Balanced Adaptive Subspace Collaboration for Mixed Pareto-Lexicographic Multi-Objective Problems with Priority Levels

Wenjing Hong
National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen, China
Shenzhen City Key Laboratory of Embedded System Design, Shenzhen University, Shenzhen, China
hongwj@szu.edu.cn

Abstract

Multi-objective Optimization Problems (MOPs) where objectives have different levels of importance in decision-making, known as Mixed Pareto-Lexicographic MOPs with Priority Levels (PL-MPL-MOPs), are increasingly prevalent in real-world applications. General-purpose Multi-Objective Evolutionary Algorithms (MOEAs) that treat all objectives equally not only increase the workload of decision-making but also suffer from computational inefficiencies due to the necessity of generating many additional solutions. Conversely, strictly adhering to Priority Levels (PLs) during optimization can easily result in premature convergence within some PLs. To address this issue, we suggest an effective Balanced Adaptive Subspace Collaboration (BASC) method in this paper. Specifically, this method decomposes the search space into sub-fronts based on PLs and utilizes a sampling mechanism that operates exclusively within subspaces formed by sub-fronts at the same PL to generate new solutions, thereby emphasizing the exploitation of individual PLs. Furthermore, a set of parameters is employed to control the strictness of adherence to each PL, with these parameters adaptively adjusted to balance exploration across different PLs. The two mechanisms are then collaboratively integrated into MOEAs. Comprehensive experimental studies on benchmark problems and a set of complex job-shop scheduling problems in semiconductor manufacturing demonstrate the competitiveness of the proposed method over existing methods.

Code — https://github.com/hongwj-lab/pl-mpl-mo

Introduction

A class of complex optimization problems widely encountered in the real world requires the simultaneous consideration of multiple objectives, each with different importance to decision-making. When these levels of importance are organized hierarchically, these problems are known as Mixed Pareto-Lexicographic Multi-objective Optimization Problems with Priority Levels (PL-MPL-MOPs) (Lai et al. 2023). That is, the objectives can be partitioned into multiple Priority Levels (PLs), where a PL has a group of objectives with the same importance (in the Pareto sense) to decision-making, while different PLs exist in lexicographic order. These problems, prevalent in areas such as integrated vehicle control systems (Khosravani et al. 2018), disaster management (Kovács and Moshtari 2019), industrial scheduling (Xu et al. 2016; de C. Bissoli, Zufferey, and Amaral 2021), and business (Zhong, Jiang, and Nielsen 2022), have been a subject of interest since as early as 1970s (Behringer 1977).

Despite the broad range of real-world demands, PL-MPL-MOPs are a particularly challenging class. For many years, there was a stagnation in academic research on this topic, and it has only been revived in recent years (Lai et al. 2023). The challenges arise not only because conflicts among multiple objectives at the same level lead to the absence of a single optimal solution, and there is currently no exact analytical method for solving these problems, but also because the presence of lexicographic priority relations among objectives adds complexity and makes optimization more difficult compared to general MOPs. Thus, Multi-Objective Evolutionary Algorithms (MOEAs) (Zhou et al. 2011) that have achieved widespread success in traditional MOPs fail to deliver satisfactory performance when applied to PL-MPL-MOPs (Lai et al. 2021b).

The use of MOEAs still provide a good starting point because their population-based search mechanism makes them inherently suitable for handling conflicts between multiple objectives. More specifically, since traditional MOEAs do not consider priority relationships, a common approach is to post-process the solution sets found by these methods based on priority relationships to obtain the final solution set (Lai et al. 2021b). However, such methods often suffer from computational inefficiency due to the need to retain many additional solutions that are irrelevant or even disturbing to decision-making during the search process. To address this issue, a recent alternative approach is to directly incorporate priority relationships into the search process of the algorithm. However, the introduction of priorities means that Pareto dominance no longer applies to evaluating the quality of solutions, necessitating algorithms specifically designed for PL-MPL-MOPs.

Although some efforts have been made to incorporate priority relationships into the search process (Fonseca and Fleming 1998; Tan, Lee, and Khor 1999; Tan et al. 2003; Lai et al. 2021a, 2023), most have failed to provide stable solutions due to the use of dominance relations that violate the transitive property for PL-MPL-MOPs. A key contribution in this area is PL-NSGA-II (Lai et al. 2021b). PL-NSGA-II was developed by introducing a dominance relation called PL-dominance that satisfies the necessary transitive property and integrating it into the popular NSGA-II (Deb et al. 2002), widely recognized in both industry and academia. While PL-NSGA-II has shown outstanding performance on PL-MPL-MOPs, it also faces challenges due to the proposed PL-dominance. That is, unlike the Pareto dominance that operates between two solutions, PL-dominance has a global dependency on all other solutions. In other words, it is not always possible to determine whether two solutions are PL-dominated before assigning fitness ranks to the entire population. However, since evolution is dynamic and the population is constantly changing, the PL-dominance relationships calculated at some given intermediate moment may be inappropriate. Consequently, strictly adhering to the PL-dominance based on the current population could disturb the search, causing premature convergence on some PLs.

Motivated by the above observations, this paper suggests an effective Balanced Adaptive Subspace Collaboration (BASC) algorithm. Specifically, this method decomposes the search space into sub-fronts based on PLs and utilizes a sampling mechanism that operates exclusively within subspaces formed by sub-fronts at the same PL to generate new solutions, thereby emphasizing the exploitation of individual PLs. Furthermore, a set of parameters is employed to control the strictness of adherence to each PL, with these parameters adaptively adjusted to balance exploration across different PLs. The two mechanisms are then collaboratively integrated into conventional MOEAs, leading to the algorithm for PL-MPL-MOPs. The main contributions of this paper are summarized as follows:

1) A novel algorithm, namely BASC, targeting PL-MPL-MOPs, a class of problems with broad applications, is proposed. It can provide a set of different and high-quality solutions that meet priority requirements for flexible decision-making.
2) Unlike existing methods that verified algorithms on only a handful of (e.g., four) problems, which might not be sufficient to draw a general conclusion, a total of 63 test problems are examined in this paper. The test problems include both synthetic problems and real-world complex job shop scheduling problems derived from semiconductor manufacturing.

Problem Definition

The general formulation of a PL-MPL-MOP (Lai et al. 2023) is as follows:

$$\begin{aligned}&\min\{f_1(\boldsymbol{x}),\, f_2(\boldsymbol{x}),\, \ldots,\, f_M(\boldsymbol{x})\}\\ &s.t.\, \boldsymbol{x} \in \Omega,\\ &\quad \mathscr{P}(f_1,\, f_2,\, \ldots,\, f_M),\end{aligned}$$

where M is the number of objectives, Ω is the feasible decision space, and $\mathscr{P}(\cdot)$ is a generic priority distribution that arranges the objectives according to the conditions below:

1. Some groups of objectives have clear precedence over others (in the lexicographic sense).
2. Within each group, objectives are of equal importance (in the Pareto sense).

The mathematical formulation is

$$\operatorname{lexmin}\left[\min\begin{pmatrix}f_1^{(1)}\\ f_2^{(1)}\\ \cdots\\ f_{m_1}^{(1)}\end{pmatrix}, \min\begin{pmatrix}f_1^{(2)}\\ f_2^{(2)}\\ \cdots\\ f_{m_2}^{(2)}\end{pmatrix}, \cdots, \min\begin{pmatrix}f_1^{(L)}\\ f_2^{(L)}\\ \cdots\\ f_{m_L}^{(L)}\end{pmatrix}\right],$$

where L denotes the number of PLs, m_i denotes the number of objectives in the ith PL, objectives in the ith PL are infinitely more important than those in the jth PL when $i < j$ holds, and $f_1^{(i)}, \cdots, f_{m_i}^{(i)}$ are objectives of equal importance.

Moreover, the Grossone methodology (Sergeyev 2017) is employed to numerically evaluate the overall performance of the Pareto and lexicographic senses. The fundamental element is the infinite unit Grossone, denoted by ①, which allows one to build numerical values composed by finite, infinite and infinitesimal components. Based on this, PL-MPL-MOPs can be reformulated as

$$\min\left[①^0\begin{pmatrix}f_1^{(1)}\\ f_2^{(1)}\\ \cdots\\ f_{m_1}^{(1)}\end{pmatrix} + ①^{-1}\begin{pmatrix}f_1^{(2)}\\ f_2^{(2)}\\ \cdots\\ f_{m_2}^{(2)}\end{pmatrix} + \cdots + ①^{1-\mathrm{L}}\begin{pmatrix}f_1^{(L)}\\ f_2^{(L)}\\ \cdots\\ f_{m_L}^{(L)}\end{pmatrix}\right],$$

where the level with the highest priority is weighted by $①^0$, the second one by $①^{-1}$, and so on until no more PLs remain, thereby giving infinitely more importance to higher levels.

Balanced Adaptive Subspace Collaboration

In this paper, an effective Balanced Adaptive Subspace Collaboration (BASC) algorithm is proposed for PL-MPL-MOPs. The main idea is to enhance parallel convergence towards multiple PLs by decomposing the search space into subspaces and to encourage the balance across different PLs by introducing a set of adaptively adjustable strictness parameters. The pseudo-code is presented in Algorithm 1 and the details are as follows.

In each iteration, the solution space is segmented into multiple sub-fronts based on PLs, and then a sampling mechanism that operates exclusively within subspaces formed by sub-fronts at the same PL is utlized to generate new solutions, thereby emphasizing the exploitation of individual PLs. Specifically, the population is first divided into a series of sub-fronts, with each sub-front representing a group of solutions that share the same index when the population is sorted based on PL-dominance. Under the Grossone methodology, the index of a solution $\boldsymbol{x}$ can be expressed as

$$\psi = r_1①^0 + r_2①^{-1} + \ldots r_L①^{-(L-1)},$$

where $r_i \in \mathbb{Z}^+$ represents the Pareto ranking at the ith PL. For brevity, we will henceforth adopt the simplified notation

$$\psi = [r_1, r_2, \ldots, r_L]$$

where no ambiguity arises. The collection of all solutions with index ψ forms sub-front F_ψ.

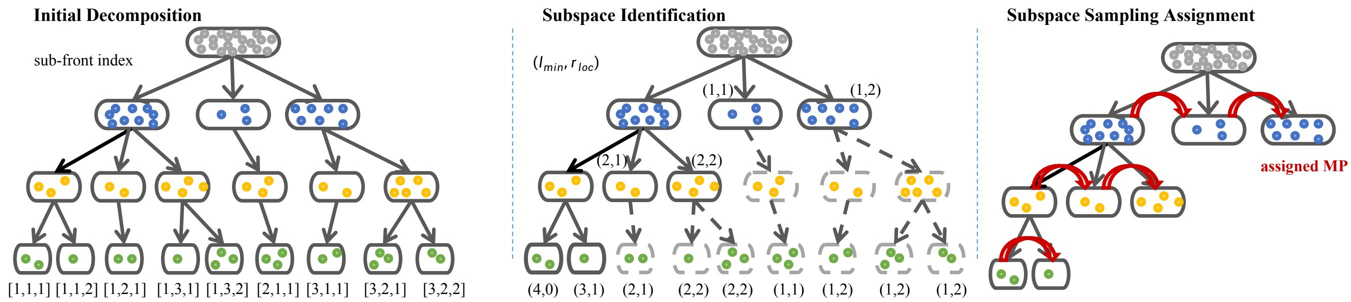

Figure 1: An example of the subspace sampling mechanism

Building on these sub-fronts, certain subspaces are dynamically identified during the evolutionary process to balance convergence and diversity. A subspace consists of a set of sub-fronts that satisfy a given constraint condition to represent a set of potential solutions. Concretely, given the lowest priority l currently under consideration, a subspace assigned with identifier l is formed by the union of sub-fronts whose indices belong to the set

$$\begin{aligned} &\{[1,\,\ldots,\,1,\,c_l,\,c_{l+1},\,\ldots,\,c_L]\} \\ &\text{s.t. } c_l > 1,\, c_{l+1} \geq 1,\,\ldots,\,c_L \geq 1. \end{aligned} \tag{1}$$

Based on this, each solution can be individually assigned a promising subspace as follows. First, given solution $\boldsymbol{x}$ belonging to sub-front F_{ψ_x} where

$$\psi_x = [\lambda_1,\,\lambda_2,\,\ldots,\,\lambda_L],$$

let l_{min} be the identifier of subspace to which $\boldsymbol{x}$ belongs to, i.e.,

$$l_{min} = \begin{cases} \min_{i=1}^{L}\{i|\lambda_i > 1\} & \sum_{i=1}^{L}\lambda_i > L \\ L+1 & otherwise. \end{cases} \tag{2}$$

Then, the neighboring solutions with better quality than $\boldsymbol{x}$ within this subspace are determined based on the Pareto ranking. That is, the set consists of solutions with a Pareto ranking of r_{loc} in the subspace with identifier l_{min}, where r_{loc} is set to $\lambda_{l_{min}} - 1$. In this way, the promising subspace of this solution, denoted as $MP(\boldsymbol{x})$, can be expressed as the set of solutions that belongs to any sub-front with index

$$\begin{aligned} &[\lambda_1,\,\lambda_2,\,\ldots,\,\lambda_{l_{min}-1},\,r_{loc},\,c_{l_{min}+1},\ldots\,c_L], \\ &\text{s.t. } c_{l_{min}+1} \geq 1,\,\ldots,\,c_L \geq 1. \end{aligned} \tag{3}$$

New solutions are sampled based on $MP(\boldsymbol{x})$ as follows. Given population P, two solutions $\boldsymbol{x}$ and $\boldsymbol{x}'$ are selected and involved in genetic operators together to generate new solutions. The first solution $\boldsymbol{x}$ is selected randomly from P and the second solution $\boldsymbol{x}'$ is selected randomly from $MP(\boldsymbol{x})$ to guide the local search, or randomly from P to encourage global exploration (Bian et al. 2023), accroding to the sub-front that $\boldsymbol{x}$ belongs to. Taking advantage of these diverse mating pools, the new sampling mechanism is expected to enhance the search by tracking multiple, distinct regions simultaneously and avoiding premature convergence. An example of the PL-subspace sampling mechanism is illustrated in Fig. 1 and the detailed steps are presented in lines 6–18 in Algorithm 1.

Furthermore, a set of parameters is employed to control the strictness of adherence to each PL. This set of parameters, indicated by matrix $\boldsymbol{\Theta}$, is adaptively adjusted to balance convergence and diversity across different PLs. Specifically, given strictness $\boldsymbol{\Theta}$ that

$$\begin{aligned} &\boldsymbol{\Theta} = (\boldsymbol{\theta}^{(1)}, \boldsymbol{\theta}^{(2)}, \cdots, \boldsymbol{\theta}^{(L)}), \\ &\boldsymbol{\theta}^{(l)} = \{\theta_l^{(1)}, \theta_l^{(2)}, \cdots, \theta_l^{(m_l)}\},\ \theta_l^{(\cdot)} \in \mathbb{R}, \end{aligned}$$

a new dominance relationship with adjustable strictness is defined as follows. Solution $\boldsymbol{u}$ is better than solution $\boldsymbol{v}$ in the lth PL if and only if

$$\begin{aligned} &\forall i \in \{1,2,\cdots,m_l\}, \quad f_i^{(l)}(\boldsymbol{v}) \geq f_i^{(l)}(\boldsymbol{u}) + \theta_i^{(l)}, \\ &\exists j \in \{1,2,\cdots,m_l\}, \quad f_j^{(l)}(\boldsymbol{v}) > f_j^{(l)}(\boldsymbol{u}) + \theta_j^{(l)}. \end{aligned} \tag{4}$$

The larger $\theta^{(l)}$ is, the more strict the lth PL is, and in case that $\boldsymbol{\Theta} = \boldsymbol{0}$, this dominance is actually the standard PL-dominance. Then, these parameters are adaptively adjusted by being initialized as zero to emphasize convergence, and being updated progressively to preserve diversity by

$$\theta_i^{(l)} = d_i^{(l)}(e^{-FE/FE_{max}} - e^{-1}), \tag{5}$$

where $d_i^{(l)}$ is the minimum objective distances between solutions in lth PL at its ith objective, FE and FE_{max} are the current and maximum fitness evaluations. This update proceeds once the following condition is satisfied: given population P' preserved periodically in previous generations, if most of the solutions of P' are comparable to or better than that of the current population P in the lth PL. In this way, the participation of solutions that are weakly PL-dominated in higher PL but have clear advantages in the lower PL can be increased to enhance exploration among different PLs. The detailed steps are presented in lines 20–32 in Algorithm 1 .

The PL-subspace sampling mechanism and the balanced adaptive PL-selection mechanisms are then collaboratively integrated into the evolutionary framework.

Experimental Setup

A total of 63 problems from three test suites are considered in the experiments. The first contains three benchmarking problems, PL-A, PL-B (Lai et al. 2021b), and

Algorithm 1: The Proposed BASC

Input: PL number L, population size N, maximum fitness evaluations FE_{max}, check κ
Output: A

Initialize P with N random solutions and set $A = P' = P$;
Set strictness $\mathbf{\Theta} = \mathbf{0}$, check level $l_c = 1$;
Set $e = N$;
while $e < FE_{max}$ **do**
  /* *PL-subspace sampling mechanism* */
  Set $Q = \emptyset$;
  for $i = 1 : N/2$ **do**
    Select $\boldsymbol{x}$ from P randomly;
    Calculate l_{min} of $\boldsymbol{x}$ base on (2);
    Calculate $MP(\boldsymbol{x})$ based on (3);
    if $l_{min} \leq L$ **then**
      Select $\boldsymbol{x}'$ from $MP(\boldsymbol{x})$ randomly;
    else
      Select $\boldsymbol{x}'$ from P randomly;
    end
    Generate new solutions q_1, q_2 using genetic operations on $\boldsymbol{x}$, $\boldsymbol{x}'$, and evaluate them;
    Set $Q = Q \cup \{q_1, q_2\}$;
  end
  /* *balanced adaptive PL-selection mechanism* */
  if $mod(e, N^2) = 0$ **then**
    Rank $P \cup P'$ with Pareto nondominated sort on objectives in the l_cth PL;
    Let r_1 and r_2 be the Pareto ranking of the κNth solution in P and P', respectively;
    if $r_1 \geq r_2$ **then**
      Update $\mathbf{\Theta}$ by (5);
      Set $l_c = l_c + 1$;
    end
    Set $P' = P$;
  end
  Rank $R = P \cup Q$ using the dominance with $\mathbf{\Theta}$ shown in (4) and the crowding distance;
  Set P as the first N solutions of R with smaller index or larger crowding distance;
  Update A by Q with the PL-dominance and the crowding distance;
  Set $e = e + N$;
end
return A

PL-C (Lai et al. 2023), specifically designed for studying PL-MPL-MOPs. The second extends a popular benchmark suite named MaF (Cheng et al. 2017), which contains 15 test problems and has a number of continuous variables ranging from 2 to 80. Two configurations of the MaF suite are examined, one with four objectives and two PLs in terms of $\min\{[(f_1, f_2) + \textcircled{1}^{-1}(f_3, f_4)]\}$ and the other with nine objectives and three PLs in terms of $\min\{[(f_1, f_2, f_3) + \textcircled{1}^{-1}(f_4, f_5, f_6) + \textcircled{1}^{-2}(f_7, f_8, f_9)]\}$, resulting in 30 test problems. The third is a set of complex job-shop scheduling problems derived from semiconductor manufacturing (Knopp, Dauzère-Pérès, and Yugma 2017; Türkyilmaz et al. 2020). This set of problems, dubbed CJSPs, contains 30 problems with four objectives and two PLs in terms of $\min\{[(f_1, f_2) + \textcircled{1}^{-1}(f_3, f_4)]\}$, where f_1 indicates the total weighted tardiness for CJSP1-CJSP15 and the total weighted completion time for CJSP16-CJSP30, f_2 indicates the total workload of machines, f_3 indicates the makespan, and f_4 indicates the weighted number of tardy jobs. It challenges algorithms to tune a large number, ranging from 225 to 2,505, of discrete variables. The problems from different classes are used to investigate the performance of the algorithms more comprehensively on different PL numbers, problem sizes, and variable types.

To evaluate the effectiveness of the proposed BASC, five state-of-the-art algorithms are involved in the comparison, namely 1) NSGA-III-post (Lai et al. 2021b), extended NSGA-III (Deb and Jain 2014) with posterior filtering; 2) MOEA/DD-post, extended MOEA/DD (Li et al. 2015) with posterior filtering; 3) NSGA-II-pre (Lai et al. 2021b), extended NSGA-II (Deb et al. 2002) with prefiltering; 4) MOEA/D-pre (Lai et al. 2021b), extended MOEA/D (Zhang and Li 2007) with prefiltering; 5) PL-NSGA-II (Lai et al. 2023), extended NSGA-II that incorporates priority. The algorithms are built on the PlatEMO platform (Tian et al. 2017) and configured as follows. The population size is set to 100, the maximum number of fitness evaluations is set to 100,000, and genetic operators are set below. In the case of PL-A, PL-B, PL-C, and MaF problems, the well-known simulated binary crossover and polynomial mutation operator (Deb 2000) are employed. In the case of CJSP problems, the typical generalized position crossover (Mattfeld 1996), precedence operation crossover (Lee, Yamakawa, and Lee 1998), and uniform crossover (Pan, Lei, and Wang 2022), are applied. The other parameters are set as their default values. For BASC, check κ is set to 0.9 based on a grid search. Each algorithm is performed independently for 21 runs.

The metric $\Delta(\cdot) = \max\{IGD(\cdot), GD(\cdot)\}$, which provides a comprehensive evaluation of balance across multiple objectives (Lai et al. 2023), is employed to evaluate the algorithm performance in the experiments. The computation of $\Delta(\cdot)$ comes after the normalization of the objective space to the unitary hyper-cube and the PL-nondominated solutions from the solution set obtained by an MOEA is used to measure the performance of the MOEA. In the case where the true optimal solutions of a PL-MPL-MOP are known in advance, a uniformly sampled set of them is used for the normalization. In the other unknown case, the optimal solution set of the ensemble of solutions obtained by all algorithms in all independent runs is employed. To have a statistically sound conclusion, the Wilcoxon signed-rank test (Wilcoxon 1992) at a 0.05 significance level is used to validate the statistical significance.

Results and Discussions

Tables 1-5 report the results of the proposed BASC and the five compared algorithms on metric $\Delta(\cdot)$, where the best results among the six algorithms are shown in bold, and the statistical test results of BASC against other algorithms are characterized by $+$, $\approx$, and $-$ to characterize that BASC achieves significantly better, not significantly different, and significantly worse results than the algorithms in the corresponding rows or columns, respectively. At the end of each

	PL-A	PL-B	PL-C	*win-tie-loss*
BASC	**[0.029, 0.000]**	**[0.001, 0.002, 0.008]**	[0.004, 0.015, 0.005]	
PL-NSGA-II	**[0.029, 0.000]** ≈	**[0.001, 0.002, 0.008]** ≈	**[0.004, 0.014, 0.005]** ≈	0-3-0
NSGA-III-post	[0.201, 0.005] +	[0.505, 0.024, 0.349] +	[0.081, 0.285, 0.096] +	3-0-0
MOEA/DD-post	[0.077, 0.001] +	[0.023, 0.015, 0.372] +	[0.147, 0.697, 0.200] +	3-0-0
NSGA-II-pre	[0.080, 0.001] +	[0.011, 0.021, 0.209] +	[0.109, 0.142, 0.131] +	3-0-0
MOEA/D-pre	[0.056, 0.001] +	[0.025, 0.032, 0.344] +	[0.858, 0.899, 0.941] +	3-0-0

Table 1: Performance in terms of $\Delta(\cdot)$ on PL-A, PL-B, and PL-C.

	BASC	PL-NSGA-II	NSGA-III-post	MOEA/DD-post	NSGA-II-pre	MOEA/D-pre
MaF1	[0.198, 0.007]	**[0.167, 0.009]** −	[0.275, 0.075] +	[0.406, 0.023] +	[0.334, 0.001] +	[0.202, 0.000] +
MaF2	**[0.009, 0.000]**	[0.010, 0.000] +	[0.062, 0.128] +	[0.057, 0.090] +	[0.012, 0.094] +	[0.012, 0.072] +
MaF3	**[0.000, 0.674]**	**[0.000, 0.674]** ≈	[0.012, 0.213] +	[0.176, 0.435] +	**[0.000, 0.674]** ≈	**[0.000, 0.674]** ≈
MaF4	[0.035, 0.001]	[0.038, 0.001] +	[0.041, 0.002] +	[0.046, 0.006] +	[0.038, 0.000] +	**[0.008, 0.000]** −
MaF5	[0.000, 0.609]	[0.000, 0.609] +	[0.000, 0.115] +	[0.099, 0.309] +	**[0.000, 0.580]** −	[0.000, 0.638] +
MaF6	[0.000, 0.000]	[0.000, 0.000] ≈	[0.000, 0.000] ≈	[0.000, 0.000] ≈	[0.000, 0.000] ≈	[0.000, 0.000] ≈
MaF7	**[0.000, 0.234]**	[0.000, 0.250] +	[0.018, 0.373] +	[0.000, 0.474] +	[0.000, 0.568] +	[0.000, 0.662] +
MaF8	[0.103, 0.131]	[0.116, 0.159] +	[0.173, 0.201] +	[0.124, 0.137] +	**[0.039, 0.030]** −	[0.089, 0.075] −
MaF9	[0.000, 0.000]	[0.000, 0.000] −	[0.000, 0.000] +	[0.000, 0.000] +	**[0.000, 0.000]** −	[0.000, 0.000] +
MaF10	[0.000, 0.125]	[0.003, 0.202] +	[0.000, 0.066] −	[0.267, 0.728] +	[0.003, 0.520] +	[0.110, 0.672] +
MaF11	[0.001, 1.320]	[0.001, 1.406] +	[0.300, 0.724] +	[0.344, 0.885] +	**[0.001, 1.266]** −	[0.002, 0.995] +
MaF12	**[0.068, 0.026]**	[0.085, 0.032] +	[0.224, 0.430] +	[0.561, 0.732] +	**[0.068, 0.026]** ≈	[0.092, 0.038] +
MaF13	**[0.000, 0.000]**	[0.003, 0.000] ≈	[0.153, 0.029] +	[0.186, 0.026] +	[0.000, 0.107] +	[0.000, 0.016] +
MaF14	[0.000, 0.025]	**[0.000, 0.022]** −	[0.000, 0.037] +	[0.002, 0.005] +	[0.000, 0.336] +	[0.000, 0.498] +
MaF15	[0.010, 0.680]	[0.009, 0.681] −	[0.021, 0.684] +	[0.016, 0.682] +	**[0.008, 0.214]** −	[0.013, 0.478] +
win-tie-loss		8-3-4	13-1-1	14-1-0	7-3-5	11-2-2

Table 2: Performance in terms of $\Delta(\cdot)$ on MaF problems with four objectives and two PLs.

	BASC	PL-NSGA-II	NSGA-III-post	MOEA/DD-post
MaF1	[0.361, 0.109, 0.006]	**[0.347, 0.123, 0.005]** −	[0.456, 0.282, 0.222] +	[0.637, 0.220, 0.634] +
MaF2	**[0.001, 0.000, 0.017]**	[0.001, 0.000, 0.036] +	[0.036, 0.158, 0.471] +	[0.059, 0.084, 0.300] +
MaF3	**[0.000, 0.000, 0.000]**	**[0.000, 0.000, 0.000]** ≈	[0.000, 0.000, 0.001] +	[0.000, 0.000, 0.000] ≈
MaF4	[0.272, 0.219, 0.041]	**[0.256, 0.237, 0.045]** −	[0.485, 0.515, 0.551] +	[0.877, 0.518, 0.963] +
MaF5	[0.000, 0.000, 0.804]	[0.000, 0.000, 0.820] +	**[0.000, 0.000, 0.135]** −	[0.000, 0.000, 0.256] −
MaF6	[0.000, 0.000, 0.996]	[0.000, 0.000, 0.996] ≈	**[0.000, 0.000, 0.000]** −	**[0.000, 0.000, 0.000]** −
MaF7	[0.000, 0.000, 0.426]	**[0.000, 0.000, 0.382]** −	[0.023, 0.023, 0.377] +	[1.085, 0.978, 0.625] +
MaF8	[0.255, 0.157, 0.189]	[0.290, 0.195, 0.211] +	[0.225, 0.206, 0.200] −	[0.255, 0.283, 0.247] +
MaF9	**[0.000, 0.000, 0.000]**	[0.000, 0.000, 0.000] +	[0.000, 0.000, 0.000] +	[0.000, 0.000, 0.000] +
MaF10	[0.005, 0.001, 0.661]	[0.006, 0.002, 0.697] +	**[0.000, 0.000, 0.126]** −	[0.000, 0.056, 0.598] +
MaF11	**[0.001, 0.000, 0.274]**	[0.001, 0.000, 0.323] +	[0.044, 0.470, 0.784] +	[0.159, 0.109, 0.754] +
MaF12	**[0.003, 0.001, 0.001]**	[0.013, 0.004, 0.003] +	[0.061, 0.226, 0.758] +	[0.042, 0.054, 0.968] +
MaF13	**[0.004, 0.000, 0.000]**	**[0.004, 0.000, 0.000]** ≈	[0.043, 0.000, 0.000] +	[0.022, 0.000, 0.000] +
MaF14	**[0.000, 0.000, 0.005]**	[0.000, 0.000, 0.007] +	[0.000, 0.079, 0.055] +	[0.000, 0.000, 0.039] +
MaF15	[0.005, 0.002, 0.192]	**[0.004, 0.002, 0.169]** −	[0.106, 0.071, 0.144] +	[0.008, 0.003, 0.166] +
win-tie-loss		8-3-4	11-0-4	12-1-2

Table 3: Performance in terms of $\Delta(\cdot)$ on MaF problems with nine objectives and three PLs.

table, a summary of the statistical test results on this set of test problems is given in the form of win-tie-loss, which indicates the number of problems for which BASC is significantly better, not significantly different, and significantly worse, respectively. Overall, the proposed algorithm demonstrated competitive performance against all compared algorithms on at least 52 of the 63 test problems, with this competitive performance being achieved in at least 25 out of the 33 benchmark problems and at least 25 out of the 30 examined real-world problems.

The first concern is the effectiveness of incorporating priority in the search process. Recall that NSGA-III-post and

	BASC	NSGA-II-pre	MOEA/D-pre
MaF1	**[0.361, 0.109, 0.006]**	[0.505, 0.088, 0.012] +	[0.509, 0.093, 0.008] +
MaF2	**[0.001, 0.000, 0.017]**	[0.001, 0.001, 0.057] +	[0.001, 0.001, 0.055] +
MaF3	**[0.000, 0.000, 0.000]**	[0.000, 0.000, 0.075] +	[0.000, 0.000, 0.078] +
MaF4	**[0.272, 0.219, 0.041]**	[0.344, 0.411, 0.036] +	[0.333, 0.397, 0.050] +
MaF5	**[0.000, 0.000, 0.804]**	[0.000, 0.000, 0.820] +	[0.000, 0.000, 0.853] +
MaF6	**[0.000, 0.000, 0.996]**	[0.000, 0.000, 0.981] +	[0.000, 0.000, 0.398] −
MaF7	**[0.000, 0.000, 0.426]**	[0.000, 0.694, 0.713] +	[0.000, 1.005, 0.869] +
MaF8	[0.255, 0.157, 0.189]	**[0.120, 0.069, 0.107]** −	[0.122, 0.098, 0.083] −
MaF9	**[0.000, 0.000, 0.000]**	[0.000, 0.000, 0.000] +	[0.000, 0.000, 0.000] +
MaF10	**[0.005, 0.001, 0.661]**	[0.000, 0.153, 0.881] +	[0.201, 0.499, 0.820] +
MaF11	**[0.001, 0.000, 0.274]**	[0.001, 0.000, 0.299] +	[0.003, 0.055, 0.331] +
MaF12	**[0.003, 0.001, 0.001]**	[0.005, 0.002, 0.001] +	[0.039, 0.013, 0.008] +
MaF13	**[0.004, 0.000, 0.000]**	[0.012, 0.000, 0.000] +	[0.019, 0.000, 0.000] +
MaF14	**[0.000, 0.000, 0.005]**	[0.000, 0.000, 0.351] +	[0.000, 0.000, 0.577] +
MaF15	**[0.005, 0.002, 0.192]**	[0.020, 0.479, 0.323] +	[0.007, 0.351, 0.464] +
win-tie-loss		14-0-1	13-0-2

Table 4: Performance in terms of $\Delta(\cdot)$ on MaF problems with nine objectives and three PLs.

	BASC	PL-NSGA-II	NSGA-III-post	MOEA/DD-post	NSGA-II-pre	MOEA/D-pre
JSP1	**[0.128, 0.440]**	[0.207, 0.476] +	[0.219, 0.266] +	[0.335, 0.530] +	[0.305, 0.608] +	[0.967, 0.858] +
JSP2	**[0.128, 0.164]**	[0.165, 0.256] +	[0.162, 0.124] +	[0.330, 0.362] +	[0.203, 0.258] +	[0.944, 0.728] +
JSP3	**[0.128, 0.163]**	[0.219, 0.152] +	[0.193, 0.132] +	[0.296, 0.249] +	[0.239, 0.160] +	[0.924, 0.623] +
JSP4	**[0.113, 0.223]**	[0.184, 0.257] +	[0.233, 0.455] +	[0.413, 0.330] +	[0.238, 0.291] +	[1.078, 0.462] +
JSP5	**[0.113, 0.180]**	[0.306, 0.188] +	[0.201, 0.454] +	[0.438, 0.243] +	[0.299, 0.267] +	[1.137, 0.669] +
JSP6	[0.236, 0.496]	[0.211, 0.457] −	**[0.109, 0.154]** −	[0.401, 0.428] +	[0.311, 0.492] +	[1.094, 0.802] +
JSP7	**[0.090, 0.279]**	[0.189, 0.315] +	[0.224, 0.278] +	[0.510, 0.307] +	[0.271, 0.288] +	[1.115, 0.652] +
JSP8	**[0.160, 0.493]**	[0.207, 0.553] +	[0.227, 0.095] +	[0.525, 0.480] +	[0.231, 0.630] +	[1.083, 0.975] +
JSP9	**[0.139, 0.354]**	[0.158, 0.341] +	[0.176, 0.287] +	[0.361, 0.364] +	[0.287, 0.458] +	[1.012, 0.844] +
JSP10	[0.521, 0.208]	[0.438, 0.367] −	[0.351, 0.249] −	**[0.068, 0.167]** −	[0.455, 0.389] −	[0.884, 0.765] +
JSP11	**[0.114, 0.494]**	[0.190, 0.466] +	[0.191, 0.103] +	[0.467, 0.518] +	[0.277, 0.643] +	[1.053, 0.935] +
JSP12	**[0.125, 0.252]**	[0.245, 0.321] +	[0.217, 0.347] +	[0.327, 0.383] +	[0.341, 0.386] +	[1.111, 0.884] +
JSP13	**[0.081, 0.065]**	[0.292, 0.159] +	[0.309, 0.473] +	[0.688, 0.191] +	[0.320, 0.185] +	[1.179, 0.741] +
JSP14	**[0.138, 0.145]**	[0.250, 0.226] +	[0.255, 0.421] +	[0.461, 0.216] +	[0.372, 0.318] +	[1.099, 0.721] +
JSP15	**[0.134, 0.108]**	[0.282, 0.147] +	[0.221, 0.421] +	[0.351, 0.126] +	[0.365, 0.225] +	[1.064, 0.705] +
JSP16	[0.190, 0.630]	[0.128, 0.623] −	**[0.114, 0.189]** −	[0.403, 0.382] +	[0.156, 0.449] −	[0.831, 0.440] +
JSP17	**[0.140, 0.017]**	[0.260, 0.062] +	[0.273, 0.000] +	[0.281, 0.104] +	[0.341, 0.059] +	[0.987, 0.535] +
JSP18	**[0.102, 0.166]**	[0.205, 0.139] +	[0.195, 0.064] +	[0.318, 0.154] +	[0.215, 0.088] +	[1.004, 0.523] +
JSP19	**[0.069, 0.049]**	[0.159, 0.022] +	[0.267, 0.010] +	[0.309, 0.060] +	[0.247, 0.070] +	[1.021, 0.372] +
JSP20	**[0.155, 0.071]**	[0.323, 0.120] +	[0.276, 0.073] +	[0.339, 0.095] +	[0.336, 0.174] +	[1.156, 0.503] +
JSP21	**[0.058, 0.220]**	[0.180, 0.235] +	[0.134, 0.009] +	[0.281, 0.201] +	[0.235, 0.222] +	[1.231, 0.637] +
JSP22	**[0.121, 0.151]**	[0.191, 0.103] +	[0.154, 0.019] +	[0.313, 0.177] +	[0.267, 0.217] +	[1.113, 0.573] +
JSP23	**[0.053, 0.043]**	[0.191, 0.082] +	[0.160, 0.149] +	[0.424, 0.057] +	[0.213, 0.074] +	[1.128, 0.203] +
JSP24	**[0.114, 0.060]**	[0.212, 0.036] +	[0.190, 0.012] +	[0.292, 0.096] +	[0.271, 0.084] +	[1.065, 0.215] +
JSP25	**[0.096, 0.166]**	[0.234, 0.132] +	[0.249, 0.085] +	[0.409, 0.214] +	[0.283, 0.178] +	[1.116, 0.649] +
JSP26	[0.327, 0.148]	[0.380, 0.169] +	[0.422, 0.197] +	**[0.121, 0.078]** −	[0.381, 0.133] +	[0.979, 0.247] +
JSP27	[0.362, 0.086]	[0.373, 0.102] +	[0.411, 0.661] +	**[0.139, 0.039]** −	[0.383, 0.075] +	[0.986, 0.095] +
JSP28	[0.225, 0.104]	[0.286, 0.088] +	[0.249, 0.075] +	**[0.145, 0.096]** −	[0.292, 0.151] +	[1.032, 0.524] +
JSP29	[0.196, 0.187]	[0.254, 0.090] +	[0.223, 0.094] +	**[0.193, 0.076]** −	[0.362, 0.231] +	[1.032, 0.433] +
JSP30	**[0.106, 0.143]**	[0.206, 0.169] +	[0.189, 0.051] +	[0.354, 0.056] +	[0.261, 0.192] +	[1.054, 0.539] +
win-tie-loss		27-0-3	27-0-3	25-0-5	28-0-2	30-0-0

Table 5: Performance in terms of $\Delta(\cdot)$ on CJSPs.

MOEA/DD-post are set to completely ignore PLs to optimize all objectives equally, and NSGA-II-pre and MOEA/D-pre are set to ignore all PLs except the first one to focus on the most important objectives, the proposed BASC and

	BASC	BASC-sampling	BASC-selection
JSP1	**[0.128, 0.440]**	**[0.128, 0.440]** $\approx$	[0.215, 0.501] $+$
JSP2	**[0.128, 0.164]**	**[0.128, 0.164]** $\approx$	[0.168, 0.121] $+$
JSP3	**[0.128, 0.163]**	[0.132, 0.156] $+$	[0.227, 0.133] $+$
JSP4	**[0.113, 0.223]**	[0.115, 0.226] $\approx$	[0.190, 0.231] $+$
JSP5	**[0.113, 0.180]**	[0.123, 0.101] $+$	[0.310, 0.199] $+$
JSP6	[0.236, 0.496]	[0.222, 0.432] $-$	**[0.188, 0.210]** $-$
JSP7	**[0.090, 0.279]**	[0.095, 0.278] $\approx$	[0.201, 0.302] $+$
JSP8	[0.160, 0.493]	**[0.160, 0.489]** $\approx$	[0.221, 0.227] $+$
JSP9	**[0.139, 0.354]**	[0.140, 0.300] $+$	[0.162, 0.293] $+$
JSP10	[0.521, 0.208]	**[0.520, 0.211]** $\approx$	**[0.372, 0.199]** $-$
w-t-l		3-6-1	8-0-2

Table 6: Performance in terms of $\Delta(\cdot)$ for BASC variants.

PL-NSGA-II are compared with them to address this concern. From Table 1, when applied to PL-A, PL-B, and PL-C, both BASC and PL-NSGA-II manage to outperform the four posterior filtering-based MOEAs. Due to the complete ignorance of PLs, NSGA-III-post and MOEA/DD-post cannot provide stable search pressure towards the regions of interest, failing to achieve competitive results at any PL. While NSGA-II-pre and MOEA/D-pre could get some good results at the first PL, taking PL-B as an example, they fail to achieve satisfactory performance at subsequent PLs due to the neglect of subsequent PLs. From Table 2, when applied to MaF problems with four objectives and two PLs, BASC and PL-NSGA-II outperforms the posterior filtering-based MOEAs in general. The benefits of incorporation of priority preference are further verified in Tables 3–4 as the numbers of objectives and PLs increase. When applied to the complex CJSP problems, as shown in Table 5, both BASC and PL-NSGA-II exhibit superior performance, and in particular, BASC performs significantly better than the four posterior filtering-based MOEAs on most of the test problems. Such results imply the effectiveness of algorithms that incorporate priority preference to focus the search on preferred regions, being more suited for solving PL-MPL-MOPs.

The second concern is the efficacy of the proposed BASC compared with PL-NSGA-II, both of which incorporate priority. From Tables 1–3 and Table 5, it can be observed that BASC shows a clear superiority over PL-NSGA-II by achieving significantly better results on 43 out of the 63 test problems and performing comparably on nine of them. More specifically, when applied to PL-A, PL-B, and PL-C, the results in Table 1 show that BASC is able to obtain comparable results to PL-NSGA-II, which are significantly better than other comparison algorithms. When applied to the MaF problems with four objectives and two PLs, as shown in Table 2, BASC continues to exhibit its superiority over PL-NSGA-II by outperforming PL-NSGA-II significantly on eight problems and only being worse on four problems. On the eight problems, BASC is shown to be able to provide better performance at either PL. For example, while BASC and PL-NSGA-II perform comparably on the first PL on MaF7, BASC performs significantly better on the second PL, resulting in a better performance. When applied to the MaF problems with larger numbers of objectives and PLs, BASC consistently outperforms PL-NSGA-II. As shown in Table 3, BASC achieves significantly better results on eight problems and comparable results on three problems out of the 15 MaF problems. Apart from the superior performance on the examined benchmarks, the competitiveness of BASC compared with PL-NSGA-II is further reported on CJSP problems derived from real-world scenarios. According to Table 5, BASC manages to obtain significantly better results on 27 out of the 30 CJSP problems, and on 17 of them BASC performs significantly better on all the PLs, implying its strength in improving performance across all PLs. In summary, the above results clearly demonstrate the competitiveness of BASC against the state-of-the-art PL-NSGA-II in dealing with PL-MPL-MOPs.

An ablative experiment is also conducted to examine the components of BASC, as shown in Table 6. One variant is used to examine the effectiveness of the PL-subspace sampling mechanism, namely BASC-sampling, and the other is to examine that of the balanced adaptive PL-selection mechanism, namely BASC-selection. The experimental results on CJSP1-CJSP10 problems show that BASC presents competitive performance on 9 of them with the use of PL-subspace sampling and performs competitively on 8 of them with the use of selection.

Conclusion

This paper presents a novel algorithm, namely BASC, targeting PL-MPL-MOPs, a class of problems with broad applications that have multiple objectives with different levels of importance in decision-making. This algorithm suggests to enhance parallel convergence towards multiple PLs by decomposing the search space into subspaces and to encourage the balance across different PLs by introducing a set of adaptively adjustable strictness parameters. Comprehensive experimental studies have conducted on a total of 63 test problems that include both synthetic problems and complex job shop scheduling problems derived from semiconductor manufacturing. The results demonstrated the competitiveness of the proposed algorithm over existing methods in providing a set of different and high-quality solutions that meet priority requirements for flexible decision-making.

In the future, several directions may be further pursued. First, it is worthy of developing more PL-MPL-MOPs for benchmarking. The performance under various problem properties, such as geometry (Huband et al. 2006) of PL optimal solutions, scalability in terms of PLs, parameter dependencies (Li et al. 2016), and differernt application scenarios (Yang et al. 2024; Hong et al. 2024), deserves in-depth analysis. Second, considering advantages of other popular MOEA frameworks such as decomposition-based (Trivedi et al. 2017) and indicator-based (Falcón-Cardona and Coello 2021) fameworks, we plan to study PL-MPL-MOPs with these frameworks. Third, attempts on extending theoretical analysis of MOEAs (Qian, Liu, and Zhou 2022; Qian et al. 2019; Bian, Qian, and Tang 2018; Qian, Yu, and Zhou 2013) to PL-MPL-MOPs would be beneficial.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China (Grant No. 62106098).

References

Behringer, F. A. 1977. Lexicographic quasiconcave multiobjective programming. *Zeitschrift für Operations Research*, 21(3): 103–116.

Bian, C.; Qian, C.; and Tang, K. 2018. A general approach to running time analysis of multi-objective evolutionary algorithms. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, Stockholm, Sweden*, 1405–1411.

Bian, C.; Zhou, Y.; Li, M.; and Qian, C. 2023. Stochastic population update can provably be helpful in multi-objective evolutionary algorithms. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence, Macao, SAR, China*, 5513–5521.

Cheng, R.; Li, M.; Tian, Y.; Zhang, X.; Yang, S.; Jin, Y.; and Yao, X. 2017. A benchmark test suite for evolutionary many-objective optimization. *Complex Intelligent Systems*, 3(1): 67–81.

de C. Bissoli, D.; Zufferey, N.; and Amaral, A. R. S. 2021. Lexicographic optimization-based clustering search metaheuristic for the multiobjective flexible job shop scheduling problem. *International Transactions in Operational Research*, 28(5): 2733–2758.

Deb, K. 2000. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, (2/4): 186.

Deb, K.; Agrawal, S.; Pratap, A.; and Meyarivan, T. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197.

Deb, K.; and Jain, H. 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4): 577–601.

Falcón-Cardona, J. G.; and Coello, C. A. C. 2021. Indicator-based multi-objective evolutionary algorithms: A comprehensive survey. *ACM Computing Surveys*, 53(2): 29:1–29:35.

Fonseca, C. M.; and Fleming, P. J. 1998. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 28(1): 26–37.

Hong, W.; Li, G.; Liu, S.; Yang, P.; and Tang, K. 2024. Multi-objective evolutionary optimization for hardware-aware neural network pruning. *Fundamental Research*, 4(4): 941–950.

Huband, S.; Hingston, P.; Barone, L.; and While, L. 2006. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5): 477–506.

Khosravani, S.; Jalali, M.; Khajepour, A.; Kasaiezadeh, A.; Chen, S.; and Litkouhi, B. 2018. Application of lexicographic optimization method to integrated vehicle control systems. *IEEE Transactions on Industrial Electronics*, 65(12): 9677–9686.

Knopp, S.; Dauzère-Pérès, S.; and Yugma, C. 2017. A batch-oblivious approach for complex job-shop scheduling problems. *European Journal of Operational Research*, 263(1): 50–61.

Kovács, G.; and Moshtari, M. 2019. A roadmap for higher research quality in humanitarian operations: A methodological perspective. *European Journal of Operational Research*, 276(2): 395–408.

Lai, L.; Fiaschi, L.; Cococcioni, M.; and Deb, K. 2021a. Handling priority levels in mixed pareto-lexicographic many-objective optimization problems. In *Proceedings of the 11th International Conference on Evolutionary Multi-Criterion Optimization, Shenzhen, China*, volume 12654, 362–374.

Lai, L.; Fiaschi, L.; Cococcioni, M.; and Deb, K. 2021b. Solving mixed Pareto-lexicographic multiobjective optimization problems: The case of priority levels. *IEEE Transactions on Evolutionary Computation*, 25(5): 971–985.

Lai, L.; Fiaschi, L.; Cococcioni, M.; and Deb, K. 2023. Pure and mixed lexicographic-Paretian many-objective optimization: State of the art. *Natural Computing*, 22(2): 227–242.

Lee, K. M.; Yamakawa, T.; and Lee, K. 1998. A genetic algorithm for general machine scheduling problems. In *Proceedings of the 2nd International Conference Knowledge-Based Intelligent Electronic Systems, Adelaide, South Australia*, 60–66.

Li, K.; Deb, K.; Zhang, Q.; and Kwong, S. 2015. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5): 694–716.

Li, K.; Omidvar, M. N.; Deb, K.; and Yao, X. 2016. Variable interaction in multi-objective optimization problems. In *Proceedings of the 14th International Conference on Parallel Problem Solving from Nature - XIV, Edinburgh, UK*, volume 9921, 399–409.

Mattfeld, D. C. 1996. *Evolutionary search and the job shop - Investigations on genetic algorithms for production scheduling*. Production and Logistics. Physica-Verlag.

Pan, Z.; Lei, D.; and Wang, L. 2022. A bi-population evolutionary algorithm with feedback for energy-efficient fuzzy flexible job shop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(8): 5295–5307.

Qian, C.; Liu, D.; and Zhou, Z. 2022. Result diversification by multi-objective evolutionary algorithms with theoretical guarantees. *Artificial Intelligence*, 309: 103737.

Qian, C.; Yu, Y.; Tang, K.; Yao, X.; and Zhou, Z. 2019. Maximizing submodular or monotone approximately submodular functions by multi-objective evolutionary algorithms. *Artificial Intelligence*, 275: 279–294.

Qian, C.; Yu, Y.; and Zhou, Z. 2013. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence*, 204: 99–119.

Sergeyev, Y. D. 2017. Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two Hilbert problems. *EMS Surveys in Mathematical Sciences*, 4(2): 219–320.

Tan, K. C.; Khor, E. F.; Lee, T. H.; and Sathikannan, R. 2003. An evolutionary algorithm with advanced goal and priority specification for multi-objective optimization. *Journal of Artificial Intelligence Research*, 18: 183–215.

Tan, K. C.; Lee, T. H.; and Khor, E. F. 1999. Evolutionary algorithms with goal and priority information for multi-objective optimization. In *Proceedings of the 1999 Congress on Evolutionary Computation, Washington, DC*, 106–116.

Tian, Y.; Cheng, R.; Zhang, X.; and Jin, Y. 2017. PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4): 73–87.

Trivedi, A.; Srinivasan, D.; Sanyal, K.; and Ghosh, A. 2017. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3): 440–462.

Türkyilmaz, A.; Senvar, Ö.; Ünal, I.; and Bulkan, S. 2020. A research survey: Heuristic approaches for solving multi-objective flexible job shop problems. *Journal of Intelligent Manufacturing*, 31(8): 1949–1983.

Wilcoxon, F. 1992. Individual comparisons by ranking methods. 196–202.

Xu, R.; Chen, H.; Liang, X.; and Wang, H. 2016. Priority-based constructive algorithms for scheduling agile earth observation satellites with total priority maximization. *Expert Systems with Applications*, 51: 195–206.

Yang, P.; Zhang, L.; Liu, H.; and Li, G. 2024. Reducing idleness in financial cloud services via multi-objective evolutionary reinforcement learning based load balancer. *Science China Information Sciences*, 67(2).

Zhang, Q.; and Li, H. 2007. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6): 712–731.

Zhong, S.; Jiang, Y.; and Nielsen, O. A. 2022. Lexicographic multi-objective road pricing optimization considering land use and transportation effects. *European Journal of Operational Research*, 298(2): 496–509.

Zhou, A.; Qu, B.; Li, H.; Zhao, S.; Suganthan, P. N.; and Zhang, Q. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1): 32–49.