aligned} \quad (8)$$

where

$$(L)_{ij} = \begin{cases} \frac{1}{n_s n_s}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s \\ \frac{1}{n_t n_t}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise} \end{cases}, \quad (9)$$

By using such derivation, equation (7) can be simplified as:

$$MMD(\mathcal{F}, X, Y) = \mathrm{tr}(KL) - \lambda\,\mathrm{tr}(K), \quad (10)$$

where

$$K = \begin{bmatrix} K_{s,s} & K_{s,t} \\ K_{t,s} & K_{t,t} \end{bmatrix}. \quad (11)$$

Since equation (10) is a semi-definite programming problem, it can be solved by constructing the following formulation:

$$\tilde{K} = (KK^{-\frac{1}{2}}\tilde{W})(\tilde{W}^T K^{-\frac{1}{2}}K) = KWW^TK, \quad (12)$$

where $W = K^{-\frac{1}{2}}\tilde{W}$. By using 12, the distance between the distributions of two sets of observations can be re-written as:

$$\mathrm{Dist}(\mathcal{D}_s, \mathcal{D}_t) = \mathrm{tr}((KWW^TK)L) = \mathrm{tr}(W^TKLKW), \quad (13)$$

Thereafter, the final optimisation problem is:

$$\begin{aligned} &\min_W \mathrm{tr}(W^TKLKW) \\ &\text{s.t.} \quad W^TKHKW = I \end{aligned}, \quad (14)$$

where $H = \mathbf{I} - (1/[s+t])\mathbf{1}\mathbf{1}^T$ and $\mathbf{I} \in \mathbb{R}^{(s+t)*(s+t)}$ is an identity matrix. Lagrangian duality can be used to solve (14) whilst W is the first $(s+t)$ eigenvalues.

C. Using TCA within MOEA/D-EGO

As discussed in Section I, we do not intend to restart the search process from scratch when the environment has been changed from time step t to time step $t+1$. On the contrary, we expect to leverage the knowledge collected from the previous time steps to enrich the training dataset in the underlying time step. To this end, we only need the following three additional procedures to extend the original MOEA/D-EGO.

- Once the environment is switched from time step t to $t+1$, we need to store all non-dominated solutions from P_e at the time step t into $\mathcal{A}$.
- We use the same experimental design method as the Step 1 of MOEA/D-EGO to sample N_T solutions from Ω. In particular, $1 < N_T \ll N_I$ given the shortage of computational resource. Then, we evaluate their objective values and set them as the new P_e at the time step t.
- At for the Step 3 of MOEA/D-EGO, instead of just using P_e to build a GP model, we use TCA to map $\mathcal{A}$ to the domain of P_e at the current time step and generate a new set of data $\mathcal{A}_t$. At the end, $P_e \bigcup \mathcal{A}_t$ is used as the training data for building the GP model.

It is worth noting that we use MOEA/D-EGO for proof-of-concept purpose. In principle, this idea can be incorporated with any surrogate-assisted EMO algorithms.

IV. Empirical Study

A. Benchmark Problems

To have a proof-of concept study of our proposed TL-MOEA/D-EGO, we choose some widely used test problems in the literature, including FDA1 to FDA5 [4], ZJZ5 to ZJZ8 [26], dMOP1 and dMOP2 [27], to constitute our benchmark set. FDA test suite represents various dynamic features introduced in Section I. dMOP1 and dMOP2 provide an extension of FDA. ZJZ test suite consists of 10 different test problem instances. In particular, ZJZ1 to ZJZ4 are similar to FDA1 to FDA5, thus only the remaining six problems are considered in our experiments.

B. Performance Metrics

To have a quantitative evaluation of the performance of different algorithms, we use the modified inverted generational distance (MIGD) proposed in [26] as the performance metric. Specifically, it is defined as:

$$MIGD = \frac{\sum_{t \in T} IGD(S_t, S_t^*)}{T} \tag{15}$$

where T is the number of time steps, S_t^* is the reference set sampled from the t-th PF and S_t is the approximated PF obtained at the end of the time step t. In particular, the vanilla IGD is calculated as:

$$IGD(S_t, S_t^*)) = \frac{\sum_{\mathbf{x}^* \in S_t^*} dist(\mathbf{x}^*, S_t)}{|S_t^*|} \tag{16}$$

where $dist(\mathbf{x}^*, S_t)$ is the Euclidean distance between the solution $\mathbf{x}^* \in S_t^*$ and its nearest neighbour of S_t in the objective space and $|S_t^*|$ is the cardinality of S_t^*.

C. Peer Algorithms

In our experiments, we choose the dynamic NSGA-II (DNSGA-II) proposed in [13] as the peer algorithm for performance comparisons. Specifically, DNSGA-II keeps on choosing 10% randomly chosen solutions as pilots to keep track of the change of the environment. In particular, if the objective function values of the current pilots are different from their last step, the change of the environment is thus captured. In DNSGA-II, there are two different strategies to respond to the change of the environment:

- *DNSGA-II-A*: When a change occurs, $\zeta\%$ of the population is replaced by randomly generated solutions.
- *DNSGA-II-B*: Similar to DNSGA-II-A, $\zeta\%$ of the solutions are mutated with rate $\eta_{NSGA}\%$ when the change is detected.

In particular, our experiments set $\zeta = 20$ and $\eta_{NSGA} = 5$.

D. Parameter Settings

The parameters associate with the algorithms considered in our experiments and also the benchmark problems are introduced as follows.

- The number of decision variables is set as $n = 10$ for all benchmark problems.
- The population size of DNSGA-II is set to 152.
- Simulated binary crossover and polynomial mutation is used as the reproduction operator. Their parameters are set as $\eta_c = \eta_m = 20$, $p_c = 1.0$ and $p_m = \frac{1}{n}$.
- In order to have a statistical sound comparison, Wilcoxon rank sum test is used to validate the statistical significance of the best results.
- There are two parameters associated with the benchmark problems to control the changing frequency and severity. Four different changing patterns are considered in our experiments, i.e., $(n_t = 10, \tau_t = 50)$, $(n_t = 10, \tau_t = 30)$, $(n_t = 10, \tau_t = 20)$, $(n_t = 20, \tau_t = 50)$. n_t and τ_t are the severity and the frequency of changes, respectively. A lower value of n_t represents larger change and a smaller value of τ_t represents faster change.

E. Results

The statistical results of IGD values for the different problems are listed in Table I, II, III and IV. The statistical test of Wilcoxon's rank-sum test for independent samples is tested for all IGD statistical values. Wilcoxon's rank-sum test is a non-parametric statistical hypothesis test used to compare two sets of samples to assess whether their population means rank differ. In this test, we assume that data are paired and come from the same population. Then compare the statistical results with the significance level at $p = 0.05$. When the statistical result is less than p, the hypothesis is rejected, that is, the proposed algorithm performs significantly better than the compared algorithms.

In our results, the performance of the TL-MOEA/D-EGO is better than the performance of the peer algorithms, especially in Test 2 (Table II). TF-MOEA/D-EGO algorithm performs

significantly better than the other comparison algorithms in 9 out of the 11 problems. This shows that the proposed algorithm can accelerate the speed of the search for Pareto Optimal. Figure 1-11 shows the IGD changing with different test problems. According to Figure II and Figure III, we can see that when τ_t gradually becomes smaller, that is, when fast changes occur, the performance of our algorithm decreases in some test problems. This is because our algorithm needs a certain number of samples as a benchmark to transfer, and it takes a certain period to collect samples. When the environment changes rapidly, detecting and tracking the latest changes is a complicated problem. Similarly, when n_t becomes larger, it indicates that the severity of the change becomes smaller, thus the effect of the algorithm using transfer learning method is not obvious.

In Fig. 1 to Fig. 4, we use 3 different lines to represent the IGD values' change of the proposed algorithm and the peer algorithms in each test case during function evaluations. In particular, our parameters set $n_t = 10$ and $\tau_t = 50$. As can be seen from the figures, except for the test problem of FDA1, the IGD value of the proposed algorithm changes less sharply change than the peer algorithms. This indicates that in most cases, the proposed algorithm can effectively track the changes of Pareto optimal.

As can be seen from Fig. 1, in the dMOP test problems, the proposed TF-MOEA/D-EGO performance is significantly better than other peer algorithms. First, in the beginning, TF-MOEA/D-EGO finds effective solutions faster than other algorithms. And in subsequent changes, the optimal solution has been continuously tracked by proposed algorithm. For the five problems of the FDA test set, the proposed algorithm performs poorly in some test cases, especially in FDA1, which shows greater fluctuations than other algorithms. Although our algorithm in FDA1 is like in the dMOP problem, the proposed algorithm converges faster than other algorithms, but the Pareto front is not captured real. This may be related to the choice of framework. We will conduct further experiments and improve in the future. In the FDA3, as shown in the Fig. 2(c), the three comparison algorithms show the same IGD change pattern. It can be considered that our algorithm is equally effective on this problem, but more attention to detail is needed to improve the performance of the algorithm. In rest tests of the FDA test set, the proposed algorithm performance is statistically significantly better than other peer algorithms. In the ZJZ test set, we tested four questions (ZJZ5 to ZJZ8). Among them, ZJZ5 to ZJZ7 are 2-objective test problems and have non-linear linkages between decision variables. It is observed that our algorithm performs significantly better than peers. The TF-MOEA/D-EGO algorithm not only has immediate results at the beginning but also remains stable in subsequent changes. ZJZ8 is a 3-objective problem with PF that remains the same at all times. As can be seen from Fig. 4(d) and Table II, our algorithm performs significantly better.

TABLE I
COMPARISON RESULTS OF MIGD VALUES OF PROPOSED ALGORITHM AND OTHER PEER ALGORITHMS WHERE $n_t = 20$ $\tau_t = 50$)

	TL-MOEA/D-EGO	DNSGA-II-A	DNSGA-II-B
FDA1	2.385E-2(1.33E-4)	**6.098E-3(3.09E-7)**‡	2.909E-2(6.11E-4)†
FDA2	**5.670E-3(3.92E-6)**	3.684E-2(1.18E-2)†	2.157E-2(1.52E-4)†
FDA3	8.998E-1(2.75E-3)	3.244E-1(2.12E-1)†	**1.572E-1(3.86E-2)**‡
FDA4	**6.345E-2(1.67E-4)**	9.755E-2(3.38E-4)†	7.339E-2(2.79E-4)†
FDA5	**8.744E-2(2.89E-4)**	1.285E-1(3.89E-3)†	2.528E-1(1.33E-2)†
ZJZ5	**2.724E-2(1.90E-2)**	3.780E-2(3.42E-5)†	1.422E-1(5.04E-3)†
ZJZ6	**3.596E-2(1.30E-3)**	3.909E-2(3.77E-5)†	6.936E-2(6.77E-5)†
ZJZ7	3.544E-2(3.02E-3)	**2.975E-2(3.49E-5)**‡	8.719E-2(7.08E-5)†
ZJZ8	**5.182E-2(6.70E-3)**	5.977E-1(1.26E-2)†	5.175E-1(1.62E-2)†
dMOP1	**1.298E-2(2.88E-4)**	1.541E-2(4.21E-5)†	1.989E-2(5.31E-5)†
dMOP2	1.330E-2(7.90E-5)	**5.960E-3(7.91E-7)**‡	1.298E-1(4.08E-2)†

† denotes that the proposed algorithm is significantly better than the compared algorithm with the Wilcoxon's rank-sum test at $p = 0.05$ significance level. ‡ denotes that the corresponding algorithm is better than proposed algorithm.

TABLE II
COMPARISON RESULTS OF MIGD VALUES OF PROPOSED ALGORITHM AND OTHER PEER ALGORITHMS WHERE $n_t = 10$ $\tau_t = 50$)

	TL-MOEA/D-EGO	DNSGA-II-A	DNSGA-II-B
FDA1	3.772E-2(3.23E-4)	**2.364E-2(1.15E-5)**‡	3.095E-2(2.37E-3)
FDA2	**6.441E-3(8.73E-6)**	1.421E-2(1.15E-4)†	1.559E-1(7.58E-7)†
FDA3	8.890E-1(7.80E-3)	5.197E-2(1.95E-2)	**4.214E-2(3.86E-4)**‡
FDA4	**8.193E-2(4.65E-4)**	1.280E-1(3.35E-4)†	9.917E-2(4.31E-5)†
FDA5	**8.616E-2(9.71E-4)**	1.609E-1(6.39E-4)†	1.489E-1(3.69E-5)†
ZJZ5	**2.326E-2(9.90E-5)**	9.499E-2(1.12E-3)†	2.146E-1(7.79E-3)†
ZJZ6	**2.316E-2(6.55E-3)**	5.881E-2(1.50E-4)†	1.450E-1(2.51E-4)†
ZJZ7	**1.433E-2(2.37E-2)**	5.249E-2(1.54E-4)†	1.535E-1(1.98E-4)†
ZJZ8	**4.645E-2(7.90E-4)**	4.989E-1(5.64E-2)†	1.116E-1(7.38E-5)†
dMOP1	**9.204E-3(1.04E-4)**	1.823E-2(1.20E-4)†	1.565E-1(6.42E-7)†
dMOP2	**2.161E-2(3.85E-4)**	5.583E-2(5.93E-4)†	3.724E-1(8.14E-2)†

† denotes that the proposed algorithm is significantly better than the compared algorithm with the Wilcoxon's rank-sum test at $p = 0.05$ significance level. ‡ denotes that the corresponding algorithm is better than proposed algorithm.

TABLE III
COMPARISON RESULTS OF MIGD VALUES OF PROPOSED ALGORITHM AND OTHER PEER ALGORITHMS WHERE $n_t = 10$ $\tau_t = 30$

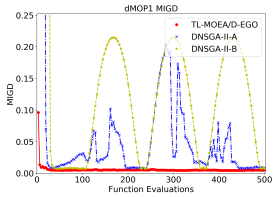
	TL-MOEA/D-EGO	DNSGA-II-A	DNSGA-II-B
FDA1	3.191E-2(2.19E-4)	**5.787E-3(1.12E-7)**‡	5.189E-2(1.60E-2)†
FDA2	**1.125E-2(7.04E-5)**	7.127E-2(9.41E-4)†	2.075E-1(8.35E-4)†
FDA3	8.818E-1(5.80E-3)	**7.808E-2(1.51E-2)**‡	3.280E-1(1.72E-3)†
FDA4	**5.130E-2(2.50E-5)**	7.832E-2(2.79E-4)	5.652E-2(1.97E-5)
FDA5	**5.657E-2(2.50E-5)**	3.876E-1(6.63E-3)†	5.148E-1(2.78E-3)†
ZJZ5	5.927E-2(1.24E-2)	**4.924E-2(1.29E-4)**‡	6.598E-1(2.22E-2)†
ZJZ6	4.138E-2(3.11E-4)	**4.026E-2(4.51E-5)**	5.570E-2(6.18E-4)
ZJZ7	1.621E-1(1.34E-1)	**3.404E-2(1.55E-5)**‡	4.101E-2(7.32E-5)
ZJZ8	**6.679E-2(5.73E-4)**	3.870E-1(1.62E-2)†	1.253E-1(3.70E-4)†
dMOP1	**1.129E-2(3.55E-5)**	5.933E-2(2.20E-3)†	2.155E-1(3.74E-7)†
dMOP2	**7.701E-2(3.06E-3)**	9.049E-2(2.73E-3)	1.521E-1(6.13E-2)

† denotes that the proposed algorithm is significantly better than the compared algorithm with the Wilcoxon's rank-sum test at $p = 0.05$ significance level. ‡ denotes that the corresponding algorithm is better than proposed algorithm.

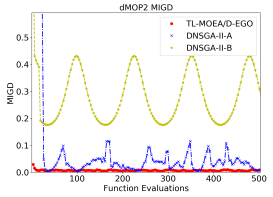
TABLE IV
COMPARISON RESULTS OF MIGD VALUES OF PROPOSED ALGORITHM AND OTHER PEER ALGORITHMS WHERE $n_t = 10$ $\tau_t = 20$)

	TL-MOEA/D-EGO	DNSGA-II-A	DNSGA-II-B
FDA1	7.786E-2(7.13E-4)	**3.184E-2(5.66E-5)**‡	4.257E-2(6.65E-3)
FDA2	**1.919E-2(4.70E-4)**	2.846E-1(8.70E-2)†	2.635E-1(8.97E-2)†
FDA3	8.856E-1(4.89E-3)	7.325E-1(3.29E-1)†	3.372E-1(8.59E-2)†
FDA4	**5.738E-2(2.50E-4)**	1.431E-1(3.40E-4)†	1.236E-1(2.75E-4)
FDA5	**5.199E-2(2.50E-5)**	1.637E-1(4.90E-4)†	1.494E-1(3.13E-5)†
ZJZ5	**5.153E-2(4.94E-4)**	2.512E-1(1.23E-2)†	7.115E-1(1.07E-1)†
ZJZ6	1.137E-1(1.62E-2)	2.028E-1(4.15E-3)†	**8.518E-2(9.07E-4)**‡
ZJZ7	9.670E-1(3.98E-1)	1.541E-1(6.41E-3)	**8.031E-2(3.05E-4)**‡
ZJZ8	**2.820E-1(3.76E-4)**	8.959E-1(5.19E-2)†	6.178E-1(1.37E-2)†
dMOP1	**1.425E-2(1.17E-4)**	3.875E-1(1.03E-1)†	3.472E-1(1.07E-1)†
dMOP2	**6.900E-2(2.72E-3)**	1.172E-1(2.15E-2)	1.565E-1(4.97E-2)

† denotes that the proposed algorithm is significantly better than the compared algorithm with the Wilcoxon's rank-sum test at $p = 0.05$ significance level. ‡ denotes that the corresponding algorithm is better than proposed algorithm.



(a) MIGD for dMOP1 Problem



(b) MIGD for dMOP2 Problem

Fig. 1. Comparison of MIGD for dMOP Problems($n_t = 10$ $\tau_t = 50$)

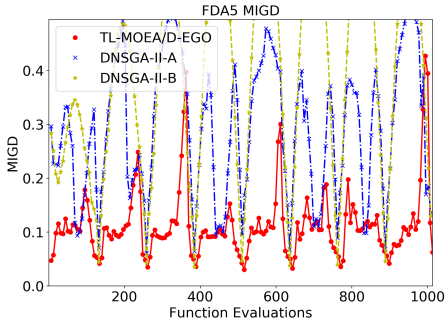


Fig. 3. Results of MIGD for FAD5 Problem($n_t = 10$ $\tau_t = 50$)

V. CONCLUSIONS

Due to the dynamic features (e.g., changing PS, PF or problem formulation), dynamic multi-objective optimisation pose significant more challenges than its static counterpart. It becomes even more complicated if the objective functions are computationally expensive. As an initial attempt along this line, we propose a surrogate assisted EA, using MOEA/D-EGO as the baseline, based on transfer learning, using TCA to implement domain adaptation, to solve dynamic MOPs with expensive objective functions. Its basic idea is to use transfer learning to jump start the surrogate model building process in order to adapt to the fast moving environment. Preliminary results on some popular dynamic multi-objective optimisation benchmark problems demonstrate the effectiveness of our proposed method.

As an early attempt along this line of research, a lot of issues remained for further investigations. For example, many other transfer learning techniques can be considered to better leverage the previous knowledge. In some cases, the landscape might have a significant change with respect to the previous time steps, it is arguable to treat all those archived data equally in transfer learning. It is interesting to study more problem specific way to aggregate the training data archived at different time steps. Many other surrogate assisted EA frameworks can be explored in future research.

REFERENCES

[1] J. G. Herrero, A. Berlanga, and J. M. M. López, "Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters," *IEEE Trans. Evolutionary Computation*, vol. 13, no. 1, pp. 151–168, 2009.

[2] J. Handl, D. B. Kell, and J. D. Knowles, "Multiobjective optimization in bioinformatics and computational biology," *IEEE/ACM Trans. Comput. Biology Bioinform.*, vol. 4, no. 2, pp. 279–292, 2007.

[3] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.

[4] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Trans. Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.

[5] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Comp. Int. Mag.*, vol. 4, no. 3, pp. 62–76, 2009.

[6] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evolutionary Computation*, vol. 23, no. 3, pp. 442–458, 2019.

[7] Q. Zhang, W. Liu, E. P. K. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with gaussian process model," *IEEE Trans. Evolutionary Computation*, vol. 14, no. 3, pp. 456–474, 2010.

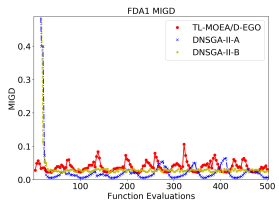
[8] T. Goel, R. Vaidyanathan, R. T. Haftka, W. Shyy, N. V. Queipo, and K. Tucker, "Response surface approximation of pareto optimal front in multi-objective optimization," *Computer methods in applied mechanics and engineering*, vol. 196, no. 4-6, pp. 879–893, 2007.

[9] W. Ponweiser, T. Wagner, D. Biermann, and M. Vincze, "Multiobjective optimization on a limited budget of evaluations using model-assisted -metric selection," in *PPSN X: Proc. of the 10th International Conference Dortmund Parallel Problem Solving from Nature*, 2008, pp. 784–794.

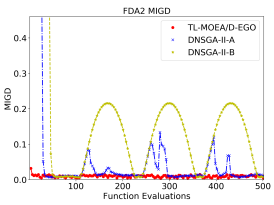
[10] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.

[11] N. Beume, B. Naujoks, and M. T. M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

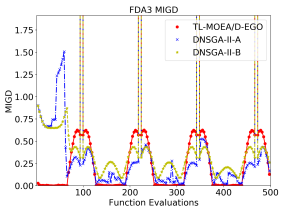
[12] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evolutionary Computation*, vol. 22, no. 1, pp. 129–142, 2018.



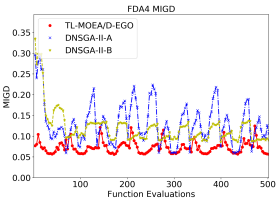
(a) MIGD for FDA1 Problem



(b) MIGD for FDA2 Problem

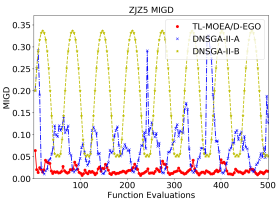


(c) MIGD for FDA3 Problem

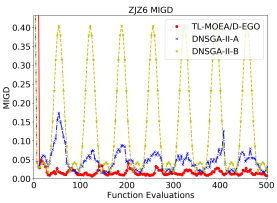


(d) MIGD for FDA4 Problem

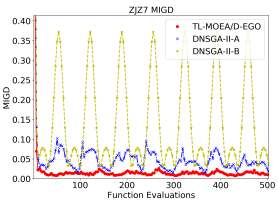
Fig. 2. Comparison of MIGD for 2 objective FDA Problems($n_t = 10$ $\tau_t = 50$)



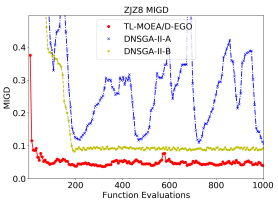
(a) MIGD for ZJZ5 Problem



(b) MIGD for ZJZ6 Problem



(c) MIGD for ZJZ7 Problem



(d) MIGD for ZJZ8 Problem

Fig. 4. Comparison of MIGD for ZJZ Problems($n_t = 10$ $\tau_t = 50$)

[13] 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 *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007, Proceedings*, 2006, pp. 803–817.

[14] 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.

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

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

[17] ——, "A multiple reference point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes," in *CEC'14: Proc. of the IEEE Congress on Evolutionary Computation*, 2014, pp. 3168–3175.

[18] J. Branke, T. Kaußler, C. Smidt, and H. Schmeck, "A multi-population approach to dynamic optimization problems," in *Evolutionary design and manufacture*. Springer, 2000, pp. 299–307.

[19] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via kalman filter prediction," *IEEE Trans. Cybernetics*, vol. 46, no. 12, pp. 2862–2873, 2016.

[20] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[21] 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.

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

[23] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, Massachusetts: The MIT Press, 2006.

[24] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[25] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *NIPS'06: Proc. of the Twentieth Annual Conference on Neural Information Processing Systems*, 2006, pp. 513–520.

[26] 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.

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