兰州理工大学

科 研 成 果 汇 总

学　　号：	201081104002
研 究 生：	盛昭君
导　　师：	李二超 教授
研究方向：	动态多目标优化
论文题目：	基于预测的动态多目标进化算法研究及应用
学　　科：	模式识别与智能系统
学　　院：	自动化与电气工程学院
入学时间：	2020 年 9 月

目 录

文献检索报告

报告编号：R2026-0100

机构：兰州理工大学

姓名：盛昭君　[201081104002]

著者要求对其在国内外学术出版物所发表的科技论著被以下数据库收录情况进行查证。

检索范围：

- 科学引文索引（Science Citation Index Expanded）： 1900年-2026年
- 工程索引（Engineering Index）： 1884年-2026年

检索结果：

检索类型	数据库	年份范围	总篇数
SCI-E 收录	SCI-EXPANDED	2025	1
EI 收录	EI-Compendex	2025	1

委托人声明：

本人委托兰州理工大学图书馆查询论著被指定检索工具收录情况，经核对检索结果，附件中所列文献均为本人论著，特此声明。

作　　者（签字）：盛昭君

完 成 人（签字）：刘　玮

完　成　日　期　：2026年3月13日

完成单位（盖章）：兰州理工大学图书馆信息咨询与学科服务部

（本检索报告仅限校内使用）

附件一：SCI-E 收录

#	作者	标题	来源出版物	文献类型	入藏号
1	**Sheng, ZJ**; Li, EC	Dynamic Multi-Objective Evolutionary Algorithm Based on Spectral Clustering Prediction	***IEEE ACCESS*** 2025, 13: 176421-176433.	J Article	WOS:001596864800046
				合计	1

第 1 条，共 1 条:
标题: Dynamic Multi-Objective Evolutionary Algorithm Based on Spectral Clustering Prediction
作者: **Sheng, ZJ** (**Sheng, Zhao-Jun**); Li, EC (Li, Erchao)
来源出版物: IEEE ACCESS **卷:** 13 **页:** 176421-176433 **出版年:** 2025
入藏号: WOS:001596864800046
文献类型: Article **出版物类型:** J
作者地址: [**Sheng, Zhao-Jun**; Li, Erchao] **Lanzhou Univ Technol**, Coll Elect Engn & Informat Engn, Lanzhou **730050**, Gansu, Peoples R China.
通讯作者地址: Li, EC (corresponding author), Lanzhou Univ Technol, Coll Elect Engn & Informat Engn, Lanzhou 730050, Gansu, Peoples R China.

附件二：EI 收录

#	作者	标题	来源出版物	文献类型	入藏号
1	**Sheng, Zhaojun**; Li, Erchao	Design of DMO Test Functions Based on PS Dynamic Changes and PF Discontinuity	***Youth Academic Annual Conference of Chinese Association of Automation, YAC*** 2025, (2025): 1794-1798.	Conference article (CA)	20260920158035
				合计	1

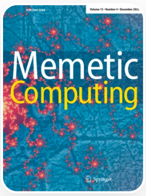
第 1 条，共 1 条:
Accession number:20260920158035
Title:Design of DMO Test Functions Based on PS Dynamic Changes and PF Discontinuity
Authors:Sheng, Zhaojun (1); Li, Erchao (1)
Author affiliation:(1) Lanzhou University of Technology, College of Electrical Engineering and Information Engineering, Lanzhou, China
Corresponding author:Sheng, Zhaojun(shengzjkk@163.com)
Source title:Youth Academic Annual Conference of Chinese Association of Automation, YAC Issue:2025;
Issue title:Proceedings - 2025 40th Youth Academic Annual Conference of Chinese Association of Automation, YAC 2025 Issue date:2025; Publication year:2025; Pages:1794-1798; Language:English; ISSN:28378598; E-ISSN:28378601; ISBN-13:9798331539481;
Document type:Conference article (CA)
Conference name:40th Youth Academic Annual Conference of Chinese Association of Automation, YAC 2025
Conference date:May 17, 2025 - May 19, 2025; Conference location:Zhengzhou, China; Conference code:212644;
Database:Compendex

End

A dividing lines–hyperplane intersection points and associated subpopulations prediction strategy for evolutionary dynamic multiobjective optimization

Regular Research Paper Published: 10 April 2026

Volume 18, article number 18, (2026) Cite this article



Memetic Computing

Aims and scope

Submit manuscript

Zhao-jun Sheng, Er-chao Li, Li-sen Wei & Chen-miao Liu

Abstract

To rapidly track the dynamic evolution of the Pareto-optimal front (PF) in dynamic multiobjective optimization problems (DMOPs), this paper proposes a dividing-line–hyperplane-intersection-based and associated subpopulation prediction strategy (ISPP). The strategy infers the changes in the Pareto-optimal set (PS) in the decision space from the dynamic characteristics of the PF in the objective space. First, hyperplane fitting and dividing-line setting strategy are used to locate and segment the PF, and to characterize

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 29 August 2025, accepted 12 September 2025, date of publication 3 October 2025, date of current version 16 October 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3617239

RESEARCH ARTICLE

Dynamic Multi-Objective Evolutionary Algorithm Based on Spectral Clustering Prediction

ZHAO-JUN SHENG AND ERCHAO LI
College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou, Gansu 730050, China
Corresponding author: Erchao Li (lecstarr@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62063019, in part by Gansu Provincial Natural Science Foundation of China under Grant 24JRRA173, and in part by Gansu Provincial Key Research and Development Program under Grant 25YFGA030.

ABSTRACT To address the challenge of rapidly tracking the new pareto optimal set (PS) after environmental changes in dynamic multi-objective optimization problems (DMOPs), this paper proposes a dynamic multi-objective evolutionary algorithm (DMOEA) based on spectral clustering prediction (SCP). The algorithm first applies spectral clustering to the PS from historical moments to automatically identify multiple subclasses of the PS. When environmental changes occur, the algorithm uses historical information to predict the centroids and shapes of each subclass, thereby constructing a new initial population. Experimental results on 14 standard dynamic test problems indicate that the proposed SCP-DMOEA outperforms four typical algorithms—PPS, MDP, KTM, and RNN—in handling various change characteristics in most test functions. These characteristics include translation or rotation changes of the PS, correlations between decision variables, and time-varying mixed pareto optimal fronts (PFs), including variations in convexity and concavity. In particular, SCP-DMOEA also shows superior performance when handling optimization problems with more complex PF changes, such as the position of the PF continuously changes over time, the objective vector oscillates among several modes, and the PFs are discontinuous.

INDEX TERMS Dynamic multi-objective optimization, evolutionary algorithm, prediction, spectral clustering.

I. INTRODUCTION

In many practical application fields, the optimization problems are dynamic multi-objective optimization problems (DMOPs), such as scheduling [1], [2], [3], [4], [5], management [6], [7], planning [8], [9], [10], design [11], [12], [13], control [14], and machine learning [15], [16]. The multiple objectives often conflict with each other, and the objective functions, constraint functions, and related parameters may change over time. Tracking the new optimal solution set after such changes is a major challenge in solving dynamic multi-objective optimization problems. Furthermore, as research on these types of problems deepens, greater challenges are posed to dynamic multi-objective optimization algorithms (DMOEAs).

The associate editor coordinating the review of this manuscript and approving it for publication was Harikrishnan Ramiah.

According to the uncertainty characteristics of DMOPs, there can be various different forms of description [17], [18]. The mathematical description of the DMOPs studied in this paper is shown in Equation 1:

$$\begin{aligned} &\min\ F(x,t) = (f_1(x,\ t),\ \ldots,\ f_m(x,t))^T \\ &\text{s.t. } x \in \prod_{i=1}^{n} [a_i, b_i] \end{aligned} \tag{1}$$

where $t \in T = \{1,\ 2,\ \ldots\}$ denotes the time instants. where $\prod_{i=1}^{n} [a_i,\ b_i] \subset \mathbb{R}^n$ refers to the decision space. $\boldsymbol{x} = (x_1,\ \ldots,\ x_n)^{\mathrm{T}}$ defines the decision variable. $F(\boldsymbol{x},t) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ consists of m objective functions, $\mathbb{R}^m$ refers to the objective space.

For two decision vectors $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \prod_{i=1}^{n} [a_i, b_i]$ of DMOPs at time t, $\boldsymbol{x}_1$ is said to Pareto dominate $\boldsymbol{x}_2$, when

$f_i(\boldsymbol{x}_1, t) \leq f_i(\boldsymbol{x}_2, t)$ for $\forall\ i = 1, 2, \cdots, m$ and $f_j(\boldsymbol{x}_1, t) < f_j(\boldsymbol{x}_2, t)$ for $\exists j = 1, 2, \cdots, m$. If there does not exist any solution $\boldsymbol{x} \in \prod_{i=1}^{n} [a_i, b_i]$ which can dominate $\boldsymbol{x}^*$ at time t, $\boldsymbol{x}^*$ is called a Pareto optimal solution and a set including all Pareto-optimal solutions is called Pareto optimal set (PS). Moreover, the corresponding objective vectors of PS are called Pareto optimal front (PF).

In the DMOPs outlined in (1), the Pareto optimal front (PF) and/or the Pareto optimal set (PS) change over time [19], [20]. Typically, the aim of dynamic multi-objective optimization algorithms is to swiftly track the changing PF (PS) when the environment changes.

Evolutionary Algorithms (EAs) possess characteristics of multi-point search, as well as adaptability and self-learning, making them highly suitable for solving complex, nonlinear problems. Since multi-objective optimization problems are often complex and nonlinear, evolutionary algorithms are particularly well-suited for solving such problems [21], [22], [23].

Currently, many dynamic response strategies for researching the PS have been proposed, among which predictive strategies have garnered particular attention. This strategy identifies patterns of change between historical environments by mining useful information from the decision space and objective space in historical environments and constructing predictive models. Therefore, predictive strategies can quickly forecast high-quality solutions in new environments and have become a research hotspot in recent years [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], which will be reviewed in detail in Section II. Based on the current research status, this paper focuses on investigating predictive strategies.

For decision variables that exhibit correlations in DMOPs, Zhou et al. [24] proposed a population-based prediction method (PPS). This method assumes that in the decision space, the PS of different environments is formed by a central point and an (m-1)-dimensional manifold. By utilizing the central point and the (m-1)-dimensional manifold of the PS from historical environments, the method predicts the entire initial population. However, when the PF of the test functions is discontinuous or when the PF changes complexly over time, such as when the PF oscillates between several patterns, the predictive strategy based on a single representative individual obtained from the PS may lead to poor prediction results.

To address the issue of PS that undergoes rotational changes in DMOPs, Rong et al. [25] proposed a multi-directional prediction method (MDP). This method utilizes k-means clustering to obtain multiple representative individuals, acquiring several evolutionary guiding directions to predict the entire initial population. However, when the PS changes over time and does not simply translate or rotate, k-means clustering may not effectively identify the shape changes of the PS, which leads to poor classification results of the PS and impacts prediction accuracy.

Assuming that the solutions in different environments are highly similar and follow an independent and identically distributed (IID) model, an online learning predictor based on Incremental Support Vector Machine (ISVM) for DMOEA was proposed in [26]. However, this assumption may not always hold in practical scenarios, thus reducing the performance of the prediction strategy. Since transfer learning can effectively handle data with non-IID characteristics, a transfer learning framework for DMOEA, Tr-DMOEA, was proposed in [27]. This framework implements transfer component analysis (TCA) in the objective space to utilize useful knowledge from past environments. However, most methods can encounter negative transfer. KTM-DMOEA [28] attempts to classify high-quality solutions from a large number of randomly generated new environments using kernel transfer principles (KTP) in dynamic environments, reducing differences in features and distributions between different environments. By modeling the distribution of elite solutions in the previous environment using kernel mean signature (KMS), this approach samples better solutions in the new environment based on dynamic change trends, aiming to alleviate negative transfer. Additionally, a DMOEA based on recurrent neural networks (RNN) [29] has been proposed, where the model learns the continuously obtained optimal solutions from all historical information and updates the RNN model using the acquired knowledge. However, the time series formed by historical information often contains linear or nonlinear growth or decline trends. Since RNN is inherently nonlinear models, it may overfit local nonlinear fluctuations while neglecting the overall linear trends, leading to prediction bias.

Based on the above research, we find that reasonably predicting and initializing the population in dynamic multi-objective optimization (DMO) can significantly improve the convergence speed of the algorithm and the quality of the solution set. The main contributions of introducing spectral clustering (SC) into the initial population prediction are as follows:

1) Handling Correlated Decision Variables: Most existing dynamic multi-objective optimization methods assume that decision variables are independent, which affects the quality of initial population predictions when dealing with correlated DMOPs. Spectral Clustering can transform the correlation between samples into reasonable similarity weights, effectively addressing DMOPs with correlated decision variables.
2) When dealing with DMOPs where the PS changes over time and does not simply translate or rotate, Spectral Clustering is more advantageous than traditional clustering algorithms, such as k-means and the EM algorithm, which assume that sample clusters are convex and spherical. By constructing a similarity matrix

and utilizing graph Laplacian eigenvectors, Spectral Clustering can identify clusters of arbitrary shapes and converge to a global optimum, demonstrating greater adaptability to data distributions. Therefore, Spectral Clustering is well-suited for clustering MOPs with complex PS structures [44]. Moreover, In practical applications, the structure of PS in optimization problems is frequently unknown, a clustering method that can effectively address MOPs with complex PSs is likely to be more applicable to real-world problems.

3) In DMOPs with discontinuous PFs, utilizing Spectral Clustering to automatically partition the PF into several connected subsets can effectively cover the discontinuous PF. At the same time, using the center of each subset as a representative individual can better accelerate the convergence rate of the population while ensuring its diversity, compared to using a single representative individual.
4) The graph-theoretic framework of Spectral Clustering inherently possesses noise tolerance. Even when suboptimal or deteriorating solutions exist in historical solutions, their influence can be weakened through feature decomposition. Thus, Spectral Clustering exhibits strong robustness.

Therefore, we propose a DMOEA based on spectral clustering prediction (SCP). The algorithm first implements spectral clustering on the PS at historical time points to automatically identify multiple subclasses of the PS. When the environment changes, the algorithm utilizes historical information to predict the centers and shapes of each subclass, thereby generating a new initial population. This approach can guide the population to increase its diversity while leveraging the similarity among individuals within the subclasses to enhance the convergence of the predicted population. Additionally, we introduce a framework that combines SCP with non-dominated sorting genetic algorithms (NSGA-III) [45] to enable the algorithm to quickly track the PS after changes in a dynamic environment. Finally, in the experimental section, We assess the performance of the algorithm by comparing the proposed prediction strategy against four standard prediction strategies.

The structure of this paper is as follows: Section II reviews the previous studies related to DMOPs. Section III comprehensively introduces the prediction strategy proposed in this research. Section IV describes the test cases, experimental settings, and the performance metrics adopted. Section V elaborates on the experimental results and analysis. Section VI summarizes the content of this paper and proposes the direction for future research.

II. RELATED WORK

In recent years, researchers have designed numerous predictive strategies to address DMOPs. For instance, Muruganantham et al. [30] proposed a prediction method based on Kalman filtering. Gee et al. [31] introduced an inverse modeling approach. Chen et al. [32] proposed a hybrid fuzzy reasoning prediction strategy. Guo et al. [33] established a collective prediction model based on three heterogeneous predictive factors to estimate the fitness values of candidates in new environments. Liang et al. [34] proposed a hybrid memory and prediction strategy based on the similarity between historical and current environments. Rong et al. [36] introduced a multimodel prediction method. Wang et al. [37] proposed a grey modeling approach. Rambabu et al. [38] developed a hybrid expert system. Cao et al. [39] presented a support vector regression model. Ma et al. [40] suggested a feature information prediction strategy. Jiang et al. [41] proposed a DMOEA based on imbalance transfer learning at the knee point, selecting only high-quality solutions for knowledge transfer to avoid ineffective transfers and reduce computational costs. Subsequently, in [42], a method combining sample geodesic flow (SGF) and a memory mechanism focused on exploring potential change patterns in different environments. Li et al. [43] introduced a DMOEA that reduces negative transfer learning through clustering.

Inspired by these studies, this paper designs the SCP-DMOEA to further explore how to efficiently convey the most valuable historical information to the current environment in order to be used for predicting the initial population.

III. SPECTRAL CLUSTERING PREDICTION STRATEGY (SCP)

A. SPECTRAL CLUSTERING OVERVIEW

Spectral Clustering refers to the algorithm that clusters points using eigenvectors of matrices derived from the data [44], [46], [47], [48], [49], [50], [51], [52]. It is a graph-based method that transforms the clustering problem into a combinatorial optimization problem. Specifically, it represents the vertices V of a graph as each sample in the dataset and connects the edges E with weights $\boldsymbol{W}$ corresponding to the similarity between the sample data. This allows the entire dataset and the relationships between data samples to be represented as an undirected weighted graph $\boldsymbol{G} = (V, W)$. Therefore, graph partitioning method can be used to solve the clustering problem of sample data sets. Further, the graph partitioning problem can be addressed through the spectral decomposition of similarity matrices or Laplacian matrices.

B. ESTABLISHMENT OF SPECTRAL CLUSTERING PREDICTION MODEL

1) THE UTILIZED SPECTRAL CLUSTERING ALGORITHM

Spectral clustering has different specific implementations based on various graph partitioning criteria. This paper adopts the minimum cut set partitioning criterion, generates the similarity matrix $\boldsymbol{W}$ using a fully connected approach based on the Gaussian kernel function (RBF), applies the *Ncut* method for graph cutting, uses the symmetric normalized Laplacian matrix representation, and employs k-means as the final clustering method.

If we cluster a current population of size N, $PS^t = \{x_1^t, x_2^t, \cdots, x_N^t\}$ into k subgroups $C_1^t, C_2^t, \cdots, C_k^t$. Here are the specific implementation steps:

a. Construct a similarity matrix W_{ij}^t for all individuals in the population PS^t, with the construction formula shown in Equation (2):

$$W_{ij}^t = \exp(-\frac{\left\| x_i^t - x_j^t \right\|_2^2}{2\sigma^2}) \tag{2}$$

where σ is the Gaussian similarity bandwidth, with $\sigma = 1$.

b.For any point v_i^t in the graph, its degree d_i^t is defined as the sum of the total weights of all edges that are connected to it, that is, the total of all elements in the i th row of the matrix W_{ij}^t. Thus, the generation formula for the angle matrix D^t is shown in Equation (3):

$$d_i^t = \sum_{j=1}^{N} w_{ij}^t \tag{3}$$

c.Construct a symmetric normalized Laplacian matrix L_{sym}^t, with the construction formula shown in Equation (4):

$$L_{sym}^t = D^{t-1/2} L^t D^{t-1/2} = I^t - D^{t-1/2} W_{ij}^t D^{t-1/2} \tag{4}$$

d. Calculate the eigenvalues of L_{sym}^t, sort them in ascending order, take the first k eigenvalues, and then compute the eigenvectors corresponding to the first k eigenvalues $u_1^t, u_2^t, \ldots, u_k^t$. Construct a matrix $U^t = \{u_1^t, u_2^t, \ldots, u_k^t\}$ using the k column vectors, $U^t \in R^{n \times k}$.

e. Let $y_i^t \in R^k$ be the vector in the i th row of U^t, where $i = 1, 2, \ldots, N$, and normalize the vectors $y_i^t \in R^k$ one by one.

f. Use the k-means algorithm to cluster $Y^t = \{y_1^t, y_2^t, \ldots, y_N^t\}$ into clusters $A_1^t, A_2^t, \cdots, A_k^t$.

e. Output the clusters $C_1^t, C_2^t, \cdots, C_k^t$. Where $C_i^t = \{j \mid y_j^t \in A_i^t\}$.

2) SUBPOPULATION CENTER POINT PREDICTION

The changes of the environment in several adjacent times are similar but not identical. If the differences between them are ignored during the prediction process, the accuracy of the prediction will be decreased.Therefore, based on fully considering the similarities between environmental changes and taking into account the differences, this paper adopts the AR(p) model to predict the central points of each subclass in the new environment [53], [54]. Here, $p = 3$, the length of the historical mean point sequence $M = 23$.

Let c_k^t be the center point of the $k-th$ subpopulation at time t, then the calculation formula is shown in Equation (5).

$$c_k^t = \frac{1}{\left|C_k^t\right|} \sum_{x_k^t \in C_k^t} x_{k,g}^t \tag{5}$$

where $\left|C_k^t\right|$ is the cardinality of C_k^t, and $x_{k,g}^t$ represents the $g-th$ individual of the $k-th$ subpopulation at time t.

The prediction formula for subpopulation center point is shown in Equation (6).

$$c_k^{t+1} = \sum_{j=0}^{p-1} a_j c_k^{t-j} + N(0, \sigma_k^c) \tag{6}$$

where a_j is the parameters of the $AR_{(p)}$ model, estimated using the least squares method, and $(0, \sigma_k^c)$ is a normal distribution with variance is σ_k^c.

The formula for calculating the variance is presented in Equation (7).

$$\sigma_k^c = \frac{1}{M-p} \sum_{k=t-M+p}^{t} [c_k^{t+1} - \sum_{j=0}^{p-1} a_j c_k^{t-j}]^2 \tag{7}$$

3) SUBPOPULATION SHAPE PREDICTION

Under relaxed conditions, and according to the Karush-Kuhn-Tucker criterion, the PS of a MOP with m objectives forms a piecewise continuous m-1 dimensional manifold [55], [56]. Therefore, in a sense, the shape of the PS at adjacent moments is similar. Thus, it is possible to predict the shape of the subpopulation in the new environment based on the subpopulation from the previous moment.

The formula for predicting the shape of the subpopulation is shown in Equation (8).

$$\tilde{x}_{k,g}^{t+1} = \tilde{x}_{k,g}^t + \left(c_k^{t+1} - c_k^t\right) + N(0, \sigma_{k,g}^m) \tag{8}$$

Where $\tilde{x}_{k,g}^{t+1}$ represents the $g-th$ individual in the subpopulation at time $t+1$, and $N(0, \sigma_{k,g}^m)$ denotes a normal distribution with a variance of $\sigma_{k,g}^m$.

The formula for calculating the variance is shown in Equation (9).

$$\sigma_{k,g}^m = \frac{1}{\left|C_k^t\right|} \sum_{x_{k,g}^t \in C_k^t} (\tilde{x}_{k,g}^t - c_k^t)^2 \tag{9}$$

C. SCP-DMOEA DETAILED DESCRIPTION

Algorithm 1 presents the SCP-DMOEA framework, which embeds SCP into the NSGA-III algorithm framework. Once a change in the environment is detected, SCP will be used to predict the initial population size, P^t, for the current environment, $P^t = \{C_1^t, C_2^t, \cdots, C_k^t\}$. Combining formulas (6) and (8), we can deduce P^t, and then the final population PS^t for the current environment is obtained through NSGA-III.

D. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational cost of SCP-DMOEA is primarily caused by the SMOEA and SCP used, as shown in Algorithm 1. First, the computational cost of SCP mainly arises from spectral clustering and subclass prediction. For spectral clustering, the computational cost mainly comes from the construction of the similarity matrix, calculation of the Laplacian matrix, eigenvalue decomposition, and k-means clustering. The construction of the similarity matrix W, as shown in equation (2), requires calculating the similarity between all

TABLE 1. Modified HE2.

Instance	Definition	Remark
modified HE2 (Type I)	$f_1(x_1) = x_1,$ $f_2(x_1,\cdots,x_n) = 1 + g \cdot h,$ $g(x_2,\cdots,x_n) = 1 + \sum_{i=2}^{n}\left(x_i - G(t)\right)^2,$ $h(f_1,g) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\cdot\sin(10\pi f_1)$ $G(t) = \begin{cases} \lvert\sin(0.1\pi t)\rvert, x_1 \in [0,0.5] \\ \lvert\cos(0.1\pi t)\rvert, x_1 \in (0.5,1] \end{cases}$ $where:\ x_i \in [0,1]$	PS: [0, 0.5] moves upward, and (0.5, 1] moves downward. PF: disconnected. $f_1 = x_1,$ $f_2 = 1 + h,$ $h = 1 - \sqrt{f_1} - f_1 \cdot \sin(10\pi f_1)$ $G(t) = \begin{cases} \lvert\sin(0.1\pi t)\rvert, x_1 \in [0,0.5] \\ \lvert\cos(0.1\pi t)\rvert, x_1 \in (0.5,1] \end{cases}$ $where:\ x_i \in [0,1]$

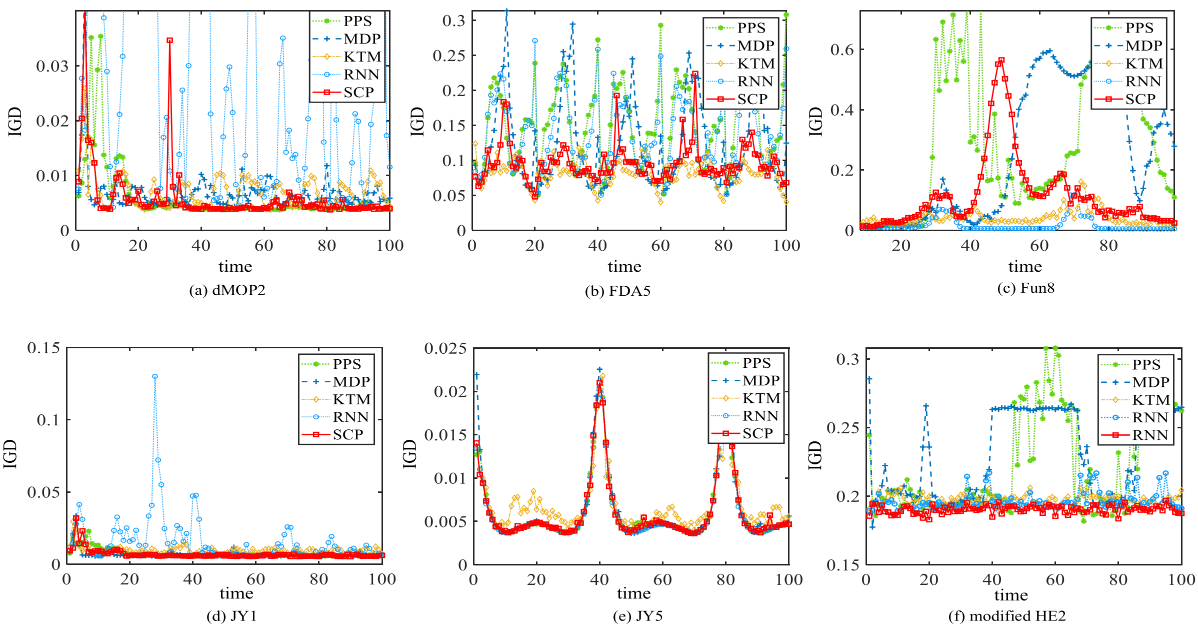


FIGURE 1. Average IGD values across 20 runs over time for PPS, MDP, KTM, RNN and SCP on (a) dMOP2 (b) FDA5 (c) Fun8 (d) JY1 (e) JY5 and (f) modified HE2.

Algorithm 1 Framework of SCP-DMOEA

Input: the size of PS, N; Number of decision variables, n; Number of objective functions, m; Number of clusters in spectral clustering, k;Gaussian similarity bandwidth, σ; The number of individuals used for change detection, N_d; The maximal number of iterations, $g_{\max}$;
Output: offspring population PS^t;
1: Detect environmental changes. If change, go to step 2; if no change, go to step 4;
2: Generate initial population P^t using the formula (6) and (8);
3: Reproduction of the offspring population PS^t by NSGA-III;
4: Check termination conditions, If $g_t > g_{\max}$, output PS^t, and end; or $g_t = g_{t+1}$, go to Step 1;

individuals. The computational complexity is $O(N^2)$, where N is the population size. The computation of the Laplacian matrix involves matrix operations, typically with a complexity of $O(N^2)$. Performing eigenvalue decomposition on the Laplacian matrix also usually has a complexity of $O(N^2)$. The complexity of clustering the eigenvectors using the k-means algorithm is $O(NIk)$, where k is the number of clusters and $I$is the number of iterations. Therefore, the worst-case overall computational complexity of spectral clustering is $O(N^2 + N^2 + N^2 + NIk) \approx O(N^2)$.

For subclass prediction, the computational complexity of forecasting the time series formed by the k subclass centroids using the AR(p) model is $O(pk)$. For predicting subclass P_k, the computational complexities of calculating the mean and covariance are $O(nkD)$ and $O(nkD^2)$, respectively, where D is the number of decision variables. Therefore, the worst-case complexity for subclass prediction is $O(pk + n_k D + n_k D^2) \approx O(ND^2)$. In summary, the worst-case computational complexity of SCP is $O(N^2 + ND^2) \approx O(N^2)$.

TABLE 2. Results OF MIGD obtained by PPS, MDP, KTM, RNN AND SCP over 20 runs.

Instance	Strategy	$0 \leq t \leq 100$	$0 \leq t \leq 20$	$21 \leq t \leq 60$	$61 \leq t \leq 100$
FDA1	PPS	0.0054	0.0110	**0.0046**	**0.0043**
	MDP	0.0058	**0.0057**	0.0052	0.0053
	KTM	**0.0053**	0.0070	0.0049	0.0047
	RNN	0.0073	0.0077	0.0073	0.0066
	SCP	0.0057	0.0081	0.0052	0.0049
FDA2	PPS	0.0064	**0.0046**	**0.0060**	0.0077
	MDP	0.0073	0.0049	0.0085	0.0073
	KTM	0.0083	0.0059	0.0095	0.0084
	RNN	0.0077	**0.0046**	0.0066	0.0083
	SCP	**0.0057**	0.0049	0.0063	**0.0054**
FDA3	PPS	0.0163	0.0105	0.0220	0.0136
	MDP	0.0136	0.0115	0.0128	0.0156
	KTM	**0.0084**	**0.0091**	**0.0085**	**0.0080**
	RNN	0.0128	0.0232	0.0136	0.0118
	SCP	0.0175	0.0106	0.0242	0.0143
FDA4	PPS	0.1012	0.1037	0.1026	0.0987
	MDP	0.1010	0.0962	0.1034	0.1009
	KTM	0.0647	**0.0342**	0.0673	0.0649
	RNN	0.1015	0.0996	0.1025	0.1016
	SCP	**0.0616**	0.0673	**0.0594**	**0.0609**
FDA5	PPS	0.1527	0.1424	0.1527	0.1580
	MDP	0.1231	0.1266	0.1197	0.1247
	KTM	**0.0892**	**0.0882**	0.0947	**0.0889**
	RNN	0.1500	0.1098	0.1513	0.1516
	SCP	0.0957	0.0961	**0.0928**	0.0984
dMOPl	PPS	0.0066	0.0087	0.0052	0.0069
	MDP	0.0064	0.0078	0.0061	0.0061
	KTM	0.0054	0.0082	0.0051	0.0055
	RNN	**0.0051**	**0.0056**	**0.0043**	0.0044
	SCP	0.0053	0.0095	0.0044	**0.0042**
dMOP2	PPS	0.0060	0.0135	**0.0041**	**0.0041**
	MDP	0.0068	**0.0090**	0.0063	0.0062
	KTM	0.0078	0.0093	0.0066	0.0065
	RNN	0.0351	0.0609	0.0518	0.0220
	SCP	**0.0059**	0.0100	0.0053	0.0045
JY1	PPS	0.0072	0.0124	0.0060	0.0060
	MDP	0.0070	**0.0096**	0.0064	0.0064
	KTM	0.0094	0.0114	0.0089	0.0090
	RNN	0.0156	0.0172	0.0206	0.0098
	SCP	**0.0069**	0.0110	**0.0059**	**0.0059**
JY2	PPS	0.0070	0.0126	0.0052	0.0059
	MDP	0.0065	**0.0083**	0.0062	0.0064
	KTM	0.0085	0.0104	0.0081	0.0080
	RNN	0.0142	0.0153	0.0174	0.0104
	SCP	**0.0061**	0.0116	**0.0048**	**0.0047**
JY5	PPS	0.0064	0.0057	0.0065	**0.0064**
	MDP	0.0065	0.0062	0.0065	0.0065
	KTM	0.0071	0.0071	0.0071	0.0071
	RNN	0.0062	0.0056	0.0063	**0.0064**
	SCP	**0.0059**	**0.0046**	**0.0047**	**0.0064**
Fun7	PPS	0.0205	0.0335	0.0245	0.0101
	MDP	0.0181	0.0292	0.0215	0.0092
	KTM	0.0553	0.0362	0.0589	0.0763
	RNN	0.0189	**0.0173**	0.0169	0.0097
	SCP	**0.0137**	0.0248	**0.0130**	**0.0090**
Fun8	PPS	0.2577	0.0118	0.2699	0.3685
	MDP	0.2372	0.0117	0.1480	0.4390
	KTM	0.0480	0.0281	0.0399	0.0481
	RNN	**0.0189**	**0.0089**	**0.0171**	**0.0203**
	SCP	0.1071	0.0174	0.1778	0.0812

TABLE 2. ***(Continued.)*** **Results OF MIGD obtained by PPS, MDP, KTM, RNN AND SCP over 20 runs.**

HE2	PPS	0.1915	0.1945	0.1874	0.1942
	MDP	0.1894	0.2077	0.1988	0.1884
	KTM	0.1884	**0.1822**	0.1912	0.1897
	RNN	0.1873	0.2014	0.1837	0.1839
	SCP	**0.1856**	0.1940	**0.1835**	**0.1836**
modified HE2	PPS	0.2223	0.2005	0.2213	0.2342
	MDP	0.2307	0.2125	0.2331	0.2374
	KTM	0.1974	0.1984	0.1967	0.1976
	RNN	0.1969	0.1995	0.1963	**0.1961**
	SCP	**0.1966**	**0.1935**	**0.1954**	0.1992

Next, the computational complexity of NSGA-III is $O(mN^2)$, as stated in [45], where m represents the number of objectives. Therefore, the worst-case overall computational complexity of the dynamic response mechanism in SCP-DMOEA is $O(N^2)$.

IV. TEST INSTANCES AND PERFORMANCE METRIC

14 benchmark functions with two or three objectives, including FDA1-FDA5 [17], dMOP1-dMOP2 [57], Fun7, Fun8 [25], HE2 [58] and modified HE2, and JY1, JY2, JY5 [59], are employed in experiment. These test functions exhibit various characteristics, including multimodality, convexity, mixed convexity and concavity, discontinuity, and nonuniformity. Modified HE2 is a Type I problem where only PS is dynamic. The PS varies over time as [0, 0.5] moves upward, and (0.5, 1] moves downward. The details of modified HE2 is listed in Table 1.

For DMOPs, when $t = (1/n_t)\lfloor \tau/\tau_t \rfloor$ occurs, a new environment emerges. In this case, τ, n_t, and τ_t represent the total number of generations, the severity, and the frequency of change, respectively. Intuitively, a smaller τ_t indicates that the environment changes more rapidly.

To ensure a fair comparison of algorithm performance, the mean inverted generational distance (MIGD) [24], [61] is employed as the performance metric. MIGD can effectively reflect the diversity and convergence of the PS obtained in the entire environment. The lower the MIGD value, the better the performance of the algorithm.

Let the true front of the DMOP be PF_{tr}^t, then the method for calculating the MIGD value is provided in Equation (10).

$$MIGD = \frac{1}{T}\sum_{t\in T}\frac{\sum_{i=1}^{|PF_{tr}^t|}\min D(PF^t, r_i^t)}{|PF_{tr}^t|} \tag{10}$$

where T represents the number of environments encountered in a single run. $D(PF^t, r_i^t)$ is the Euclidean distance between the i-th point in PF_{tr}^t and the PF^t obtained in the objective space through the algorithm.

In PF_{tr}^t, for the optimization problem of two-objective, 1000 uniformly distributed points are taken, and for three-objective optimization problems, 10000 uniformly distributed points are taken. Each experiment is performed independently 20 times, followed by an analysis of the statistical results.

V. EXPERIMRNTAL RESULTS AND ANALYSIS

A. COMPARISON OF ALGORITHMS AND PARAMETER SETTINGS

In order to conduct a comprehensive comparison with the mainstream strategies for responding to changes, we selected four common DMOEAs, including PPS-DMOEA [24], MDP-DMOEA [25], KTM-DMOEA [28], and RNN-DMOEA [29], are employed for comparison. All the specific parameters of the DMOEA being compared were configured according to the recommendations in their original literatures [24], [25], [28], and [29].

Furthermore, among all the compared algorithms, NSGA-III

was used as the static optimization algorithm within each environment to achieve a fair comparison. In NSGA-III, the crossover probability was set to 0.7, while the mutation probability was set to 0.3. The population sizes are configured to be 100 for bi-objective test instances and 200 for three-objective ones. To detect environmental changes, 5% of the individuals are selected. For all benchmarks, the search space dimension is 10, and each run involves 100 environmental changes.Here $n_t = 10$, $\tau_t = 20$.

B. COMPARISON OF PERFORMANCE EVALUATION RESULTS

Dynamic multiobjective optimization problems and most environmental prediction tasks generally exhibit stagewise evolution characteristics. In the initial stage, data are relatively scarce, and model training is insufficient, making it crucial to test the algorithm's adaptability and noise resistance. As the process enters the intermediate stage, the data accumulate, and the model begins to stabilize; Therefore, the focus shifts to evaluating the algorithm's convergence speed and transitional adaptation ability. In the later stage, with ample data, predictive methods mature, and attention turns to their long-term stability and generalization performance. An in-depth analysis was conducted on how the quantity and quality of stored historical information affect the prediction strategies, this paper divides the entire dynamic process into three periods: $0 \leq t \leq 20$ (initial), $21 \leq t \leq 60$ (intermediate), and $61 \leq t \leq 100$ (later), while also presenting results for the overall period $0 \leq t \leq 100$, with the environmental dynamics changing to $n_t = 10$, $\tau_t = 20$. By segmenting

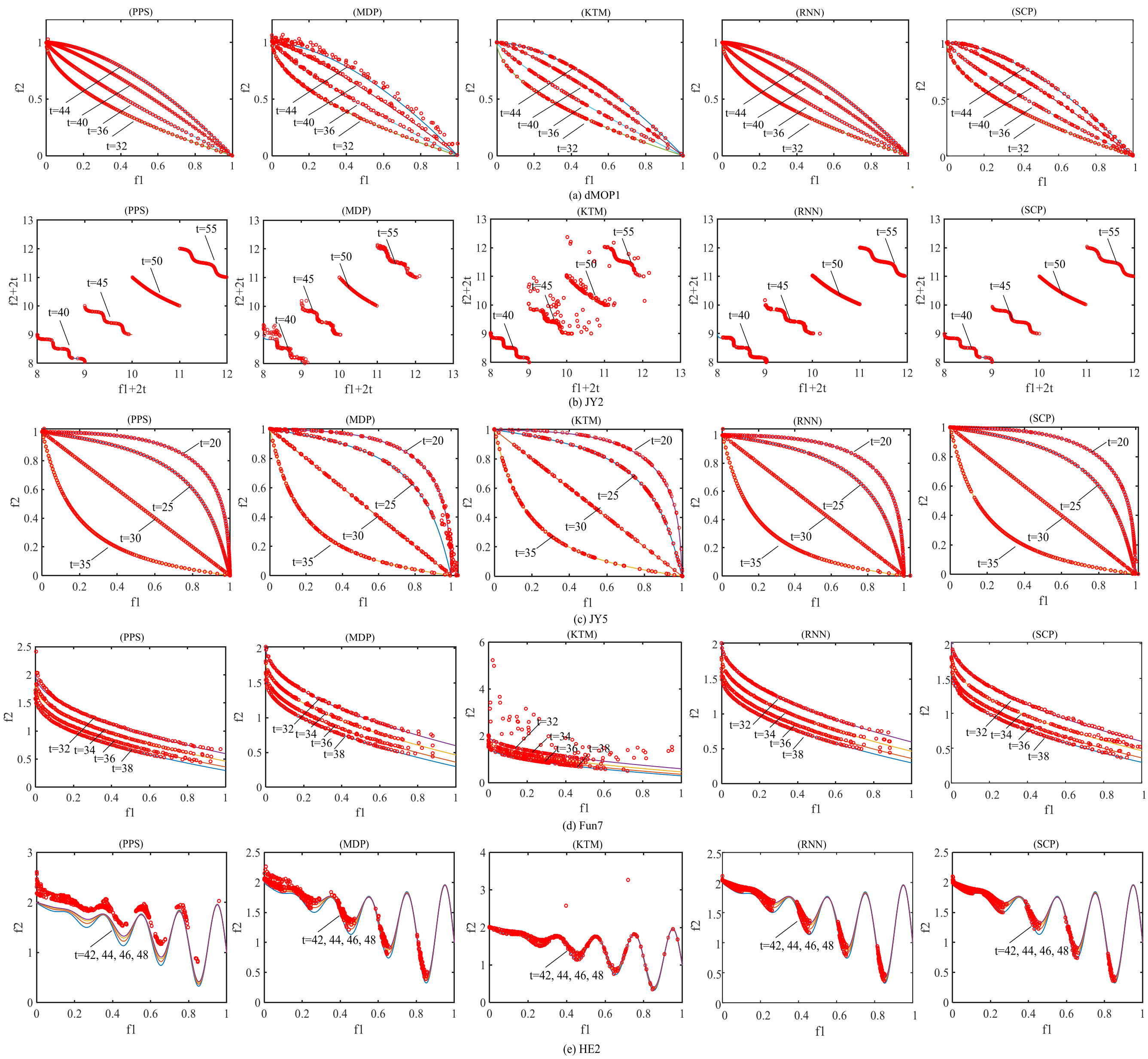


FIGURE 2. Initial populations obtained by PPS, MDP, KTM, RNN and SCP at different times, respectively with the lowest MIGD values among 20 runs on (a) dMOP1 (b) JY2 (c) JY5 (d) Fun7 (e) HE2.

algorithm performance according to these stages in this paper, we are able to reveal the advantages and disadvantages of each algorithm at different phases directly and investigate the impact of historical data accumulation on prediction performance. Moreover, this type of segmentation analysis provides valuable insights for algorithm selection in practical applications.

The performance of the MIGD is compared across these segments. Specifically, Table 2 presents the statistical results of each algorithm's MIGD after 20 independent runs in the respective time intervals, and Figure 1 illustrates the dynamic curves of the MIGD values over time for PPS, MDP, KTM, RNN and SCP on (a) dMOP2, (b) FDA5, (c) Fun8, (d) JY1, (e) JY5, and (f) modified HE2. By evaluating algorithm performance in different evolutionary stages, this segmented analysis comprehensively reveals the strengths and weaknesses of various prediction strategies, providing valuable guidance for algorithm optimization and practical system selection.

It can be seen from Table 2 that, the test functions for PSs translate over time and the mixed PFs in terms of convexity and concavity that change over time, such as the FDA test suite and dMOPs test suite, are considered. When $0 \leq t \leq 100$, SCP and KTM have certain advantages in handling three-dimensional test functions. Furthermore, for FDA3, the prediction performance of KTM is significantly

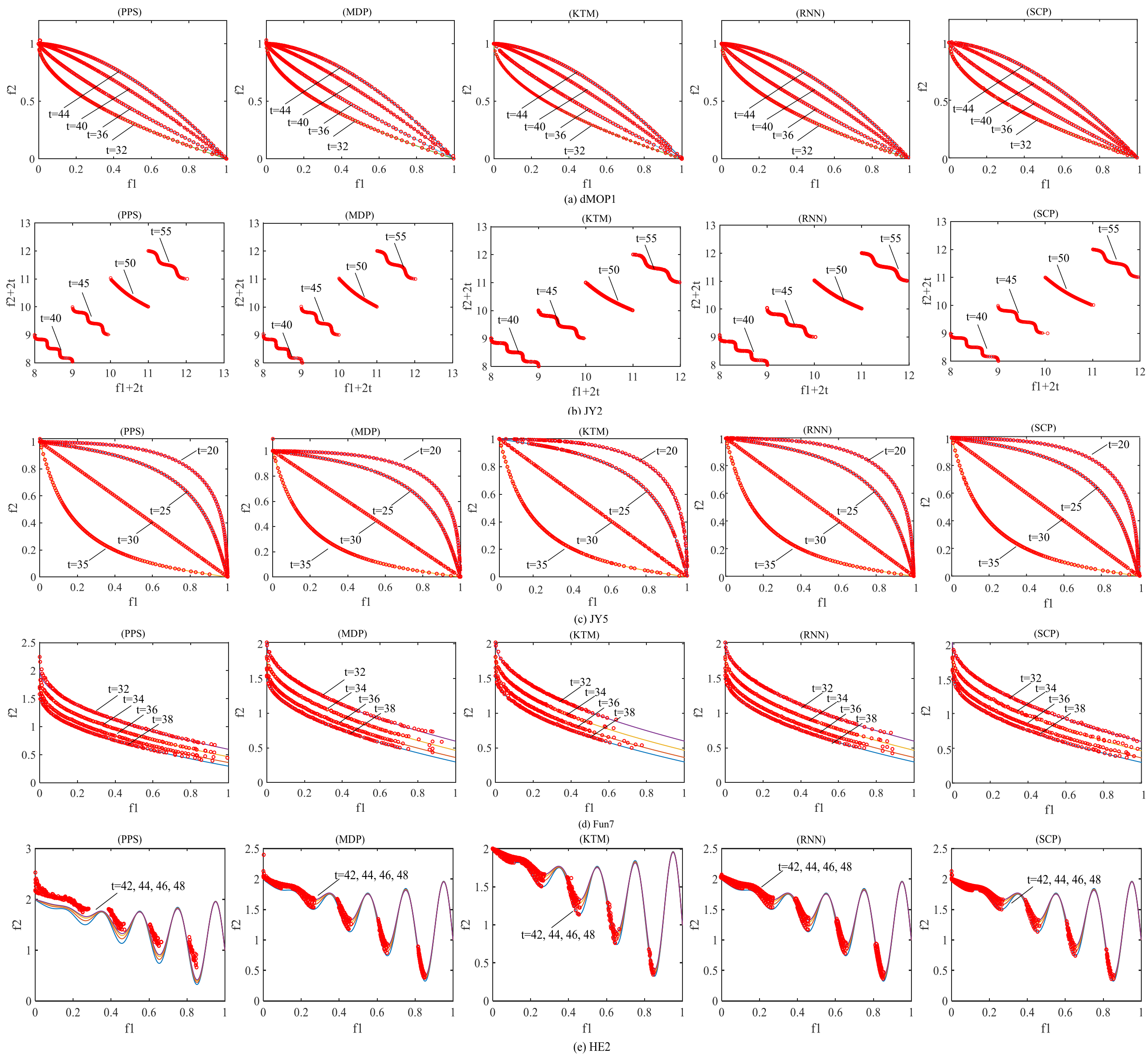


FIGURE 3. **Final populations obtained by PPS, MDP, KTM, RNN and SCP at different times, respectively with the lowest MIGD values among 20 runs on (a) dMOP1 (b) JY2 (c) JY5 (d) Fun7 (e) HE2.**

better than those of the other four methods, mainly because the distribution of the PS under different environments for FDA3 is highly suitable for predictions using the KTM method. For dMOP2, the prediction performance of RNN is significantly worse than those of the other four methods, primarily because RNN may overfit local (non)linear fluctuations. During the time periods $0 \leq t \leq 20$, $21 \leq t \leq 60$, and $61 \leq t \leq 100$, different algorithms are influenced by historical information to varying degrees, leading to discrepancies in prediction performance across different stages. PPS and RNN are particularly affected by historical information, resulting in relatively weak prediction performance in the early stages of environmental change. As seen in Figure 1, for dMOP2, the performance curves of PPS, MDP, KTM, and SCP fluctuate at the beginning of the run but stabilize when $21 \leq t \leq 100$, with the performance curve of SCP noticeably lower than that of PPS, MDP, and KTM. For FDA5, the performance curves of PPS, MDP, RNN, and SCP all experience fluctuations when $0 \leq t \leq 100$, and the fluctuations of SCP significantly decrease when $t \geq 61$.

As can be seen from Table 2, for test functions with more complex PF variations compared to the FDA test suite and the dMOPs test suite, such as JY1, JY2, and JY5. (where JY1 and JY2 have PS changing over time, and the objective vector oscillates among several modes). When $0 \leq t \leq 100$, the prediction performance of SCP is superior to those of the other four comparative methods. For JY1 and JY2, the prediction performance of RNN is significantly worse than that

TABLE 3. Results of MIGD Obtained by SCP with Different Numbers of clusters in spectral clustering *k* over 20 Runs.

numbers of clusters in spectral clustering k	1	2	3	4	**5**	6	7	8
dMOP2	0.0155	0.0123	0.0108	0.0080	**0.0059**	0.0082	0.0121	0.0132
JY1	0.0084	0.0072	0.0069	0.0064	**0.0067**	0.0069	0.0068	0.0070

TABLE 4. Results of MIGD Obtained by SCP with Different Numbers of AR Model Order *p* over 20 Runs.

AR model order p	1	2	**3**	4	5	6
dMOP2	0.0369	0.0073	**0.0059**	0.0057	0.0056	0.0055
JY1	0.0384	0.0072	**0.0067**	0.0068	0.0067	0.0068

of the other four methods, mainly because RNN may overfit local (non)linear fluctuations. Moreover, when $0 \leq t \leq 20$, due to the relatively complex PF changes, the prediction performance of all prediction strategies is comparatively poor when historical information is insufficient. In Figure 1, for JY1, when $0 \leq t \leq 100$, the performance curve of RNN fluctuates to varying degrees, with significant fluctuations when $21 \leq t \leq 60$. In contrast, the performance curves of the other prediction strategies remain overall stable.

As shown in Table 2, for Fun7, the prediction performance of SCP is superior to those of the other four comparative methods during different time periods. For Fun8, RNN shows significantly better prediction performance than the other four methods during different time periods. The reason for this phenomenon is that the PS of Fun7 rotates along the origin of the coordinate system, with its center point constantly changing over time, whereas the PS of Fun8 rotates around the PS center, a fixed center point, over time. Furthermore, how to set reasonable similarity weights during spectral clustering is crucial for effectively handling DMOPs with correlated decision variables. As seen in Figure 1, for Fun8, the performance curves of KTM and RNN show overall stable variations, with RNN performing notably worse than the other four algorithms. The performance curve of PPS fluctuates to varying degrees throughout the dynamic changes, while MDP experiences significant fluctuations when $61 \leq t \leq 100$, and SCP exhibits larger fluctuations between $40 \leq t \leq 60$.

From Table 2, it can be observed that for the discontinuous PF test problems HE2 and the improved HE2, the prediction performances of KTM, RNN, and SCP are significantly superior to those of PPS and MDP across environmental changes and various stages of these changes. Furthermore, SCP outperforms KTM and RNN slightly, indicating that SCP is well-suited for handling discontinuous PF test problems. Additionally, as seen in Figure 1, for the improved HE2, when $0 \leq t \leq 100$, the performance curves of KTM, RNN, and SCP are notably lower than those of PPS and MDP, and their overall trends tend to stabilize.

To visualize the quality of the obtained populations, we also depict the initial and final populations with the lowest MIGD metric values at different times for (a) dMOP1, (b) JY2, (c) JY5, (d) Fun7, and (e) HE2 in Figures 2 and 3, respectively.

These figures show that, while the approximation of PFs for dMOP1 and MDP is not as good as other prediction strategies, the approximation of PFs for JY2 and Fun7 by KTM is not as good as other prediction strategies, and the approximation of PFs for JY5 by MDP and KTM is not as good as other prediction strategies. However, after 20 generations, they can all approximate the PF well. But for HE2, even after 20 generations, PPS still cannot approximate the PFs, mainly because the set computational cost is not sufficient to solve a static multi-objective optimization problem.

1) *Sensitivity Study*

In SCP, we needs to determine two important parameters, namely the number of spectral clusters k in SCP and the order p of the AR model. The value of k affects the modeling accuracy of the PS: a larger k value leads to a finer classification of PS types but increases sensitivity to noise. The value of p affects the ability of the model to capture trends across various dimensions of the subclass centroids: higher-order models are prone to overfitting, while lower-order models may overlook nonlinear variations. In this section, the sensitivity of SCP to parameters k and p is investigated by applying different values of k and p in the test instances dMOP2 and JY1. In order to facilitate a direct comparison of the effects produced by different values of k and p on SCP performance, Tables 3 and 4 list the MIGD statistical results obtained from 20 independent runs for fixed $p = 3$ with varying k and fixed k= 5 with varying p, with lower values indicating better overall performance.

From Table 3, it can be seen that the performance of SCP varies across different problems with respect to the number of spectral clusters k. dMOP2 is highly sensitive to changes in k, as k increases, the MIGD value first gradually decreases and then gradually increases again. JY1, on the other hand, exhibits relatively robust performance. Overall, when k is set to 5, SCP achieves the best overall performance. Thus, this paper recommends setting k to 5. From Table 4, it can be observed that regarding the order p of the AR model, as p increases, the MIGD values for dMOP2 and JY1 decrease rapidly, demonstrating high sensitivity. This indicates that the selection of parameter p has a significant impact on the optimization outcome, lower orders may not adequately capture the features of the data, while higher orders can significantly enhance the stability of the results. Overall, when p is set to 3, SCP achieves the best overall performance. Thus, this paper recommends setting p to 3. With this configuration, the test problems can ensure good convergence while avoiding excessively high computational complexity.

VI. CONCLUSION AND FUTURE WORK

Considering that spectral clustering can implicitly utilize the connectivity and regularity of PS to achieve its clustering analysis, a dynamic multi-objective evolutionary algorithm based on spectral clustering prediction (SCP-DMOEA) is designed. The algorithm uses spectral clustering to group the population into representative subclasses, then predicts the center point and shape of each subclass based on historical information, thereby generating an entirely new initial population after environmental changes. The predicted new population can effectively approximate the PS, accelerating the algorithm's convergence in the new environment.

We used 14 benchmark functions, dividing the entire dynamic process into three periods: initial, intermediate, and later, while also presenting results for the overall period. We compared the performance of SCP with four typical prediction strategies (PPS, MDP, KTM, and RNN) from the perspective of the quantity and quality of historical information in different stages and the entire dynamic process. In addition, we evaluated the time cost of SCP-DMOEA and performed a sensitivity analysis of its two key parameters.

The experimental results show that the SCP-DMOEA proposed in this paper can indeed enhance the prediction efficiency and accelerate the convergence speed of the algorithm when solving DMOPs with different variation characteristics. It performs better on most test problems when dealing with the following situations: translation or rotation changes of the PS; correlations between decision variables; time-varying mixed PFs, including changes in convexity and concavity. When dealing with optimization problems with more complex changes in PFs, such as the position of the PF continuously changes over time, the objective vector oscillates among several modes, and the PFs are discontinuous, it exhibits superior performance compared to the comparative algorithms. However, some limitations still exist. For DMOPs with correlations between decision variables, the difficulty of setting similarity weights during spectral clustering will increase, thus affecting prediction accuracy. For example, on Fun8, SCP performs worse than RNN. Furthermore, in the process of generating an initial population through manifold prediction, the weak correlation between submanifolds and the limited ability to transfer historical features cause SCP to perform worse than KTM on FDA3. In the future, SCP-DMOEA will be further improved to better address complex DMOPs in the real world, such as the response speed of controllers and power supply issues.

ACKNOWLEDGMENT

Data Availability statement: The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for Publication: Informed consent was obtained from all individual participants included in the study.

Conflict of Interests: All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Authors' contributions: Zhao-Jun Sheng (Methodology, Writing-original draft). Erchao Li (Writing review and editing, Conceptualization, and Supervision).

APPENDIX

TABLE OF ABBREVIATION

Full forms	Abbreviation
dynamic multi-objective optimization problems	DMOPs
evolutionary algorithm	EA
spectral clustering prediction	SCP
pareto optimal front	PF
pareto optimal set	PS
population-based prediction method	PPS
multi-directional prediction method	MDP
kernel transfer principles	KTP
kernel mean signature	KMS
recurrent neural networks	RNN
mean inverted generational distance	MIGD
non-dominated sorting genetic algorithms	NSGA-III

REFERENCES

[1] K. Deb, N. U. B. Rao, and S. Karthik, "Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling," in *Proc. EMO* (Lecture Notes in Computer Science), vol. 4403. Berlin, Germany: Springer, 2007, pp. 803–817, doi: 10.1007/978-3-540-70928-2_60.

[2] M. B. Abello, L. T. Bui, and Z. Michalewicz, "An adaptive approach for solving dynamic scheduling with time-varying number of tasks—Part II," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 1711–1718, doi: 10.1109/CEC.2011.5949821.

[3] M. B. Abello, L. T. Bui, and Z. Michalewicz, "An adaptive approach for solving dynamic scheduling with time-varying number of tasks—Part I," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 1703–1710, doi: 10.1109/CEC.2011.5949820.

[4] S.-L. Jiang, Q. Liu, I. D. L. Bogle, and Z. Zheng, "A self-learning based dynamic multi-objective evolutionary algorithm for resilient scheduling problems in steelmaking plants," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 2, pp. 832–845, Apr. 2023, doi: 10.1109/TASE.2022.3168385.

[5] Z. Zhou, J. Chang, Z. Hu, J. Yu, and F. Li, "A modified PSO algorithm for task scheduling optimization in cloud computing," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 24, p. e4970, 2018.

[6] A. K. Hutzschenreuter, P. A. N. Bosman, and H. La Poutré, "Evolutionary multiobjective optimization for dynamic hospital resource management," in *Proc. EMO* (Lecture Notes in Computer Science), vol. 5467. Berlin, Germany: Springer, 2009, pp. 320–334, doi: 10.1007/978-3-642-01020-0_27.

[7] J. Tang, S. Alam, C. Lokan, and H. A. Abbass, "A multi-objective evolutionary method for dynamic airspace re-sectorization using sectors clipping and similarities," in *Proc. IEEE Congr. Evol. Comput.*, 2012, pp. 1–8, doi: 10.1109/CEC.2012.6253008.

[8] A. Isaacs, V. Puttige, T. Ray, W. Smith, and S. Anavatti, "Development of a memetic algorithm for dynamic multi-objective optimization and its applications for online neural network modeling of UAVs," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IEEE World Congr. Comput. Intell.)*, Jun. 2008, pp. 548–554, doi: 10.1109/IJCNN.2008.4633847.

[9] L. T. Bui and Z. Michalewicz, "An evolutionary multi-objective approach for dynamic mission planning," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–8.

[10] Y. Zhang, T. Zhao, L. Wu, and Z. Cui, "Interval many-objective dynamic charging planning in wireless rechargeable sensor networks," *Concurrency Comput., Pract. Exper.*, vol. 38, no. 19, p. e8150, 2024.

[11] P. P. Y. Wu, D. Campbell, and T. Merz, "Multi-objective four-dimensional vehicle motion planning in large dynamic environments," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 41, no. 3, pp. 621–634, Jun. 2011.

[12] P. Di Barba, "Dynamic multiobjective optimization: A way to the shape design with transient magnetic fields," *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 962–965, Jun. 2008.

[13] F. V. C. Martins, E. G. Carrano, E. F. Wanner, R. H. C. Takahashi, and G. R. Mateus, "A dynamic multiobjective hybrid approach for designing wireless sensor networks," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 1145–1152.

[14] W. Ke, Y. Bai, H. Li, K. Chen, and Q. Yuan, "Control of stepping downstairs for humanoid robot based on dynamic multi-objective optimization," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 5, p. e5999, 2021.

[15] K. Kim, R. I. McKay, and B. R. Moon, "Multiobjective evolutionary algorithms for dynamic social network clustering," in *Proc. GECCO*, 2010, pp. 1179–1186.

[16] B. S. Rabil, R. Sabourin, and E. Granger, "Watermarking stack of grayscale face images as dynamic multi-objective optimization problem," in *Proc. MDA*, 2011, pp. 63–77.

[17] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, Oct. 2004, doi: 10.1109/TEVC.2004.831456.

[18] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005, doi: 10.1109/TEVC.2005.846356.

[19] K. Miettinen, *Nonlinear Multiobjective Optimization*. Dordrecht, The Netherlands: Kluwer Academic, 1999.

[20] J. Branke, *Evolutionary Optimization in Dynamic Environments*. Dordrecht, The Netherlands: Kluwer Academic, 2002.

[21] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

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

[23] P. Gholamnezhad, A. Broumandnia, and V. Seydi, "An improved model-based evolutionary algorithm for multi-objective optimization," *Concurrency Comput., Pract. Exper.*, vol. 36, no. 10, p. e6566, 2024.

[24] A. Zhou, Y. Jin, and Q. Zhang, "A population prediction strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, 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 Trans. Cybern.*, vol. 49, no. 9, pp. 3362–3374, Sep. 2019, doi: 10.1109/TCYB.2018.2842158.

[26] D. Xu, M. Jiang, W. Hu, S. Li, R. Pan, and G. G. Yen, "An online prediction approach based on incremental support vector machine for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 690–703, Aug. 2022.

[27] M. Jiang, Z. Huang, L. Qiu, W. Huang, and G. G. Yen, "Transfer learning-based dynamic multiobjective optimization algorithms," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 501–514, Aug. 2018.

[28] Q. Lin, Y. Ye, L. Ma, M. Jiang, and K. C. Tan, "Dynamic multiobjective evolutionary optimization via knowledge transfer and maintenance," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 2, pp. 936–949, Feb. 2024.

[29] Y. Hu, J. Ou, P. N. Suganthan, W. Pedrycz, R. Wang, J. Zheng, and Y. Song, "Dynamic multi-objective optimization algorithm guided by recurrent neural network," *IEEE Trans. Evol. Comput.*, Jun. 2024, doi: 10.1109/TEVC.2024.3419892.

[30] A. Muruganantham, K. C. Tan, and P. Vadakkepat, "Evolutionary dynamic multiobjective optimization via Kalman filter prediction," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2862–2873, Dec. 2016, doi: 10.1109/TCYB.2015.2490738.

[31] S. B. Gee, K. C. Tan, and C. Alippi, "Solving multiobjective optimization problems in unknown dynamic environments: An inverse modeling approach," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4223–4234, Dec. 2017.

[32] D. Chen, F. Zou, R. Lu, and X. Wang, "A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization," *Swarm Evol. Comput.*, vol. 43, pp. 147–165, Dec. 2018.

[33] Y. Guo, H. Yang, M. Chen, J. Cheng, and D. Gong, "Ensemble prediction-based dynamic robust multi-objective optimization methods," *Swarm Evol. Comput.*, vol. 48, pp. 156–171, Aug. 2019.

[34] Z. Liang, S. Zheng, Z. Zhu, and S. Yang, "Hybrid of memory and prediction strategies for dynamic multiobjective optimization," *Inf. Sci.*, vol. 485, pp. 200–218, Jun. 2019.

[35] D. Gong, B. Xu, Y. Zhang, Y. Guo, and S. Yang, "A similarity-based cooperative co-evolutionary algorithm for dynamic interval multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 142–156, Feb. 2020.

[36] M. Rong, D. Gong, W. Pedrycz, and L. Wang, "A multimodel prediction method for dynamic multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 290–304, Apr. 2020.

[37] C. Wang, G. G. Yen, and M. Jiang, "A grey prediction-based evolutionary algorithm for dynamic multiobjective optimization," *Swarm Evol. Comput.*, vol. 56, Aug. 2020, Art. no. 100695.

[38] R. Rambabu, P. Vadakkepat, K. C. Tan, and M. Jiang, "A mixture-of-experts prediction framework for evolutionary dynamic multiobjective optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 5099–5112, Dec. 2020.

[39] L. Cao, L. Xu, E. D. Goodman, C. Bao, and S. Zhu, "Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 305–319, Apr. 2020.

[40] X. Ma, J. Yang, H. Sun, Z. Hu, and L. Wei, "Feature information prediction algorithm for dynamic multi-objective optimization problems," *Eur. J. Oper. Res.*, vol. 295, no. 3, pp. 965–981, 2021.

[41] M. Jiang, Z. Wang, H. Hong, and G. G. Yen, "Knee point-based imbalanced transfer learning for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 117–129, Feb. 2021.

[42] M. Jiang, Z. Wang, L. Qiu, S. Guo, X. Gao, and K. C. Tan, "A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3417–3428, Jul. 2021.

[43] J. Li, T. Sun, Q. Lin, M. Jiang, and K. C. Tan, "Reducing negative transfer learning via clustering for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1102–1116, Oct. 2022.

[44] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, pp. 395–416, Aug. 2007.

[45] Y. Vesikar, K. Deb, and J. Blank, "Reference point based NSGA-III for preferred solutions," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1587–1594, doi: 10.1109/SSCI.2018.8628819.

[46] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, 2001, pp. 1–8.

[47] D. Verma and M. Meila, "A comparison of spectral clustering algorithms," Univ. Washington, Seattle, WA, USA, Tech. Rep. UWCSE030501, 2003, pp. 1–18.

[48] V. R. Ramachandran, S. K. Jacobs, A. H. Firpi, H. J. Mitchell, N. H. Tzeng, and B. M. Rodriguez, "Spectral comparison using k-means clustering," in *Proc. IEEE Aerosp. Conf.*, Mar. 2014, pp. 1–10.

[49] M. T. Law, R. Urtasun, and R. S. Zemel, "Deep spectral clustering learning," in *Proc. 34th Int. Conf. Mach. Learn.*, Jul. 2017, pp. 1985–1994. PMLR.

[50] M. B. Blaschko and C. H. Lampert, "Correlational spectral clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[51] H. Jia, S. Ding, X. Xu, and R. Nie, "The latest research progress on spectral clustering," *Neural Comput. Appl.*, vol. 24, nos. 7–8, pp. 1477–1486, Jun. 2014.

[52] S. Wang, H. Zhang, Y. Zhang, A. Zhou, and P. Wu, "A spectral clustering-based multi-source mating selection strategy in evolutionary multi-objective optimization," *IEEE Access*, vol. 7, pp. 131851–131864, 2019.

[53] I. Hatzakis and D. Wallace, "Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach," in *Proc. 8th Annu. Conf. Genetic Evol. Comput.*, 2006, pp. 1201–1208.

[54] C. Chatfield, *The Analysis of Time Series: An Introduction*. Boca Raton, FL, USA: CRC Press, 2004.

[55] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed. New York, NY, USA: Springer, 2013.

[56] C. Hillermeier, *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Basel, Switzerland: Birkhäuser, 2001.

[57] C.-K. Goh and K. Chen Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 103–127, Feb. 2009, doi: 10.1109/TEVC.2008.920671.

[58] M. Helbig and A. P. Engelbrecht, "Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2011, pp. 2047–2054.

[59] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 198–211, Jan. 2017.

ZHAO-JUN SHENG received the bachelor's degree in automation and the master's degree in rail transit signal and control from Lanzhou Jiaotong University, Gansu, China, in 2008 and 2017, respectively. She is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems with Lanzhou University of Technology. Her current research interests include multi-objective optimization algorithms, machine learning, and their applications in design and optimization.

ERCHAO LI received the master's degree in control science and engineering from Yanshan University, Hebei, China, in 2005, and the Ph.D. degree in control science and engineering from Lanzhou University of Technology, Gansu, China, in 2011. He is currently a Professor with the Department of Automation, School of Electrical and Information Engineering, Lanzhou University of Technology. His main research interests include intelligent optimization theory, methods, and applications; environmental perception, modeling, and control of intelligent robots; and modeling and operational optimization of integrated energy systems.

2025 40th Youth Academic Annual Conference of Chinese Association of Automation (YAC) | 979-8-3315-0330-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/YAC66630.2025.11150061

Design of DMO Test Functions Based on PS Dynamic Changes and PF Discontinuity

Zhaojun Sheng
College of Electrical Engineering and Information Engineering
Lanzhou University of Technology
Lanzhou, China
shengzjkk@163.com

Erchao Li
College of Electrical Engineering and Information Engineering
Lanzhou University of Technology
Lanzhou, China
lecstarr@163.com

***Abstract*—In view of the widespread practical needs of DMOPs, we propose a new set of benchmark test functions. By introducing translation or rotation changes with varying frequencies and directions, we capture the complex dynamics of PS and, on this basis, enrich the discontinuity of PF. Two typical DMOA are selected for testing, and the MIGD is used to evaluate algorithm performance. The results show that the discontinuity of PF and the complex shape changes in PS significantly increase the difficulty of problem-solving, while also revealing the limitations of existing algorithms in adapting to sudden or discontinuous dynamic environments. These findings provide valuable references for future algorithm improvements and for real-world applications of DMOPs.**

***Keywords*—DMO, Test Function Design, Dynamic Changes of PS, Discontinuity of PF**

I. INTRODUCTION

DMO problems (DMOPs) are widely found in real-world industrial applications and scientific research, making the successful solving of such problems highly significant in practice. Benchmark test functions for DMO play a crucial role in promoting the development of DMO algorithms (DMOAs) to solve real-world problems. Therefore, designing test functions that better reflect the dynamic characteristics of real optimization problems is particularly important.

Building on existing dynamic multi-objective benchmark test functions, we designs a new set of benchmark test functions. Not only do these functions provide researchers with effective tools for analyzing and improving algorithm performance, but they also further promote the extensive application of algorithms in solving real optimization problems. The mathematical description of the DMOPs studied in this paper is given by Equation (1):

$$\begin{aligned} &\min\ F(x,t) = \left(f_1(x,t),\ldots,f_m(x,t)\right)^T \\ &\text{s.t. } x \in \prod_{i=1}^{n}\left[a_i,b_i\right] \end{aligned} \tag{1}$$

Where $t \in T = \{1, 2, \ldots\}$ denotes the time instants. where $\prod_{i=1}^{n}[a_i,b_i] \subset R^n$ refers to the decision space. $x = (x_1,\ldots,x_n)^{\mathrm{T}}$ defines the decision variable. $F(x,t): R^n \times R \to R^m$ consists of m objective functions, R^m refers to the objective space.

For a given environment *t*, if there is no solution $x' \in \prod_{i=1}^{n}\left[a_i,b_i\right]$, such that $f_j(x',t) \le f_j(x,t)\left(j=1,2,\ldots,m\right)$ and $F(x',t) \ne F(x,t)$, then $x \in \prod_{i=1}^{n}\left[a_i,b_i\right]$ is called the Pareto Set (PS). The corresponding objective function values of the PS are called the Pareto Front (PF).

From Equation (1), we see that DMOPs not only involve multiple conflicting objectives but may also have time-varying decision variables, constraints, or objective functions, causing the PS and/or PF to change over time. Consequently, DMOAs must quickly track the continuously changing PF and/or PS, and find a set of solutions with good diversity and convergence that approximate the new PF and/or PS.

II. RELATED WORK

Researchers have already designed several benchmark test functions for DMO. Based on the classical static multi-objective optimization problems DTLZ [2] and ZDT [3], Farina et al. [1] pioneered a benchmark design method considering both dynamics and multi-objectivity. Building on this approach, FDA [4] was introduced as a set of dynamic multi-objective test functions, which had a profound impact on benchmark research for DMO and provided an important reference for subsequent work.

In [5], based on the FDA series of test functions, random dynamic changes were introduced to propose dMOPs, making the test functions closer to real-world problems. However, in both the FDA test functions and dMOPs, the decision variables are uncorrelated, and their PS only undergoes translation changes. Their PF exhibits convex, linear, concave, or a mixture of the three shapes in a dynamic manner. Moreover, the movement pattern of both PS and PF is vertical, meaning these two sets of functions have relatively simple and homogeneous dynamic changes; they are highly similar to each other and lack diversity, potentially limiting their capacity to comprehensively evaluate the performance of DMOAs.

In [6], after conducting in-depth research on existing test functions, the HE series of test functions with multiple discontinuous PF segments was proposed, where the PS undergoes translation changes, increasing the complexity of solving these dynamic problems. Drawing on the characteristics of the aforementioned benchmark test functions, this paper introduces new modes of PS movement and rotation and further enriches the discontinuity of the PF, thus designing a new set of test functions for more comprehensive dynamic multi-objective benchmark testing, aiming to evaluate the strengths and weaknesses of DMOAs more thoroughly.

III. PROPOSED TEST FUNCTIONS

In this section, we design a set of test functions that exhibit more complex dynamic changes by introducing new PS

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

movement and rotation patterns while further enriching PF discontinuities. The PS may involve translations or rotations at different frequencies and directions, and during the dynamic changes, the PF may exhibit jumps or breaks.

By analyzing the dynamic test function FDA, the dynamic properties in the decision variable space can be described by Equation (2):

$$\begin{aligned} &Z(x_i) = Z\left(x_i - S\left(x_1\right) G\left(t\right)\right), \\ &S\left(x_1\right) = \left|x_1 - \varphi\right|, \\ &G(t) = \begin{cases} \left|\sin(w_1 \pi t)\right|, x_1 \in \left[0, \mu\right] \\ \left|\cos(w_2 \pi t)\right|, x_1 \in \left(\mu, 1\right] \end{cases} \end{aligned} \tag{2}$$

If the PS undergoes the same frequency but different directions of translation, then in $S(x_1)$ we set $\varphi = x_1$ - 1, and $w_1 = w_2$.

If the PS undergoes different frequencies and different directions of translation, then in $S(x_1)$ we set $\varphi = x_1$ - 1, and $w_1 \neq w_2$.

If the PS undergoes rotation at the same frequency and in the same direction, then in $S(x_1)$ we set $\varphi = 0$, and $w_1 \neq w_2$.

If the PS undergoes rotation at different frequencies and in different directions, then in $S(x_1)$ we set $\varphi \neq 0$, and $w_1 \neq w_2$.

From this, we can conclude that PF may exhibit discontinuous jumps during dynamic changes.

Based on the above analysis, we proposes the following test functions:

LSF1, LSF2, and LSF3: Two-objective test functions in which PS undergoes translations or rotations at different frequencies and directions, and PF shows discontinuous changes.

LSF6: A three-objective test function in which the PS changes at the same frequency and direction, and the PF changes continuously.

Moreover, by introducing the dynamic changes of PS into the HE series of test functions, the resulting PF in dynamic changes may exhibit breaks, i.e., discontinuities in PF. From the HE series, we design two test functions, LSF4 and LSF5. Table I shows the six new test functions proposed in this paper, where n is the dimension of the decision variable space.

TABLE I. SIX NEWLY PROPSED TEST INSTANCES

Instance	Definition	Remarks
LSF1 (Type I)	$f_1(x_1,\ldots,x_n) = (1+g)\cdot\cos(0.5\pi x_1)$, $f_2(x_1,\ldots,x_n) = (1+g)\cdot\sin(0.5\pi x_1)$, $g(x_1,\ldots,x_n) = \sum_{i=2}^{n}\left(x_i - G(t)\right)^2$ $G(t) = \begin{cases} \left\|\sin(0.5\pi t)\right\|, x_1 \in [0,0.5] \\ \left\|\cos(0.5\pi t)\right\|, x_1 \in (0.5,1] \end{cases}$ $where: x_i \in [0,1]$	PS: [0, 0.5] moves upward, and (0.5, 1] moves downward with the same frequency. PF: disconnected. $f_1 = \cos(0.5\pi x_1)$, $f_2 = \sin(0.5\pi x_1)$, $where: x_i \in [0,1]$
LSF2 (Type II)	$f_1(x_1,\ldots,x_n) = (1+g)\cdot\cos(0.5\pi x_1)$, $f_2(x_1,\ldots,x_n) = (1+g)\cdot\sin(0.5\pi x_1)$, $g(x_1,\ldots,x_n) = G(t) + \sum_{i=2}^{n}\left(x_i - G(t)\right)^2$ $G(t) = \begin{cases} \left\|\sin(0.5\pi t)\right\|, x_1 \in [0,0.5] \\ \left\|\sin(0.6\pi t)\right\|, x_1 \in (0.5,1] \end{cases}$ $where: x_i \in [0,1]$	PS: [0, 0.5] moves upward, and (0.5, 1] moves downward with different frequencies. PF: disconnected (shifted along different directions with different frequencies based on $f_1 = f_2$ as the boundary). $f_1 = (1+G(t))\cdot\cos(0.5\pi x_1)$, $f_2 = (1+G(t))\cdot\sin(0.5\pi x_1)$, $G(t) = \begin{cases} \left\|\sin(0.5\pi t)\right\|, x_1 \in [0,0.5] \\ \left\|\sin(0.6\pi t)\right\|, x_1 \in (0.5,1] \end{cases}$ $where: x_i \in [0,1]$
LSF3 (Type II)	$f_1(x_1,\ldots,x_n) = (1+g)\cdot\cos(0.5\pi x_1)$, $f_2(x_1,\ldots,x_n) = (1+g)\cdot\sin(0.5\pi x_1)$, $g(x_1,\ldots,x_n) = G(t) + \sum_{i=2}^{n}\left(x_i - G(t)\cdot\left\|x_1 - 0.5\right\|\right)^2$ $G(t) = \begin{cases} \left\|\sin(0.5\pi t)\right\|, x_1 \in [0,0.5] \\ \left\|\sin(0.6\pi t)\right\|, x_1 \in (0.5,1] \end{cases}$ $where: x_1 \in [0,1], x_i \in [-1,1]$	PS: with 0.5 as the center, rotate the left side clockwise and the right side counterclockwise at different frequencies. PF: disconnected (shifted along different directions with different frequencies based on $f_1 = f_2$ as the boundary). $f_1 = (1+G(t))\cdot\cos(0.5\pi x_1)$, $f_2 = (1+G(t))\cdot\sin(0.5\pi x_1)$, $G(t) = \begin{cases} \left\|\sin(0.5\pi t)\right\|, x_1 \in [0,0.5] \\ \left\|\sin(0.6\pi t)\right\|, x_1 \in (0.5,1] \end{cases}$ $where: x_1 \in [0,1], x_i \in [-1,1]$

LSF4 (Type I)	$f_1(x_1)=x_1,$ $f_2(x_1,\cdots,x_n)=1+g\cdot h,$ $g(x_2,\cdots,x_n)=1+\sum_{i=2}^{n}\left(x_i-G(t)\right)^2,$ $h(f_1,g)=1-\sqrt{\frac{f_1}{g}}-\frac{f_1}{g}\cdot\sin(10\pi f_1)$ $G(t)=\begin{cases}\lvert\sin(0.1\pi t)\rvert, x_1\in[0,0.5]\\ \lvert\cos(0.1\pi t)\rvert, x_1\in(0.5,1]\end{cases}$ $where:\ x_i\in[0,1]$	PS: [0, 0.5] moves upward, and (0.5, 1] moves downward with same frequencies. PF: disconnected. $f_1=x_1,$ $f_2=1+h,$ $h=1-\sqrt{f_1}-f_1\cdot\sin(10\pi f_1)$ $G(t)=\begin{cases}\lvert\sin(0.1\pi t)\rvert, x_1\in[0,0.5]\\ \lvert\cos(0.1\pi t)\rvert, x_1\in(0.5,1]\end{cases}$ $where:\ x_i\in[0,1]$
LSF5 (Type I)	$f_1(x_1)=x_1,$ $f_2(x_1,\cdots,x_n)=1+g\cdot h,$ $g(x_2,\cdots,x_n)=1+\sum_{i=2}^{n}\left(x_i-G(t)\cdot x_1\right)^2,$ $h(f_1,g)=1-\sqrt{\frac{f_1}{g}}-\frac{f_1}{g}\cdot\sin(10\pi f_1)$ $G(t)=\lvert\sin(0.5\pi t)\rvert$ $where:\ x_i\in[0,1]$	PS: rotate counterclockwise around the origin of the coordinate system. PF: disconnected. $f_1=x_1,$ $f_2=1+h,$ $h=1-\sqrt{f_1}-f_1\cdot\sin(10\pi f_1)$ $G(t)=\lvert\sin(0.5\pi t)\rvert$ $where:\ x_i\in[0,1]$
LSF6 (Type I)	$f_1(x_1,\cdots,x_n)=(1+g)\cdot\cos(0.5\pi x_1)\cdot\cos(0.5\pi x_2),$ $f_2(x_1,\cdots,x_n)=(1+g)\cdot\cos(0.5\pi x_1)\cdot\sin(0.5\pi x_2),$ $f_3(x_1,x_3,\cdots,x_n)=(1+g)\cdot\sin(0.5\pi x_1),$ $g(x_1,x_3,\cdots,x_n)=\sum_{i=3}^{n}\left(x_i-G(t)\cdot x_1\right)^2$ $G(t)=\lvert\sin(0.5\pi t)\rvert$ $where:\ x_i\in[0,1]$	PS: rotate counterclockwise around the origin of the coordinate system. PF: continuous. $f_1=\cos(0.5\pi x_1)\cdot\cos(0.5\pi x_2),$ $f_2=\cos(0.5\pi x_1)\cdot\sin(0.5\pi x_2),$ $f_3=\sin(0.5\pi x_1),$ $G(t)=\lvert\sin(0.5\pi t)\rvert$ $where:\ x_i\in[0,1]$

IV. EXPERIMENTAL STUDY

A. Experimental Setup

Two typical DMOAs are used to evaluate the proposed test functions: population prediction strategy (PPS) [7] and Multidirectional Prediction Approach (MDP) [8]. Parameter settings are as follows:

(1) The MOEA uses NSGA-III, with a crossover probability of 0.7 and a mutation probability of 0.3.

(2) For two-objective dynamic optimization problems, the population size $N = 100$ and the decision variable dimension $n = 10$. For three-objective dynamic optimization problems, $N = 200$ and $n = 20$.

(3) 5% of individuals are used for detecting environmental changes. $T = 100$ environmental changes are set, where $n_t = 10$, $\tau_t = 30$, and each test function is run independently 20 times. The problems change every 30 generations.

B. Performance Indicator

We use the Main Inverted Generational Distance (MIGD) [9] as a performance indicator. Let the true Pareto front and the Pareto front obtained by DMOA in the t-th environment be denoted as PF^t and P^t respectively. Based on IGD[10], the MIGD of DMOA is defined as:

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

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

$$IGD=\frac{\sum_{x\in PF^t}\min d\left(x,P^t\right)}{\left|PF^t\right|}$$

where $d(x,P^t)$ is the Euclidian distance between x and the point in P^t. $\left|PF^t\right|$ is the cardinality of PF^t.

On the true PF, 1000 evenly distributed points are taken for two-objective problems and 10,000 evenly distributed points for three-objective problems. Each test function is run independently 20 times to compute the MIGD values of different DMOAs.

C. Prediction Results and Algorithm Performance Analysis

Table II shows the statistical MIGD results over 20 runs. The mean values for $0 \le t \le 100$, $0 \le t \le 20$, $21 \le t \le 60$, and $61 \le t \le 100$ are presented, with the best results in bold.

TABLE II. RESULTS OF MIGD OBTAINED BY PPS, MDP OVER 20 RUNS

Instance	Strategy	$0 \le t \le 100$	$0 \le t \le 20$	$21 \le t \le 60$	$61 \le t \le 100$
FDA1	PPS	**0.0047**	0.0070	**0.0042**	**0.0041**
	MDP	0.0048	**0.0049**	0.0047	0.0047
FDA2	PPS	**0.0053**	**0.0046**	**0.0053**	**0.0057**
	MDP	0.0059	0.0047	0.0060	0.0065
FDA3	PPS	0.0122	0.0087	0.0135	0.0126

	MDP	**0.0095**	**0.0076**	**0.0101**	**0.0099**
dMOPl	PPS	0.0057	**0.0065**	0.0054	0.0055
	MDP	**0.0054**	0.0069	**0.0050**	**0.0050**
dMOP2	PPS	**0.0049**	0.0081	**0.0042**	**0.0041**
	MDP	0.0053	**0.0058**	0.0052	0.0052
HE2	PPS	**0.1777**	**0.1667**	0.1810	**0.1806**
	MDP	0.1783	0.1704	**0.1790**	0.1807
LSF1	PPS	0.1021	0.0881	0.0974	0.1139
	MDP	**0.0155**	**0.0320**	**0.0115**	**0.0127**
LSF2	PPS	**0.0410**	0.0433	**0.0515**	**0.0309**
	MDP	0.0434	**0.0421**	0.0562	0.0321
LSF3	PPS	0.0765	0.0558	0.1094	0.0539
	MDP	**0.0387**	**0.0259**	**0.0504**	**0.0342**
LSF4	PPS	**0.2037**	0.2411	**0.1953**	**0.1934**
	MDP	0.2364	**0.2202**	0.2413	0.2395
LSF5	PPS	**0.2040**	**0.2064**	**0.2073**	**0.1996**
	MDP	0.2257	0.2359	0.2168	0.2294
LSF6	PPS	**0.0640**	**0.0726**	**0.0623**	**0.0614**
	MDP	0.0814	0.0861	0.0835	0.0770

From Table 2, we can draw the following conclusions:

(1) Both algorithms generally perform well on test functions with continuous PF, leading to lower MIGD values.

In the initial stage ($0 \leq t \leq 20$), MDP outperforms PPS. MDP adapts to changes in PF more quickly, while PPS, limited by historical information, shows weaker performance. As time goes on, the performance of both algorithms converges, indicating that in a simple dynamic environment, differences in predictive strategies have a minimal impact on final performance.

(2) For test functions with discontinuous PF, the adaptation difficulty significantly increases, and the MIGD values are generally higher than those in tests with continuous PF. The discontinuity of PF and the complex dynamics of PS reduce the prediction accuracy, indicating that existing algorithms lack adequate mechanisms to handle PF discontinuities and complex PS dynamics, revealing certain limitations.

(3) In LSF1 and LSF3, MDP slightly outperforms PPS in all time periods. In LSF2, LSF4, and LSF5, PPS and MDP perform similarly overall. PPS performs poorly when the environment changes drastically, mainly because its reliance on historical data restricts its ability to adapt to sudden changes. MDP handles PF discontinuities better but still has limitations in scenarios with highly complex PS changes.

V. Conclusion

Focusing on the shape changes of PS and the discontinuity of PF, we propose a new set of DMO benchmark test functions and tests two typical algorithms using these functions. The experimental results show that introducing PS shape changes and PF discontinuities significantly increases the difficulty of solving DMO problems, while also posing new challenges for the design of DMOAs. In future research, not only should more realistic test functions be designed, but algorithm design should also be further improved—such as by constructing nonlinear models or incorporating machine learning-based prediction methods—to handle PF discontinuities more effectively, and by employing diverse population initialization strategies or hybrid search mechanisms to enhance adaptation to complex PS dynamics.

Acknowledgment

Data Availability statement The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for Publication Informed consent was obtained from all individual participants included in the study.

Conflict of Interests All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Authors' contributions: Zhaoju Sheng (Methodology, Writing-original draft). Erchao Li (Writing review and editing, Conceptualization, Supervision).

References

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

[2] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, "Scable test problems for evolutionary multi-objective optimization," KanGAL Report 2001001, Kanpur Genetic Algorithms Lab (KanGAL), Indian Inst. Technol., 2001.

[3] E. Zitzler, K. Deb, L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," Evol. Comput., vol. 8, no. 2, pp. 173– 195, 2000.

[4] Goh C K, Tan K C. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(1): 103-127.

[5] C. -K. Goh and K. C. Tan (2009). A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization, in IEEE TEC, vol. 13, no. 1, pp. 103-127, doi: 10.1109/TEVC.2008.920671.

[6] Helbig, M., & Engelbrecht, A. P. (2011). Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation. In 2011 IEEE Congress of Evolutionary Computation (CEC), pp. 2047-2054.

[7] A. Zhou, Y. Jin, and Q. Zhang (2014). A population prediction strategy for evolutionary dynamic multiobjective optimization, M IEEE Trans. Cybem., vol. 44, no. 1, pp. 40-53.

[8] M. Rong, D. Gong, W. Pedrycz, and L. Wang (2020). A multimodel prediction method for dynamic multiobjective evolutionary optimization. IEEE Trans. Evol. Comput., vol. 24, no. 2, pp. 290–304.

[9] Sierra, M.R., Coello Coello, C.A. (2005). Improving PSO-based multi-objective optimization using crowding, mutation and □-dominance, in Proc. EMO, LNCS 3410, pp. 505-519, doi: 10.1007/978-3-540-31880-4_35.

[10] Yuan Y, Xu H, Wang B, et al. Balancing convergence and diversity in decomposition-based many-objective Optimizers[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(2): 180-198.

Memetic Computing (2026) 18:18
https://doi.org/10.1007/s12293-026-00499-1

REGULAR RESEARCH PAPER

A dividing lines-hyperplane intersection points and associated subpopulations prediction strategy for evolutionary dynamic multiobjective optimization

Zhao-jun Sheng[1] · Er-chao Li[1] · Li-sen Wei[1] · Chen-miao Liu[1]

Received: 11 September 2024 / Accepted: 11 March 2026

Abstract

To rapidly track the dynamic evolution of the Pareto-optimal front (PF) in dynamic multiobjective optimization problems (DMOPs), this paper proposes a dividing-line–hyperplane-intersection-based and associated subpopulation prediction strategy (ISPP). The strategy infers the changes in the Pareto-optimal set (PS) in the decision space from the dynamic characteristics of the PF in the objective space. First, hyperplane fitting and dividing-line setting strategy are used to locate and segment the PF, and to characterize the changing trend of the PF through intersection point sequences at different time steps. Subsequently, the PF is partitioned into multiple subsets through local association operations at the intersection points, establishing a mapping relationship between the PF subsets and the PS subpopulations, where each subpopulation is modeled by a subcenter and a submanifold. When environmental changes occur, the sequences of subcenters are used to estimate the new subcenters, and historical submanifolds are utilized to estimate the new submanifolds, thereby constructing the initial population. In addition, six new test functions are designed, incorporating PS rotation and translation patterns as well as discontinuous PF characteristics. The proposed ISPP-DMOEA is compared with PPS, MDP, KTM, and RNN on 16 test problems, and the results show that the proposed algorithm performs well on the majority of test functions. In particular, it demonstrates distinct advantages on test functions characterized by discontinuous PFs and those undergoing translational or rotational changes at varying frequencies.

Keywords Dynamic multiobjective optimization · Evolutionary algorithm · Prediction · Hyperplane fitting · Dividing line-hyperplane intersection point · Subpopulation

1 Introduction

Dynamic multiobjective optimization problems (DMOPs) refer to problems where the objectives or constraints change in a dynamic environment, and multiple (often conflicting) objectives need to be optimized simultaneously[1, 2]. There are numerous DMOPs in various application fields, such as dynamic path planning [3], industrial scheduling [4], system control[5, 6], investment management [7], planning [8, 9], and machine learning [10]. Furthermore, as research into these problems has progressed, greater challenges have been posed to dynamic multiobjective evolutionary algorithms (DMOEAs).

The mathematical description of the DMOPs we study is shown in Eq. (1) [11]:

$$
\begin{aligned}
&\min\ F(x,t) = (f_1(x,t),\ \ldots,\ f_m(x,t))^T \\
&s.t.\ x \in \prod_{i=1}^{n} [a_i, b_i]
\end{aligned}
\tag{1}
$$

where $t \in T = \{1, 2, \ldots\}$ denotes the time instants, $\prod_{i=1}^{n} [a_i, b_i] \subset \mathbb{R}^n$ refers to the decision space, $x = (x_1, \ldots, x_n)^T$ defines the decision variable. $F(\boldsymbol{x}, t) : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ consists of m objective functions, and $\mathbb{R}^m$ refers to the objective space.

✉ Er-chao Li
lecstarr@163.com

[1] College of Electrical Engineering and Information Engineering, Lanzhou University of Technology, Lanzhou Gansu 730050, China

For two decision vectors $x_1, x_2 \in \prod_{i=1}^{n} [a_i, b_i]$ of DMOPs at time t, $\boldsymbol{x}_1$ is said to Pareto dominate $\boldsymbol{x}_2$ when $f_i(x_1, t) \leq f_i(x_2, t)$ for $\forall i = 1, \ldots, m$ and $f_j(x_1, t) < f_j(x_2, t)$ for $\exists j = 1, \ldots, m$. If there is no solution $x \in \prod_{i=1}^{n} [a_i, b_i]$ that can dominate $\boldsymbol{x}^*$ at time t, $\boldsymbol{x}^*$ is called a Pareto-optimal solution, and a set including all Pareto-optimal solutions is called a Pareto-optimal set (PS). Moreover, the corresponding objective vectors of the PS are called the Pareto-optimal front (PF).

Evolutionary algorithms (EAs) are characterized by multipoint search capabilities, adaptability, and self-learning, making them among the most powerful and widely studied methods for solving multiobjective optimization problems [12].

Currently, many dynamic response strategies have been proposed for researching PSs, among which predictive strategies have received particular attention. This strategy identifies historical patterns of change by mining useful information from the decision space and objective space in historical environments and constructing predictive models. As a result, predictive strategies can quickly predict high-quality solutions in new environments, making them a popular research topic in recent years [13–35], which will be reviewed in detail in Sect. 2. Based on the current state of research, this paper focuses on predictive strategies.

Assuming that any two continuous changes in the decision space are similar and linear, many predictive strategies based on linear models have been proposed [13–17]. For example, Zhou et al. [13] proposed a population-based prediction method (PPS), which assumes that the PS in different environments is composed of a central point and an $m-1$ manifold, using an AR model to predict the entire initial population. However, when the relationship between any two continuous changes violates strict linear assumptions, such as PS with random or nonlinear variation patterns, predictive strategies based on linear models from individual representative individuals obtained from PS may lead to poor prediction results.

Given the assumption of high similarity among solutions across different environments and the independent and identically distributed (IID) nature of these solutions, numerous predictive strategies utilizing nonlinear models have been developed [18–23]. An example is the ISVM-DMOEA, a DMOP solver that employs an incremental support vector machine (ISVM) as an online learning predictor to find high-quality solutions in novel environments [23]. However, practical assumptions often fall short, leading to diminished performance for these predictive strategies.

In response, DMOEAs that leverage transfer learning have been designed in recent years [24–34]. Transfer learning can effectively address data that exhibit non-IID characteristics. For example, the Tr-DMOEA framework, introduced in [24], applies transfer component analysis (TCA) in the objective space to harness valuable insights from previous environments. By integrating the knowledge transfer strategy known as ETO into the DMOEA, the efficiency of solving DMOPs can be significantly enhanced. Nonetheless, it is important to note that many of these approaches encounter issues with negative transfer.

Additionally, from the perspective of making full use of historical information, a DMOEA based on recurrent neural networks (RNNs) was proposed [35], where the model learns the best solutions obtained continuously from all historical information and then updates the RNN model using the learned knowledge. However, the time series composed of historical information often contains linear or nonlinear trends of increase or decrease, whereas the RNN is essentially a nonlinear model that may overfit local nonlinear fluctuations and ignore the overall linear trends, resulting in prediction bias.

Based on the above research, when the environment or objective function changes, the existing literature typically starts from the decision space to investigate the variation characteristics of the PS under different environments, summarizing patterns of change and designing predictive models to forecast the initial population. Finally, MOEAs are utilized to optimize the initial population of these solutions, obtain the PS, and further obtain the PF [36].

However, when the environment or objective function changes, the initial consequence is that the PF in the DMOPs changes, which in turn causes variations in the PS. Therefore, the core of this change lies in the fact that the alteration of the PF requires us to research the PS further. If we can accurately predict the trend of changes in the PF, we can infer the evolution patterns of the PS in reverse. Based on this idea, we propose a prediction strategy called the dividing lines–hyperplane intersection points and associated subpopulations prediction strategy (ISPP). The intersection points between a dividing line, which is uniformly distributed in the fixed objective space (unchanged by environmental variations), and the PF not only represent the current PF but also reflect the trend of PF changes over time through a time series of intersection points along a dividing line. Each intersection point's nearby cluster of points on the PF corresponds to a subpopulation in the PS; thus, the trend of changes in the intersection point subset on the PF can also reflect the trend of changes in the associated subpopulation in the decision space (determined by the intersection points). This allows us to predict the changes in the PS through the variations in the PF in the objective space. This strategy effectively overcomes the impact of linear models on the quality of predicted solutions in the following two cases: a lack of representative points sampled on the PF in optimization problems and sampling representative points from the PS when the PS exhibits random or nonlinear variation patterns.

In this algorithm, the key aspect is determining the intersection points between the PF and the dividing line. To achieve this, we use a least-squares hyperplane to approximate the PF, which allows us to find the intersection points between the hyperplane and the dividing line. This results in points on the PF linked to each intersection point, which we then treat as a subset within the objective space. Predictions are made for the corresponding subpopulations in the decision space that relate to each subset of the objective space, thereby generating the initial population of the decision space for the current environment. Finally, the ISPP is integrated with NSGA-III [37] to derive the current PS.

Furthermore, although the above studies primarily analyze continuous DMOPs with PFs under different environments (e.g., the FDA series [38], dMOP series [39], and Fun series [17]) and have achieved good predictive results, relatively little research has been conducted on DMOPs with discontinuous PFs (e.g., HE series [40]); moreover, the current testing functions addressing PF discontinuity are quite scarce, which adversely affects the comprehensive performance analysis of the proposed prediction strategy. Therefore, based on existing dynamic multiobjective optimization test functions, we have designed six optimization problems with discontinuous PFs and PSs that translate or rotate at different frequencies.

The main contributions of this paper are as follows:

(1) Currently, most prediction strategies directly start from the PS to achieve population prediction after environmental changes, employing methods such as center point prediction, transfer learning, and distribution sampling, while neglecting the correlation between the changes in PF characteristics caused by environmental changes and the variations in the PS solution set. This paper proposes a predictive strategy that reversely deduces the correlated characteristic changes in the decision space PS from the dynamic characteristic changes of the PF in the objective space of DMOPs, aiming to quickly track the dynamic evolution of the PF as the environment changes. Specifically, the strategy performs subspace segmentation on the PF by fitting the intersection points of hyperplanes and dividing lines to track the characteristic evolution of complex changes such as (non)continuity in the PF. It associates PF characteristic subsets with PS characteristic subsets to obtain subpopulations in the decision space that correspond one-to-one with each objective space characteristic subset, thereby achieving dynamic tracking of PF features through subpopulation predictions.

(2) (2) We designed six new testing functions by introducing new PS translation and rotation patterns, as well as a discontinuous PF. In these new testing functions, the PS includes translation or rotation changes of different frequencies and directions, whereas the PF may experience jumps or breaks during dynamic changes. This research not only supplements existing testing functions but also expands the application of testing functions, making it possible to solve more complex real-world DMOPs. Moreover, this study broadens the scope of performance evaluation for different algorithms. By conducting experiments on 10 well-known test problems and our six newly proposed test problems, we thoroughly validated the performance of ISPP-DMOEA. Compared with four typical DMOEAs, PPS [13], MDP [17], KTM [34], and RNN [35], ISPP-DMOEA not only performs well on the 10 well-known test problems but also shows significant advantages on the six new test problems, as confirmed through the experiments presented in Sect. 4.

This paper is organized as follows: Sect. 2 discusses prior research concerning DMOPs. Section 3 elaborates on the proposed forecasting strategy. Section 4 presents the test instances, including six newly created test cases, as well as the experimental setup and performance metrics employed. Section 5 presents the experimental results and provides an analysis. Finally, Sect. 6 concludes with a summary and outlines directions for future research.

2 Related work

In recent years, predictive strategies have been classified into three primary categories:

Predictive strategies based on linear models: Wu et al. [14] introduced a directed search strategy (DSS) comprising two components: DSS1 for adapting to environmental changes and DSS2 for local searching, which predicts the entire population. Zou et al. [15] developed a special point-based prediction strategy (SPPS) that leverages a feedforward center point to forecast the main body of the population while using additional special points to aid in predicting the entire set. Muruganantham et al. [16] proposed a Kalman filtering-based prediction method that accurately tracks the Pareto front (PF). Rong et al. [17] presented a multidirectional prediction method (MDP) that uses multiple representative individuals to derive several evolution-guided directions for predicting the entire initial population.

Predictive strategies based on nonlinear models: Gee et al. introduced an inverse modeling method [18]. Chen et al. introduced a hybrid fuzzy inference prediction strategy [19].

Ma et al. proposed a feature information prediction strategy [20]. A previous study [21] embedded a support vector regression (SVR) predictor into DMOEA to uncover nonlinear relationships between various environments, facilitating the tracking of a moving population set (PS). Additionally, an approach that combines DMOEA with an inverse Gaussian process (IGP) model, termed IGP-DMOEA, was developed [22] to identify environmental change patterns in the objective space and estimate PS positions through backward mapping.

Predictive strategies based on transfer learning: These strategies enhance solution generation in new environments by transferring knowledge such as features, distributions [28], individuals [25], search processes [29], and mapping relationships [30]. An imbalance transfer learning DMOEA that focuses on knee points was introduced in [26], which selectively transmits high-quality solutions for knowledge transfer, minimizing ineffective transfers and reducing computational costs. Another DMOEA aimed at reducing negative transfer through clustering was proposed in [27]. Additionally, a method based on sample geodesic flows (SGFs), combined with a memory mechanism, explores potential changing patterns across different environments [28].

Integrating the knowledge transfer strategy of ETO into DMOEA can significantly enhance the efficiency of solving DMOPs. However, many methods overlook the issue of negative transfer. Notably, IT-DMOEA [25], CT-DMOEA [27], and KTM-DMOEA [34] seek to address negative transfer in dynamic environments. For example, IT-DMOEA enhances solution diversity through a presearch operation, which helps mitigate solution degradation and alleviate negative transfer, although it demands more function evaluations and may underperform with limited evaluation instances. CT-DMOEA restricts knowledge transfers to similar clusters to reduce negative transfer effects. KTM-DMOEA classifies high-quality solutions from numerous randomly generated new environments using KTP, minimizing feature and distribution differences between environments. By modeling elite solution distributions from prior environments with KMS, superior solutions in accordance with dynamic change trends can be obtained, effectively mitigating negative transfer issues.

Inspired by these studies, this paper designs the ISPP-DMOEA and formulates six new test functions based on existing dynamic multiobjective optimization problems.

3 A dividing lines–hyperplane intersection points and associated subpopulations prediction strategy (ISPP)

In this section, we elaborate on how to apply ISPP to predict the initial population in a new environment. The core idea of ISPP is to utilize the historical information of the PF in the objective space to predict the initial population that is close to the PS at the current moment. To achieve this, we first use hyperplane fitting to approximate the spatial distribution of the PF at different time instances. Next, we set partition lines to achieve a uniform partitioning of the objective space. We then solve for the intersection points between the hyperplane and the partition lines. Through local relational operations at these intersection points, we divide the PF into multiple subsets, thereby realizing the localization and partitioning of the PF space. On the same partition line, subsets from different time instances can quickly and accurately track the local trends of the PF changes. Because we have set multiple partition lines, we can efficiently and accurately observe the overall trend of the PF changes. Finally, we determine the decision space subpopulations corresponding to subsets associated with the intersection points of the hyperplane and partition lines, treating these subpopulations as composed of subcenters and submanifolds. The time series formed by the subpopulation centers at different time instances is used to estimate the subpopulation centers in the new environment. Moreover, we use submanifolds at different time instances to estimate the submanifolds in the new environment, thereby obtaining the entire initial population for the current environment.

This section uses a bi-objective dynamic optimization problem to illustrate ISPP, as shown in Fig. 1. Figure 1a describes the process in a two-dimensional objective space where first, the PF is approximated by a line using the least squares method; then, rays are set to divide the two-dimensional objective space uniformly, passing through the origin. Finally, the intersection points between the line and the rays are determined. Through local association operations at these intersection points, the PF is divided into multiple subsets, achieving spatial localization and segmentation of the PF, thus enabling rapid and accurate tracking of both the overall and local changes in the PF. This approach has advantages in tackling DMOPs with complex characteristics, such as discontinuous PFs. Figure 1b depicts the corresponding subpopulations in the low-dimensional decision space associated with each intersection point, represented by subcenter points and submanifolds. Through the predictions made by the subcenter points and submanifolds, the entire initial population is obtained.

Figures 2, 3, 4 and 5 utilize several test functions with typical characteristics to depict the PF and low-dimensional space PS in an ideal environment, further illustrating the basic

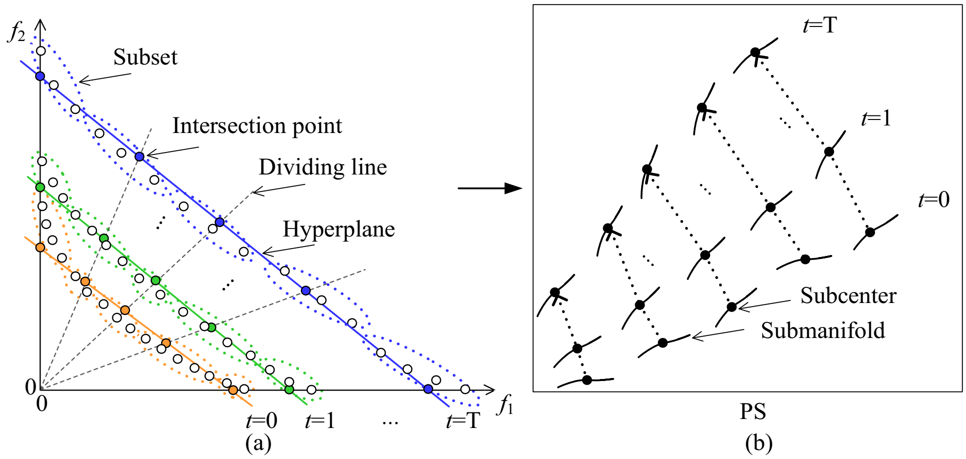


Fig. 1 Prediction process description of ISPP

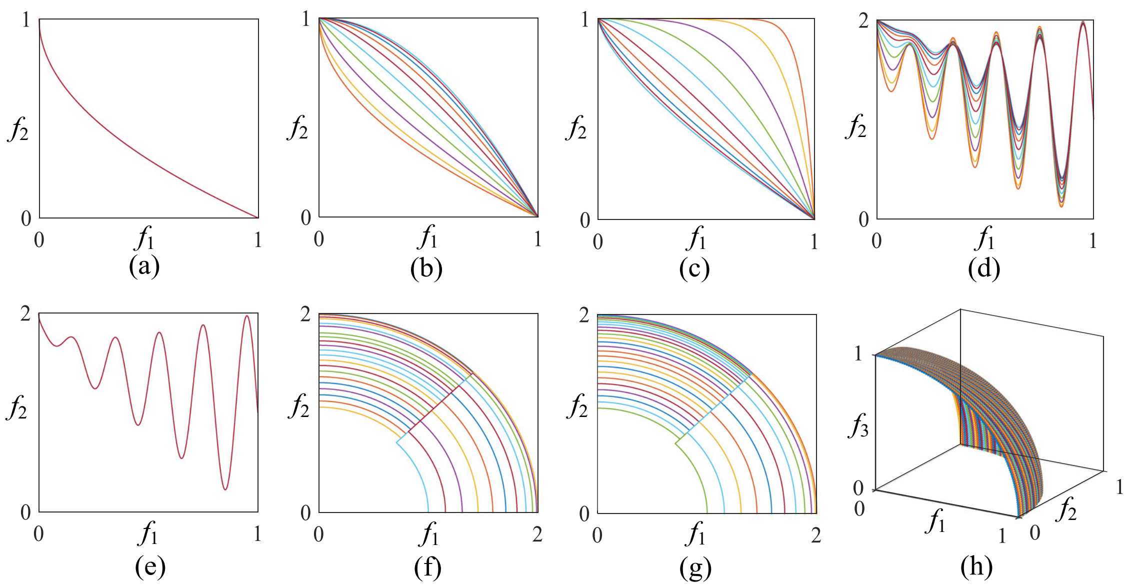


Fig. 2 Ideal PF and changing trend

idea of ISPP. The selected test functions include (a) FDA1, (b) dMOP2, (c) Fun8, (d) HE2, (e) LSF5, (f) LSF2, (g) LSF3, and (h) LSF6, with time points $t = 0, 2, 4, \ldots, 40$ and a time interval of 2. Using LSF2 as an example, we elaborate on the basic concept of ISPP.

Figure 2f: Describes the ideal PF of LSF2, which exhibits discontinuous changes (with the change characteristic varying at different frequencies and in different directions, with $f_1 = f_2$ as the boundary)

Figure 3f: Depicts the PS in the low-dimensional space for LSF2 (with change characteristics such that [0,0.5] increases from bottom to top and (0.5,1] decreases from top to bottom, moving at different frequencies). The overall trend of the population center points does not align with the local trends of the population; in fact, at certain moments, they exhibit opposite trends

Figure 4f: Describes a fixed segmentation line set to divide the objective space uniformly, passing through the origin, and calculates the intersection points with the ideal PF to achieve both intersection point localization and segmentation, yielding subsets associated with the intersection points of the objective space. The time series variation of the intersection points associated subsets along the segmentation line accurately characterizes both the overall and local change trends of the PF

Figure 5f: Illustrates the subpopulations in the decision space for LSF2. The subsets associated with the intersection points in the objective space correspond one-to-one with the subpopulations in the decision space. This implies that the change trends of the PF and PS maintain overall and local consistency through the intersection point associations along the segmentation line. This reflects not only the local change

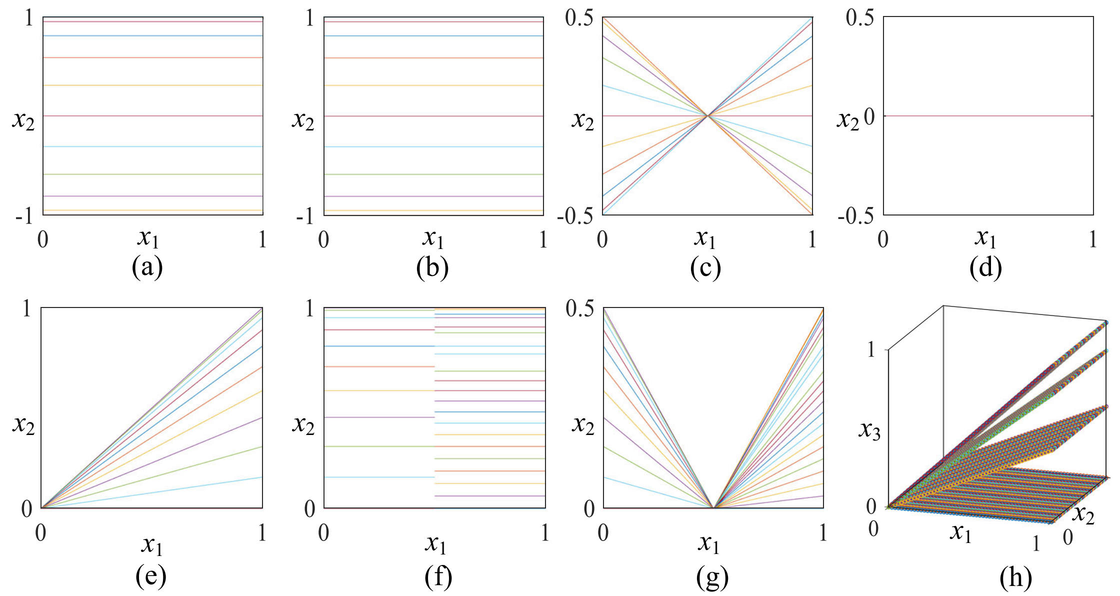


Fig. 3 PS in low-dimensional space and changing trend

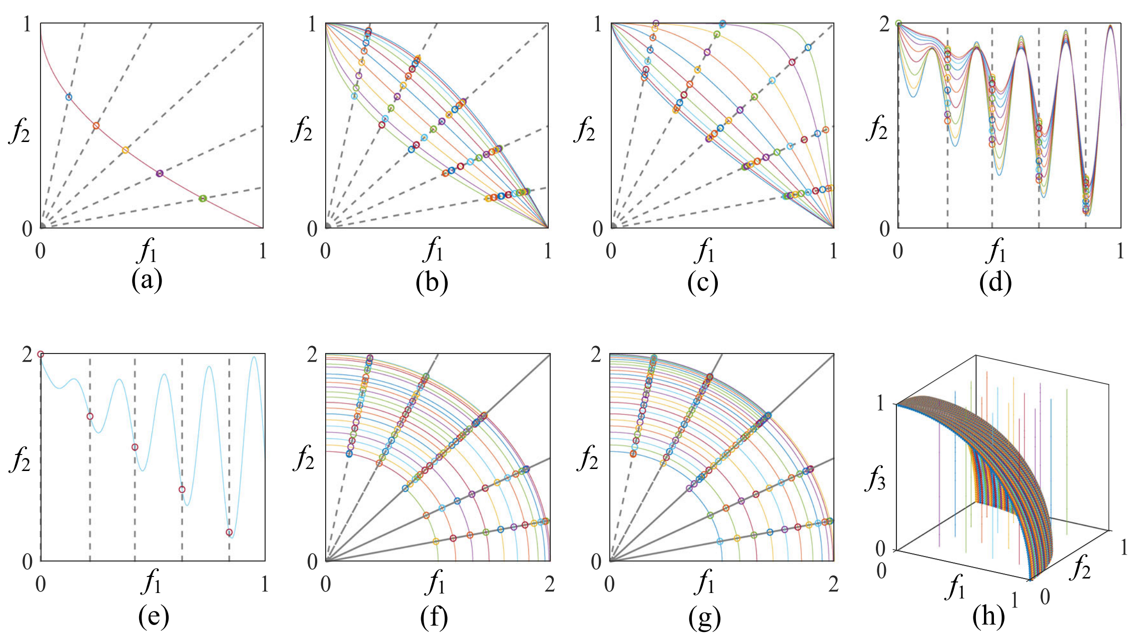


Fig. 4 Setting of the dividing lines in the objective space, along with the associated subsets related to intersection points with the PF and their changing trends

characteristics of the PF but also those of the PS. Therefore, by predicting using the PF-associated subpopulations, a higher-quality initial population can be obtained, resulting in better prediction outcomes

3.1 PF hyperplane fitting

Based on the distribution characteristics of the PF, applying hypersurface fitting after detecting environmental changes may predict a higher-quality initial population. However, considering the computational complexity of the algorithm, this paper employs hyperplane approximation for the PF. This section selects an m-dimensional dynamic optimization problem to elucidate the hyperplane approximation process for the PF.

This paper uses the least squares method to fit a hyperplane to a set of points in an m-dimensional objective space. The detailed steps for solving the hyperplane $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$

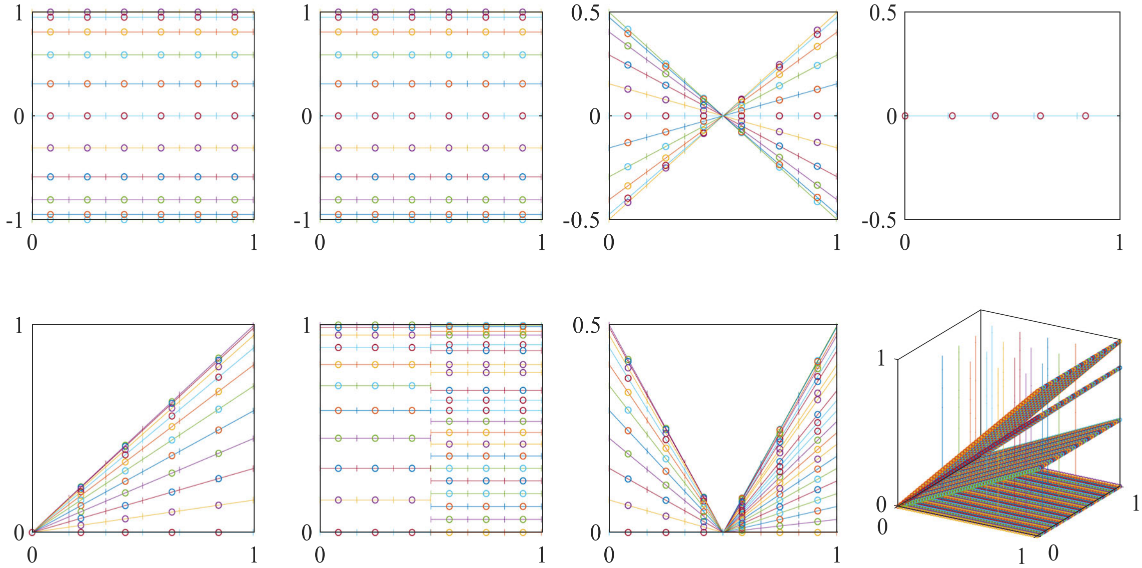


Fig. 5 Associated subpopulations in the decision space and their changing trend

fitting in high-dimensional space (among them, $\boldsymbol{x} \in \mathbb{R}^{m-1}$, $\boldsymbol{w} \in \mathbb{R}^{m-1}$, and the scalar intercept $b \in \mathbb{R}$) are provided below.

(1) Given n samples $\{(\boldsymbol{x}_i, y_i)\}$, find the parameters $(\boldsymbol{w}, b)$ that make the hyperplane $f(\boldsymbol{x}_i) = \boldsymbol{\omega}^T \boldsymbol{x}_i + b$ approximate the observed values y_i.

(2) Construct the sum of squared errors function: $J(\omega, b) = \sum\limits_{i=1}^{n} \left(y_i - \left(\omega^T x_i + b\right)\right)^2$.

(3) Construct the parameter vector $\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\omega} \\ b \end{pmatrix} \in \mathbb{R}^{m+1}$, the matrix $X = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ \vdots & 1 \\ x_2^T & 1 \end{pmatrix} \in \mathbb{R}^{m+1}$, and the target vector $y = (y_1, \cdots, y_n)^T \in \mathbb{R}^n$.

Thus, we can obtain: $J(\theta) = \|X\theta - y\|_2^2 = (X\theta - y)^T (X\theta - y)$.

(4) Obtain the normal equations

Differentiate the above equation with respect to the parameter $\boldsymbol{\theta}$ and set it to zero:

$$\nabla_\theta J = 2X^T(X\theta - y) = 0$$

From this, we can derive the normal equations: $X^T X\theta = X^T y$.

(5) Parameter solution

Explicit solution: $\theta^* = \left(X^T X\right)^{-1} X^T y$. For singular or ill-conditioned cases, the singular value decomposition method can be used for calculation. Let $X = USV^T$, where $X^+ = VS^+U^T$ is the pseudoinverse; then, $\theta^* = X^+ y$.

(6) Fitted hyperplane: $f^*(x) = \omega^{*T} x + b^*$. The fitting quality can be evaluated through residuals $r = X\theta^* - y$, the coefficient of determination R^2, or the standard deviation.

Thus, we obtain the equation of the hyperplane, which approximates the PF through this plane.

3.2 PF localization and division

Based on the distinct dynamic characteristics of the PF, this paper proposes two strategies for configuring dividing lines in the objective space to achieve PF localization, segmentation, and trend tracking. For DMOPs where the PF translates parallel to the coordinate axes or is discontinuous, a vertical uniform dividing line configuration method is adopted. Conversely, for DMOPs where the PF translates non-parallel to the coordinate axes or involves rotation, a uniform dividing line configuration method passing through the origin is employed. Furthermore, by combining these with the hyperplane equation to determine the intersection points between the dividing lines and the hyperplane, the localization and segmentation of the PF are achieved. The changing trend of the PF is then characterized by the intersection point sequences at different time steps.

An excessive number of dividing lines results in a waste of computational resources, while an insufficient number compromises the convergence and distribution (diversity) of the predicted initial population. This paper recommends setting the total number of dividing lines as follows, $K \leq \lfloor N / 10 \rfloor$, where N represents the population size and K denotes the number of dividing lines. The specific configuration methods for these two types of dividing lines are detailed below:

(1) The vertical uniform dividing line strategy involves configuring a set of dividing lines parallel to the m-th objective axis based on a sparse grid. The set of dividing lines L^1 can be defined as:

$$L^1 = \{(g, 0) + \beta \cdot e_m \big| g \in \Omega_{grid}, \beta > 0\} \tag{2}$$

where the sparse grid Ω_{grid} is a set of uniformly distributed discrete points in the $(m-1)$-dimensional subspace, defined as: $\Omega_{grid} = \{(k_1\delta_g, k_2\delta_g, \ldots, k_{m-1}\delta_g)|k_i \in \{0, 1, \ldots, H-1\}\}$.

Here, H denotes the number of grid points in each dimension. $\delta_g = \frac{1}{H}$ represents the grid step size. $\boldsymbol{g} = (g_1, g_2, \ldots, g_{m-1})$ denotes a grid point on the sparse uniform grid Ω_{grid} within the $(m-1)$-dimensional objective subspace. em is the unit vector of the m-th axis, and β represents the extension length parameter of the partition lines.

(2) The uniform dividing line passing through the origin strategy involves configuring a set of dividing lines that pass through the origin and connect to a structured set of points determined by Das and Dennis's simplex-lattice design method [41]. The set of partition lines L^2 can be defined as:

$$L^2 = \{\gamma \cdot z \big| z \in \Omega_{ref}, \gamma > 0\} \tag{3}$$

where the reference point set Ω_{ref} is a set of uniformly distributed discrete points on the m-dimensional normalized unit hyperplane (based on Das and Dennis's method), defined as: $\Omega_{ref} = \left\{(h_1\delta_r, h_2\delta_r, \ldots, h_m\delta_r)\middle| h_i \in \{0, 1, \ldots, p\}, \sum_{i=1}^{m} h_i = p\right\}$.

Here, p denotes the number of divisions on each objective axis, and $\delta_r = \frac{1}{p}$ represents the partition step size. $\boldsymbol{z} = (z_1, z_2, \ldots, z_m)$ denotes the reference point vector on the normalized unit hyperplane, and γ represents the extension length parameter of the dividing lines.

The dividing lines configured by the above two methods ensure the uniformity of the partition in the objective space, thereby effectively locating, segmenting, and tracking the dynamic changing trends of the PF.

3.3 PF intersection point-associated subpopulation generation

The change in the objective subset associated with the intersection point of the PF along the dividing line can reflect the change trend of the PF, and the subpopulation of the decision space corresponding to the objective subset associated with the intersection point can reflect the change trend of the PS so that the objective space subset and the subpopulation of the decision space correspond one-to-one; then, the subpopulation of the decision space is generated through the association information between them. Because objective functions in multiobjective optimization problems often have different dimensions and ranges, directly calculating the Euclidean distance between an individual and the intersection point in the objective space affects the association effects. To eliminate these effects, data normalization is required to ensure that the objective function values are on the same scale, facilitating comprehensive comparison and evaluation. The most typical data normalization method is normalization. The min–max normalization method is particularly suitable for scaling data to a specific range (usually [0, 1]) while maintaining the relative differences between features, making it easy to understand and interpret. Therefore, this paper chooses the min–max normalization method to normalize the objective space. The normalization formula is as follows:

$$f'_j(\boldsymbol{x}) = \frac{f_j(\boldsymbol{x}) - \min(f_j(\boldsymbol{x}))}{\max(f_j(\boldsymbol{x})) - \min(f_j(\boldsymbol{x}))} \tag{4}$$

Here, $j = 1, 2, \ldots, m$, m is the dimension of the objective functions, $\min(f_j(\boldsymbol{x}))$ represents the origin of the objective space coordinates, and $\max(f_j(\boldsymbol{x}))$ is the intersection point of the hyperplane with the coordinate axis of the objective function $f_j(\boldsymbol{x})$.

In the normalized objective space, the individuals associated with the same intersection point are classified into the same subset. The association is carried out by calculating the Euclidean distance between the individuals and intersection points. An individual is associated with the intersection point that has the minimum distance, and the individuals associated with the same intersection point become a subpopulation.

Algorithm 1 Pseudocode of generating subpopulation

Input: the current approximate PF, PF^t; the current approximate PS, P^t;
Output: the current subpopulations, P_k^t;

1: Hyperplane fitting results in a hyperplane fitting equation;
2: According to the change characteristics of PF^t and the number of optimization objectives, the dividing lines are set by $K \le \lfloor N/10 \rfloor$;
3: Find intersection points I_k^t of the hyperplane and the dividing lines;
4: Normalize the objective space using the min–max normalization method;
5: Calculate the Euclidean distance between each individual in PF^t and each intersection point I_k^t; an individual will be associated with the intersection point from which it has the minimum distance, and then obtain the subpopulation, P_k^t corresponding to the subset in the PF^t associated with each intersection point.

3.4 PS subpopulation prediction

In this work, a subpopulation is considered to be composed of a subcenter and a submanifold. When environmental changes are detected, the AR(p) model is used to predict both the subcenter and submanifold separately. The parameters of the AR(p) model are estimated using the least squares method to obtain each initial subpopulation for the new environment and subsequently derive the entire initial population for the current environment.

Let P_k^t be the final subpopulation after optimization at time t, $P_k^t = \{\boldsymbol{x}_k^t\}$, where $\boldsymbol{c}_k^t$ is the center of supopulation P_k^t at time t. The calculation formula for the center of the k-th subpopulation at time t is given by

$$\boldsymbol{c}_k^t = \frac{1}{\left|P_k^t\right|} \sum_{\boldsymbol{x}_k^t \in P_k^t} \boldsymbol{x}_k^t \tag{5}$$

where $\left|P_k^t\right|$ is the cardinality of P_k^t.

Let S_k^t be the k-th submanifold at time t with its center at the origin, $S_k^t = \{\tilde{\boldsymbol{x}}_k^t\}$; then,

$$P_k^t \cong \boldsymbol{c}_k^t + S_k^t \tag{6}$$

That is,

$$\boldsymbol{x}_k^t \cong \boldsymbol{c}_k^t + \tilde{\boldsymbol{x}}_k^t \tag{7}$$

3.4.1 PS Subcenter prediction

In the early stage of environmental changes, predictions are made using autoregressive (AR) models of different orders. Specifically, for the first two time points of environmental changes $t = 1, 2$, a randomly generated initial population is used for dynamic multiobjective optimization. For $t = 3, 4$, and 5, an AR(1) model is used; for $t = 6, 7$, and 8, an AR(2) model is employed; and for $t \ge 9$, an AR(3) model is applied. The length of the historical mean point sequence is $M = 23$. For details on the properties of AR(p) models, see references [42] and [43]. The prediction formula for the subpopulation centers is given by

$$c_{k,i}^{t+1} = \sum_{q=0}^{p-1} a_{qi} c_{k,i}^{t-q} + N(0, \sigma_{k,i}^c) \tag{8}$$

where $c_{k,i}^{t+1}$ is the value of the i-th dimension of the center point of the k-th subpopulation at time $t+1$. a_{qi} represents the parameters of the AR(p) model, which are estimated by least squares.

Within the first t time points, the centers of the k-th subpopulation at each time point form a historical center point sequence c_k^{t-r}, $r = 0, 1, \ldots, M-1$. The variance calculation formula is given by $\sigma_{k,i}^c = \frac{1}{M-p} \sum_{k=t-M+p}^{t} [c_{k,i}^{t+1} - \sum_{q=0}^{p-1} a_{qi} c_{k,i}^{t-q}]^2$.

3.4.2 PS submanifold prediction

In dynamic multiobjective optimization problems, which are influenced by constraints and objective functions, the PS is often concentrated in specific regions of the decision or objective space and exhibits certain geometric structures. According to the manifold theory proposed by Lee et al. [44], the PS in the decision space usually forms a piecewise continuous low-dimensional manifold structure. This structure reveals the underlying distribution patterns of the solution set in high-dimensional space, which is of great theoretical and practical significance for understanding algorithm behaviors and improving optimization performance.

Mathematically, let $X \subset \mathbb{R}^n$ be the decision space. If $S \subset X$ is an m-dimensional submanifold of X, then for any $\boldsymbol{x} \in S$, there exists a neighborhood U and a smooth embedding mapping ϕ from $\mathbb{R}^m$ to X such that $\phi(\mathbb{R}^n) \cap U = S \cap U$. In short, a submanifold is a low-dimensional space in the ambient space that locally preserves the Euclidean structure and differentiability, and around every point, it is homeomorphic to $\mathbb{R}^m$.

In the proposed ISPP method, we first segment the PF in the objective space and fit hyperplanes and then map each division point and its neighborhood in the objective space to a corresponding subpopulation in the decision space. ISPP assumes and exploits the fact that the historical optimal solutions within each subpopulation are not only clustered around a central point but also distributed along specific directions or structures. This local structure can be modeled as a submanifold.

Under mild conditions, according to the Karush–Kuhn–Tucker (KKT) conditions, the PS of an MOP with m objectives forms an $(m-1)$-dimensional piecewise continuous manifold [45]. Therefore, the two consecutive PSs at adjacent time steps often exhibit high structural similarity. Based on this property, the submanifold at the previous time step can be used to predict the submanifold at the next time step.

The prediction formula for the submanifold S_k^{t+1} in the new environment is

$$\tilde{x}_{k,i}^{t+1} = \tilde{x}_{k,i}^{t} + N(0, \sigma_{k,i}^{s}) \tag{9}$$

The formula for calculating the variance $\sigma_{k,i}^{s}$ is $\sigma_{k,i}^{s} = \frac{1}{n}\frac{1}{|S_k^t|}\sum_{\tilde{x}_{k,i}^{t}\in S_k^t}\min_{\tilde{x}_{k,i}^{t-1}\in S_k^{t-1}}(\tilde{x}_{k,i}^{t} - \tilde{x}_{k,i}^{t-1})^2$.

3.5 PS initial solution generation

Let PS^{t+1} be the initial population at time $t+1$ and let PS_k^{t+1} be the initial subpopulation at time $t+1$. Then, based on Eqs. (6), (8), and (9), we can obtain PS^{t+1}.

Algorithm 2 Pseudocode of generating the initial population

Input: the current subpopulation, PS_k^t;
Output: the next initial population, PS^{t+1};
1: Let $\boldsymbol{c}_k^t$ be the center of the k - th subpopulation. To obtain the center $\boldsymbol{c}_k^t$, use Formula (6).
2: Let S_k^t be the k-th submanifold with the center at the origin. To obtain the submanifold S_k^t, use Formulas (6) and (7).
3: Detecting environmental changes, at $t = 1$ and $t = 2$, randomly generate the initial populations.
4: When $t \geq 2$, use the stationary time series $\left\{\boldsymbol{c}_k^1, \boldsymbol{c}_k^2, \cdots, \boldsymbol{c}_k^{t-1}, \boldsymbol{c}_k^t\right\}$ to predict the center of the k-th subpopulation at $t+1$ using Formula (8).
5: According to the KKT conditions, predict the submanifold of the k-th subpopulation at $t+1$ using Formula (9)
6: Let PS^{t+1} be the initial population at time $t+1$, and PS_k^{t+1} be the initial subpopulation at time $t+1$. Then, the initial population PS^{t+1} can be obtained from Equations (6), (8), and (9).

3.6 ISPP procedure

This paper combines the proposed prediction strategy ISPP with the static multiobjective optimization method NSGA-III to obtain the PS in the new environment. Algorithm 3 describes the specific process of the proposed DMOA.

Algorithm 3 Framework of ISPP

Input: the size of PS, N; Number of decision variables, n; Number of objective functions, m; Number of dividing lines, K; Number of individuals used for change detection, N_d; Maximal number of iterations, $g_{\max}$;

Output: offspring population P^t;

1: Detect environmental changes. If changes detected, go to step 2; if no changes detected, go to step 4;

2: Initialize new population Using Algorithms 1 and 2;

3: Reproduce the offspring population P^t by NSGA-III;

4: Check termination conditions. If $g_t > g_{\max}$, output P^t, and end; or $g_t = g_{t+1}$, go to Step 1.

3.7 Computational complexity analysis

The computational cost associated with ISPP-DMOEA results primarily from SMOEA (NSGA-III) and the dynamic response mechanism, as illustrated in Algorithm 3. First, the computational complexity of NSGA-III is given by $O(mN^2)$, where m represents the number of objectives and N denotes the population size, as indicated in [39]. For the dynamic response mechanism, costs mainly stem from determining the intersection points between the dividing lines and hyperplanes, as well as from predicting the corresponding subpopulations. According to Algorithm 1, the complexity of fitting the PF using a least squares method to find the intersection points between dividing lines and hyperplanes is $O(m^2N + m^3)$. The complexity for calculating the intersection points of the hyperplane with$K$ dividing lines is $O(L)$, and the complexity for assigning the N individuals in the PF to subsets corresponding to the K intersection points via a nearest neighbor approach is $O(N)$. Thus, the worst-case complexity for intersections between the dividing lines and hyperplanes can be approximated as $O(m^2N + m^3 + L + N) \approx O(m^2N)$. As shown in Algorithm 2, for predicting the associated subpopulations, the complexity for forecasting the time series composed of the K centers using the AR(p) model is $O(pL)$. For the prediction of the associated subpopulation P_k at intersection point k (where $(1 \leq k \leq K)$), calculating the mean and covariance incurs complexities of $O(nkD)$ and $O(nkD^2)$, respectively, where D is the number of decision variables. Therefore, the worst-case complexity for predicting associated subpopulations is $O(pL + ND + ND^2) \approx O(ND^2)$. In summary, the overall worst-case computational complexity of the dynamic response mechanism within ISPP-DMOEA is $O(ND^2)$.

4 Test instances and performance metric

4.1 Test instances

We compared the performance of the algorithm on 16 test functions with two or three objectives, including six newly proposed test functions. These test functions include three dynamic two-objective test functions, FDA1, FDA2, and FDA3, where the PF is continuous and the PS translates in the same direction at the same frequency, as well as two dynamic three-objective test functions, FDA4 and FDA5 [38]. In addition, two dynamic two-objective test functions, dMOP1 and dMOP2 [39], are used. Two dynamic two-objective test functions, Fun7 and Fun8 [17], are also included, where the PF is continuous and the PS rotates in the same direction at the same frequency. Finally, it includes the dynamic two-objective test function HE2 [40], where the PF is discontinuous and the PS translates in the same direction at the same frequency. The information regarding the six proposed test functions is detailed in Table 1. Variants LSF1, LSF2, and LSF3 are based on the two-objective dynamic test functions from the FDA test series, whereas variants LSF4 and LSF5 are based on HE2, and variant LSF6 is based on FDA4.

LSF1 is a type I optimization problem, where the PS varies over time as follows: [0, 0.5] moves upward, and (0.5, 1] moves downward with the same frequency. The PF remains disconnected. LSF2 is a type II optimization problem, where the PS changes over time as follows: [0, 0.5] moves upward,

Table 1 Six newly proposed test instances

Instance	Definition	Remarks
LSF1 (type I)	$f_1(x_1, \cdots, x_n) = (1+g)\cdot\cos(0.5\pi x_1)$, $f_2(x_1, \cdots, x_n) = (1+g)\cdot\sin(0.5\pi x_1)$, $g(x_1, \cdots, x_n) = \sum_{i=2}^{n}(x_i - G(t))^2$ $G(t) = \begin{cases} \lvert\sin(0.5\pi t)\rvert, x_1 \in [0, 0.5] \\ \lvert\cos(0.5\pi t)\rvert, x_1 \in (0.5, 1] \end{cases}$ $where: x_i \in [0, 1]$	PS: [0, 0.5] moves upward, and (0.5, 1] moves downward with the same frequency PF: disconnected $f_1 = \cos(0.5\pi x_1)$, $f_2 = \sin(0.5\pi x_1)$, $where: x_i \in [0, 1]$
LSF2 (type II)	$f_1(x_1, \cdots, x_n) = (1+g)\cdot\cos(0.5\pi x_1)$, $f_2(x_1, \cdots, x_n) = (1+g)\cdot\sin(0.5\pi x_1)$, $g(x_1, \cdots, x_n) = G(t) + \sum_{i=2}^{n}(x_i - G(t))^2$ $G(t) = \begin{cases} \lvert\sin(0.5\pi t)\rvert, x_1 \in [0, 0.5] \\ \lvert\sin(0.6\pi t)\rvert, x_1 \in (0.5, 1] \end{cases}$ $where: x_i \in [0, 1]$	PS: [0, 0.5] moves upward, and (0.5, 1] moves downward with different frequencies PF: disconnected (shifted along different directions with different frequencies based on $f_1 = f_2$ as the boundary) $f_1 = (1+G(t))\cdot\cos(0.5\pi x_1)$, $f_2 = (1+G(t))\cdot\sin(0.5\pi x_1)$, $G(t) = \begin{cases} \lvert\sin(0.5\pi t)\rvert, x_1 \in [0, 0.5] \\ \lvert\sin(0.6\pi t)\rvert, x_1 \in (0.5, 1] \end{cases}$ $where: x_i \in [0, 1]$
LSF3 (type II)	$f_1(x_1, \cdots, x_n) = (1+g)\cdot\cos(0.5\pi x_1)$, $f_2(x_1, \cdots, x_n) = (1+g)\cdot\sin(0.5\pi x_1)$, $g(x_1, \cdots, x_n) = G(t) + \sum_{i=2}^{n}(x_i - G(t)\cdot\lvert x_1 - 0.5\rvert)^2$ $G(t) = \begin{cases} \lvert\sin(0.5\pi t)\rvert, x_1 \in [0, 0.5] \\ \lvert\sin(0.6\pi t)\rvert, x_1 \in (0.5, 1] \end{cases}$ $where: x_1 \in [0, 1], x_i \in [-1, 1]$	PS: with 0.5 as the center, rotate the left side clockwise and the right side counterclockwise at different frequencies PF: disconnected (shifted along different directions with different frequencies based on $f_1 = f_2$ as the boundary) $f_1 = (1+G(t))\cdot\cos(0.5\pi x_1)$, $f_2 = (1+G(t))\cdot\sin(0.5\pi x_1)$, $G(t) = \begin{cases} \lvert\sin(0.5\pi t)\rvert, x_1 \in [0, 0.5] \\ \lvert\sin(0.6\pi t)\rvert, x_1 \in (0.5, 1] \end{cases}$ $where: x_1 \in [0, 1], x_i \in [-1, 1]$
LSF4 (type I)	$f_1(x_1) = x_1$, $f_2(x_1, \cdots, x_n) = 1 + g\cdot h$, $g(x_2, \cdots, x_n) = 1 + \sum_{i=2}^{n}(x_i - G(t))^2$, $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\cdot\sin(10\pi f_1)$ $G(t) = \begin{cases} \lvert\sin(0.1\pi t)\rvert, x_1 \in [0, 0.5] \\ \lvert\cos(0.1\pi t)\rvert, x_1 \in (0.5, 1] \end{cases}$ $where: x_i \in [0, 1]$	PS: [0, 0.5] moves upward, and (0.5, 1] moves downward with same frequencies PF: disconnected $f_1 = x_1$, $f_2 = 1 + h$, $h = 1 - \sqrt{f_1} - f_1\cdot\sin(10\pi f_1)$ $G(t) = \begin{cases} \lvert\sin(0.1\pi t)\rvert, x_1 \in [0, 0.5] \\ \lvert\cos(0.1\pi t)\rvert, x_1 \in (0.5, 1] \end{cases}$ $where: x_i \in [0, 1]$
LSF5 (type I)	$f_1(x_1) = x_1$, $f_2(x_1, \cdots, x_n) = 1 + g\cdot h$, $g(x_1, \cdots, x_n) = 1 + \sum_{i=2}^{n}(x_i - G(t)\cdot x_1)^2$, $h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\cdot\sin(10\pi f_1)$ $G(t) = \lvert\sin(0.5\pi t)\rvert$ $where: x_i \in [0, 1]$	PS: rotate counterclockwise around the origin of the coordinate system PF: disconnected $f_1 = x_1$, $f_2 = 1 + h$, $h = 1 - \sqrt{f_1} - f_1\cdot\sin(10\pi f_1)$ $G(t) = \lvert\sin(0.5\pi t)\rvert$ $where: x_i \in [0, 1]$

Table 1 (continued)

Instance	Definition	Remarks
LSF6 (type I)	$f_1(x_1, \cdots, x_n) = (1+g)\cdot\cos(0.5\pi x_1)\cdot\cos(0.5\pi x_2)$, $f_2(x_1, \cdots, x_n) = (1+g)\cdot\cos(0.5\pi x_1)\cdot\sin(0.5\pi x_2)$, $f_3(x_1, x_3, \cdots, x_n) = (1+g)\cdot\sin(0.5\pi x_1)$, $g(x_1, x_3, \cdots, x_n) = \sum_{i=3}^{n}(x_i - G(t)\cdot x_1)^2$ $G(t) = \lvert\sin(0.5\pi t)\rvert$ $where:\ x_i \in [0, 1]$	PS: rotate counterclockwise around the origin of the coordinate system PF: continuous $f_1 = \cos(0.5\pi x_1)\cdot\cos(0.5\pi x_2)$, $f_2 = \cos(0.5\pi x_1)\cdot\sin(0.5\pi x_2)$, $f_3 = \sin(0.5\pi x_1)$, $G(t) = \lvert\sin(0.5\pi t)\rvert$ $where:\ x_i \in [0, 1]$

and (0.5, 1] moves downward with different frequencies. The PF is disconnected (shifted along different directions with different frequencies and based on $f_1 = f_2$ as the boundary). LSF3 is also a type II optimization problem, where the PS varies over time as follows: with 0.5 as the center, it rotates the left side clockwise and the right side counterclockwise at different frequencies. The PF is disconnected (shifted along different directions with different frequencies and based on $f_1 = f_2$ as the boundary). LSF4 is a type I optimization problem, where the PS varies over time as [0, 0.5] moves upward, and (0.5, 1] moves downward with the same frequency. The PF remains disconnected. LSF5 is a type I optimization problem, where the PS rotates counterclockwise around the origin of the coordinate system. The PF remains disconnected. LSF6 is a type I optimization problem, where the PS rotates counterclockwise around the origin of the coordinate system, and the PF is continuous.

4.2 Performance metric

In this paper, the mean inverted generational distance (MIGD) [13] is used to evaluate the convergence and diversity of PSs obtained from different prediction strategies. Let PF^t and P^t represent the true PF and the PF obtained by the DMOA under the t-th environment, respectively. Based on the inverted generational distance (IGD) [46], the MIGD of the DMOA is defined as:

$$MIGD = \frac{1}{|T|}\sum_{t\in T} IGD(PF^t, P^t) \tag{10}$$

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

$$IGD = \frac{\sum_{x\in PF^t} \min d\left(x, P^t\right)}{|PF^t|}$$

Thus, the MIGD can comprehensively evaluate both the convergence and distribution of the PS obtained by the optimization algorithm, where $d(x, P^t)$ is the Euclidean distance between x and the point in P^t. $\left|PF^t\right|$ is the cardinality of PF^t. The lower the MIGD value is, the better the tracing performance.

For the true PF, 1000 uniformly distributed points are taken for two-objective optimization problems, and 10,000 uniformly distributed points are taken for three-objective optimization problems. A test instance is run independently 20 times, and the MIGD for different optimization algorithms is calculated.

5 Experimental results and analysis

5.1 Compared algorithms and parameter settings

In this study, four competitive DMOEAs are selected for comparison with ISPP-DMOEA, namely, PPS-DMOEA [13], MDP-DMOEA [17], KTM-DMOEA [34], and RNN-DMOEA [35]. All algorithms utilize the same static solver, NSGA-III. The specific parameters for each of the compared DMOEAs are configured according to the recommendations provided in their respective original publications [13, 17, 34, 35]. The time variable t for all test problems is expressed as $t = \left(1/n_t\right)\left\lfloor \tau/\tau_t \right\rfloor$, where n_t represents the severity of environmental changes and τ_t denotes the frequency of these changes, with τ indicating the generation counter.

The parameter settings are as follows.

(1) The MOEA uses NSGA-III, the crossover probability is 0.7, and the mutation probability is 0.3.
(2) For a dynamic optimization problem with two objectives, the population size is $N = 100$. For a dynamic optimization problem with three objectives, the population size is $N = 200$. For all test instances, the dimension of the search space is 10.

(3) For the dynamic optimization problem with two objectives, dividing lines are set as $K = 10$. Among these, FDA1, FDA2, FDA3, dMOP1, dMOP2, Fun8, F7, LSF1, LSF2, and LSF3 use Method 1 to set the dividing lines. Fun7, HE2, LSF4, and LSF5 use Method 2 to set the dividing lines. For the dynamic optimization problem with three objectives, FDA4, FDA5, and LSF6, dividing lines are set as $K = 5*5= 25$.
(4) When 5% of the individuals are selected for environmental change detection, 100 environmental changes are selected in each run. The environmental dynamics are set to $(n_t, \tau_t) \in \{(5, 20), (10, 20), (10, 30)\}$, each test instance is run independently 20 times, and the problems change every 30 generations.

5.2 Comparison of performance evaluation results

This section first visualizes the prediction performance of ISPP using 30 consecutive time points with an interval of 2. The selected test instances are (a) dMOP2, (b) Fun7, (c) LSF3, and (d) LSF5, where $t = 60, 62, 64, \ldots, 90$, $t + 1 = 61, 63, 65, \ldots, 91$.$n_t = 10$, $\tau_t = 30$. Figure 6 shows the final PF and PS at time t, and Fig. 7 depicts the predicted initial PF and PS at time $t + 1$.

Dynamic multiobjective optimization problems and most environmental prediction tasks generally exhibit stagewise evolution characteristics. In the initial stage, data are relatively scarce, and model training is insufficient, making it crucial to test the algorithm's adaptability and noise resistance. As the process enters the intermediate stage, the data accumulate, and the model begins to stabilize; therefore, the focus shifts to evaluating the algorithm's convergence speed and transitional adaptation ability. In the later stage, with ample data, predictive methods mature, and attention turns to their long-term stability and generalization performance. To conduct an in-depth analysis of how the quantity and quality of stored historical information affect prediction strategies, this paper divides the entire dynamic process into three periods: $0 \leq t \leq 20$ (initial), $21 \leq t \leq 60$ (intermediate), and $61 \leq t \leq 100$ (later), while also presenting results for the overall period $0 \leq t \leq 100$, with the environmental dynamics changing to $n_t = 10$, $\tau_t = 30$. By segmenting algorithm performance according to these stages in this paper, we are able to reveal the advantages and disadvantages of each algorithm at different phases directly and investigate the impact of historical data accumulation on prediction performance. Moreover, this type of segmentation analysis provides valuable insights for algorithm selection in practical applications.

The performance of the MIGD is compared across these segments. Specifically, Table 2 presents the statistical results of each algorithm's MIGD after 20 independent runs in the respective time intervals, and Fig. 8 illustrates the dynamic curves of the MIGD values over time. By evaluating algorithm performance in different evolutionary stages, this segmented analysis comprehensively reveals the strengths and weaknesses of various prediction strategies, providing valuable guidance for algorithm optimization and practical system selection.

According to the different variation characteristics of PF and PS, all the test problems are divided into seven categories, and then the experimental results are discussed.

Category 1: Test instances where the PF is continuous and the PS translates in the same direction at the same frequency. Examples include FDA1, FDA2, FDA3, FDA4, FDA5, dMOP1, dMOP2, and LSF6.

Category 2: Test instances where the PF is continuous and the PS rotates in the same direction at the same frequency. Examples include Fun7 and Fun8.

Category 3: Test instances where the PF is disconnected and the PS translates in different directions at the same frequency. Example: LSF1.

Category 4: Test instances where the PF is disconnected and the PS translates in different directions at different frequencies. Example: LSF2.

Category 5: Test instances where the PF is disconnected and the PS rotates in different directions at different frequencies. Example: LSF3.

Category 6: Test instances where the PF is disconnected and the PS rotates in different directions at the same frequency. Example: LSF4.

Category 7: Test instances where the PF is disconnected and the PS rotates in the same direction at the same frequency. Example: LSF5.

(1) Experimental results and discussion of the first-category test problems.

As shown in Table 2, for some test problems, when $0 \leq t \leq 20$, the prediction results of the MDP and KTM outperform those of the other comparative algorithms. The reason for this phenomenon is mainly due to the limited historical information stored by PPS, RNN, and ISPP during the initial stage. When $21 \leq t \leq 60$, $61 \leq t \leq 100$, and $0 \leq t \leq 100$, the prediction performance of ISPP is better than those of PPS, MDP, KTM, and RNN for most test problems. Among them, when $0 \leq t \leq 100$, for FDA3, the prediction performance of KTM is significantly better than those of the other four methods, mainly because the distribution of the PS under different environments for FDA3 is very suitable for predictions using the KTM method. For dMOP2, the prediction performance of RNN is significantly worse than those of the other four methods, mainly because RNN may overfit the local (non)linear fluctuations.

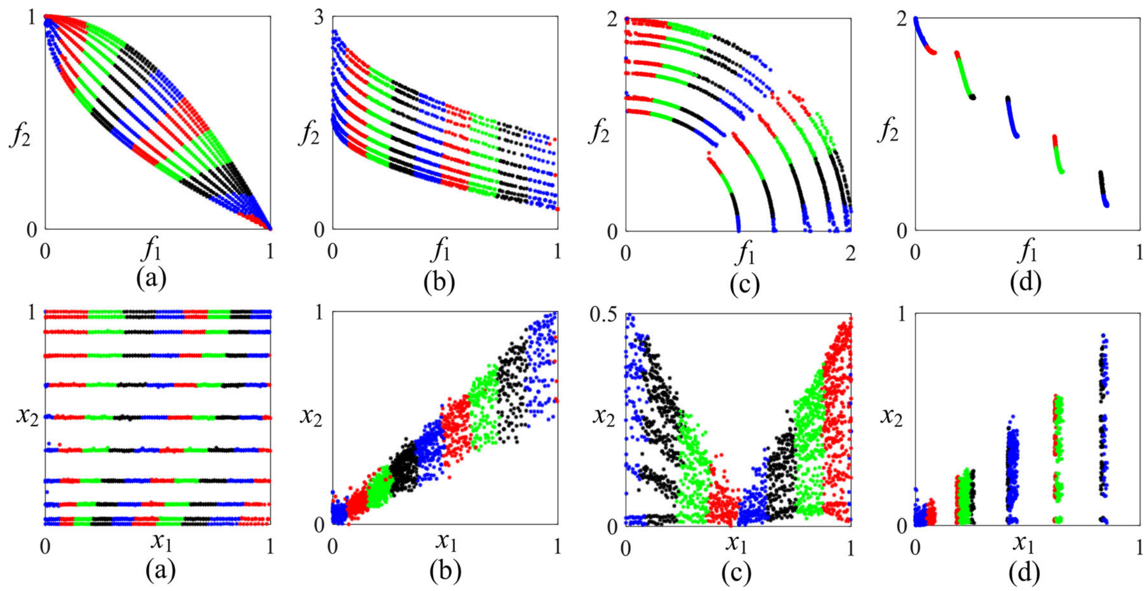


Fig. 6 Final PF and PS at time t

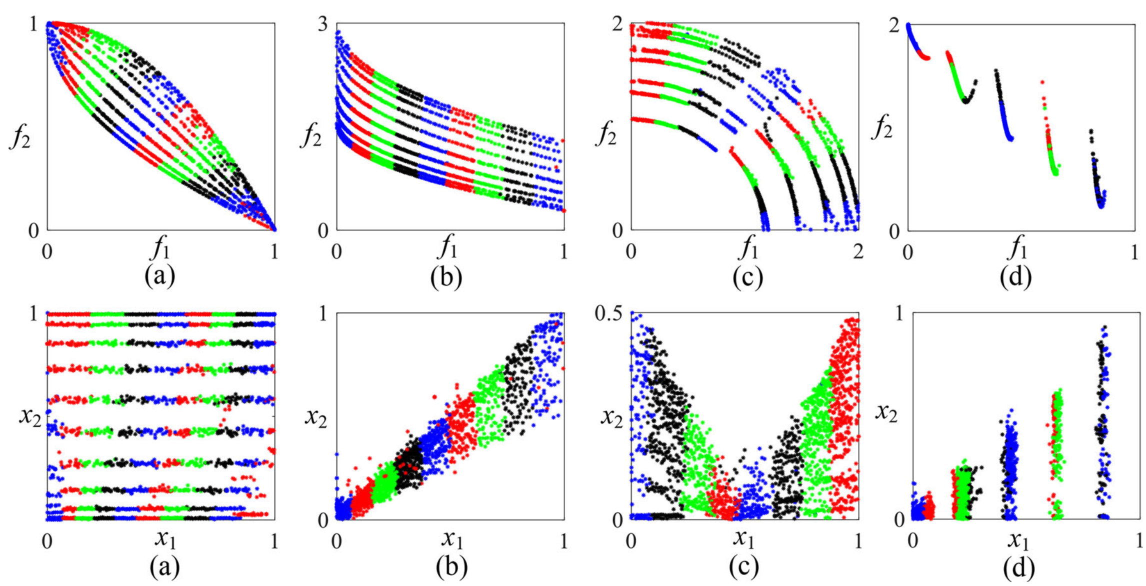


Fig. 7 Predicted initial PF and PS at time t + 1

(2) Experimental results and discussion of the second-category test problems.

As shown in Table 2, for Fun7, the prediction performance of ISPP is superior to those of the other four comparative methods during different time periods. For Fun8, except when $0 \leq t \leq 20$, where ISPP outperforms the other four methods, RNN shows significantly better prediction performance than the other four methods when $21 \leq t \leq 60$, $61 \leq t \leq 100$, and $0 \leq t \leq 100$. The reason for this phenomenon is that the PS of Fun7 rotates around the origin, with its center point constantly changing, whereas the PS of Fun8 rotates around a fixed center point.

(3) Experimental results and discussion of the third-category test problems.

As shown in Table 2, for LSF1, when $0 \leq t \leq 20$, $21 \leq t \leq 60$, $61 \leq t \leq 100$, and $0 \leq t \leq 100$, the prediction performances of ISPP and MDP are comparable; both outperform the other three comparative algorithms, with ISPP being slightly better overall. This result indicates that although the historical information stored by ISPP is limited during the

Table 2 Results of MIGD obtained by PPS, MDP, KTM, RNN, and ISPP over 20 runs

Instance	Strategy	$0 \le t \le 100$	$0 \le t \le 20$	$21 \le t \le 60$	$61 \le t \le 100$
FDA1	PPS	0.0047	0.0070	0.0042	0.0041
	MDP	0.0048	**0.0049**	0.0047	0.0047
	KTM	0.0049	0.0056	0.0048	0.0047
	RNN	0.0053	0.0063	0.0055	0.0052
	ISPP	**0.0045**	0.0066	**0.0039**	**0.0040**
FDA2	PPS	0.0053	0.0046	0.0053	0.0057
	MDP	0.0059	0.0047	0.0060	0.0065
	KTM	0.0058	0.0043	0.0070	0.0053
	RNN	0.0070	**0.0042**	0.0059	0.0094
	ISPP	**0.0050**	0.0048	**0.0048**	**0.0052**
FDA3	PPS	0.0122	0.0087	0.0135	0.0126
	MDP	0.0095	**0.0076**	0.0101	0.0099
	KTM	**0.0070**	**0.0076**	0.0068	0.0069
	RNN	0.0091	0.0101	0.0097	0.0082
	ISPP	0.0080	0.0135	**0.0063**	**0.0068**
FDA4	PPS	0.0751	0.0775	0.0765	0.0734
	MDP	0.0543	0.0495	0.0521	0.0590
	KTM	0.0511	**0.0467**	0.0534	**0.0511**
	RNN	0.0729	0.0710	0.0739	0.0729
	ISPP	**0.0507**	0.0510	**0.0496**	0.0516
FDA5	PPS	0.1111	0.1110	0.1113	0.1109
	MDP	0.0880	0.0856	0.0878	0.0893
	KTM	**0.0817**	0.0802	**0.0825**	0.0815
	RNN	0.1133	0.1078	0.1146	0.1148
	ISPP	0.0844	**0.0796**	0.0913	**0.0799**
dMOPl	PPS	0.0057	0.0065	0.0054	0.0055
	MDP	0.0054	0.0069	0.0050	0.0050
	KTM	0.0050	0.0075	0.0042	0.0045
	RNN	0.0048	0.0081	0.0040	**0.0040**
	ISPP	**0.0044**	**0.0056**	**0.0040**	0.0041
dMOP2	PPS	0.0049	0.0081	0.0042	0.0041
	MDP	0.0053	**0.0058**	0.0052	0.0052
	KTM	0.0058	0.0074	0.0053	0.0055
	RNN	0.0129	0.0170	0.0130	0.0106
	ISPP	**0.0046**	0.0078	**0.0038**	**0.0037**
Fun7	PPS	0.0104	0.0162	0.0096	0.0083
	MDP	0.0092	0.0127	0.0081	0.0085
	KTM	0.0296	0.0134	0.0380	0.0292
	RNN	0.0101	0.0122	0.0110	0.0083
	ISPP	**0.0087**	**0.0112**	**0.0078**	**0.0081**
Fun8	PPS	0.0313	0.0115	0.0299	0.0425
	MDP	0.0220	0.0087	0.0271	0.0285
	KTM	0.0423	0.0100	0.0709	0.0299
	RNN	**0.0164**	0.0079	**0.0176**	**0.0195**
	ISPP	0.0217	**0.0070**	0.0270	0.0236
HE2	PPS	0.1777	0.1667	0.1810	0.1806
	MDP	0.1783	0.1704	0.1790	0.1807
	KTM	0.1786	0.1643	0.1822	0.1823
	RNN	0.1810	0.1875	0.1792	0.1796
	ISPP	**0.1762**	**0.1640**	**0.1780**	**0.1774**
LSF1	PPS	0.1021	0.0881	0.0974	0.1139
	MDP	0.0155	0.0320	0.0115	0.0127
	KTM	0.0767	0.0978	0.0696	0.0733
	RNN	0.0366	0.0546	0.0304	0.0338
	ISPP	**0.0154**	**0.0290**	**0.0112**	**0.0112**

Table 2 (continued)

Instance	Strategy	$0 \leq t \leq 100$	$0 \leq t \leq 20$	$21 \leq t \leq 60$	$61 \leq t \leq 100$
LSF2	PPS	0.0765	0.0558	0.1094	0.0539
	MDP	0.0387	0.0259	0.0504	0.0342
	KTM	0.0727	0.0340	0.1066	0.0581
	RNN	0.1044	0.1240	0.0772	0.1219
	ISPP	**0.0378**	**0.0250**	**0.0501**	**0.0311**
LSF3	PPS	0.0410	0.0433	0.0515	0.0309
	MDP	0.0434	0.0406	0.0562	0.0321
	KTM	0.0465	0.0382	0.0592	0.0387
	RNN	0.0385	0.0391	**0.0483**	0.0287
	ISPP	**0.0384**	**0.0366**	0.0498	**0.0279**
LSF4	PPS	0.2037	0.2411	0.1953	0.1934
	MDP	0.2364	0.2202	0.2413	0.2395
	KTM	0.1938	0.1941	0.1939	0.1936
	RNN	0.1928	0.1943	0.1923	0.1924
	ISPP	**0.1916**	**0.1916**	**0.1914**	**0.1917**
LSF5	PPS	0.2040	0.2064	0.2073	0.1996
	MDP	0.2257	0.2359	0.2168	0.2294
	KTM	0.2097	0.2119	0.2118	0.2065
	RNN	0.1938	**0.1931**	0.1939	0.1942
	ISPP	**0.1934**	0.1932	**0.1933**	**0.1937**
LSF6	PPS	0.0640	0.0726	0.0623	0.0614
	MDP	0.0814	0.0861	0.0835	0.0770
	KTM	0.0658	0.0735	0.0643	0.0635
	RNN	0.0617	0.0763	0.0592	0.0568
	ISPP	**0.0584**	**0.0672**	**0.0579**	**0.0546**

The bold values highlight the statistically optimal mean results of MIGD achieved by different prediction strategies on each test instance over 20 runs across various stages of the entire dynamic process

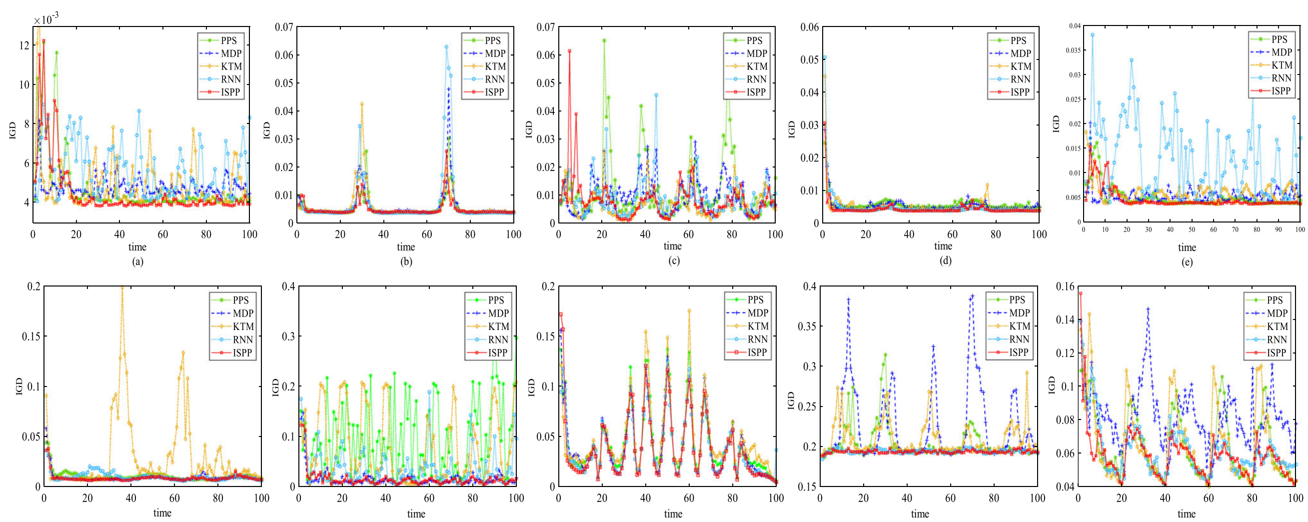


Fig. 8 Average IGD values over 20 runs versus time for PPS, MDP, and ISPP on **a** FDA1, **b** FDA2, **c** FDA3, **d** dMOPl, **e** dMOP2, **f** Fun7, **g** LSF1, **h** LSF3, (i) LSF5, and (j) LSF6

initial stage, the quality of the initial population produced by ISPP predictions is still better for this type of test function.

(4) Experimental results and discussion of the fourth-category test problems.

The experimental results are similar to those in (3). Table 2 shows that for LSF2, when $0 \le t \le 20$, $21 \le t \le 60$, $61 \le t \le 100$, and $0 \le t \le 100$, the prediction performances of ISPP and MDP are comparable; both outperform the other three comparative algorithms, with ISPP being slightly better overall.

(5) Experimental results and discussion of the fifth-category test problems.

Table 2 shows that for LSF3, when $0 \le t \le 20$, $21 \le t \le 60$, $61 \le t \le 100$, and $0 \le t \le 100$, the prediction performances of ISPP and RNN are comparable, and both outperform the other three comparative algorithms. RNN has a clear advantage in terms of prediction performance, mainly because the PS of LSF3 rotates around a fixed center point.

(6) Experimental results and discussion of the sixth-category test problems.

The experimental results are similar to those in (5). Table 2 shows that for LSF4, when $0 \le t \le 20$, $21 \le t \le 60$, $61 \le t \le 100$, and $0 \le t \le 100$, the prediction performances of ISPP and RNN are comparable; both outperform the other three comparative algorithms, with ISPP being slightly better overall.

(7) Experimental results and discussion of the seventh-category test problems.

The experimental results are similar to those in (5). Table 2 shows that for LSF5, when $0 \le t \le 20$, $21 \le t \le 60$, $61 \le t \le 100$, and $0 \le t \le 100$, the prediction performances of ISPP and RNN are comparable; both outperform the other three comparative algorithms, with ISPP being slightly better overall.

Additionally, from the overall data distribution in Table 2, the following three points can be observed:

(1) For different prediction strategies, the MIGD values for continuous test functions are lower than those for discontinuous test functions. As the variation characteristics of the PS become more complex, the MIGD values increase.
(2) For test functions where the PS changes when it rotates around a center point, the prediction performance of PPS is poor. This is because PPS describes the PS using a single center point and an $m-1$-dimensional manifold. Although the center point changes little under different environments, merely predicting the center of the PS does not sufficiently capture the trends in PS variation.
(3) For discontinuous test functions, the prediction performance of KTM is poor. This is primarily due to the sampling method used by KTM, which can introduce poor solutions by sampling from the distribution of constructed elite solutions, thereby affecting prediction performance. For continuous test functions, however, KTM can achieve better distributed solutions, especially in tracking the boundary points of the target space more effectively.

To clearly observe the quality of the obtained initial populations, we also draw the initial populations obtained by PPS, MDP, KTM, RNN and ISPP with the lowest MIGD metric values at different times on (a) FDA2, (b) dMOP1, (c) Fun7, (d) LSF2, (e) LSF3, (f) HE2, and (g) LSF5 in Fig. 9.

Figure 9 shows that for all test functions, the quality of the initial population predicted by ISPP is better. Furthermore, the quality of the initial predicted population for continuous PF test functions is significantly higher than those for discontinuous PF test functions. The more complex the variation characteristics of the PS are, the more the advantages of ISPP are demonstrated.

To further observe the quality of the final population obtained, we plotted the MIGD values of the lowest final populations for some test functions in Fig. 10, which revealed that when the static optimization methods are consistent, after running 30 generations on FDA2, dMOP1, Fun7, and LSF5, the quality of the final populations produced by all five methods is good; however, the distribution of the final population obtained by ISPP is better. After running 30 generations on LSF2, the final populations produced by the four comparative algorithms (PPS, MDP, KTM, RNN) have poor distributions, whereas the final population produced by ISPP is of good quality. The main reason for this is the different initial population qualities predicted by the five prediction strategies, leading to differences in the final tracking performance.

To assess the influence of the severity and frequency of environmental changes on the effectiveness of prediction strategies, Table 3 presents the MIGD statistical results from 20 independent runs for three different parameter combinations: $(n_t, \tau_t) \in \{(5, 20), (10, 20), (10, 30)\}$. The statistically optimal mean results for each test instance are highlighted in bold. To ensure valid conclusions, the Wilcoxon rank-sum test is conducted at a significance level of 0.05. In the second-to-last row of Table 3, the symbols " + ",

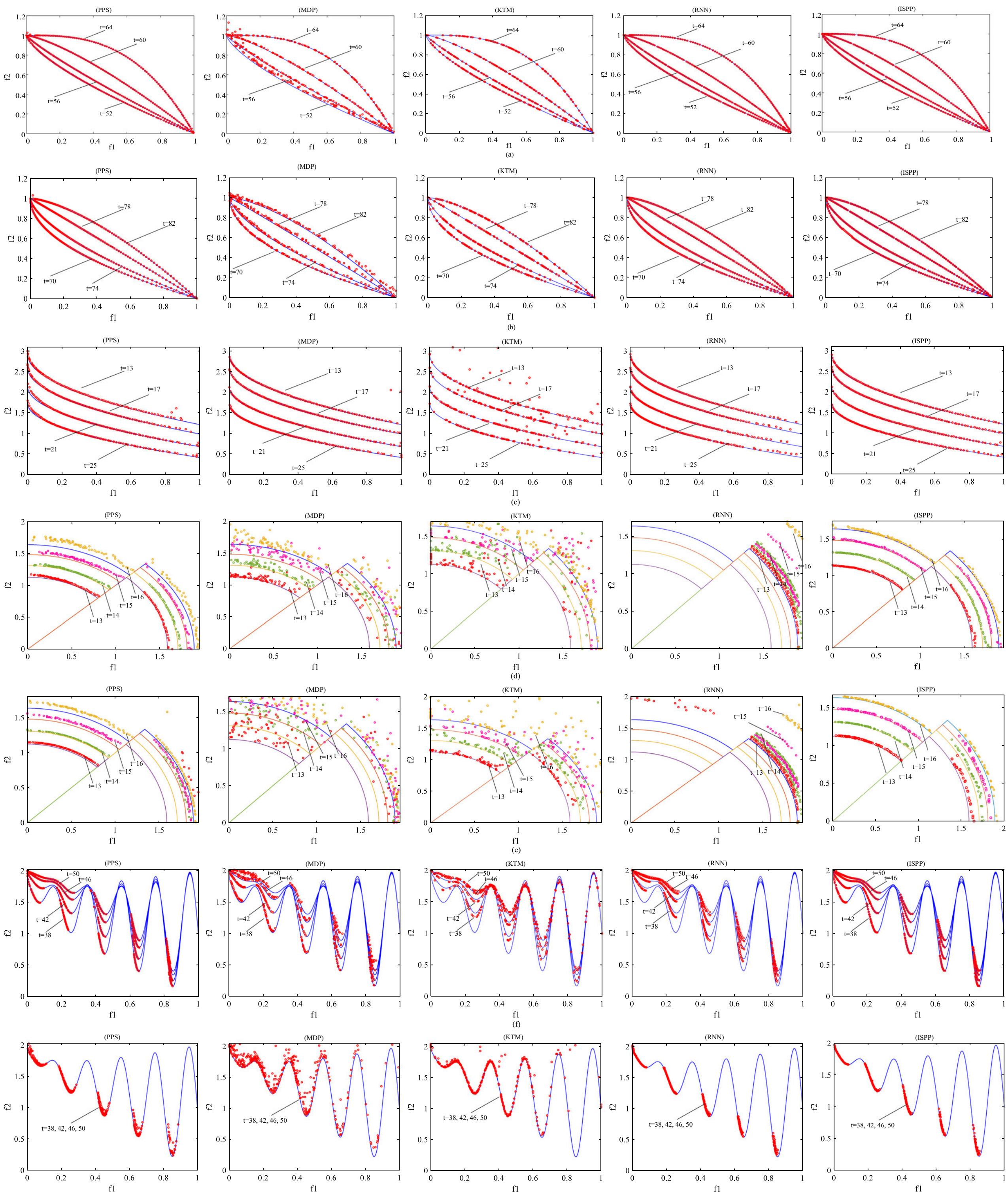


Fig. 9 Initial populations obtained by PPS, MDP, and ISPP at different times with the lowest MIGD values among 20 runs on **a** FDA2, **b** dMOP1, **c** Fun7, **d** LSF2, **e** LSF3, **f** HE2, and **g** LSF5

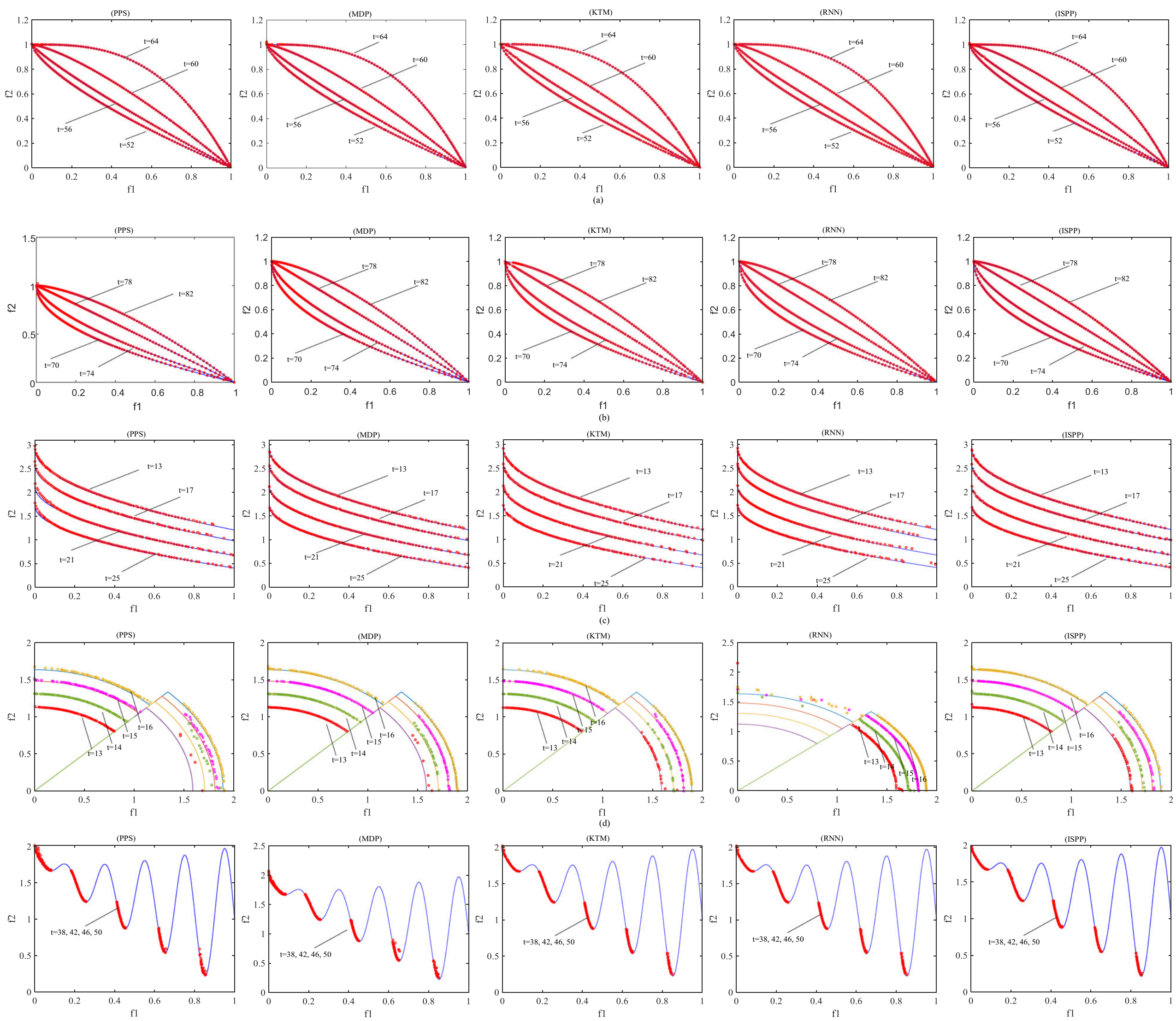


Fig. 10 Final populations obtained by PPS, MDP, and ISPP at different times with the lowest MIGD values among 20 runs on **a** FDA2, **b** dMOP1, **c** Fun7, **d** LSF2, and **e** LSF5

" = ", and "–" summarize the outcomes where ISPP-NSGA-III outperforms, performs comparably to, or underperforms relative to its competitors. The last row indicates the percentage of the best statistically significant mean results achieved by each algorithm across all test instances.

As shown, ISPP achieved the best results in 39 out of 48 scenarios when tackling the selected test problems, whereas PPS, MDP, KTM, and RNN had the best results in only 0, 0, 5, and 4 cases, respectively. This suggests that ISPP performs better at tracking the PF in most scenarios. In particular, under most dynamic conditions for FDA1, FDA2, FDA4, FDA5, dMOP1, dMOP2, Fun7, HE2, and the LSF series, ISPP significantly outperformed the other algorithms, demonstrating a clear advantage in adapting to changes in PF structures.

However, ISPP performed somewhat worse than its best competitors on FDA3 and Fun8, mainly because of the specific characteristics of these cases. In particular, the PS distribution in FDA3 is well suited for prediction using the KTM method, whereas the rotational nature of the PS around the center point in Fun8 is more efficiently handled by the nonlinear predictor RNN. Overall, compared with the four other algorithms, ISPP exhibits superior optimization performance and shows lower sensitivity to variations in frequency and severity.

5.3 Sensitivity study

In the ISPP, there are two important parameters that users need to identify: the number of dividing lines (k) in the ISPP

Table 3 Results of MIGD obtained by PPS, MDP, KTM, RNN, and ISPP over 20 runs with three dynamic setups

Problem	$(n_t,\ \tau_t)$	PPS	MDP	KTM	RNN	ISPP
FDA1	(10,30)	0.0047(=)	0.0048(=)	0.0049(=)	0.0053(+)	**0.0045**
	(10,20)	0.0054(=)	0.0058(=)	0.0053(=)	0.0073(+)	**0.0052**
	(5,20)	0.0082(+)	0.0092(+)	0.0080(+)	0.0095(+)	**0.0073**
FDA2	(10,30)	0.0053(=)	0.0059(+)	0.0058(+)	0.0070(+)	**0.0050**
	(10,20)	0.0064(+)	0.0073(+)	0.0083(+)	0.0077(+)	**0.0055**
	(5,20)	0.0076(+)	0.0086(+)	0.0089(+)	0.0092(+)	**0.0064**
FDA3	(10,30)	0.0122(+)	0.0095(+)	**0.0070(-)**	0.0091(+)	0.0080
	(10,20)	0.0163(+)	0.0136(+)	**0.0084(-)**	0.0128(+)	0.0116
	(5,20)	0.0861(+)	0.0461(+)	**0.0207(-)**	0.0429(+)	0.0316
FDA4	(10,30)	0.0751(+)	0.0543(+)	0.0511(=)	0.0729(+)	**0.0507**
	(10,20)	0.1012(+)	0.0789(+)	0.0647(+)	0.1015(+)	**0.0605**
	(5,20)	0.1618(+)	0.1231(+)	**0.0691(-)**	0.1561(+)	0.0769
FDA5	(10,30)	0.1111(+)	0.0880(+)	**0.0817(-)**	0.1133(+)	0.0844
	(10,20)	0.1527(+)	0.1231(+)	0.0892(+)	0.1500(+)	**0.0878**
	(5,20)	0.2486(+)	0.1555(+)	0.1121(+)	0.2438(+)	**0.0958**
dMOPl	(10,30)	0.0057(+)	0.0054(+)	0.0050(+)	0.0048(=)	**0.0044**
	(10,20)	0.0066(+)	0.0064(+)	0.0054(+)	0.0051(=)	**0.0048**
	(5,20)	0.0071(+)	0.0069(+)	0.0070(+)	0.0062(+)	**0.0049**
dMOP2	(10,30)	0.0049(=)	0.0053(+)	0.0058(+)	0.0129(+)	**0.0046**
	(10,20)	0.0060(+)	0.0068(+)	0.0078(+)	0.0351(+)	**0.0055**
	(5,20)	0.0150(+)	0.0203(+)	0.0129(+)	0.1259(+)	**0.0083**
Fun7	(10,30)	0.0104(+)	0.0092(+)	0.0296(+)	0.0101(+)	**0.0087**
	(10,20)	0.0205(+)	0.0181(+)	0.0723(+)	0.0218(+)	**0.0135**
	(5,20)	0.2412(+)	0.2371(+)	0.2559(+)	0.4199(+)	**0.1386**
Fun8	(10,30)	0.0313(+)	0.0220(=)	0.0423(+)	**0.0164(-)**	0.0217
	(10,20)	0.2577(+)	0.1676(+)	0.0480(+)	**0.0189(-)**	0.0256
	(5,20)	0.3235(+)	0.2372(+)	0.0621(+)	**0.0476(-)**	0.0562
HE2	(10,30)	0.1777(+)	0.1783(+)	0.1786(+)	0.1810(+)	**0.1762**
	(10,20)	0.1915(+)	0.1894(+)	0.1844(+)	0.1845(+)	**0.1828**
	(5,20)	0.2001(+)	0.1995(+)	0.2010(+)	0.1988(+)	**0.1983**
LSF1	(10,30)	0.1021(+)	0.0155(=)	0.0767(+)	0.0366(+)	**0.0154**
	(10,20)	0.1079(+)	0.0253(=)	0.1119(+)	0.0477(+)	**0.0249**
	(5,20)	0.1176(+)	0.0531(+)	0.1196(+)	0.1064(+)	**0.0493**
LSF2	(10,30)	0.0765(+)	0.0387(+)	0.0727(+)	0.1044(+)	**0.0378**
	(10,20)	0.0857(+)	0.0490(+)	0.1006(+)	0.1385(+)	**0.0444**
	(5,20)	0.1846(+)	0.0872(+)	0.1130(+)	0.1399(+)	**0.0730**
LSF3	(10,30)	0.0410(+)	0.0434(+)	0.0465(+)	0.0385(=)	**0.0384**
	(10,20)	0.0460(+)	0.0504(+)	0.0538(+)	0.0463(+)	**0.0440**
	(5,20)	0.0527(+)	0.0525(+)	0.0552(+)	0.0484(+)	**0.0468**
LSF4	(10,30)	0.2037(+)	0.2364(+)	0.1938(+)	0.1928(+)	**0.1916**
	(10,20)	0.2176(+)	0.2367(+)	0.1946(+)	**0.1929(-)**	0.1934
	(5,20)	0.2223(+)	0.2417(+)	0.2078(+)	0.2083(+)	**0.2017**
LSF5	(10,30)	0.2040(+)	0.2257(+)	0.2097(+)	0.1938(=)	**0.1934**
	(10,20)	0.2106(+)	0.2283(+)	0.2156(+)	0.2051(+)	**0.2007**
	(5,20)	0.2504(+)	0.2356(+)	0.2306(+)	0.2109(+)	**0.2064**
LSF6	(10,30)	0.0640(+)	0.0814(+)	0.0658(+)	0.0617(+)	**0.0584**
	(10,20)	0.0782(+)	0.0963(+)	0.0748(+)	0.0770(+)	**0.0578**
	(5,20)	0.1030(+)	0.0966(+)	0.0970(+)	0.0934(+)	**0.0661**
+/=/-		44/4/0	43/5/0	40/3/5	40/4/4	—
Best/all		0/48	0/48	5/48	4/48	39/48

The bold values highlight the statistically optimal mean results of MIGD achieved by different prediction strategies on each test instance over 20 runs under three different parameter combinations: $(n_t,\ \tau_t) \in \{(5, 20), (10, 20), (10, 30)\}$

Table 4 Results of MIGD obtained by ISPP with different numbers of dividing lines k over 20 runs

Numbers of dividing lines k	1	5	10	**15**	20	25	30
dMOP2	0.0065	0.0057	0.0055	**0.0048**	0.0051	0.0056	0.0061
LSF2	0.1261	0.0456	**0.0446**	0.0475	0.0457	0.0405	0.0416

The bold values indicate the statistically best mean results of MIGD obtained by ISPP with different numbers of dividing lines k over 20 runs

Table 5 Results of MIGD obtained by ISPP with different numbers of AR model order p over 20 runs

AR model order p	1	2	**3**	4	5	6
dMOP2	0.0261	0.0070	**0.0055**	0.0063	0.0064	0.0080
LSF2	0.0950	0.0422	**0.0446**	0.0568	0.0563	0.0648

The bold values indicate the statistically best mean results of MIGD obtained by ISPP with different numbers of AR model order p over 20 runs

and the order of the AR model (p). The value of k affects the modeling accuracy of the PF: a larger k allows for a more detailed characterization of the PF morphology but increases sensitivity to noise. The value of p affects the trend-capturing ability across dimensions at the center point: higher-order models are prone to overfitting, whereas lower-order models may overlook nonlinear changes.

In this section, the sensitivity of the ISPP to parameters k and p was studied by applying different values of k and p in the test instances dMOP2 and LSF2.

To compare the effects of different k and p values on ISPP performance visually, Tables 4 and 5 present the MIGD statistical results from 20 independent runs, with a fixed AR model order of $p = 3$ for different k values and a fixed number of dividing lines of $k = 15$ for different p values, where lower values indicate better overall performance.

As shown in Table 4, the number of dividing lines k is set in the range of 10–20, with the optimal $k = 15$. As shown in Table 5, the AR model order is set in the range of 2–3, with the optimal $p = 3$. Under this configuration, the test problems can ensure good convergence while avoiding excessive computational complexity.

6 Conclusions and future work

In this paper, we propose a dividing line–hyperplane intersection and associated subpopulation prediction strategy for evolutionary dynamic multiobjective optimization, referred to as ISPP-DMOEA, to increase the ability of MOEAs to handle DMOPs. ISPP captures and divides the PF by fitting a hyperplane and setting evenly distributed dividing lines, thereby thoroughly capturing the overall characteristics of the PF as it changes over time. Through local association operations of PF intersection points, the PS is divided into multiple subpopulations, accurately reflecting the trend of PS changes over time and quickly predicting the initial population with better distribution and convergence.

To test the performance of ISPP comprehensively, we designed six optimization problems based on existing dynamic multiobjective optimization test functions, featuring discontinuous PFs and PFs that translate or rotate at different frequencies. We compared ISPP with four typical algorithms (PPS, MDP, KTM, and RNN) from the perspectives of the quantity and quality of historical information, as well as the severity and frequency of environmental changes. Furthermore, we assessed the time cost of ISPP-DMOEA and conducted sensitivity analyses on its two key parameters.

The experimental results show that ISPP performs well on most test functions, especially in problems with discontinuous PFs and PSs, where the PS undergoes translations or rotations at different frequencies, demonstrating a clear competitive advantage. Nevertheless, certain limitations remain. Because ISPP adopts an AR model to predict subpopulation centers—a linear prediction method—it excels at forecasting trends but may fail to capture nonlinear features in environments with pronounced nonlinear changes, thus affecting predictive accuracy. For instance, ISPP falls short of RNN on Fun8. In addition, during the process of generating the initial population through submanifold prediction, weak correlations between submanifolds and limited capability for historical feature transfer result in ISPP being outperformed by KTM on the FDA3. The success of KTM on FDA3 suggests that introducing knowledge transfer mechanisms can increase ISPP's adaptability in the early stages of environmental changes. Similarly, the strong performance of RNN on Fun8 implies that integrating the AR(p) model with RNN could further improve ISPP's ability to solve dynamic optimization problems under diverse environments, especially when there are substantial population differences and pronounced nonlinear trends in the population distribution.

Future work will focus on improving and extending ISPP-DMOEA to better address complex dynamic multiobjective optimization problems in real-world scenarios.

Author contributions Zhaoju Sheng (Methodology, Writing-original draft). Erchao Li (Writing review and editing, Conceptualization, Supervision). Li-sen Wei (Figure and Table Preparation). Chen-miao Liu (Data Management).

Funding This work was supported by the National Natural Science Foundation of China (Nos. 62063019), and Gansu Provincial Natural Science Foundation of China (Nos. 24JRRA173). Gansu Province Excellent Doctoral Foundation of China (Nos. 24JRRA205).

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Consent for publication Informed consent was obtained from all individual participants included in the study.

References

1. Raquel C, Yao X (2013) Dynamic multi-objective optimization: A survey of the state-of-the-art. Evolutionary Computation for Dynamic Optimization Problems. Springer, Heidelberg, Germany, pp 85–106
2. Wang B-C, Li H-X, Zhang Q, Wang Y (2021) Decomposition-based multiobjective optimization for constrained evolutionary optimization. IEEE Trans Syst, Man, Cybern, Syst 51(1):574–587
3. Liu M (2016) Robotic online path planning on point cloud. IEEE Trans Cybern 46(5):1217–1228
4. Du W, Zhong W, Tang Y, Du W, Jin Y (2019) High-dimensional robust multi-objective optimization for order scheduling: a decision variable classification approach. IEEE Trans Ind Inform 15(1):293–304
5. Han H, Zhang L, Liu H-X, Yang C, Qiao J (2021) Intelligent optimal control system with flexible objective functions and its applications in wastewater treatment process. IEEE Trans Syst, Man, Cybern, Syst 51(6):3464–3476
6. Fu J, Zou C, Zhang M, Lu X, Li Y (2023) Multiobjective dynamic optimization of nonlinear systems with path constraints. IEEE Transactions on Systems, Man, and Cybernetics: Systems 53(3):1530–1542
7. BC Barroso, FG Ferreira, GP Hanaoka, FD Paiva, and RT Cardoso (2017) Composition of investment portfolios through a combinatorial multiobjective optimization model using CVaR. Proc. IEEE Congr. Evol. Comput. 1795–1802
8. LT Bui and Z Michalewicz (2010) An evolutionary multiobjective approach for dynamic mission planning in Proc. IEEE CEC. 1–8
9. Wu PP-Y, Campbell D, Merz T (2011) Multiobjective four-dimensional vehicle motion planning in large dynamic environments. IEEE Trans Syst, Man, Cybem B, Cybern 41(3):621–634
10. BS Rabil, R Sabourin, and E Granger (2011) Watermarking stack of grayscale face images as a dynamic multiobjective optimization problem. Proc. MDA. 63–77
11. Jin Y, Branke J (2005) Evolutionary optimization in uncertain environments: a survey. IEEE Trans Evol Comput 9(3):303–317. https://doi.org/10.1109/TEVC.2005.846356
12. Nguyen TT, Yang S, Branke J (2012) Evolutionary dynamic optimization: a survey of the state of the art. Swarm Evol Comput 6:1–24
13. Zhou A, Jin Y, Zhang Q (2014) A population prediction strategy for evolutionary dynamic multiobjective optimization. M IEEE Trans Cybem 44(1):40–53
14. Wu Y, Jin Y, Liu X (2015) Directed search strategy for evolutionary dynamic multiobjective optimization. Soft Comput 19(11):3221–3235
15. Li Q, Zou J, Yang S et al (2019) A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. Soft Comput 23:3723–3739. https://doi.org/10.1007/s00500-018-3033-0
16. Muruganantham A, Tan KC, Vadakkepat P (2016) Evolutionary dynamic multiobjective optimization via Kalman filter prediction. IEEE Trans Cybern 46(12):2862–2873. https://doi.org/10.1109/TCYB.2015.2490738
17. Rong M, Gong D, Zhang Y, Jin Y, Pedrycz W (2019) Multidirectional prediction approach for dynamic multiobjective optimization problems. IEEE Trans Cybern 49(9):3362–3374. https://doi.org/10.1109/TCYB.2018.2842158
18. Gee SB, Tan KC, Alippi C (2017) Solving multiobjective optimization problems in unknown dynamic environments: an inverse modeling approach. IEEE Trans Cybern 47(12):4223–4234
19. Chen D, Zou F, Lu R, Wang X (2018) A hybrid fuzzy inference prediction strategy for dynamic multi-objective optimization. Swarm Evol Comput 43:147–165
20. Ma X, Yang J, Sun H, Hu Z, Wei L (2021) Feature information prediction algorithm for dynamic multi-objective optimization problems. Eur J Oper Res 295(3):965–981
21. Cao L, Xu L, Goodman ED, Bao C, Zhu S (2020) Evolutionary dynamic multiobjective optimization assisted by a support vector regression predictor. IEEE Trans Evol Comput 24(2):05–319
22. Zhang H, Ding J, Jiang M, Tan KC, Chai T (2022) Inverse Gaussian process modeling for evolutionary dynamic multiobjective optimization. IEEE Trans Cybern 52(10):11240–11253
23. Xu D, Jiang M, Hu W, Li S, Pan R, Yen GG (2022) An online prediction approach based on incremental support vector machine for dynamic multiobjective optimization. IEEE Trans Evol Comput 26(4):690–703
24. Jiang M, Huang Z, Qiu L, Huang W, Yen GG (2018) Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Trans Evol Comput 22(4):501–514
25. Jiang M, Wang Z, Guo S, Gao X, Tan KC (2021) Individual-based transfer learning for dynamic multiobjective optimization. IEEE Trans Cybern 51(10):4968–4981
26. Jiang M, Wang Z, Hong H, Yen GG (2021) Knee point-based imbalanced transfer learning for dynamic multiobjective optimization. IEEE Trans Evol Comput 25(1):117–129
27. Li J, Sun T, Lin Q, Jiang M, Tan KC (2022) Reducing negative transfer learning via clustering for dynamic multiobjective optimization. IEEE Trans Evol Comput 26(5):1102–1116
28. Jiang M, Wang Z, Qiu L, Guo S, Gao X, Tan KC (2021) A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning. IEEE Trans Cybern 51(7):3417–3428

29. Zhao Q, Yan B, Shi Y, Middendorf M (2022) Evolutionary dynamic multiobjective optimization via learning from historical search process. IEEE Trans Cybern 52(7):6119–6130
30. Feng L, Zhou W, Liu W, Ong Y-S, Tan KC (2022) Solving dynamic multiobjective problem via autoencoding evolutionary search. IEEE Trans Cybern 52(5):2649–2662
31. Chen G, Guo Y, Huang M, Gong D, Yu Z (2022) A domain adaptation learning strategy for dynamic multiobjective optimization. F Sci 606:328–349
32. Yan L et al (2023) Manifold clustering-based prediction for dynamic multiobjective optimization. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2023.101254
33. Ye Y, Lin Q, Ma L, Wong K-C, Gong M, Coello Coello CA (2022) Multiple source transfer learning for dynamic multiobjective optimization. F Sci 607:739–757
34. Lin Q, Ye Y, Ma L, Jiang M, Tan KC (2024) Dynamic multiobjective evolutionary optimization via knowledge transfer and maintenance. IEEE Trans Syst Man Cybern Syst 54(2):936–949
35. Hu Y, Ou J, Suganthan PN, Pedrycz W, Wang R, Zheng J, and Song Y (2024) Dynamic multi-objective optimization algorithm guided by recurrent neural network. IEEE Transactions on Evolutionary Computation. 1–14
36. Zhou W, Feng L, Tan KC et al (2021) Evolutionary Search With Multiview Prediction for Dynamic Multiobjective Optimization. IEEE Trans Evol Comput 26(5):911–925
37. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: solving problems with box constraints. IEEE Trans Evol Comput 18(4):577–601
38. Farina M, Deb K, Amato P (2004) Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans Evol Comput 8(5):425–442. https://doi.org/10.1109/TEVC
39. Goh C-K, Tan KC (2009) A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE TEC 13(1):103–127. https://doi.org/10.1109/TEVC.2008.920671
40. Helbig M, and Engelbrecht AP (2011) Archive management for dynamic multi-objective optimisation problems using vector evaluated particle swarm optimisation. IEEE Congress of Evolutionary Computation (CEC). 2047–2054
41. Das I, Dennis JE (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657
42. I Hatzakis and D Wallace (2006) Topology of anticipatory populations for evolutionary dynamic multiobjective optimization. Proc. 11th AIAA/ISSMO Multidisciplinary Anal. Optimization Conf. 1944–1953
43. Chatfictd C (2004) The Analysis of Time Series: An Introduction. CRC Press, Boca Raton, FL, USA
44. Lee JM (2013) Introduction to Smooth Manifolds, 2nd edn. Springer, New York
45. Hillermeier C (2001) Nonlinear Multiobjective Optimization-A Generalized Homotopy Approach. Birkhauser
46. Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model based multiobjective estimation of distribution algorithm. IEEE Trans Evol Comput 12(1):41–63. https://doi.org/10.1109/TEVC.2007.894202