

Evolutionary Dynamic Multi-objective Optimization Via Regression Transfer Learning

Zhenzhong WANG, Min JIANG*, Xing GAO*, Liang FENG, Weizhen HU, and Kay Chen TAN

Abstract

Dynamic multi-objective optimization problems (DMOPs) remain a challenge to be settled, because of conflicting objective functions change over time. In recent years, transfer learning has been proven to be a kind of effective approach in solving DMOPs. In this paper, a novel transfer learning based dynamic multi-objective optimization algorithm (DMOA) is proposed called regression transfer learning prediction based DMOA (RTLP-DMOA). The algorithm aims to generate an excellent initial population to accelerate the evolutionary process and improve the evolutionary performance in solving DMOPs. When an environmental change is detected, a regression transfer learning prediction model is constructed by reusing the historical population, which can predict objective values. Then, with the assistance of this prediction model, some high-quality solutions with better predicted objective values are selected as the initial population, which can improve the performance of the evolutionary process. We compare the proposed algorithm with three state-of-the-art algorithms on benchmark functions. Experimental results indicate that the proposed algorithm can significantly enhance the performance of static multi-objective optimization algorithms and is competitive in convergence and diversity.

Keywords

evolutionary algorithm, dynamic multi-objective optimization, transfer learning, regression prediction

I. INTRODUCTION

Many optimization problems in the real world [1] [2] 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) [3]. For example, in the design of job scheduling systems [4], a number of decision variables, such as procedures, components, and operation time, are involved, which determine objective functions of energy consumption, production, and stability. These conflicting objective functions always change with time. Hence, efficient DMOAs should rapidly arrange scheduling schemes according to the changing environments, and this ability is critical to robust scheduling systems.

Z.WANG and M. JIANG and X.GAO and W.HU are with the school of informatics, Xiamen University, China, Fujian, 361005. Min JIANG and Xing GAO are corresponding authors and email: minjiang@xmu.edu.cn, gaoxing@xmu.edu.cn.

L.FENG is with the College of Computer Science, Chongqing University, Chongqing, China 400044.

KC TAN is with the Department of Computer Science, City University of Hong Kong.

In recent years, in order to solving DMOPs, a variety of DMOAs have been proposed. These existing methods can be roughly grouped into the following three categories: The first category of DMOAs is based on maintaining diversity. Gong *et al.* [5] proposed a general framework to decompose decision variables into two subpopulations according to the interval similarity between each decision variable and interval parameters, and a strategy on the basis of change intensity is adopted to track the POF. In [6], Jiang *et al.* developed a framework based on domain adaptive and non-parametric estimation to keep the exploration-exploitation of DMOPs in terms of temporal and spatial views. The second is a memory-based method. Chen *et al.* [7] implemented a dynamic two-archive strategy to simultaneously maintain two co-evolving populations. One population is concerned on convergence while the other focuses on diversity. Branke *et al.* [8] proposed a memory scheme to enhance the evolutionary process. In this algorithm, some excellent solutions are saved which can be used for guiding towards to optimal solutions. The third category of DMOAs is based on prediction. Muruganantham *et al.* [9] presented a population prediction strategy based on the Kalman filter technique. The Kalman filter technique [10] can guide the search for new Pareto-optimal solutions to generate a large number of high-quality initial individuals. Then, the algorithm finds the optimal at this moment based on a decomposition-based differential evolution algorithm. Rong *et al.* [11] presented a prediction model to track the moving POS by clustering the whole population into several subpopulations. In addition, the number of clusters depends on the intensity of environmental change. Zhou *et al.* [12] proposed a population prediction method to predict a whole population instead of predicting some isolated points. The algorithm uses center points to predict the next center point, and the previous manifolds are used to estimate the next manifold. The optimal population at this moment is determined based on a decomposition-based differential evolution algorithm. Hu *et al.* [13] designed a promising approach based on Incremental Support Vector Machine (ISVM) [14] classifier in solving DMOPs, the ISVM is trained from the past Pareto-optimal set, then high-quality initial individuals are filtered through the trained ISVM. Jiang *et al.* [15] presented a framework based on transfer learning [16] to predict an effective initial population for solving DMOPs. The transfer component analysis (TCA) [16] is used in this framework for the domain adaptation problem [17].

Traditional machine learning approaches are usually based on the assumption that the samples follow the Independent Identically Distributed (IID). Nevertheless, this hypothesis will be broken when dealing with DMOPs, since the solution distribution fails to satisfy the IID hypothesis. Although there is a DMOA based on transfer learning. However, it leads to poor diversity when samples clustering in the high dimensional latent space created by TCA.

In this paper, a regression transfer learning prediction based DMOA (RTLP-DMOA) is proposed. The algorithm aims to generate an excellent initial population to enhance the ability of existing multi-objective optimization algorithms for DMOPs. When the environment has changed, a regression transfer learning prediction model is constructed by utilizing the historical population information which can predict objective values in the new environment. Then, with the assistance of this regression prediction model, some high-quality solutions with better predicted objective values can be identified and selected as an initial population, which can improve the individuals' performance of the evolutionary process significantly.

The contributions of this work are as follows: 1) The proposed algorithm can make full use of historical

information and predict high-quality initial population to improve the evolutionary performance of the existing static multi-objective optimization algorithms (SMOAs) in solving DMOPs. 2) The proposed algorithm can overcome the difficulty that solution distributions fail to meet the IID hypothesis. Compared with other prediction methods, the RTLP-DMOA is promising.

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 RTLP-DMOA. Section III gives the designed RTLP-DMOA in detail. In Section IV, experimental results and analysis are shown. Conclusions are drawn in Section V.

II. PRELIMINARY STUDIES

A. Dynamic Multi-objective Optimization

The mathematical form of DMOPs is as follows:

$$\min F(x, t) = \langle f_1(x, t), f_2(x, t), \dots, f_m(x, t) \rangle \quad (1)$$

where $x \in \Omega$, and $x = \langle x_1, x_2, \dots, x_n \rangle$ is the n -dimensional decision vector, and t is the environment variable. $F = \langle f_1, f_2, \dots, f_m \rangle$ is the m -dimensional objective vector. The goal of DMOAs is to find solutions at environment t so that all objectives are as small as possible. Nevertheless, one solution cannot satisfy the minimum of all conflicting objectives. Hence, a trade-off method called Pareto dominance is introduced to compare these solutions. The set of optimal trade-off solutions is called the Pareto-optimal solutions (POS) in the decision space and the Pareto-optimal front (POF) in the objective space [18].

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

$$\begin{cases} f_j(x_2, t) \geq f_j(x_1, t), & \forall j = 1, \dots, m \\ f_j(x_2, t) > f_j(x_1, t), & \exists j = 1, \dots, m \end{cases} \quad (2)$$

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

$$DPOS = \{x^* | \nexists x, x \succ_t x^*\}, \quad (3)$$

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

Definition 3: (Dynamic Pareto-Optimal Front, DPOF) DPOF is the Pareto-optimal front of the DPOS for the DMOPs at the environment t

$$DPOF = \{y^* | y^* = F(x^*, t), x^* \in DPOS\}.$$

B. TrAdaboost.R2

TrAdaboost [19] is a classification algorithm based on the boosting method. The aim of TrAdaBoost is to filter out dissimilar samples in the past source domain to those in the target domain. In this way, TrAdaboost improves the classification accuracy. The source data set is combined with the target domain set to form a single data set.

At each boosting step, TrAdaBoost increases the relative weights of target instances that are misclassified. When a source instance is misclassified, however, its weight is decreased. In this way, TrAdaBoost makes use of those source instances that are most similar to the target data while ignoring those that are dissimilar. In [20], the authors introduce TrAdaboost-based algorithms for transfer regression task, called TrAdaboost.R2.

TrAdaboost.R2 is an ensemble method in which each weak regression hypothesis h_i ($i = 1, \dots, K$) can map the source domain data set X_{source} and the target domain data X_{target} to $Y \in R$. A strong regression hypothesis h is determined by combining these weak hypotheses. In each training round, TrAdaboost.R2 increases the relative weights of instances from the target domain. Meanwhile, TrAdaboost.R2 decreases the weights of the instances from the source domain. When the regression error of a instance caused by h_i is large, h_i has a substantial influence on the changing weight of the instance. In this way, TrAdaboost.R2 reuses source instances that are most similar to the target data and ignores those that are dissimilar. In the next round, these modified weights are inputted into the next regression hypothesis h_{i+1} , instances that are dissimilar to the target domain weaken their impacts of learning process, and instances with large weights help the learning algorithm in training better regressions.

III. PROPOSED ALGORITHM

The framework of RTLP-DMOA is illustrated in Algorithm 1. In brief, RTLP-DMOA initializes randomly a population $initPop$ with size N , and then executes a SMOA to optimize the population at environment $t = 0$. If environmental changes are detected, the environment variable t is updated as $t = t + 1$. Then, the last population P_{t-1} is inputted into the procedure of regression transfer. In the procedure of regression transfer, a regression hypothesis h^t is determined with historical information which can predict objective vectors of individuals at the new environment. Next, in the procedure called initial population prediction, the h^t is employed to predict the objective vectors and some high-quality individuals are selected according to their predictive objective vectors. These individuals are regarded as an excellent initial population $initPop$ and inputted into a SMOA to accelerate the evolutionary process. The details of RTLP-DMOA are presented in the following section.

A. Regression Transfer

The regression transfer process returns a strong regression hypothesis h^t for environment t . The strong regression hypothesis h^t adapts to the solution distribution at current environment. When an individual x is given, $h^t(x)$ outputs a predicted objective vector of x . Therefore, in the subsequent process of RTLP-DMOA, an excellent individual x with better predicted objective vectors $h^t(x)$ can be selected as a member of the initial population.

The strong regression hypothesis h^t is integrated with several weak regression hypotheses h_i^t ($i = 1, \dots, K$, K is the maximum number of iterations for training). These weak regression hypotheses are trained with the past population information. The last population P_{t-1} combined with their objective values $F_{t-1}(P_{t-1})$ are regarded as source domain set D_{source} . The target domain set D_{target} is comprised of \hat{P}_t which is sampled from $U(a, b)$ in the current decision space and their objective values $F_t(\hat{P}_t)$, where a and b are the lower bound and upper bound of the decision variable at environment t . D_{source} and D_{target} are combined into a set D as the training data.

Algorithm 1: RTLP-DMOA

Input: The dynamic multi-optimization problem $F_t(\cdot)$, the population size N , a *SMOA*

Output: the population P_t of $F_t(\cdot)$

- 1 Initialize the environment variable $t = 0$;
- 2 Initialize randomly a population $initPop$ with N ;
- 3 $P_t = \text{SMOA}(initPop, F_t)$;
- 4 **while** the environment has changed **do**

 - 5 $t = t + 1$;
 - 6 $h^t = \text{Regression-Transfer}(F_t, F_{t-1}, P_{t-1})$;
 - 7 $initPop = \text{Initial-Population-Prediction}(h^t, N)$;
 - 8 $P_t = \text{SMOA}(initPop, F_t)$;
 - 9 **return** P_t

- 10 **end**

The process for training weak regression hypotheses is as follows: First of all, the weight vector $w_i^1(x)$ is initialized as $\frac{1}{|D|}$, $w_i^t(x)$ denotes the weight of x for training h_i^t at environment t . In the main training loop, for training h_i^t , a Support Vector Regression (SVR) [21] is implemented as a basic learner to obtain the weak regression hypothesis h_i^t from D and w_i^t . Then, the adjusted error $e_i^t(x)$ of each individual $x \in D$ for training h_i^t is calculated as

$$e_i^t(x) = \begin{cases} \frac{|F_{t-1}(x) - h_i^t(x)|}{E_i^t}, & x \in D_{source} \\ \frac{|F_t(x) - h_i^t(x)|}{E_i^t}, & x \in D_{target} \end{cases}, \quad (4)$$

where E_i^t is the maximum error, it is described as

$$\begin{aligned} E_i^t = \max\{\max\{F_{t-1}(x) - h_i^t(x) | x \in D_{source}\}, \\ \max\{F_t(x) - h_i^t(x) | x \in D_{target}\}\}. \end{aligned} \quad (5)$$

The $e_i^t(x)$ is bigger when the difference between the predicted object vector $h_i^t(x)$ and the true objective vector $F(x)$ become bigger, and the adjusted error ϵ_i^t for h_i^t is calculated as

$$\epsilon_i^t = \sum_{x \in D_{target}} e_i^t(x) w_i^t(x). \quad (6)$$

When e_i^t is small, ϵ_i^t becomes smaller. Next, the weight vector is updated according to e_i^t and ϵ_i^t : If a training individual from the D_{source} has a bigger e_i^t , the individual may be more dissimilar to the distribution of the target domain. Therefore, its training weight must be reduced more. However, if a training individual from the target domain has a bigger e_i^t , then its training weight should be increased more for h_i^t to adapt the target domain. So, the weights can be updated as

$$w_{i+1}^t(x) = \begin{cases} w_i^t(x) \beta^{e_i^t(x)}, & x \in D_{source} \\ w_i^t(x) \beta_i^{-e_i^t(x)}, & x \in D_{target} \end{cases}, \quad (7)$$

where $\beta_i = \epsilon_i^t / (1 - \epsilon_i^t)$, and $\beta = 1 / (1 + \sqrt{2 \ln |D_{source}| / K})$. In this way, individuals adapted to the solution distribution of the target domain have large weights; otherwise, they have small weights. Then, modified weights w_{i+1}^t are inputted into next SVR to learn h_{i+1}^t . Thus, in the next round, individuals with low weights that are dissimilar to the target domain weaken their impacts of the learning process and those with large weights will help the learning algorithm train better regression hypotheses. These weak regression hypotheses ($h_i^t, i = 1, \dots, K$) may gradually adapt to the target domain. After K iterations, we obtain the final $\lceil \frac{K}{2} \rceil$ weak regression hypotheses and combine them to acquire a strong regression h^t .

The details of regression transfer are shown in Procedure REGRESSION TRANSFER.

Procedure(Regression Transfer)

Input: Dynamic optimization functions F_t and F_{t-1} , the population P_{t-1}

Output: A regression hypothesis h^t

- 1 $D_{source} = \{P_{t-1}, F_{t-1}(P_{t-1})\};$
- 2 Sample \hat{P}_t from $U(a, b)$;
- 3 $D_{target} = \{\hat{P}_t, F_t(\hat{P}_t)\};$
- 4 $D = D_{source} \cup D_{target};$
- 5 Initialize the weight vector $w_i^1 = \frac{1}{|D|};$
- 6 Set the maximum number of iterations K ;
- 7 **for** $i = 1, \dots, K$ **do**
- 8 Call a SVR with D and w_i^t , and get a regression hypothesis $h_i^t : X \rightarrow R^m$;
- 9 Calculate the adjusted error according to (4) for each individuals;
- 10 Calculate the adjusted error of h_i^t according to (6);
- 11 Update the weight vector according to (7);
- 12 **end**
- 13 **return** $h^t =$ the weighted median of $h_i^t(x)$ for the final $\lceil \frac{K}{2} \rceil$, and using $\ln(1/\beta_i)$ as the weight for hypothesis $h_i^t(x)$.

B. Initial Population Prediction

In this section, the initial population prediction is utilized to identify some excellent solutions as the initial population with the assistance of h^t .

To begin with, a test population P_t^{test} is sampled from $U(a, b)$, where a and b are the lower bound and upper bound of the decision variable at environment t . Then, objective values $h^t(P_t^{test})$ are predicted, and the non-dominated front \mathcal{F} can be determined by fast non-dominated sort [22] according to predicted objective values. Then, we select the first i non-dominated fronts as $initPop$ and limit the size of $initPop$ does not exceed the population size N . Next, some Gaussian noises are added to $initPop$ until the population size is N . The initial population $initPop$ is to accelerate the evolutionary process and improve the evolutionary performance for the current environment.

Procedure(INITIAL POPULATION PREDICTION)

Input: A regression hypothesis h^t , the population size N

Output: An initial population $initPop$

- 1 Sample P_t^{test} from $U(a, b)$;
- 2 $initPop = \emptyset$;
- 3 Use h^t to predict the objective value $h^t(P_t^{test})$;
- 4 $\mathcal{F} = \text{fast-non-dominated-sort}(P_t^{test}, h^t(P_t^{test}))$;
- 5 $i = 1$;
- 6 **while** $|initPop| + |\mathcal{F}_i| \leq N$ **do**
- 7 $initPop = initPop \cup \mathcal{F}_i$; // \mathcal{F}_i represents the i -th non-dominated front.
- 8 $i = i + 1$;
- 9 **end**
- 10 Add Gaussian noise to $initPop$ until the population size is N ;
- 11 **return** $initPop$

The details of initial population prediction are presented in Procedure INITIAL POPULATION PREDICTION.

IV. EXPERIMENTS

A. Compared Algorithms

Three DMOAs used for comparison in the experiment are as follows:

- 1) dCOEA [23]: It is a DMOA with dynamic competition-cooperative co-evolution.
- 2) PPS [24]: It is a DMOA based on population prediction.
- 3) SGEA [25]: It is a DMOA based on steady state and maintaining population diversity.

B. Test Problems

All compared algorithms are evaluated on 8 benchmark DMOPs selected from FDA [26] and DMOP [23]. The FDA benchmark comprises FDA1, FDA2, FDA3, FDA4, and FDA5. The DMOP benchmark contains dMOP1, dMOP2, and dMOP3.

DMOPs is divided into three categories: Type I problem indicates POS changes, but the POF does not change. Type II problem indicates changes in POS and POF. Type III problem implies the POF changes but the POS does not change.

FDA1, FDA4, and dMOP3 belong to Type I problem. FDA3, FDA5, and dMOP2 belong to Type II problem. Type III contains FDA2 and dMOP1.

The dynamics of a DMOP is controlled by

$$t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_t} \right\rfloor, \quad (8)$$

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

C. Performance Indicators

1) The Inverted Generational Distance (IGD) metric [27] can measure the convergence of obtained solutions. A small IGD value represents the convergence of the solution is improved. IGD is defined as

$$IGD(POF^*, POF) = \frac{1}{N} \sum_{p^* \in POF^*} \min_{p \in POF} \|p^* - p\|^2, \quad (9)$$

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

The MIGD [7] metric is a variant of IGD. The MIGD can be described as the average of the IGD values in all environments during a run.

$$MIGD(POF_t^*, POF_t) = \frac{1}{|T|} \sum_{t \in T} IGD(POF_t^*, POF_t), \quad (10)$$

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

2) The Maximum Spread (MS) [24] can quantify the extent of obtained solutions covers the true POF. A large MS value indicates additional coverage for the true POF by solutions obtained by the algorithm. MS is calculated as follows:

$$MS = \sqrt{\frac{1}{m} \sum_{k=1}^m \left[\frac{\min[F_k^{max}, f_k^{max}] - \max[F_k^{min}, f_k^{min}]}{F_k^{max} - F_k^{min}} \right]^2}, \quad (11)$$

where F_k^{max} and F_k^{min} represents maximum and minimum of k -th objective in true POF, respectively; and f_k^{max} and f_k^{min} represent the maximum and minimum of k -th objective in the obtained POF, respectively. This metric is also modified for evaluating DMOAs.

D. Parameter Settings

Parameter settings in RTLP-DMOA are as follows: We set the size of the population N to 100 and set the number of iterations K for training h^t to 10. The size of \hat{P}_t and P_t^{test} are set to 50 and 500, respectively. We choose RM-MEDA [27] as the SMOA optimizer for RTLP-DMOA, and the number of cluster is 4 in RM-MEDA. The parameters in SVR are set by default [28].

Consistent with the experimental configuration in this study [25]: We fix the n_t to 10. The frequency of change τ_t values are 5, and 10. The number of iterations of compared algorithms is $3 \times n_t \times \tau_t + 50$, of which 50 are the number of iterations at the initial time. Hence, in each population of configurations, the problem is changed by $3 \times n_t$ times.

E. Experimental Results

Experimental comparison results of RTLP-DMOA with other three state of the art DMOAs. MIGD values and MS values are presented in Tables I and Table II, respectively. The best metric values are highlighted in bold.

As the experimental results show, in Table I, the proposed RTLP-DMOA performs better than the other three algorithms in 9 out of 16 test instances for MIGD values. It clearly shows that the proposed RTLP-DMOA performs

TABLE I
MEAN AND STANDARD DEVIATION VALUES OF MIGD METRIC FOR DIFFERENT DYNAMIC TEST SETTINGS

Problem	τ_t, n_t	RTLP-RM-MEDA	dCOEA	PPS	SGEA
FDA1	(5,10)	0.0051(0.0013)	0.0661(0.0128)	0.2061(0.0769)	0.0338(0.0081)
	(10,10)	0.0049(0.0011)	0.0413(0.0068)	0.0476(0.0204)	0.0132(0.0025)
FDA2	(5,10)	0.0228(0.0046)	0.0774(0.0390)	0.0888(0.0348)	0.0121(0.0014)
	(10,10)	0.0223(0.0529)	0.0491(0.0329)	0.0619(0.0107)	0.0083(0.0006)
FDA3	(5,10)	0.1425(0.0066)	0.2640(0.0355)	0.4143(0.0101)	0.0612(0.0327)
	(10,10)	0.1455(0.0081)	0.1910(0.0338)	0.2003(0.0183)	0.0405(0.0180)
FDA4	(5,10)	0.1116(0.0092)	0.1604(0.0066)	0.3191(0.0203)	0.1603(0.0642)
	(10,10)	0.1189(0.0091)	0.1296(0.0048)	0.2196(0.0215)	0.1241(0.0664)
FDA5	(5,10)	0.3615(0.0027)	0.4387(0.0469)	0.6577(0.0318)	0.5221(0.0395)
	(10,10)	0.3612(0.0053)	0.3691(0.0403)	0.5037(0.0355)	0.4002(0.0088)
DMOP1	(5,10)	0.0469(0.0620)	0.0702(0.0157)	0.4182(0.1674)	0.0136(0.0079)
	(10,10)	0.0495(0.0085)	0.0395(0.0066)	0.0499(0.0091)	0.0084(0.0057)
DMOP2	(5,10)	0.0425(0.0101)	0.1103(0.0207)	0.1563(0.0126)	0.0345(0.0036)
	(10,10)	0.0427(0.0097)	0.0850(0.0098)	0.4293(0.0195)	0.0162(0.0005)
DMOP3	(5,10)	0.0047(0.0081)	0.0512(0.0101)	0.1717(0.0804)	0.1734(0.0858)
	(10,10)	0.0044(0.0077)	0.0287(0.0123)	0.1134(0.0079)	0.1252(0.0143)

better than the compared algorithms on FDA1, FDA4, FDA5, and DMOP3 under all configurations for the MIGD values. We can find that RTLP-DMOA achieves a good performance of MIGD values for tri-objective problems. This is because the prediction method based on the transfer learning method have a strong ability to explore complicated different solution distributions. However, it performs worse than SGEA for FDA3, DMOP1, and DMOP2 under all dynamic test settings. Experimental results of MIGD values indicate that the proposed RTLP-DMOA maintains better convergence over the other three state of the art DMOAs under most test functions.

It can be clearly found from the Table II that the proposed RTLP-DMOA obtains the best results in 13 out of 16 instances for MS values. Apart from FDA3 and DMOP1, RTLP-DMOA performs better than the compared algorithms under all configurations. It is worth noting that RTLP-DMOA achieves the maximum value of MS on tri-objective problems: FDA4 and FDA5. Nevertheless, RTLP-DMOA is a little worse than SGEA on FDA3. Overall, the diversity of solutions obtained by RTLP-DMOA are extremely better than the other three algorithms in most case.

F. Discussion

In this subsection, we perform a comparative experiment to verify whether the combination with the regression transfer learning prediction can improve performance. We compare RTLP-RM-MEDA with RM-MEDA. RM-MEDA is originally used to solve the static multi-objective problem and not applicable for DMOPs. Table III indicates that RTLP-RM-MEDA performs better than RM-MEDA in all test functions at $n_t = 10$ and $\tau_t = 5$ configuration

TABLE II
MEAN AND STANDARD DEVIATION VALUES OF MS METRIC FOR DIFFERENT DYNAMIC TEST SETTINGS

Problem	τ_t, n_t	RTLP-RM-MEDA	dCOEA	PPS	SGEA
FDA1	(5,10)	0.9983(0.0026)	0.8697(0.0249)	0.8721(0.0333)	0.9441(0.0378)
	(10,10)	0.9985(0.0024)	0.8921(0.0211)	0.9635(0.0149)	0.9782(0.0110)
FDA2	(5,10)	0.9988(0.0047)	0.8267(0.0505)	0.9013(0.0497)	0.9934(0.0053)
	(10,10)	0.9939(0.0036)	0.8672(0.0285)	0.9356(0.0121)	0.9930(0.0034)
FDA3	(5,10)	0.8809(0.0035)	0.5031(0.0427)	0.6001(0.0404)	0.8843(0.0711)
	(10,10)	0.8585(0.0253)	0.5873(0.0356)	0.6180(0.0299)	0.9437(0.0775)
FDA4	(5,10)	1.0000(0.0000)	0.9649(0.7774)	0.9984(0.0008)	0.9997(0.0001)
	(10,10)	1.0000(0.0000)	0.9702(0.0063)	0.9990(0.0001)	0.9996(0.0001)
FDA5	(5,10)	1.0000(0.0000)	0.9304(0.0380)	0.9974(0.0024)	0.9997(0.0001)
	(10,10)	1.0000(0.0000)	0.9551(0.0369)	0.9979(0.0039)	0.9995(0.0001)
DMOP1	(5,10)	0.9961(0.0011)	0.8643(0.0414)	0.9301(0.0667)	0.9555(0.0305)
	(10,10)	0.9823(0.0006)	0.8881(0.0255)	0.9782(0.0339)	0.9849(0.0179)
DMOP2	(5,10)	0.9962(0.0028)	0.7556(0.0563)	0.8513(0.0139)	0.9502(0.0130)
	(10,10)	0.9980(0.0263)	0.8145(0.0253)	0.9600(0.0147)	0.9810(0.0004)
DMOP3	(5,10)	0.9969(0.0013)	0.8782(0.0136)	0.8559(0.0315)	0.5031(0.0248)
	(10,10)	0.9991(0.0014)	0.9104(0.0093)	0.8880(0.0183)	0.5838(0.0296)

for MIGD values. The RTLP-RM-MEDA improves the RM-MEDA for MIGD values by 22.66%96.39%. Table IV indicates that RTLP-RM-MEDA performs better than RM-MEDA in all test instances for MS values. RTLP-RM-MEDA improves the RM-MEDA for MS values by 0.08%39.88%. The ablation study reveals that the designed regression transfer learning prediction can significantly improve the performance of SMOAs.

TABLE III
MEAN AND STANDARD DEVIATION VALUES OF MIGD METRIC FOR DIFFERENT PROBLEMS AT $n_t = 10$ AND $\tau_t = 5$

Problem	RM-MEDA	RTLP-RM-MEDA
FDA1	0.1309(0.0287)	0.0051(0.0013)
FDA2	0.1429(0.0333)	0.0228(0.0046)
FDA3	0.2110(0.0285)	0.1425(0.0066)
FDA4	0.1691(0.0140)	0.1116(0.0092)
FDA5	0.5522(0.0160)	0.3615(0.0027)
DMOP1	0.4187(0.0689)	0.0469(0.0620)
DMOP2	0.0696(0.0149)	0.0425(0.0101)
DMOP3	0.0235(0.0125)	0.0047(0.0081)

TABLE IV
MEAN AND STANDARD DEVIATION VALUES OF MS METRIC FOR DIFFERENT PROBLEMS AT $n_t = 10$ AND $\tau_t = 5$

Problem	RM-MEDA	RTLP-RM-MEDA
FDA1	0.8515(0.0365)	0.9983(0.0026)
FDA2	0.9447(0.0098)	0.9988(0.0047)
FDA3	0.6634(0.1033)	0.8809(0.0035)
FDA4	0.9992(0.0002)	1.0000(0.0000)
FDA5	0.9988(0.0002)	1.0000(0.0000)
DMOP1	0.7121(0.0759)	0.9961(0.0011)
DMOP2	0.9274(0.0190)	0.9962(0.0028)
DMOP3	0.9692(0.0105)	0.9969(0.0013)

V. CONCLUSION

This paper has proposed the RTLP-DMOA in solving DMOPs. When the environment has changed, a regression hypothesis which adapts to the solution distribution for predicting objective values is deduced. Then, excellent individuals are identified according to their predicted objective values and selected as an initial population, which can improve the performance of the evolutionary process.

From experimental comparison results, the proposed RTLP-DMOA is very competitive in most test functions. In our future work, we will integrate some advanced machine learning methods into evolutionary computing to enhance the evolutionary performance of existing static multi-objective optimization algorithms and solve the real world problems [29] [30].

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant No.61673328) and Shenzhen Scientific Research and Development Funding Program (Grant No. JCYJ20180307123637294).

REFERENCES

- [1] 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.
- [2] 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.
- [3] 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.
- [4] W. Du, W. Zhong, Y. Tang, W. Du, and Y. Jin, “High-dimensional robust multi-objective optimization for order scheduling: A decision variable classification approach,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 293–304, Jan 2019.
- [5] 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*, pp. 1–1, 2019.

- [6] 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.
- [7] R. Chen, K. Li, and X. Yao, "Dynamic multiobjectives optimization with a changing number of objectives," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 157–171, Feb 2018.
- [8] J. Branke, "Memory enhanced evolutionary algorithms for changing optimization problems," in *Proceedings of the 1999 Congress on Evolutionary Computation*. Institute of Electrical and Electronics Engineers.
- [9] 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.
- [10] G. Welch, *Kalman Filter*. Boston, MA: Springer US, 2014, pp. 435–437.
- [11] 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.
- [12] 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, 2014.
- [13] 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.
- [14] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano, and S. Li, "Incremental support vector learning for ordinal regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 7, pp. 1403–1416, July 2015.
- [15] 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.
- [16] 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.
- [17] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Systems*, vol. 80, pp. 14 – 23, 2015, 25th anniversary of Knowledge-Based Systems.
- [18] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.
- [19] W. Dai, Q. Yang, G. R. Xue, and Y. Yu, "Boosting for transfer learning," in *International Conference on Machine Learning*, 2007.
- [20] D. Pardoe and P. Stone, "Boosting for regression transfer," in *International Conference on Machine Learning*, 2010.
- [21] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing-Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2007.
- [22] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002.
- [23] C.-K. 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, 2009.
- [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] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 65–82, 2017.
- [26] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: test cases, approximations, and applications," *IEEE Transactions on evolutionary computation*, vol. 8, no. 5, pp. 425–442, 2004.
- [27] Q. Zhang, A. Zhou, and Y. Jin, "Rm-medea: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
- [28] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.
- [29] J. Min, C. Zhou, and S. Chen, "Embodied concept formation and reasoning via neural-symbolic integration," *Neurocomputing*, vol. 74, no. 1, pp. 113–120, 2010.
- [30] W. Yin, J. Min, Z. Huang, C. Fei, and C. Zhou, "An np-complete fragment of fibring logic," *Annals of Mathematics and Artificial Intelligence*, vol. 75, no. 3-4, pp. 391–417, 2015.