symmetry

MDPI

Article

An Improved NSGA-III with a Comprehensive Adaptive Penalty Scheme for Many-Objective Optimization

Xinghang Xu [1], Du Cheng [2], Dan Wang [1], Qingliang Li [3] and Fanhua Yu [3,]*

[1] College of Computer Science and Technology, Beihua University, Jilin 132013, China; qixiangxxh123@gmail.com (X.X.); wangdanjl_jl@163.com (D.W.)
[2] School of Artificial Intelligence, Jilin University, Changchun 130012, China; chengdu23@mails.jlu.edu.cn
[3] College of Computer Science and Technology, Changchun Normal University, Changchun 130032, China; liqingliang@ccsfu.edu.cn
* Correspondence: yufanhua@beihua.edu.cn

Abstract: Pareto dominance-based algorithms face a significant challenge in handling many-objective optimization problems. As the number of objectives increases, the sharp rise in non-dominated individuals makes it challenging for the algorithm to differentiate their quality, resulting in a loss of selection pressure. The application of the penalty-based boundary intersection (PBI) method can balance convergence and diversity in algorithms. The PBI method guides the evolution of individuals by integrating the parallel and perpendicular distances between individuals and reference vectors, where the penalty factor is crucial for balancing these two distances and significantly affects algorithm performance. Therefore, a comprehensive adaptive penalty scheme was proposed and applied to NSGA-III, named caps-NSGA-III, to achieve balance and symmetry between convergence and diversity. Initially, each reference vector's penalty factor is computed based on its own characteristic. Then, during the iteration process, the penalty factor is adaptively adjusted according to the evolutionary state of the individuals associated with the corresponding reference vector. Finally, a monitoring strategy is designed to oversee the penalty factor, ensuring that adaptive adjustments align with the algorithm's needs at different stages. Through a comparison involving benchmark experiments and two real-world problems, the competitiveness of caps-NSGA-III was demonstrated.

Keywords: many-objective optimization; penalty-based boundary intersection; comprehensive adaptive penalty scheme; NSGA-III

check for updates

Citation: Xu, X.; Cheng, D.; Wang, D.; Li, Q.; Yu, F. An Improved NSGA-III with a Comprehensive Adaptive Penalty Scheme for Many-Objective Optimization. *Symmetry* **2024**, *16*, 1289. https://doi.org/10.3390/sym16101289

Academic Editor: Shangce Gao, Yirui Wang and Yang Yu

Received: 16 August 2024
Revised: 19 September 2024
Accepted: 29 September 2024
Published: 1 October 2024

1. Introduction

Many real-world optimization problems often involve multiple conflicting objectives to be optimized simultaneously, which are referred to as multi-objective optimization problems (MOPs). Multi-objective evolutionary algorithms (MOEAs) are powerful tools for addressing MOPs as they can extensively explore the decision space and evaluate and select individuals to achieve the optimization of multiple objectives. MOEAs can be broadly categorized into three categories: Pareto dominance-based approaches [1–3], decomposition-based approaches [4–6], and indicator-based approaches [7–9]. Among these, Pareto dominance-based algorithms, such as NSGA-II and SPEA2, have been widely applied to various engineering and real-world problems, such as path decision [10], flight optimization [11], and distributed generation system planning [12]. However, these algorithms encounter considerable difficulties when addressing many-objective optimization problems (MaOPs) involving more than three objectives. The primary challenge arises from the sharp increase in the proportion of non-dominated individuals within the population as the dimensionality of objectives increases. This results in the Pareto dominance relationship becoming ineffective, making the algorithm's convergence not guaranteed.

To address the aforementioned issues, extensive research has been conducted, which can be primarily categorized into three approaches. The first approach is improvements to the Pareto dominance relationship, such as ϵ-dominance [13], hpaEA [14], and PeEA [15]. Specifically, ϵ-dominance is a novel dominance relation that divides the objective space into multiple hyperboxes, each containing at most one solution, thereby maintaining a balance between convergence and diversity. The hpaEA identifies non-dominated solutions with significant Pareto front tendencies as prominent and then refines these non-dominated solutions using hyperplanes formed by them and their neighbors, thereby relaxing the Pareto dominance relationship. PeEA is a method for estimating the shape of the Pareto front (PF) by guiding the search process with a curvature-based approach, which handles the issue of selection pressure loss in problems with varying PF shapes. The second approach is to define new diversity criteria. Deb and Jain [3] proposed NSGA-III, which uses a reference point-based method to increase selection pressure and maintain solution diversity. The third approach is the adoption of collaborative strategies, such as θ-NSGA-III [16] and SSCEA [17]. θ-NSGA-III builds on NSGA-III by incorporating the θ-dominance relationship to balance the algorithm's convergence and diversity. SSCEA is a coevolutionary method that combines indicator-based and Pareto dominance-based approaches.

Notably, the θ-dominance relationship in θ-NSGA-III makes use of the PBI method [4]. The PBI method effectively balances algorithm convergence and diversity, with its performance primarily dependent on the penalty factor θ. A smaller θ promotes convergence, while a larger θ promotes diversity. However, setting a fixed penalty factor based on empirical knowledge does not guarantee the performance of PBI when dealing with Pareto fronts (PFs) of different shapes. An excessively large penalty factor may result in a uniformly distributed PF, which may not represent the true PF. Conversely, an excessively small penalty factor could lead to the loss of boundary individuals. Therefore, adaptive adjustment of the penalty factor is necessary. Yang et al. [18] proposed two penalty schemes: the adaptive penalty scheme (APS) and the subproblem-based penalty scheme (SPS). In APS, all subproblems have the same θ, which progressively increases as the iterations proceed, gradually shifting the focus from convergence to diversity. In SPS, penalty values are calculated based on the subproblems. To prevent the loss of boundary individuals during iterations, larger penalty values are assigned to boundary subproblems to emphasize diversity, while smaller penalty values are set for intermediate subproblems to emphasize convergence. Additionally, based on population and weight vector distribution information, Han et al. [19] proposed a dynamic penalty scheme. Specifically, when a subproblem is farther from the associated individuals and neighboring subproblems, the penalty factor is increased to enhance diversity. Conversely, the penalty factor is decreased to enhance convergence.

The adaptive penalty schemes enhance the algorithm's performance in handling MaOPs. Different penalty schemes involve the algorithm's needs at different stages, characteristics of subproblems, and the state of population evolution. However, existing works do not comprehensively consider these factors, which could be more beneficial for designing penalty schemes. In summary, we proposed a comprehensive adaptive penalty scheme and applied it to NSGA-III (caps-NSGA-III) to balance the convergence and diversity of the algorithm. The main contributions of this paper are as follows:

1. An adaptive penalty scheme is proposed. Each penalty factor is initially calculated based on its reference vector and is then adaptively adjusted according to the evolutionary state of the individuals associated with that reference vector during the iteration process.
2. A monitoring strategy is proposed, in which the adaptive penalty scheme is monitored and adjusted to meet the algorithm's needs at different stages. For example, if diversity adjustment is performed during the algorithm's early stage (convergence phase), this is considered a violation. Once the violation handling criterion is met, a convergence operation is performed.

3. Through comparisons with five state-of-the-art many-objective evolutionary algorithms on benchmark function experiments and two real-world applications, the competitiveness of caps-NSGA-III is demonstrated.

The rest of the paper is structured as follows: Basic definitions and improvement motivations are introduced in Section 2. The proposed caps-NSGA-III algorithm is presented in Section 3. The experimental setup and results are described in Section 4. Finally, Section 5 concludes this paper and discusses some future work.

2. Related Work

2.1. Basic Definitions

A multi-objective optimization problem (MOP) is typically considered as a minimization problem and can be mathematically defined as follows [20]:

$$\begin{aligned} \min F(x) &= (f_1(x), f_2(x), \cdots, f_m(x))^T \\ \text{subject to} \quad & x \in \Omega \end{aligned} \tag{1}$$

where $x = (x_1, x_2, \cdots, x_n)^T$ is an n-dimensional decision variable vector from the decision space Ω and m is the number of objectives. $F : \Omega \to R^m$ is the vector of m objective function values, and R^m represents the objective space. When $m > 3$, the problem is termed a many-objective optimization problem (MaOP).

Pareto dominance

For two solutions $x_1, x_2 \in \Omega$, x_1 is said to Pareto dominate x_2 ($x_1 \prec x_2$), if $f_i(x_1) \leq f_i(x_2)$, for every $i \in \{1, 2, \cdots, m\}$, and $f_j(x_1) < f_j(x_2)$, for at least one index $j \in \{1, 2, \cdots, m\}$.

Pareto optimal

For a decision vector $x^* \in \Omega$, if there does not exist another vector $x \in \Omega$ such that $x \prec x^*$, then x^* is Pareto optimal.

Pareto set

The Pareto set (PS) is defined as

$$PS = \{x \in \Omega \mid \ x \text{ is Pareto optimal}\}$$

Pareto front

The Pareto front (PF) is defined as

$$PF = \{f(x) \in R^m \mid x \in PS\}$$

2.2. NSGA-III

The Non-dominated Sorting Genetic Algorithm III (NSGA-III) is one of the most outstanding many-objective evolutionary algorithms available today. It has proven effective in addressing various engineering optimization problems, leading to significant outcomes [21–23]. However, as a Pareto dominance-based algorithm, it has a significant limitation. With increasing dimensionality of objectives, the Pareto non-dominance relation gradually fails, making the algorithm's convergence not guaranteed. Additionally, in the selection process of NSGA-III, only those individuals closest to the reference vectors are considered. While this approach can achieve good diversity, the convergence is unsatisfactory. Therefore, there is a need to enhance the consideration of convergence.

2.3. PBI and SPS

The penalty-based boundary intersection (PBI) method can balance convergence and diversity, and its computation is as follows:

$$g^{pbi}(x \mid \omega, Z^*) = d_1 + \theta d_2 \tag{2}$$

where x is the decision vector, ω is the reference vector, Z^* is the ideal point in the objective space, θ is penalty factor, d_1 is the projection of $F(x)$ onto ω, and d_2 is the perpendicular distance from $F(x)$ to ω. The calculations for d_1 and d_2 are as follows:

$$d_1 = \frac{\| (F(x) - Z^*)^T w \|}{\| w \|} \tag{3}$$

$$d_2 = \| F(x) - Z^* - d_1 \frac{w}{\| w \|} \| \tag{4}$$

where $F(x)$ is the objective vector for the individual x. d_1 and d_2 are illustrated in Figure 1.

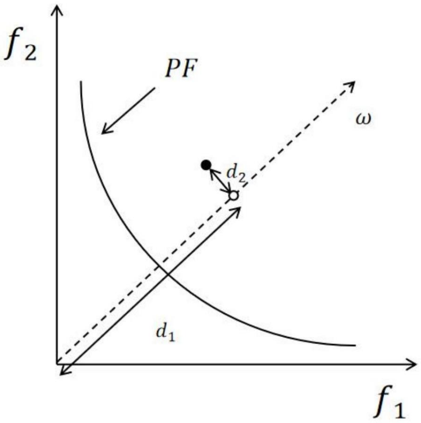


Figure 1. Illustration of distances d_1 and d_2.

The penalty factor in PBI significantly impacts its performance. The subproblem-based penalty scheme (SPS) is a method for calculating penalty factors as follows:

$$\begin{gathered} \theta_i = e^{\alpha \beta_i} \\ \beta_i = \max_{1 \le j \le m} \omega_i^j - \min_{1 \le j \le m} \omega_i^j \end{gathered} \tag{5}$$

where i is the ith reference vector, α is a control parameter for the magnitude of the penalty, and β_i is the difference between the maximum and minimum component of i.

Within SPS, boundary weight vectors and intermediate weight vectors have different penalty factors θ. For boundary weight vectors, especially near the coordinate axes, β_i approaches 1 and θ_i takes a higher value, which emphasizes diversity. For the intermediate weight vectors, where all components are nearly equal, β_i approaches 0 and θ_i approaches 1, emphasizing convergence. It is worth noting that after initialization, θ values remain constant during the iteration, which could potentially lead to the abandonment of valuable solutions.

In Figure 2, $\omega = (0.1, 0.9)^T$ is a boundary weight vector, and segment $A - B$ is part of the true PF. As seen in Figure 2, for ω and the associated individuals a and b, it is evident that individual b is closer to the expected PF segment than individual a. However, due to g^{pbi} of b being greater than that of a, individual b is not selected. This indicates that an excessive focus on the weight vectors themselves, while neglecting the algorithm's need for individuals with strong convergence in the early stage, and a blind pursuit of diversity, may impair the algorithm's convergence performance.

In Figure 3, $\omega = (0.5, 0.5)^T$ is an intermediate weight vector. Individuals a and b are two points associated with ω, located near the segment $A - B$ of the true PF. As seen in Figure 3, for ω and the associated individuals a and b, it is evident that individual b is closer to ω. However, because the g^{pbi} of a is less than that of b, the more diverse individual b is abandoned. This indicates that an excessive focus on the weight vectors themselves, while neglecting the algorithm's need for individuals with strong diversity in the late stage, and a blind pursuit of convergence, could result in a decrease in solution diversity.

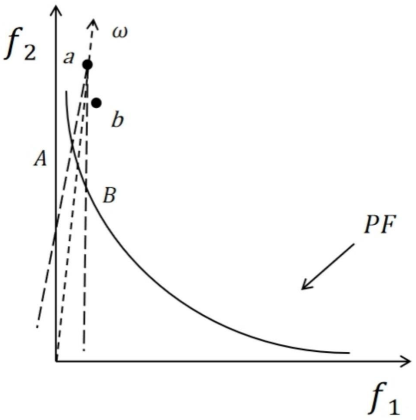


Figure 2. Illustration of the limitation of the fixed θ value in SPS using boundary weight vector.

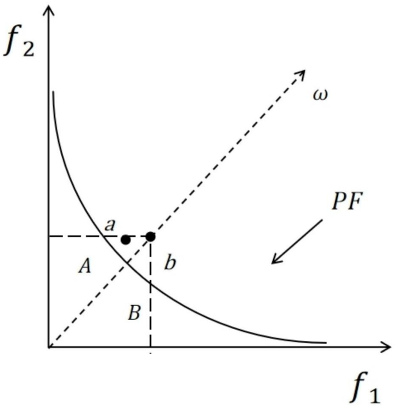


Figure 3. Illustration of the limitation of the fixed θ value in SPS using intermediate reference vector.

2.4. Motivation

Pareto-based methods have limitations in addressing many-objective optimization problems (MaOPs). Researchers have extensively explored improvements by improving the Pareto dominance relationship, defining new diversity criteria, and adopting collaborative approaches. In fact, collaborative approaches essentially combine the Pareto dominance relationship with additional convergence metrics. Solutions are initially ranked based on Pareto dominance and further selected according to convergence metrics. For example, the knee point proposed by Zhang et al. [24] and the grid dominance measure proposed by Yang et al. [25] are both additional convergence metrics. Additionally, the PBI method used in θ-NSGA-III is also a convergence-related metric. It calculates the parallel distance (measuring convergence) and the perpendicular distance (measuring diversity) between solutions and reference vectors, adjusting the importance of both through a penalty factor. Notably, the penalty factor significantly affects the algorithm's performance, and a fixed value based on experience may not be suitable for all types of problems. To address this issue, Yang et al. [18] proposed two adaptive penalty schemes: APS and SPS. APS adjusts the penalty factor based on the algorithm's needs at different stages. In the early stage, it focuses on convergence, and as the iterations progress, the penalty factor gradually increases to emphasize diversity. SPS sets the penalty factors based on the characteristics of each subproblem, with central subproblems focusing on convergence and boundary subproblems emphasizing diversity. Additionally, Han et al. [19] proposed an adaptive penalty scheme that adjusts the penalty factor based on the evolutionary state of the population. However, existing methods do not comprehensively consider these aspects. Therefore, we propose a comprehensive penalty scheme. Based on the limitations of the SPS method described in Section 2.3 and the adaptive adjustment strategy for the penalty factor based on the population's evolutionary state, we propose an adaptive penalty scheme to improve the performance of the algorithm in handling MaOPs, as detailed in Section 3.2. Additionally, there is a key threshold in the adaptive penalty scheme that determines the algorithm's convergence and diversity behavior. This threshold should not be fixed but should be adjusted throughout the entire evolutionary process. Therefore, we propose a monitoring strategy to adjust the threshold, allowing the adaptive penalty scheme to meet the algorithm's varying requirements for convergence and diversity at different stages, thus improving its ability to handle MaOPs, as detailed in Section 3.3.

3. Proposed Algorithm

The main framework of caps-NSGA-III is shown in Algorithm 1. First, the reference vectors are generated and the initial penalty factors are calculated. Next, the population is initialized using chaotic mapping, after which the iteration process begins. First, the offspring population Q_t is generated using NSGA-III's genetic operators and then combined with P_t to form the population R_t. After, non-dominated sorting is utilized to divide R_t into various Pareto-based non-domination levels, with the final layer denoted as layer l. Next, starting from F_1, each Pareto layer is added to the set S_t until the number of individuals in S_t is greater than or equal to N. If the number equals N, the next iteration begins. Otherwise, individuals from the first l layers are stored, and a PBI distance-based NSGA-III selection procedure (replacing the perpendicular distance with the PBI distance (PBI value)) is executed in layer l to select the remaining required individuals. After generating the population P_{t+1}, each penalty factor is adaptively adjusted, and the monitoring strategy is then executed.

Algorithm 1 General framework of caps-NSGA-III

```
Input: $N$ (Population size), $M$ (Number of objectives), $V$ (Number of decision variables),
    $ub$ (Upper bounds of decision variables), $lb$ (Lower bounds of decision variables), $MFEs$
    (Maximum number of fitness evaluations), $\varepsilon$ (Threshold of convergence metric), $VF$
    (Violation factor)
Output: population $P$
$Z \leftarrow$ Generate Reference Vectors()
$\theta_0 \leftarrow$ Calculate the initial penalty factor //SPS
$P_0 \leftarrow$ Chaotic mapping population initialization($N, V, ub, lb$) //Algorithm 2
while termination condition is not met do
    $Q_t \leftarrow$ Genetic Operator($P_t$)
    $R_t = P_t \cup Q_t$
    $(F_1, F_2, \cdots)$ = Non-dominated-sort($R_t$)
    repeat
    $S_t = S_t \cup F_i$ and $i = i + 1$
    until $|S_t| \geqslant N$
    Last front to be included: $F_l = F_i$
    if $|S_t| = N$ then
        $P_{t+1} = S_t$, break
    else
        $P_{t+1} = \cup_{j=1}^{l-1} F_j$
        $P_{t+1} \leftarrow$ The selection process of NSGA-III based on PBI distance.
    end if
    $\theta_{t+1} \leftarrow$ Adaptive penalty factor($P_t, P_{t+1}, \theta_t, t, MFEs, N, Z, \varepsilon$) //Algorithm 3
    $[\varepsilon, \theta_{t+1}, VF] \leftarrow$ Monitoring strategy($P_t, P_{t+1}, \theta_t, \theta_{t+1}, VF, \varepsilon$) //Algorithm 4
end while
```

3.1. Chaotic Mapping Population Initialization

With increasing objective dimensions, the initial population may exhibit duplication or clustering, leading to reduced diversity. Therefore, we introduced a widely used chaotic mapping method [26] to initialize the population [27,28]. The Logistic equation is a typical chaotic mapping system [29] and is calculated as follows:

$$xl = \mu x(1 - x) \tag{6}$$

where x is a random number in [0, 1], and μ is a logistic control parameter, a random floating-point number in the range [0, 4].

The specific process for chaotic mapping initialization of the population is detailed in Algorithm 2.

Algorithm 2 Chaotic mapping population initialization (N, V, ub, lb)

```
Input: $N, V, ub, lb$
Output: $P_0$
$P_0 = \emptyset$
for $i = 1 : | N |$ do
    for $j = 1 : | V |$ do
        $x_j = \mathrm{rand}(0, 1)$
        $xl_j = \mu x_j(1 - x_j)$
        $x_{i,j} = lb_j + (ub_j - lb_j)xl_j$
    end for
    $P_0 = P_0 \cup x_i$
end for
```

3.2. Adaptive Penalty Factor

In the SPS method, after initializing the penalty factor (θ) it remains fixed, which may lead to the abandonment of some excellent individuals and affect the algorithm's performance. Therefore, we proposed an adaptive method based on SPS. Initially, each reference vector's penalty factor is computed based on its own characteristic. Then, during the iteration process, the penalty factor is adaptively adjusted according to the evolutionary state of the individuals associated with the corresponding reference vector. Specifically, for reference vector i, the PBI of the centroid of the individuals associated with i is calculated in P_t and P_{t+1}, respectively. The d_1 of PBI is used as a convergence indicator: a smaller d_1 means the next-generation centroid is closer to the true PF, suggesting that individuals associated with i are converging. When d_1 changes relatively significantly, the θ value is reduced to emphasize convergence. Conversely, the θ value is increased to emphasize diversity. The detailed process is outlined in Algorithm 3.

Algorithm 3 Adaptive penalty factor $(P_t, P_{t+1}, \theta_t, t, MFEs, N, Z, \varepsilon)$

```
Input: $P_t$ , $P_{t+1}$ , $\theta_t$ , $t$ , $MFEs$ , $N$, $Z$ , $\varepsilon$
Output: $\theta_{t+1}$
$CP = P_t \cup P_{t+1}$
Perform associated operation on CP.
for $i = 1 : | Z |$ do
    $AS_t \leftarrow$ Find the set of individuals in $P_t$ associated with $Z(i)$
    $AS_{t+1} \leftarrow$ Find the set of individuals in $P_{t+1}$ associated with $Z(i)$
    Calculate the centroids $c_t$ and $c_{t+1}$ of $AS_t$ and $AS_{t+1}$, respectively.
    Calculate $d_1$, $d_2$, and $PBI$ for $c_t$ and $c_{t+1}$, respectively.
    if $d_{1,c_t} - d_{1,c_{t+1}} / d_{1,c_t} > \varepsilon$ then
        $\theta_{t+1} = \theta_t - \frac{t*N}{MFEs} \times \frac{|PBI_{c_{t+1}} - PBI_{c_t}|}{PBI_{c_t}}$
    else
        $\theta_{t+1} = \theta_t + \frac{t*N}{MFEs} \times \frac{|PBI_{c_{t+1}} - PBI_{c_t}|}{PBI_{c_t}}$
    end if
end for
```

3.3. Monitoring Strategy

Algorithms have varying requirements at different stages. The adaptive adjustment of the penalty factor should align with the algorithm's needs, with controlling the threshold for increasing or decreasing θ being crucial, as it directly impacts the algorithm's convergence and diversity behavior. To achieve symmetry between convergence and diversity, we proposed a monitoring strategy. Specifically, if a diversity adjustment is performed during the convergence phase, it is considered a violation. When the predefined violation factor is exceeded, convergence adjustment is then performed. Similarly, during the diversity phase, if the number of convergence adjustments exceeds the violation factor, a diversity adjustment is then performed. The specific process is detailed in Algorithm 4.

Algorithm 4 Monitoring strategy($P_t, P_{t+1}, \theta_t, \theta_{t+1}, VF, \varepsilon$)

```
Input: $P_t$ , $P_{t+1}$ , $\theta_t$ , $\theta_{t+1}$ , $VF$ , $\varepsilon$
Output: $\varepsilon$, $\theta_{t+1}$, $VF$
Phase = 0 //Phase = 0 represents the early stage of the algorithm.
$[d_{1,t}, d_{1,t+1}]$ = Calculate the sum of $d_1$ for individuals' PBI in $P_t$ and $P_{t+1}$.
if $t > T/2$ then
    if $d_{1,t}$ - $d_{1,t+1}/d_{1,t} < 10^{-3}$ then
        Phase = 1 // Phase = 1 represents the late stage of the algorithm.
    end if
end if
if Phase = 0 then
    if $\theta_{t+1} - \theta_t > 0$ then
        $VF = VF - 1$ //Violation
    end if
    if $VF = 0$ then
        $\varepsilon = \varepsilon - \mid cr_d \mid$ * rand(0, 1) //$cr_d$ is the rate of change of $d_1$ in the centroid's PBI at the first violation.
        $\theta_{t+1} = \theta_{t+1} - \mid cr_{pbi} \mid$ * rand(0, 1) //$cr_{pbi}$ is the sum of the rate of change of the PBI of the centroid for the three violations.
    end if
else
    if $\theta_{t+1} - \theta_t < 0$ then
        $VF = VF - 1$ //Violation
    end if
    if $VF = 0$ then
        $\varepsilon = \varepsilon + \mid cr_d \mid$ * rand(0, 1)
        $\theta_{t+1} = \theta_{t+1} + \mid cr_{pbi} \mid$ * rand(0, 1)
    end if
end if
```

3.4. Complexity Analysis

The time complexity calculation for caps-NSGA-III primarily includes chaotic mapping population initialization (Algorithm 2), non-dominated sorting (line 7), environmental selection (line 16), adaptive penalty factor (Algorithm 3), and monitoring strategy (Algorithm 4). Assuming the population size is N, the number of objectives is M, the number of decision variables is V, and the number of reference vectors is K. Consequently, the time complexity of chaotic mapping population initialization is $O(NV)$ and the time complexity of non-dominated sorting is $O(MN^2)$. The main components of environmental selection include normalization, PBI calculation, and niche selection. The time complexity for normalization is $O(MN^2)$, for PBI calculation is $O(MNK)$, and for niche selection is $O(N^2)$ (in the worst case, selecting N individuals). Since $K \leq N$, the time complexity of environmental selection is $O(MN^2)$. Additionally, the time complexity of the adaptive penalty factor (Algorithm 3) is $O(NK)$ and that of the monitoring strategy (Algorithm 4) is O(K). In summary, the overall time complexity of caps-NSGA-III is $O(MN^2)$.

4. Experimental Studies

4.1. Benchmark Function and Algorithm Parameter Settings

4.1.1. Benchmark Function Settings

The experiments utilize the widely used DTLZ [30] and WFG [31] test suites. For DTLZ, we focus only on the DTLZ1 to DTLZ4 problems, similar to NSGA-III. The decision variable V is defined as $V = M + r - 1$, where the objective dimension M ranges from 3 to 15 and the parameter r is 10 (or 5 for DTLZ1). For WFG, we consider all problems and define the decision variable V as $V = k + l$, where the position-related variable $k = 2(M - 1)$ and the distance-related variable $l = 20$.

4.1.2. Algorithm Parameter Settings

The experiment compared five many-objective evolutionary algorithms: SSCEA, PeEA, hpaEA, θ-NSGA-III, and NSGA-III. These five algorithms were all designed to address the limitations of the traditional Pareto dominance method in handling many-objective optimization problems, covering the three approaches mentioned in Section 1. Among them, SSCEA and θ-NSGA-III, like caps-NSGA-III, belong to the third approach; NSGA-III belongs to the second approach; and PeEA and hpaEA belong to the first approach. Our aim is to validate the effectiveness of caps-NSGA-III by comparing it with similar methods as well as those from the other two approaches. The algorithms' parameter settings involved in this study are as follows:

1. Population size settings: The population size is determined by the parameter H and the objective dimension M, with specific settings detailed in Table 1. We use the method of Das and Dennis [32] to generate reference vectors. When M exceeds 3, the method of Deb and Jain [3] is employed.
2. Runs and termination criteria: The number of runs is 20 for each instance, with the termination criteria for the algorithms defined as the maximum number of fitness evaluations, as detailed in Table 2.
3. Crossover and mutation operator settings: The crossover probability is 1, with the distribution index set to 30 (20 for SSCEA, PeEA, and hpaEA). The mutation probability is $1/V$, with a distribution index of 20.
4. Parameter Settings: All algorithms use the parameter settings from the original studies. In caps-NSGA-III, $\mu = 4$ for the Logistic equation and $\alpha = 4$ for SPS. The initial threshold is defined as $cd * rand$, where cd is the initial rate of change of d_1 of the centroid associated with the reference vector, and the violation factor is set to 3.

Table 1. Settings of population size.

M	H	Population Size
3	12	91
5	6	210
8	$H_1 = 3, H_2 = 2$	156
10	$H_1 = 3, H_2 = 2$	275
15	$H_1 = 3, H_2 = 1$	135

H_1 and H_2 represent the number of divisions for the boundary layer and the inner layer, respectively.

Table 2. Settings of termination condition.

Test Instance	M = 3	M = 5	M = 8	M = 10	M = 15
DTLZ1-DTLZ4	18,200	42,000	31,200	55,000	27,000
WFG1-WFG9	36,400	157,500	234,000	550,000	405,000

4.2. Performance Metrics

4.2.1. Inverted Generational Distance (IGD)

Let P represent the set of points on the final obtained PF and P^* represent a set of points uniformly spread over the true PF. The IGD [33] is then calculated as follows:

$$\text{IGD}(P, P^*) = \frac{\sum_{i=1}^{|P|} d(P_i, P^*)}{|P^*|} \tag{7}$$

where $|P|$ is the number of individuals in set P, $d(P_i, P^*)$ is the minimum Euclidean distance from the solution P_i to P^*, and $|P^*|$ is the number of individuals in set P^*.

4.2.2. Hypervolume (HV)

Let P represent the set of points on the final obtained PF and $Z = (z_1, z_2, \cdots, z_m)^T$ represent an m-dimensional reference point in the objective space that is dominated by all Pareto optimal points. The HV [34] is then calculated as follows:

$$\text{HV}(P, Z) = Volume(\cup_{F \in P} [f_1, z_1] * \cdots * [f_m, z_m]) \tag{8}$$

4.3. Results and Discussion

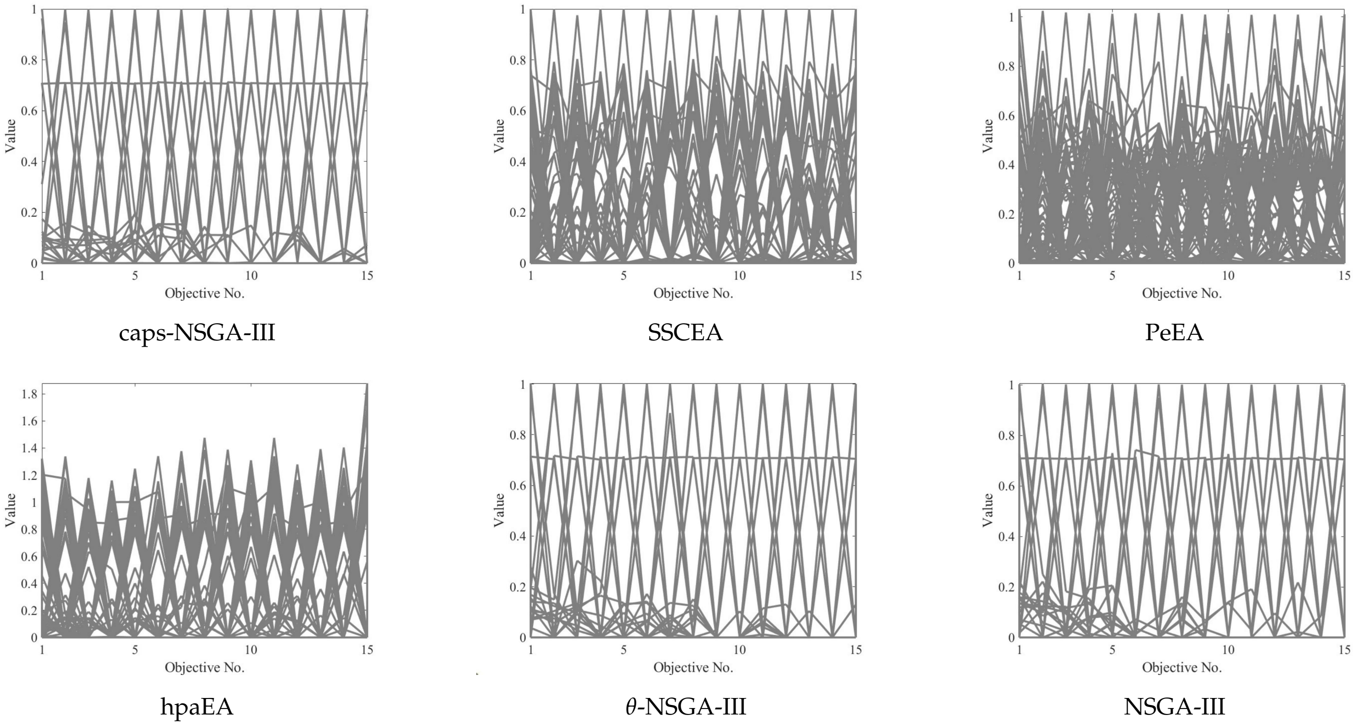
4.3.1. Results on DTLZ Suite

The IGD values obtained by the six algorithms under different DTLZ1-4 instances are shown in Table 3, with the best results highlighted. According to Table 3, caps-NSGA-III excels in 10 out of 20 tests on DTLZ1-4. Especially on DTLZ3, caps-NSGA-III outperforms other algorithms in tests with all objectives except for the 10-objective instance. Additionally, SSCEA works well on the eight-objective and ten-objective DTLZ1 instances, the 10-objective DTLZ2 instance, and the 10-objective DTLZ3 instance. PeEA excels on the 10-objective DTLZ4 instance. The hpaEA demonstrates superior performance on the three-objective DTLZ2 instance and on the three-objective and five-objective DTLZ4 instances. θ-NSGA-III excels on the 15-objective DTLZ2 instance. NSGA-III performs well on the three-objective DTLZ1 instance.

Figure 4 presents the final PFs generated by six algorithms on the 15-objective DTLZ4 instance. As seen in Figure 4, caps-NSGA-III's overall performance surpasses other algorithms. The diversity performance of θ-NSGA-III and NSGA-III is slightly inferior. Notably, hpaEA exhibits poor convergence, with the maximum function values for all objectives exceeding 1.

4.3.2. Results on WFG Suite

Table 4 presents the HV values obtained by the six algorithms on the WFG suite, with the best results highlighted. According to Table 4, caps-NSGA-III excels in 24 out of 45 tests. Notably, it outperforms other algorithms across all objectives on the WFG3 and WFG8 test problems. Additionally, SSCEA, hpaEA, θ-NSGA-III, and NSGA-III achieve the best performance in nine, two, eight, and two instances, respectively. Among them, SSCEA performs well on WFG2 and WFG4. Additionally, it also demonstrates superior results on the eight-objective and fifteen-objective WFG1 instances. The hpaEA performs excellently only on the three-objective and five-objective WFG9 instances. θ-NSGA-III demonstrates excellent performance on WFG6 and achieves superior results in high-dimensional (10- and 15-objective) tests on WFG5 and WFG7. Additionally, it also attains superior performance on the 10-objective WFG9 instance. Finally, NSGA-III shows excellent performance only on the 10-objective WFG6 instance and the 15-objective WFG9 instance. Notably, PeEA does not show any advantages in the comparisons.



caps-NSGA-III

SSCEA

PeEA

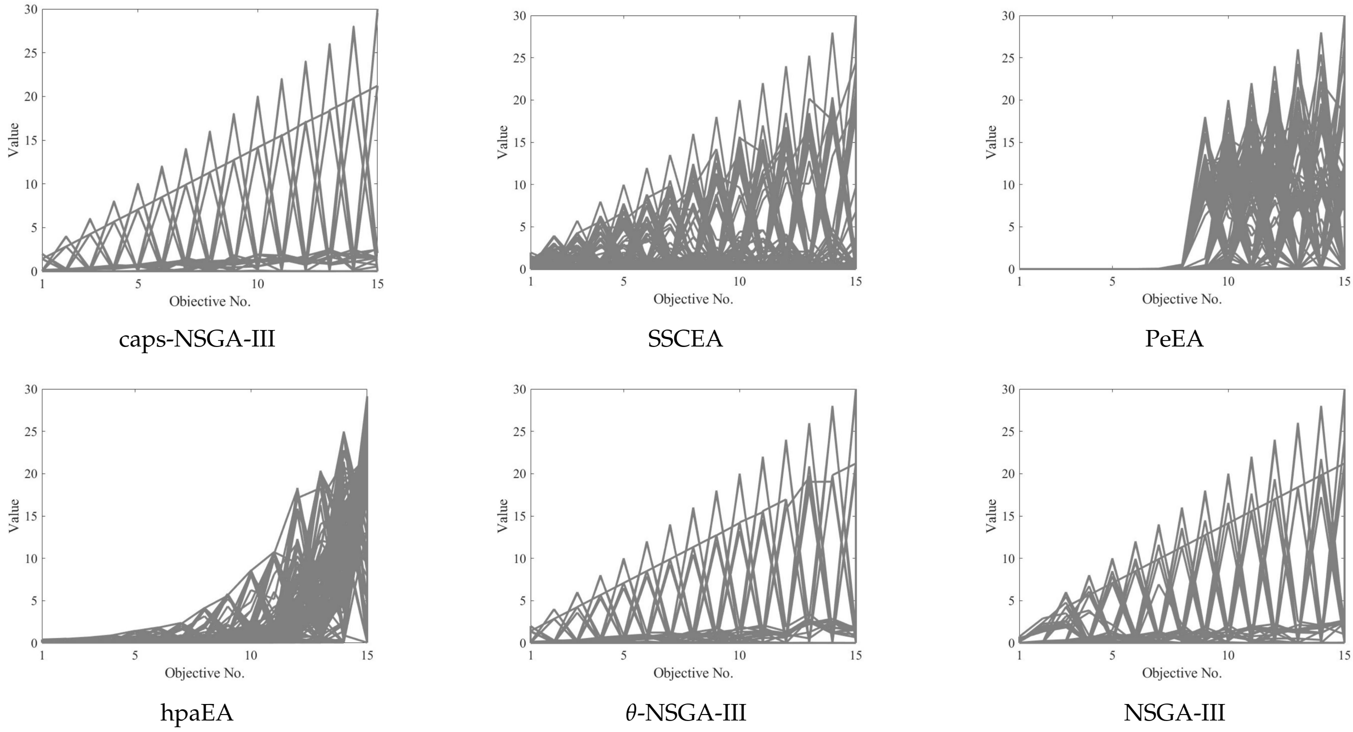
hpaEA

θ-NSGA-III

NSGA-III

Figure 4. Parallel coordinates of the non-dominated fronts obtained by the six algorithms on the 15-objective DTLZ4 instance.

Figure 5 presents the final PFs generated by six algorithms on the 15-objective WFG4 instance. As seen in Figure 5, caps-NSGA-III's overall performance surpasses the other algorithms. θ-NSGA-III and NSGA-III exhibit marginally lower performance compared to caps-NSGA-III. Notably, hpaEA shows poor diversity, as does PeEA for objectives 2–8.



caps-NSGA-III

SSCEA

PeEA

hpaEA

θ-NSGA-III

NSGA-III

Figure 5. Parallel coordinates of the non-dominated fronts obtained by the six algorithms on the 15-objective WFG4 instance.

Table 3. The IGD values (mean and standard deviation) obtained by caps-NSGA-III and other algorithms on the DTLZ test suite.

Problem	M	NSGA-III	θ-NSGA-III	hpaEA	PeEA	SSCEA	caps-NSGA-III
DTLZ1	3	1.464×10^{-1} (1.63×10^{-1})	2.360×10^{-1} (3.12×10^{-1})	1.989×10^{-1} (1.78×10^{-1})	2.046×10^{-1}(2.26×10^{-1})	2.705×10^{-1} (2.68×10^{-1})	3.515×10^{-1} (1.02×10^{-4})
	5	2.084×10^{-1} (1.73×10^{-1})	1.145×10^{-1} (1.39×10^{-1})	2.180×10^{-1} (1.79×10^{-1})	7.526×10^{-2}(5.66×10^{-2})	7.342×10^{-2} (5.84×10^{-2})	5.592×10^{-2} (5.34×10^{-3})
	8	9.122×10^{-1} (3.31×10^{-3})	4.305×10^{-1} (3.29×10^{-1})	1.277×10^{0} (7.22×10^{-1})	1.937×10^{-1}(1.21×10^{-1})	1.162×10^{-1} (4.63×10^{-2})	4.199×10^{-1} (2.14×10^{-1})
	10	2.996×10^{-1} (1.99×10^{-1})	2.458×10^{-1} (1.98×10^{-1})	3.023×10^{0} (1.15×10^{0})	1.443×10^{-1}(6.18×10^{-2})	1.070×10^{-1} (4.45×10^{-3})	1.855×10^{-1} (4.61×10^{-2})
	15	8.530×10^{-1} (5.76×10^{-1})	5.601×10^{-1} (2.78×10^{-1})	3.703×10^{0} (9.28×10^{-1})	3.380×10^{-1}(1.79×10^{-1})	3.411×10^{-1} (1.86×10^{-1})	3.357×10^{-1} (1.54×10^{-1})
DTLZ2	3	5.449×10^{-2} (2.41×10^{-5})	5.448×10^{-2} (1.59×10^{-5})	5.361×10^{-2} (1.78×10^{-1})	6.471×10^{-2}(4.45×10^{-3})	5.795×10^{-2} (1.05×10^{-3})	8.223×10^{-1} (8.50×10^{-2})
	5	1.651×10^{-1} (1.01×10^{-4})	1.651×10^{-1} (8.64×10^{-4})	1.674×10^{-1} (2.00×10^{-2})	1.763×10^{-1}(3.24×10^{-3})	1.652×10^{-1} (8.64×10^{-4})	1.651×10^{-1} (4.99×10^{-5})
	8	4.017×10^{-1} (8.92×10^{-2})	3.235×10^{-1} (2.50×10^{-3})	5.267×10^{-1} (5.52×10^{-2})	3.670×10^{-1}(4.28×10^{-3})	3.477×10^{-1} (6.92×10^{-3})	3.217×10^{-1} (5.15×10^{-3})
	10	4.655×10^{-1} (4.49×10^{-2})	4.266×10^{-1} (2.39×10^{-3})	6.238×10^{-1} (4.50×10^{-2})	4.214×10^{-1}(2.35×10^{-2})	3.870×10^{-1} (4.82×10^{-3})	4.319×10^{-1} (1.50×10^{-2})
	15	6.527×10^{-1} (2.14×10^{-2})	6.172×10^{-1} (1.01×10^{-2})	9.358×10^{-1} (3.55×10^{-2})	7.521×10^{-1}(5.69×10^{-2})	6.432×10^{-1} (5.64×10^{-2})	6.393×10^{-1} (1.65×10^{-2})
DTLZ3	3	1.302×10^{1} (5.49×10^{0})	8.846×10^{0} (4.42×10^{0})	1.534×10^{1} (9.83×10^{0})	5.328×10^{0}(2.97×10^{0})	1.742×10^{1} (1.32×10^{1})	9.933×10^{-1} (2.10×10^{-4})
	5	4.914×10^{0} (4.09×10^{0})	5.605×10^{0} (3.04×10^{0})	5.359×10^{0} (3.03×10^{0})	3.833×10^{0}(3.20×10^{0})	3.242×10^{0} (2.08×10^{0})	1.020×10^{0} (8.38×10^{-1})
	8	1.360×10^{1} (6.24×10^{0})	1.281×10^{1} (9.54×10^{0})	3.358×10^{1} (5.52×10^{-2})	1.202×10^{1}(8.69×10^{0})	7.162×10^{0} (4.01×10^{0})	7.148×10^{0} (4.64×10^{0})
	10	9.363×10^{0} (5.06×10^{0})	6.361×10^{0} (5.17×10^{0})	5.414×10^{1} (1.83×10^{1})	3.762×10^{0}(3.26×10^{0})	2.846×10^{0} (2.72×10^{0})	6.693×10^{0} (6.93×10^{0})
	15	2.385×10^{1} (9.98×10^{0})	1.421×10^{1} (5.60×10^{0})	8.530×10^{1} (2.28×10^{1})	9.575×10^{0}(6.39×10^{0})	9.853×10^{0} (6.53×10^{0})	9.567×10^{0}(4.62×10^{0})
DTLZ4	3	5.449×10^{-2} (1.82×10^{-5})	5.449×10^{-2} (1.17×10^{-5})	5.352×10^{-2} (3.59×10^{-4})	6.149×10^{-2}(1.10×10^{-3})	5.765×10^{-2} (7.68×10^{-4})	8.135×10^{-1} (9.64×10^{-2})
	5	1.651×10^{-1} (1.21×10^{-4})	1.650×10^{-1} (6.17×10^{-5})	1.612×10^{-1} (7.55×10^{-4})	1.729×10^{-1}(2.05×10^{-3})	1.674×10^{-1} (1.24×10^{-3})	1.651×10^{-1} (5.96×10^{-5})
	8	3.260×10^{-1} (1.62×10^{-3})	3.285×10^{-1} (2.50×10^{-2})	3.594×10^{-1} (1.26×10^{-2})	3.656×10^{-1}(2.85×10^{-3})	3.510×10^{-1} (1.01×10^{-3})	3.251×10^{-1} (1.17×10^{-3})
	10	4.351×10^{-1} (3.31×10^{-3})	4.312×10^{-1} (2.58×10^{-3})	5.156×10^{-1} (1.71×10^{-2})	4.080×10^{-1}(4.42×10^{-3})	4.081×10^{-1} (8.42×10^{-3})	4.375×10^{-1} (2.63×10^{-3})
	15	6.333×10^{-1} (1.63×10^{-2})	6.320×10^{-1} (4.47×10^{-3})	6.479×10^{-1} (1.09×10^{-2})	1.115×10^{0}(8.88×10^{-2})	8.627×10^{-1} (1.01×10^{-1})	6.317×10^{-1} (1.30×10^{-2})

Table 4. The HV values (mean and standard deviation) obtained by caps-NSGA-III and other algorithms on the WFG test suite.

Problem	M	NSGA-III	θ-NSGA-III	hpaEA	PeEA	SSCEA	caps-NSGA-III
WFG1	3	6.746×10^{-1} (4.34×10^{-3})	7.122×10^{-1} (6.21×10^{-4})	4.859×10^{-1} (3.13×10^{-3})	7.268×10^{-1} (2.14×10^{-3})	5.905×10^{-1} (6.46×10^{-3})	8.025×10^{-1} (3.81×10^{-2})
	5	8.553×10^{-1} (2.47×10^{-4})	8.708×10^{-1} (5.37×10^{-3})	8.731×10^{-1} (3.08×10^{-2})	9.199×10^{-1} (3.20×10^{-3})	8.955×10^{-1} (2.78×10^{-3})	9.313×10^{-1} (4.33×10^{-2})
	8	7.840×10^{-1} (2.16×10^{-3})	8.034×10^{-1} (6.43×10^{-3})	7.872×10^{-1} (6.17×10^{-3})	9.086×10^{-1} (4.86×10^{-2})	9.113×10^{-1} (9.39×10^{-2})	8.838×10^{-1} (7.07×10^{-2})
	10	9.340×10^{-1} (7.66×10^{-3})	9.466×10^{-1} (4.35×10^{-3})	9.142×10^{-1} (6.78×10^{-2})	9.224×10^{-1} (1.95×10^{-2})	9.773×10^{-1} (9.64×10^{-2})	9.858×10^{-1} (2.45×10^{-3})
	15	8.056×10^{-1} (3.21×10^{-3})	7.754×10^{-1} (6.42×10^{-3})	7.859×10^{-1} (1.28×10^{-3})	9.217×10^{-1} (4.57×10^{-3})	9.770×10^{-1} (2.11×10^{-3})	9.197×10^{-1} (1.80×10^{-1})
WFG2	3	8.319×10^{-1} (9.62×10^{-2})	8.153×10^{-1} (5.79×10^{-2})	7.941×10^{-1} (8.36×10^{-3})	8.112×10^{-1} (8.71×10^{-3})	8.428×10^{-1} (6.33×10^{-3})	9.046×10^{-1} (8.40×10^{-3})
	5	9.497×10^{-1} (4.38×10^{-3})	9.235×10^{-1} (5.24×10^{-3})	8.624×10^{-1} (9.77×10^{-2})	9.046×10^{-1} (9.32×10^{-2})	9.583×10^{-1} (9.31×10^{-2})	9.794×10^{-1} (6.59×10^{-4})
	8	8.961×10^{-1} (8.95×10^{-3})	3.862×10^{-1} (9.31×10^{-2})	5.058×10^{-1} (1.23×10^{-3})	9.236×10^{-1} (6.64×10^{-2})	9.571×10^{-1} (6.58×10^{-2})	9.318×10^{-1} (2.11×10^{-2})
	10	9.503×10^{-1} (4.25×10^{-3})	8.669×10^{-1} (6.82×10^{-2})	9.394×10^{-1} (2.32×10^{-2})	4.051×10^{-1} (5.26×10^{-3})	9.751×10^{-1} (6.29×10^{-3})	9.352×10^{-1} (1.89×10^{-2})
	15	8.237×10^{-1} (6.82×10^{-3})	7.129×10^{-1} (8.41×10^{-3})	8.087×10^{-1} (2.19×10^{-2})	8.865×10^{-1} (2.81×10^{-2})	9.296×10^{-1} (9.02×10^{-2})	3.027×10^{-1} (9.79×10^{-2})
WFG3	3	3.446×10^{-1} (6.55×10^{-3})	3.407×10^{-1} (5.49×10^{-3})	3.324×10^{-1} (3.46×10^{-3})	3.247×10^{-1} (2.14×10^{-3})	3.835×10^{-1} (1.49×10^{-3})	4.631×10^{-1} (2.72×10^{-2})
	5	1.086×10^{-1} (3.97×10^{-2})	1.444×10^{-1} (3.18×10^{-3})	1.149×10^{-1} (5.51×10^{-3})	1.605×10^{-1} (5.76×10^{-2})	1.610×10^{-1} (5.48×10^{-3})	6.142×10^{-1} (5.04×10^{-3})
	8	0.000×10^{0} (0.00×10^{0})	4.469×10^{-4} (3.13×10^{-1})	0.000×10^{0} (0.00×10^{0})	1.681×10^{-2} (5.95×10^{-2})	6.349×10^{-3} (3.97×10^{-3})	4.317×10^{-1} (1.93×10^{-2})
	10	0.000×10^{0} (0.00×10^{0})	0.000×10^{0} (0.00×10^{0})	0.000×10^{0} (0.00×10^{0})	5.147×10^{-1} (4.72×10^{-3})	0.000×10^{0} (0.00×10^{0})	5.335×10^{-1} (2.63×10^{-3})
	15	0.000×10^{0} (0.00×10^{0})	0.000×10^{0} (0.00×10^{0})	0.000×10^{0} (0.00×10^{0})	0.000×10^{0} (0.00×10^{0})	0.000×10^{0} (0.00×10^{0})	3.237×10^{-1} (1.15×10^{-1})

Table 4. *Cont.*

Problem	M	NSGA-III	θ-NSGA-III	hpaEA	PeEA	SSCEA	caps-NSGA-III
WFG4	3	5.456×10^{-1} (2.75×10^{-2})	5.471×10^{-1} (4.57×10^{-3})	5.056×10^{-1} (3.25×10^{-3})	5.338×10^{-1} (4.66×10^{-3})	5.496×10^{-1} (1.27×10^{-3})	5.474×10^{-1} (1.87×10^{-3})
	5	7.879×10^{-1} (3.62×10^{-3})	7.898×10^{-1} (2.54×10^{-3})	6.468×10^{-1} (4.13×10^{-3})	7.693×10^{-1} (5.52×10^{-3})	7.983×10^{-1} (9.47×10^{-3})	7.858×10^{-1} (2.85×10^{-2})
	8	8.918×10^{-1} (4.33×10^{-3})	9.039×10^{-1} (6.47×10^{-3})	5.615×10^{-1} (2.39×10^{-3})	7.963×10^{-1} (3.82×10^{-3})	9.065×10^{-1} (7.16×10^{-3})	8.996×10^{-1} (4.05×10^{-3})
	10	9.327×10^{-1} (5.74×10^{-3})	9.561×10^{-1} (4.16×10^{-2})	6.008×10^{-1} (6.79×10^{-3})	6.811×10^{-1} (3.81×10^{-3})	9.570×10^{-1} (9.47×10^{-2})	9.508×10^{-1} (5.43×10^{-1})
	15	$9.596 \times 10^{-1}1$ (2.06×10^{-2})	9.779×10^{-1} (8.30×10^{-3})	4.659×10^{-1} (9.78×10^{-3})	8.226×10^{-1} (2.24×10^{-1})	9.568×10^{-1} (9.30×10^{-3})	9.843×10^{-1} (1.63×10^{-3})
WFG5	3	5.073×10^{-1} (3.95×10^{-3})	5.069×10^{-1} (2.59×10^{-3})	5.001×10^{-1} (5.29×10^{-2})	5.006×10^{-1} (5.64×10^{-3})	5.009×10^{-1} (4.43×10^{-3})	5.289×10^{-1} (3.58×10^{-3})
	5	7.517×10^{-1} (3.07×10^{-4})	7.529×10^{-1} (5.81×10^{-3})	7.330×10^{-1} (4.27×10^{-3})	7.240×10^{-1} (4.75×10^{-3})	7.200×10^{-1} (8.26×10^{-2})	7.628×10^{-1} (1.86×10^{-3})
	8	8.496×10^{-1} (5.24×10^{-2})	8.536×10^{-1} (4.16×10^{-3})	7.366×10^{-1} (2.87×10^{-2})	8.086×10^{-1} (2.01×10^{-2})	8.067×10^{-1} (5.70×10^{-1})	8.599×10^{-1} (1.06×10^{-3})
	10	8.980×10^{-1} (5.87×10^{-2})	9.020×10^{-1} (5.42×10^{-2})	7.794×10^{-1} (9.41×10^{-2})	3.546×10^{-1} (8.19×10^{-2})	8.517×10^{-1} (2.63×10^{-1})	8.942×10^{-1} (2.45×10^{-3})
	15	9.082×10^{-1} (5.29×10^{-2})	9.147×10^{-1} (5.33×10^{-2})	5.713×10^{-1} (3.89×10^{-3})	8.136×10^{-1} (1.41×10^{-3})	8.275×10^{-1} (6.59×10^{-1})	9.137×10^{-1} (6.11×10^{-2})
WFG6	3	5.137×10^{-1} (3.89×10^{-3})	5.128×10^{-1} (3.11×10^{-3})	5.060×10^{-1} (3.39×10^{-3})	4.963×10^{-1} (1.16×10^{-3})	5.141×10^{-1} (2.64×10^{-2})	5.198×10^{-1} (2.01×10^{-3})
	5	7.600×10^{-1} (6.58×10^{-2})	7.618×10^{-1} (4.33×10^{-3})	7.439×10^{-1} (5.80×10^{-3})	7.132×10^{-1} (5.01×10^{-3})	7.592×10^{-1} (2.63×10^{-3})	7.422×10^{-1} (3.49×10^{-3})
	8	8.610×10^{-1} (1.13×10^{-2})	8.661×10^{-1} (6.52×10^{-3})	7.972×10^{-1} (1.55×10^{-3})	7.190×10^{-1} (4.79×10^{-3})	8.654×10^{-1} (8.67×10^{-2})	8.384×10^{-1} (6.63×10^{-3})
	10	9.124×10^{-1} (1.38×10^{-2})	9.120×10^{-1} (1.57×10^{-2})	8.471×10^{-1} (2.72×10^{-2})	7.681×10^{-1} (1.52×10^{-2})	9.091×10^{-1} (1.19×10^{-1})	8.732×10^{-1} (5.79×10^{-3})
	15	9.171×10^{-1} (1.67×10^{-2})	9.259×10^{-1} (6.66×10^{-3})	6.445×10^{-1} (1.44×10^{-2})	8.495×10^{-1} (2.79×10^{-2})	9.130×10^{-1} (1.31×10^{-1})	8.808×10^{-1} (9.39×10^{-3})
WFG7	3	5.391×10^{-1} (1.35×10^{-3})	5.395×10^{-1} (5.71×10^{-3})	5.376×10^{-1} (1.16×10^{-2})	5.371×10^{-1} (4.55×10^{-3})	4.923×10^{-1} (7.26×10^{-3})	5.517×10^{-1} (3.93×10^{-3})
	5	8.000×10^{-1} (1.19×10^{-3})	8.011×10^{-1} (9.44×10^{-4})	7.732×10^{-1} (2.13×10^{-3})	7.721×10^{-1} (9.63×10^{-4})	7.268×10^{-1} (7.09×10^{-1})	8.020×10^{-1} (5.04×10^{-3})
	8	8.261×10^{-1} (5.41×10^{-3})	9.045×10^{-1} (7.63×10^{-4})	7.343×10^{-1} (6.05×10^{-2})	8.761×10^{-1} (4.35×10^{-3})	8.023×10^{-1} (2.19×10^{-1})	9.124×10^{-1} (3.75×10^{-3})
	10	8.754×10^{-1} (6.43×10^{-3})	9.623×10^{-1} (7.11×10^{-1})	7.757×10^{-1} (7.26×10^{-3})	9.240×10^{-1} (3.08×10^{-3})	8.762×10^{-1} (9.03×10^{-1})	9.535×10^{-1} (2.63×10^{-3})
	15	9.652×10^{-1} (8.18×10^{-3})	9.866×10^{-1} (5.81×10^{-2})	5.897×10^{-1} (6.42×10^{-2})	8.825×10^{-1} (4.57×10^{-2})	8.792×10^{-1} (7.39e+)	9.860×10^{-1} (1.10×10^{-2})
WFG8	3	4.470×10^{-1} (2.88×10^{-3})	4.533×10^{-1} (3.71×10^{-3})	4.463×10^{-1} (5.16×10^{-3})	4.367×10^{-1} (5.16×10^{-3})	4.684×10^{-1} (4.88×10^{-2})	5.068×10^{-1} (2.15×10^{-2})
	5	6.730×10^{-1} (2.19×10^{-3})	6.748×10^{-1} (3.51×10^{-2})	6.730×10^{-1} (4.42×10^{-3})	6.411×10^{-1} (6.07×10^{-3})	6.720×10^{-1} (7.46×10^{-1})	7.522×10^{-1} (2.85×10^{-2})
	8	7.529×10^{-1} (9.69×10^{-3})	7.468×10^{-1} (1.19×10^{-2})	6.789×10^{-1} (7.65×10^{-3})	7.544×10^{-1} (1.39×10^{-2})	7.533×10^{-1} (6.26×10^{-1})	8.863×10^{-1} (1.23×10^{-2})
	10	8.520×10^{-1} (8.49×10^{-3})	8.405×10^{-1} (1.75×10^{-2})	7.473×10^{-1} (9.28×10^{-3})	8.529×10^{-1} (8.73×10^{-2})	8.398×10^{-1} (2.86×10^{-1})	9.440×10^{-1} (5.43×10^{-1})
	15	8.635×10^{-1} (1.42×10^{-1})	8.852×10^{-1} (5.24×10^{-2})	6.432×10^{-1} (1.49×10^{-2})	8.744×10^{-1} (1.77×10^{-2})	8.946×10^{-1} (3.21e+)	9.769×10^{-1} (4.26×10^{-3})
WFG9	3	4.674×10^{-1} (2.05×10^{-3})	4.678×10^{-1} (1.84×10^{-2})	5.006×10^{-1} (2.28×10^{-3})	4.685×10^{-1} (5.27×10^{-3})	4.566×10^{-1} (3.02×10^{-1})	4.971×10^{-1} (9.71×10^{-4})
	5	6.667×10^{-1} (2.16×10^{-2})	6.747×10^{-1} (3.41×10^{-3})	6.814×10^{-1} (2.56×10^{-2})	6.678×10^{-1} (8.07×10^{-3})	6.196×10^{-1} (2.06×10^{-1})	6.799×10^{-1} (2.85×10^{-2})
	8	7.042×10^{-1} (1.68×10^{-2})	7.138×10^{-1} (1.93×10^{-2})	6.716×10^{-1} (2.38×10^{-2})	6.970×10^{-1} (9.09×10^{-3})	6.328×10^{-1} (3.49×10^{-1})	7.347×10^{-1} (3.04×10^{-3})
	10	7.616×10^{-1} (3.73×10^{-3})	7.860×10^{-1} (4.18×10^{-2})	7.288×10^{-1} (8.77×10^{-3})	5.620×10^{-1} (4.07×10^{-3})	6.921×10^{-1} (6.34×10^{-2})	7.627×10^{-1} (5.43×10^{-1})
	15	7.475×10^{-1} (2.78×10^{-2})	7.445×10^{-1} (4.05×10^{-2})	5.730×10^{-1} (1.87×10^{-3})	7.103×10^{-1} (5.01×10^{-2})	6.172×10^{-1} (2.77×10^{-1})	7.337×10^{-1} (5.93×10^{-3})

Overall, caps-NSGA-III performs the best in many-objective tests for DTLZ1-4 and WFG1-9, particularly in DTLZ3, WFG3, and WFG8. This demonstrates the effectiveness of the penalty scheme we proposed in tackling complex many-objective problems. The adaptive penalty scheme enhances the algorithm's performance, and the monitoring strategy further harmonizes its convergence and diversity performance across different stages.

4.4. Real-World Problem Applications

Two real-world cases were selected for experimental comparison: car side impact [35,36] and water resource planning [35,36]. Among these, water resource planning refers to the optimal planning problem for urban storm drainage systems. It involves three variables (local detention storage capacity, maximum treatment rate, and maximum allowable overflow rate) and five objective functions, which include costs (drainage network, storage facility, treatment facility, and expected flood damage) and expected economic loss due to flood. The car side impact problem involves seven variables, including the thicknesses of the B-Pillars, floor, crossmembers, door beam, roof rail, etc. It aims to achieve three objectives: minimizing the car's weight, minimizing the public force experienced by a passenger, and minimizing the average velocity of the V-Pillar responsible for withstanding impact load. As recommended by Tanabe and Ishibuchi [35], an additional objective function was added to each problem: minimizing the total constraint violation.

For both problems, the population sizes are 100 and 150, respectively, and the termination conditions are 10,000 and 21,000 fitness evaluations, respectively. Other algorithm settings remain consistent with those outlined in Section 4.1.2. Due to the unavailability of the true PF, the HV metric is employed for evaluation, with the obtained HV results presented in Table 5. According to Table 5, caps-NSGA-III exhibits the best performance for both problems. This demonstrates the algorithm's effectiveness in addressing real-world problems.

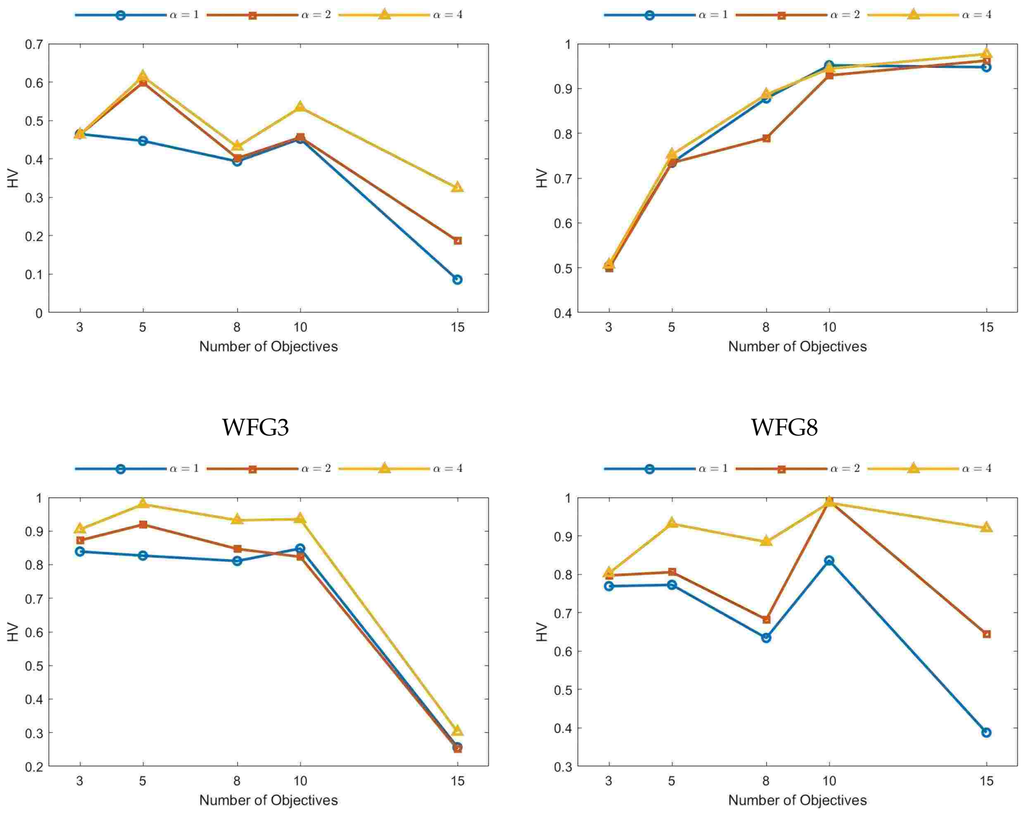
Table 5. The HV values (mean and standard deviation) obtained by caps-NSGA-III and other algorithms on real-world problems.

Problem	NSGA-III	θ-NSGA-III	SPS-NSGA-III	hpaEA	PeEA	SSCEA	caps-NSGA-III
Car side impact	2.677×10^{-2} (1.27×10^{-3})	2.537×10^{-2} (1.05×10^{-3})	2.653×10^{-2} (1.20×10^{-3})	1.795×10^{-2} (2.31×10^{-3})	1.779×10^{-2} (2.50×10^{-3})	2.323×10^{-2} (1.56×10^{-3})	2.732×10^{-2} (1.11×10^{-3})
Water resource planning	1.097×10^{-1} (2.35×10^{-2})	6.780×10^{-2} (2.69×10^{-2})	1.105×10^{-1} (2.24×10^{-2})	1.221×10^{-1} (1.68×10^{-2})	1.192×10^{-1} (2.81×10^{-2})	1.217×10^{-1} (1.73×10^{-2})	1.249×10^{-1} (2.70×10^{-2})

4.5. Parameter Sensitivity Analysis

To investigate the impact of parameters on caps-NSGA-III, we conducted a sensitivity analysis on the parameter α in the SPS method with values of 1, 2, and 4. The experimental setup is consistent with Section 4. We selected four test problems—WFG3 (linear PF), WFG8 (concave PF), WFG2 (convex PF), and WFG1 (mixed PF)—to identify the optimal value of α.

Figure 6 presents the HV line plots for different values of α across various types of PF. It can be observed from the figure that different values of α indeed affect the performance of caps-NSGA-III. Although performance differences on WFG8 are small, indicating that concave problems may be less influenced by the penalty factor θ, caps-NSGA-III overall performs best when $\alpha = 4$. On the whole, caps-NSGA-III achieves the best results when $\alpha = 4$, especially on WFG2, where the difference compared to other α values is the most significant. This may be because the boundary regions of convex problems are harder to identify, and larger values of α can lead to higher θ values for boundary vectors, thereby helping to maintain diversity.



WFG3 WFG8

WFG2 WFG1

Figure 6. Analysis of the impact of α on caps-NSGA-III for the WFG1, WFG2, WFG3, and WFG8 problems. The blue line represents $\alpha = 1$, the orange line represents $\alpha = 2$, and the yellow line represents $\alpha = 4$.

4.6. Ablation Studies

To verify the effectiveness of the adaptive penalty factor and monitoring strategy components, we conducted a series of ablation experiments on the three-, eight-, and fifteen-objective WFG test suites, with the experimental setup consistent with that described in Section 4. The algorithms compared include the original PBI-NSGA-III, SPS-NSGA-III, adaptive penalty factor-NSGA-III (AP-NSGA-III), and caps-NSGA-III, which combines the adaptive penalty factor with the monitoring strategy. Among these, PBI-NSGA-III represents the baseline algorithm, SPS-NSGA-III is used to validate the effectiveness of the SPS component, AP-NSGA-III is used to validate the effectiveness of the adaptive penalty factor component, and caps-NSGA-III is used to validate the effectiveness of the monitoring strategy component.

Table 6 presents the HV values obtained by the four algorithms on the WFG suite, with the best results highlighted. The results show that caps-NSGA-III performed the best, achieving the top performance in 20 out of 27 comparisons. It is followed by AP-NSGA-III, which achieved the best performance in four comparisons, and then SPS-NSGA-III, which achieved the best performance in three comparisons. Notably, PBI-NSGA-III did not show any advantages in the comparisons. By comparing the four algorithms both as a whole and individually, it can be observed that the introduction of each component indeed improved the performance of the algorithms.

Table 6. The HV values (mean and standard deviation) obtained by caps-NSGA-III and other algorithms on the WFG test suite.

Problem	M	PBI-NSGA-III	SPS-NSGA-III	AP-NSGA-III	caps-NSGA-III
WFG1	3	7.477×10^{-1} (4.26×10^{-2})	7.780×10^{-1} (2.67×10^{-2})	7.881×10^{-1} (3.08×10^{-2})	8.025×10^{-1} (3.81×10^{-2})
	8	7.488×10^{-1} (1.63×10^{-1})	9.174×10^{-1} (1.06×10^{-1})	9.340×10^{-1} (7.05×10^{-2})	8.838×10^{-1} (7.07×10^{-2})
	15	7.686×10^{-1} (2.48×10^{-1})	8.863×10^{-1} (1.90×10^{-1})	8.637×10^{-1} (2.15×10^{-1})	9.197×10^{-1} (1.80×10^{-1})
WFG2	3	8.001×10^{-1} (9.38×10^{-2})	8.311×10^{-1} (8.54×10^{-2})	9.018×10^{-1} (1.15×10^{-2})	9.046×10^{-1} (8.40×10^{-3})
	8	7.856×10^{-1} (6.63×10^{-2})	8.466×10^{-1} (1.14×10^{-1})	9.018×10^{-1} (2.94×10^{-2})	9.318×10^{-1} (2.11×10^{-2})
	15	3.303×10^{-1} (7.13×10^{-2})	4.540×10^{-1} (9.23×10^{-2})	4.467×10^{-1} (1.45×10^{-1})	3.027×10^{-1} (9.79×10^{-2})
WFG3	3	4.481×10^{-1} (3.28×10^{-2})	4.595×10^{-1} (2.52×10^{-2})	4.889×10^{-1} (1.79×10^{-2})	4.631×10^{-1} (2.72×10^{-2})
	8	4.141×10^{-1} (3.33×10^{-2})	4.107×10^{-1} (2.54×10^{-2})	4.116×10^{-1} (2.42×10^{-2})	4.317×10^{-1} (1.93×10^{-2})
	15	3.851×10^{-1} (4.09×10^{-2})	3.998×10^{-1} (5.18×10^{-2})	3.931×10^{-1} (5.01×10^{-2})	3.237×10^{-1} (1.15×10^{-1})
WFG4	3	5.463×10^{-1} (1.76×10^{-3})	5.471×10^{-1} (1.69×10^{-3})	5.473×10^{-1} (1.42×10^{-3})	5.474×10^{-1} (1.87×10^{-3})
	8	8.615×10^{-1} (7.47×10^{-2})	8.664×10^{-1} (6.15×10^{-2})	8.797×10^{-1} (5.39×10^{-2})	8.996×10^{-1} (4.05×10^{-3})
	15	9.720×10^{-1} (1.41×10^{-2})	9.721×10^{-1} (1.42×10^{-2})	9.517×10^{-1} (5.01×10^{-2})	9.843×10^{-1} (1.63×10^{-3})
WFG5	3	5.273×10^{-1} (3.53×10^{-3})	5.273×10^{-1} (2.97×10^{-3})	5.277×10^{-1} (3.03×10^{-3})	5.289×10^{-1} (3.58×10^{-3})
	8	8.482×10^{-1} (4.92×10^{-2})	8.449×10^{-1} (9.32×10^{-2})	8.510×10^{-1} (3.27×10^{-2})	8.599×10^{-1} (1.06×10^{-3})
	15	8.913×10^{-1} (6.73×10^{-2})	8.766×10^{-1} (8.30×10^{-2})	8.526×10^{-1} (1.05×10^{-1})	9.137×10^{-1} (6.11×10^{-2})
WFG6	3	5.194×10^{-1} (2.75×10^{-3})	5.194×10^{-1} (1.89×10^{-3})	5.202×10^{-1} (2.95×10^{-3})	5.198×10^{-1} (2.01×10^{-3})
	8	8.371×10^{-1} (6.49×10^{-3})	8.395×10^{-1} (8.29×10^{-3})	8.325×10^{-1} (4.62×10^{-3})	8.384×10^{-1} (6.63×10^{-3})
	15	8.764×10^{-1} (2.02×10^{-2})	8.762×10^{-1} (1.64×10^{-2})	8.802×10^{-1} (1.12×10^{-2})	8.808×10^{-1} (9.39×10^{-3})
WFG7	3	5.522×10^{-1} (3.62×10^{-3})	5.518×10^{-1} (2.96×10^{-3})	5.525×10^{-1} (1.46×10^{-3})	5.517×10^{-1} (3.93×10^{-3})
	8	9.045×10^{-1} (3.69×10^{-2})	8.499×10^{-1} (4.61×10^{-2})	8.700×10^{-1} (7.61×10^{-2})	9.124×10^{-1} (3.75×10^{-3})
	15	9.811×10^{-1} (4.26×10^{-3})	9.809×10^{-1} (4.52×10^{-3})	9.630×10^{-1} (3.74×10^{-2})	9.860×10^{-1} (1.10×10^{-2})
WFG8	3	4.992×10^{-1} (1.72×10^{-2})	4.989×10^{-1} (1.92×10^{-2})	4.991×10^{-1} (1.80×10^{-2})	5.068×10^{-1} (2.15×10^{-2})
	8	7.833×10^{-1} (2.17×10^{-2})	7.892×10^{-1} (2.03×10^{-2})	7.471×10^{-1} (1.77×10^{-2})	8.863×10^{-1} (1.23×10^{-2})
	15	8.837×10^{-1} (6.71×10^{-2})	9.002×10^{-1} (1.04×10^{-2})	9.424×10^{-1} (4.41×10^{-2})	9.769×10^{-1} (4.26×10^{-3})
WFG9	3	4.966×10^{-1} (1.09×10^{-3})	4.968×10^{-1} (1.62×10^{-3})	4.963×10^{-1} (1.22×10^{-3})	4.971×10^{-1} (9.71×10^{-4})
	8	7.010×10^{-1} (9.82×10^{-2})	7.247×10^{-1} (7.30×10^{-2})	7.134×10^{-1} (6.39×10^{-2})	7.347×10^{-1} (3.04×10^{-3})
	15	7.150×10^{-1} (6.20×10^{-2})	6.856×10^{-1} (8.63×10^{-2})	6.445×10^{-1} (8.38×10^{-2})	7.337×10^{-1} (5.93×10^{-3})
Best/Worse		3/24	5/22	6/21	

Best means that the algorithm's HV value is better than caps-NSGA-III, while Worse means it is worse than caps-NSGA-III.

5. Conclusions

In this paper, an improved algorithm named caps-NSGA-III is proposed. This algorithm integrates NSGA-III with a comprehensive adaptive penalty scheme to balance convergence and diversity. Specifically, chaotic mapping is used in caps-NSGA-III to initialize the population, achieving a more uniform distribution and enhanced population diversity. Additionally, in the initial stage, penalty factors are computed according to the properties of the reference vectors themselves. Then, during iterations, these penalty factors are adaptively adjusted according to the evolutionary state of the individuals associated with the corresponding reference vectors, thereby better balancing the algorithm's convergence and diversity. Simultaneously, controlling the threshold for convergence or diversity operation is crucial. To meet the algorithm's needs at different stages, we proposed a monitoring strategy. The adaptive adjustment of penalty factors is monitored during algorithm iterations to achieve adaptive threshold changes, thereby further enhancing the algorithm's performance. Caps-NSGA-III's effectiveness was demonstrated through comparisons with five many-objective evolutionary algorithms. Additionally, the capability of caps-NSGA-III to address practical issues was also validated through comparative experiments on real-world problems. Although our proposed caps-NSGA-III shows improvements in solving MaOPs, its performance on large-scale MaOPs requires further study. Therefore, future work will focus on further exploring caps-NSGA-III and delving into the following aspects: decision variable analysis (DVA [37]), cooperative coevolution (CC) frameworks (incorporating various grouping methods like random grouping [38], dynamic grouping [39], and differential grouping [40]), and problem transformation techniques (such as the Weighted Optimization Framework, WOF [41]) to enhance its performance on MaOPs involving a large number of decision variables.

Author Contributions: Conceptualization, X.X., D.C. and F.Y.; methodology, X.X., D.C. and F.Y.; software, X.X. and D.C.; validation, X.X.; formal analysis, X.X., D.C. and F.Y.; investigation, X.X.; resources, D.C. and F.Y.; data curation, D.W.; writing—original draft preparation, X.X.; writing—review and editing, X.X., D.C. and F.Y.; visualization, X.X. and Q.L.; supervision, F.Y.; project administration, F.Y.; funding acquisition, F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Jilin Provincial Science and Technology Development Plan Project under grant 20220203184SF and the General Project of Graduate Innovation Program at Beihua University ([2023]051).

Data Availability Statement: All of the data are in the article, no other new data are created.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PBI	Penalty-based boundary intersection
caps-NSGA-III	Comprehensive adaptive penalty scheme-NSGA-III
MOPs	Multi-objective optimization problems
MOEAs	Multi-objective evolutionary algorithms
MaOPs	Many-objective optimization problems
PF	Pareto front
PS	Pareto set
APS	Adaptive penalty scheme
SPS	Subproblem-based penalty scheme
cr_d	Rate of change of d_1 in the centroid's PBI at the first violation
cr_{pbi}	Sum of the rate of change of the PBI of the centroid for the three violations
IGD	Inverted generational distance
HV	Hypervolume

References

1. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
2. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK Rep.* **2001**, *103*.
3. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]
4. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
5. Asafuddoula, M.; Ray, T.; Sarker, R. A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Trans. Evol. Comput.* **2014**, *19*, 445–460. [CrossRef]
6. Bao, C.; Gao, D.; Gu, W.; Xu, L.; Goodman, E.D. A new adaptive decomposition-based evolutionary algorithm for multi-and many-objective optimization. *Expert Syst. Appl.* **2023**, *213*, 119080. [CrossRef]
7. Bader, J.; Zitzler, E. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evol. Comput.* **2011**, *19*, 45–76. [CrossRef]
8. Sun, Y.; Yen, G.G.; Yi, Z. IGD indicator-based evolutionary algorithm for many-objective optimization problems. *IEEE Trans. Evol. Comput.* **2018**, *23*, 173–187. [CrossRef]
9. Yuan, J.; Liu, H.L.; Yang, S. An adaptive parental guidance strategy and its derived indicator-based evolutionary algorithm for multi-and many-objective optimization. *Swarm Evol. Comput.* **2024**, *84*, 101449. [CrossRef]
10. Zhu, C.; Zhu, X. Multi-objective path-decision model of multimodal transport considering uncertain conditions and carbon emission policies. *Symmetry* **2022**, *14*, 221. [CrossRef]
11. Yang, W.; Wen, X.; Wu, M.; Bi, K.; Yue, L. Three-Dimensional Conflict Resolution Strategy Based on Network Cooperative Game. *Symmetry* **2022**, *14*, 1517. [CrossRef]
12. Ney, R.; Canha, L.; Adeyanju, O.; Arend, G. Multi-objective optimal planning of distributed energy resources using SPEA2 algorithms considering multi-agent participation. In Proceedings of the 2019 54th International Universities Power Engineering Conference (UPEC), Bucharest, Romania, 3–6 September 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–6.
13. Laumanns, M.; Thiele, L.; Deb, K.; Zitzler, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evol. Comput.* **2002**, *10*, 263–282. [CrossRef] [PubMed]
14. Chen, H.; Tian, Y.; Pedrycz, W.; Wu, G.; Wang, R.; Wang, L. Hyperplane assisted evolutionary algorithm for many-objective optimization problems. *IEEE Trans. Cybern.* **2019**, *50*, 3367–3380. [CrossRef]

15. Li, L.; Yen, G.G.; Sahoo, A.; Chang, L.; Gu, T. On the estimation of pareto front and dimensional similarity in many-objective evolutionary algorithm. *Inf. Sci.* **2021**, *563*, 375–400. [CrossRef]
16. Yuan, Y.; Xu, H.; Wang, B. An improved NSGA-III procedure for evolutionary many-objective optimization. In Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, Vancouver, BC, Canada, 12–16 July 2014; pp. 661–668.
17. Liu, G.; Pei, Z.; Liu, N.; Tian, Y. Subspace segmentation based co-evolutionary algorithm for balancing convergence and diversity in many-objective optimization. *Swarm Evol. Comput.* **2023**, *83*, 101410. [CrossRef]
18. Yang, S.; Jiang, S.; Jiang, Y. Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes. *Soft Comput.* **2017**, *21*, 4677–4691. [CrossRef]
19. Han, D.; Du, W.; Du, W.; Jin, Y.; Wu, C. An adaptive decomposition-based evolutionary algorithm for many-objective optimization. *Inf. Sci.* **2019**, *491*, 204–222. [CrossRef]
20. Srinivas, N.; Deb, K. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **1994**, *2*, 221–248. [CrossRef]
21. Wang, Y.; Chen, C.; Tao, Y.; Wen, Z.; Chen, B.; Zhang, H. A many-objective optimization of industrial environmental management using NSGA-III: A case of China's iron and steel industry. *Appl. Energy* **2019**, *242*, 46–56. [CrossRef]
22. Liu, F.; Liu, J.; Yan, X. Solving the asymmetry multi-objective optimization problem in PPPs under LPVR mechanism by Bi-level programing. *Symmetry* **2020**, *12*, 1667. [CrossRef]
23. Liu, Y.; You, K.; Jiang, Y.; Wu, Z.; Liu, Z.; Peng, G.; Zhou, C. Multi-objective optimal scheduling of automated construction equipment using non-dominated sorting genetic algorithm (NSGA-III). *Autom. Constr.* **2022**, *143*, 104587. [CrossRef]
24. Zhang, X.; Tian, Y.; Jin, Y. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2014**, *19*, 761–776. [CrossRef]
25. Yang, S.; Li, M.; Liu, X.; Zheng, J. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans. Evol. Comput.* **2013**, *17*, 721–736. [CrossRef]
26. Schuster, H.G.; Just, W. *Deterministic Chaos: An Introduction*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
27. Gutiérrez, A.; Lanza, M.; Barriuso, I.; Valle, L.; Domingo, M.; Perez, J.; Basterrechea, J. Comparison of different pso initialization techniques for high dimensional search space problems: A test with fss and antenna arrays. In Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP), Rome, Italy, 11–15 April 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 965–969.
28. Guo, H.; Zhu, D.; Zhou, C.; Zou, C. DNA sequences design under many objective evolutionary algorithm. *Clust. Comput.* **2024**, *27*, 14167–14183. [CrossRef]
29. May, R.M. Simple mathematical models with very complicated dynamics. *Nature* **1976**, *261*, 459–467. [CrossRef]
30. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 105–145.
31. Huband, S.; Hingston, P.; Barone, L.; While, L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans. Evol. Comput.* **2006**, *10*, 477–506. [CrossRef]
32. Das, I.; Dennis, J.E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **1998**, *8*, 631–657. [CrossRef]
33. Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C.M.; Da Fonseca, V.G. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.* **2003**, *7*, 117–132. [CrossRef]
34. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [CrossRef]
35. Tanabe, R.; Ishibuchi, H. An easy-to-use real-world multi-objective optimization problem suite. *Appl. Soft Comput.* **2020**, *89*, 106078. [CrossRef]
36. Palakonda, V.; Kang, J.M.; Jung, H. Benchmarking Real-World Many-Objective Problems: A Problem Suite With Baseline Results. *IEEE Access* **2024**, *12*, 49275–49290. [CrossRef]
37. Trivedi, A.; Srinivasan, D.; Sanyal, K.; Ghosh, A. A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Trans. Evol. Comput.* **2016**, *21*, 440–462. [CrossRef]
38. Yang, Z.; Tang, K.; Yao, X. Large scale evolutionary optimization using cooperative coevolution. *Inf. Sci.* **2008**, *178*, 2985–2999. [CrossRef]
39. Li, X.; Yao, X. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans. Evol. Comput.* **2011**, *16*, 210–224.
40. Mei, Y.; Omidvar, M.N.; Li, X.; Yao, X. A competitive divide-and-conquer algorithm for unconstrained large-scale black-box optimization. *ACM Trans. Math. Softw. (TOMS)* **2016**, *42*, 1–24. [CrossRef]
41. Zille, H.; Ishibuchi, H.; Mostaghim, S.; Nojima, Y. A framework for large-scale multiobjective optimization based on problem transformation. *IEEE Trans. Evol. Comput.* **2017**, *22*, 260–275. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.