Shift-based Penalty for Evolutionary Constrained Multiobjective Optimization and Its Application

Zhongwei Ma and Yong Wang, *Senior Member, IEEE*

***Abstract*—This paper presents a new constraint-handling technique (CHT), called shift-based penalty (ShiP), for solving constrained multiobjective optimization problems. In ShiP, infeasible solutions are first shifted according to the distributions of their neighboring feasible solutions. The degree of shift is adaptively controlled by the proportion of feasible solutions in the current parent and offspring populations. Then, the shifted infeasible solutions are penalized based on their constraint violations. This two-step process can encourage infeasible solutions to approach/enter the feasible region from diverse directions in the early stage of evolution, and guide diverse feasible solutions toward the Pareto optimal solutions in the later stage of evolution. Moreover, ShiP can achieve an adaptive transition from both diversity and feasibility in the early stage of evolution to both diversity and convergence in the later stage of evolution. ShiP is flexible and can be embedded into three well-known multiobjective optimization frameworks. Experiments on benchmark test problems demonstrate that ShiP is highly competitive with other representative CHTs. Further, based on ShiP, we propose an archive-assisted constrained multiobjective evolutionary algorithm (CMOEA), called ShiP$^+$, which outperforms two other state-of-the-art CMOEAs. Finally, ShiP is applied to the vehicle scheduling of urban bus line successfully.**

***Index Terms*—Constrained multiobjective optimization, constraint-handling techniques, evolutionary algorithms, penalty, shift.**

I. Introduction

WITHOUT loss of generality, a constrained multiobjective optimization problem (CMOP), in the case of minimization, can be formulated as follows [1]–[5]:

$$\min\ \vec{F}(\vec{x}) = \big(f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})\big)^T$$
$$\text{s.t.} \begin{cases} g_j(\vec{x}) \le 0,\ j = 1, \dots, n_g \\ h_j(\vec{x}) = 0,\ j = n_g + 1, \dots, n_g + n_h \\ x_k^{low} \le x_k \le x_k^{upp},\ k = 1, \dots, n \end{cases} \tag{1}$$

where $\vec{x} = (x_1, \dots, x_n)$ is an n-dimensional decision vector defined in the decision space S, x_k ($k \in \{1, \dots, n\}$) is the kth decision variable, x_k^{low} and x_k^{upp} are the lower and upper bounds of x_k, respectively, $\vec{F}$ is an objective vector consisting of m conflicting objectives, $g_j(\vec{x})$ is the jth inequality constraint, $h_j(\vec{x})$ is the $(j - n_g)$th equality constraint, and n_g and n_h are the numbers of inequality and equality constraints, respectively.

This work was supported by the National Natural Science Foundation of China under Grants 61976225 and 61673397. (*Corresponding author: Yong Wang*)

The authors are with the School of Automation, Central South University, Changsha 410083, China. (Email: mzw_cemo@csu.edu.cn; ywang@csu.edu.cn)

Generally, the constraint violation of $\vec{x}$ on the jth constraint is defined as:

$$CV_j(\vec{x}) = \begin{cases} \max(0, g_j(\vec{x})), & j = 1, \dots, n_g \\ \max(0, |h_j(\vec{x})| - \delta), & j = n_g + 1, \dots, n_g + n_h \end{cases} \tag{2}$$

where δ is a small tolerance value to slightly relax equality constraints. $\vec{x}$ is called a feasible solution if its total constraint violation, i.e.,

$$CV(\vec{x}) = \sum_{j=1}^{n_g+n_h} CV_j(\vec{x}) \tag{3}$$

is equal to 0; otherwise, $\vec{x}$ is called an infeasible solution. Thus, the feasible region is defined as

$$\Omega = \{\vec{x} \in S \mid CV(\vec{x}) = 0\}. \tag{4}$$

If a feasible solution is not dominated by any other feasible solution, it is called a Pareto optimal solution of a CMOP. The set of Pareto optimal solutions is called the Pareto optimal set. The image of the Pareto optimal set in the objective space is called constrained Pareto front (PF).

It is well known that many optimization problems in the real world are CMOPs by nature [6]–[9]. Evolutionary algorithms (EAs) are a kind of powerful population-based optimization algorithms. However, existing multiobjective EAs (MOEAs) are mainly developed to deal with unconstrained MOPs, that is, they lack key mechanisms to handle constraints [10]. Therefore, many constraint-handling techniques (CHTs) are embedded into MOEAs, enabling them to solve CMOPs [11].

Current CHTs can be roughly classified into three kinds: methods based on penalty functions [12], methods based on the separation of objectives and constraints [13], and methods based on multiobjective optimization [14]. However, current CHTs usually lack a powerful capability to guide the population toward the feasible region with good diversity in the early stage of evolution and to maintain sufficient feasible solutions with good diversity and convergence in the later stage of evolution. Moreover, their capability to provide a transition from both diversity and feasibility in the early stage of evolution to both diversity and convergence in the later stage of evolution is also limited.

Motivated by the above consideration, a shift-based penalty function, called ShiP, is designed in this paper. During the course of evolution, infeasible solutions are shifted by taking their *local feasible nadirs* as reference positions, and the degree of shift is adaptively controlled by the feasibility proportion of the current parent and offspring populations. Afterward, these infeasible solutions are further penalized based on their constraint violations. The main contributions of this paper can be summarized as follows:

- In principle, ShiP is an adaptive penalty function and the similar idea (i.e., the shift by using local feasible nadirs) has not appeared previously. ShiP considers the distributions of feasible solutions in the objective space, with the aim of supporting promising infeasible solutions to approach/enter the feasible region from different directions. If enough feasible solutions have been found, the shift and penalty will severely decrease the fitness of infeasible solutions, thus giving more chances to diverse feasible solutions to converge toward the Pareto optimal solutions. Moreover, based on the feasibility proportion, ShiP provides an adaptive transition from both diversity and feasibility in the early stage of evolution to both diversity and convergence in the later stage of evolution. Therefore, ShiP has the potential to alleviate the aforementioned issues in current CHTs.
- The implementation of ShiP is simple and flexible. We have embedded ShiP into three well-known multiobjective optimization frameworks: NSGA-II [13], MOEA/D [15], and IBEA [16]. They are representatives of MOEAs based on Pareto dominance, decomposition, and indicator, respectively [17]. ShiP does not add any significant computational burden to them. We have also investigated its effectiveness by comparing it with other CHTs on these frameworks.
- The extension and application of ShiP are discussed. On one hand, ShiP has been extended by adding an archive-assisted strategy. In this strategy, an archive is updated in a way that is complementary to ShiP, making the archive and the population evolve cooperatively. Experiments have validated that the archive-assisted ShiP outperforms peer algorithms on highly-constrained benchmark cases. On the other hand, we have also applied ShiP to a real-world CMOP: the vehicle scheduling of urban bus line. On this real-world case, ShiP exhibits good performance.

The rest of this paper is organized as follows. Section II introduces the related work. Section III presents the details of ShiP. The experimental studies of ShiP and its archive-assisted variant are carried out in Section IV. Then, the vehicle scheduling problem is solved by ShiP in Section V. Finally, Section VI concludes this paper.

II. Related Work

Next, we briefly overview the three kinds of CHTs for CMOPs in the EA community.

A. Methods Based on Penalty Functions

As traditional penalty functions always require a careful tuning of penalty factors [18], some self-adaptive penalty functions have been developed. Woldesenbet *et al.* [12] modified the objective in each dimension to the sum of a distance measure and a penalty function:

$$f_i'(\vec{x}) = d_i(\vec{x}) + p_i(\vec{x}). \tag{5}$$

For the ith objective, $d_i(\vec{x})$, i.e., the distance measure, is defined as follows:

$$d_i(\vec{x}) = \begin{cases} \tilde{CV}(\vec{x}), & \text{if } P_{fea} = 0 \\ \sqrt{\tilde{f}_i^2(\vec{x}) + \tilde{CV}^2(\vec{x})}, & \text{otherwise} \end{cases} \tag{6}$$

and $p_i(\vec{x})$, i.e., the penalty function, is formulated as follows:

$$p_i(\vec{x}) = (1 - P_{fea})X_i(\vec{x}) + P_{fea}Y_i(\vec{x}) \tag{7}$$

where P_{fea} is the proportion of feasible solutions in the population; $\tilde{f}_i$ and $\tilde{CV}$ are the normalized objective in the ith dimension and normalized constraint violation of $\vec{x}$, respectively; $X_i(\vec{x})$ is equal to 0 if $P_{fea} = 0$, otherwise $X_i(\vec{x}) = \tilde{CV}(\vec{x})$; and $Y_i(\vec{x})$ is equal to 0 if $\tilde{CV}(\vec{x}) = 0$, otherwise $Y_i(\vec{x}) = \tilde{f}_i(\vec{x})$.

Jiao *et al.* [19] employed the similar idea in their design. The modified objective in the ith dimension is formulated as follows:

$$f_i'(\vec{x}) = \begin{cases} \sqrt{\tilde{f}_i^2(\vec{x}) + \tilde{CV}^2(\vec{x})}, & \text{if } P_{fea} = 0 \\ P_{fea}\tilde{f}_i(\vec{x}) + (1 - P_{fea})\tilde{CV}(\vec{x}), & \text{otherwise} \end{cases} \tag{8}$$

where the meaning of P_{fea}, $\tilde{f}_i$, and $\tilde{CV}$ is the same as in [12].

Jan and Zhang [20] developed a penalty function in MOEA/D. The penalty is added to the Tchebycheff aggregation function (denoted as g^{te}) as follows:

$$g_{ap}^{te}(x|\lambda, o^*) = \begin{cases} g^{te}(x|\lambda, o^*) + s_1 CV^2(\vec{x}), & \text{if } CV(\vec{x}) < \tau \\ g^{te}(x|\lambda, o^*) + s_1\tau^2 + s_2(CV(\vec{x}) - \tau), & \text{otherwise} \end{cases} \tag{9}$$

where s_1 and s_2 are two control parameters, $s_1 \ll s_2$, and o^* is a reference point. In this manner, the penalty on an infeasible solution would increase promptly if its constraint violation is greater than a threshold, i.e., τ.

In [21], Jan *et al.* further improved this method by eliminating s_1 and s_2:

$$f_i^{ap}(\vec{x}) = \begin{cases} f_i(\vec{x}) + P_{inf}^2 CV^2(\vec{x}), & \text{if } CV(\vec{x}) < \tau \\ f_i(\vec{x}) + P_{inf}^2\tau^2 + P_{inf}(CV(\vec{x}) - \tau), & \text{otherwise} \end{cases} \tag{10}$$

where P_{inf} is the proportion of infeasible solutions in the population and τ is equal to the average constraint violation of all infeasible solutions.

B. Methods Based on the Separation of Objectives and Constraints

Some CHTs compare objectives and constraints separately. CDP [13] is the simplest CHT that belongs to this kind. It uses the following three rules in the pair-wise comparison:

- When comparing two feasible solutions, the one that Pareto dominates the other is selected;
- When comparing a feasible solution with an infeasible solution, the feasible one is selected;
- When comparing two infeasible solutions, the one with smaller CV is selected.

Obviously, CDP always prefers feasible solutions to infeasible ones. Due to its simple structure, it has been widely used to solve CMOPs. Wei and Wang [22] used CDP in infeasible elitist-based particle swarm optimization. Liu and Wang [23] combined CDP with MOEA/D [15], where each subproblem has a temporary register composed of the infeasible solutions with lower constraint violations and better objective values. Fan *et al.* [24] utilized the angle information among solutions to adjust the dominance relationship, and designed an angle-based CDP in MOEA/D.

Takahama and Sakai [25] proposed ϵ constrained method, where a decreasing ϵ-level is designed to relax the constraint violations of infeasible solutions. Given two solutions $\vec{a}$ and $\vec{b}$, $\vec{a}$ is better than $\vec{b}$, if one of the following conditions meets:

- both $CV(\vec{a})$ and $CV(\vec{b})$ are smaller than ϵ, and $\vec{a}$ Pareto dominates $\vec{b}$;
- $CV(\vec{a})$ is smaller than ϵ, while $CV(\vec{b})$ is greater than ϵ;
- both $CV(\vec{a})$ and $CV(\vec{b})$ are greater than ϵ, and $CV(\vec{a}) < CV(\vec{b})$.

If ϵ decreases to 0, ϵ constrained method is equivalent to CDP [26]. Yang *et al.* [27] incorporated ϵ constrained method and an adaptive operator selection into MOEA/D. Fan *et al.* [28] proposed an improved version that adjusts the ϵ-level dynamically based on the ratio of feasible solutions. Recently, Fan *et al.* designed a push and pull strategy, where the ϵ-level is controlled in a more elaborate manner [29].

Runarsson and Yao [30] designed a stochastic ranking method. In this method, when comparing two solutions, parameter P_f is used to introduce some information of objective. To be specific, two solutions are compared based on objective values with the probability of P_f, while they are compared based on constraint violations with the probability of $(1 - P_f)$. Geng *et al.* [31] combined the stochastic ranking method with an infeasible elitist-based MOEA, where the original objectives are transformed into a scalar value based on nondomination level and crowding distance. In addition, Jan and Khanum [32] embedded the stochastic ranking method into MOEA/D.

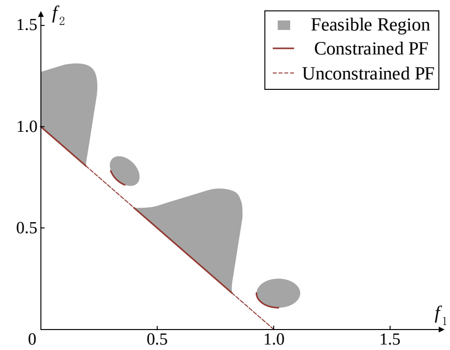
C. Methods Based on Multiobjective Optimization

Methods based on multiobjective optimization balance constraints and objectives via transforming constraints into one or more additional objectives. Based on their early work [33], Ray *et al.* [14] proposed an infeasibility driven EA (called IDEA), which adds a new objective based on a measure of constraint violation. Vieira *et al.* [34] transformed constraints into two objectives: one is based on a penalty function and the other is equal to the number of violated constraints. Long [35] explicitly constructed three new objectives to quantify a solution's convergence, diversity, and feasibility, respectively. Wang *et al.* [36] proposed an adaptive tradeoff model, which divides the search process into three scenarios. If there is no feasible solution, the population is ranked by nondominated sorting with an additional objective defined by constraint violation.

III. Proposed Approach

A. Motivation

When solving CMOPs by EAs, it is necessary to deal with various constraints and optimize multiple conflicting objectives simultaneously, which gives rise to three major challenges in the field of evolutionary constrained multiobjective optimization. The first challenge is how to achieve both diversity and feasibility. Many CMOPs have complicated nonlinear constraints, which generally make the feasible region disjoint in the objective space. A bi-objective example is given in Fig. 1. In the early stage of evolution, there are few feasible solutions in the population. Therefore, it is a critical task to motivate infeasible solutions with good diversity to approach/enter the disjoint feasible parts. The second challenge is how to achieve both diversity and convergence. It is because the ultimate aim is to converge to a representative Pareto optimal set, the image of which is evenly distributed on the constrained PF. In particular, it is important to keep the final population feasible as infeasible solutions will be eliminated in the decision making. Additionally, the third challenge is how to provide a transition from "diversity & feasibility" to "diversity & convergence". It is not difficult to understand since after we obtain diverse feasible solutions, we need to make them become well-converged and well-distritbuted Pareto optimal solutions in the end.

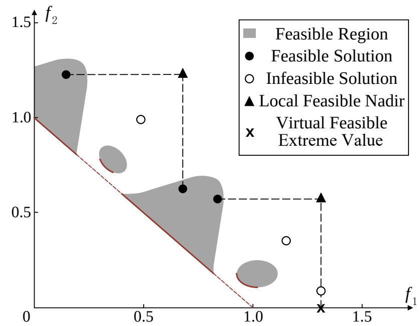


(a)

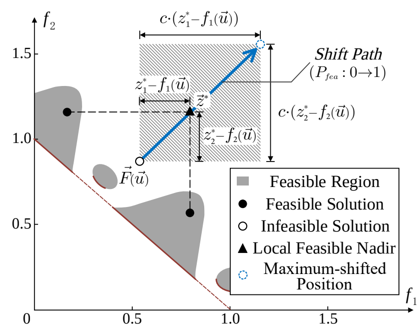
Fig. 1. Illustrations about a CMOP with two objectives and disjoint feasible parts.

However, with respect to methods based on penalty functions, in the early stage of evolution, the solution selection depends largely on the degree of constraint violation. This bias toward constraints will lead to the loss of diversity and premature convergence. In addition, in the later stage of evolution, methods based on penalty functions may maintain some infeasible solutions with slight constraint violations in the population since they generally adopt the feasibility proportion to balance objectives and constraints. Thus, it is hard to achieve both diversity and convergence. Therefore, methods based on penalty functions face the first and second challenges. In terms of methods based on the separation of objectives and constraints and methods based on multiobjective optimization, they face the third challenge due to the fact that they do not use feedback information (such as the feasibility proportion) to guide the evolution.

Based on the above consideration, we design a shift-based penalty function in this paper, called ShiP, which includes a shift measure and a penalty measure.



(a)



(b)

Fig. 2. Illustrations about (a) the calculation of local feasible nadir, and (b) the shift of infeasible solution $\vec{u}$ with $\vec{F}(\vec{u}) = (f_1(\vec{u}), f_2(\vec{u}))$ based on the local feasible nadir (i.e., $\vec{z}^* = (z_1^*, z_2^*)$).

B. Shift-based Penalty (ShiP)

In the shift measure, we propose a new concept, i.e., local feasible nadir. Given infeasible solution $\vec{u}$ and its objective vector $\vec{F}(\vec{u}) = \big(f_1(\vec{u}), f_2(\vec{u}), \dots, f_m(\vec{u})\big)^T$, we first find the feasible solutions whose values in the ith ($i \in \{1, \dots, m\}$) objective are bigger than $f_i(\vec{u})$. Suppose that the number of such feasible solutions is k_i and the corresponding k_i values in the ith objective are: $v_{i,1}, \dots, v_{i,k_i}$. Let $z_i^* = \min\{v_{i,1}, \dots, v_{i,k_i}\}$. The feasible solution, whose value in the ith objective is equal to z_i^*, is called a neighbouring feasible solution of $\vec{u}$. It is clear that there are totally m neighbouring feasible solutions of $\vec{u}$. Then, $\vec{z}^* = (z_1^*, \dots, z_m^*)$ is called the local feasible nadir of $\vec{u}$. Note that if $k_i = 0$, z_i^* is equal to the biggest value in the ith objective of the population, which is called a virtual feasible extreme value in this paper. Still taking Fig. 1 as an example, we show the local feasible nadirs of three infeasible solutions in Fig. 2(a). As shown in Fig. 2(a), some infeasible solutions may share the same neighbouring feasible solutions and local feasible nadir.

Based on $\vec{z}^*$, the shift of $\vec{u}$ on the ith objective is defined as follows:

$$S_i(\vec{u}) = f_i(\vec{u}) + c \cdot P_{fea} \cdot (z_i^* - f_i(\vec{u})) \tag{11}$$

where P_{fea} is the proportion of feasible solutions in the current parent and offspring populations, and c is an arbitrary constant bigger than 1. Fig. 2(b) shows the shift of an infeasible solution based on (11). It can be seen that the degree of shift is controlled by P_{fea}. When $P_{fea} = 1$, the infeasible solution will get the maximum amount of shift. Importantly, the shift measure has the following properties:

Case 1: if $c{\cdot}P_{fea} < 1$ (i.e., $P_{fea} < 1/c$), a shifted infeasible solution still has an opportunity to be nondominated with the neighbouring feasible solutions in the objective space.

Case 2: if $c \cdot P_{fea} \geq 1$ (i.e., $P_{fea} \geq 1/c$), an infeasible solutions will be shifted to the area that is dominated by its neighbouring feasible solutions in the objective space.

c is set to 2 in this paper, that is, the threshold of P_{fea} to trigger the transition between the above two cases is 0.5. The rationality behind this settings is that since P_{fea} is the proportion of feasible solutions in the current parent and offspring populations, $P_{fea} = 0.5$ means that after the environmental selection, the population for the next generation is likely to be entirely feasible.

In addition to the shift measure, a penalty measure is used to further distinguish infeasible solutions. In terms of the ith objective, the penalty of $\vec{u}$ is defined based on its constraint violation:

$$P_i(\vec{u}) = CV(\vec{u}). \tag{12}$$

Finally, the ith objective of each individual in the population is formulated as:

$$f_i'(\vec{x}) = \begin{cases} S_i(\vec{x}) + P_i(\vec{x}), & \text{if } P_{fea} > 0 \wedge CV(\vec{x}) > 0 \\ f_i(\vec{x}), & \text{otherwise} \end{cases} \tag{13}$$

From (13), it is obvious that feasible solutions will not experience any shift and penalty.

C. Analysis of Principle

Next, we analyze the working principle of ShiP based on (13):

1) When the population is entirely infeasible, ShiP only uses the original objectives. By taking advantage of the information of the original objectives, it is helpful to maintain the diversity and explore the search space [29]. One may be interested in why ShiP can find feasible solutions in this manner. The reasons are twofold. First, due to the fact that constraints have been ignored, the population will converge toward the unconstrained PF. During this process, the population will inevitably meet the feasible region as shown in Fig. 1. Second, the mating selection based on constraint violation can help the population to yield some feasible offspring during the evolution.

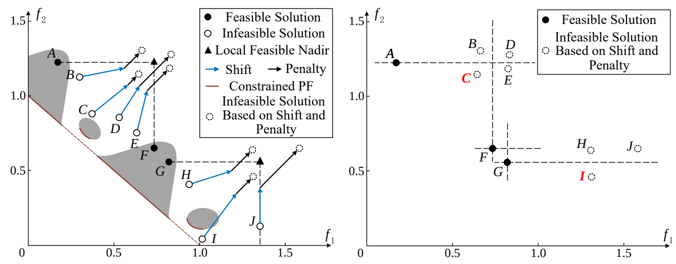
2) When $0 < P_{fea} < 1/c$, by utilizing the distribution information of feasible solutions, ShiP is able to identify promising infeasible solutions in the areas defined by the neighbouring feasible solutions and local feasible nadirs. To illustrate this, we present a hypothetical population and simulate the shift and penalty measures. In this scenario, there are ten solutions denoted as A–J. Among them, three are feasible solutions (i.e., A, F, and G) and seven are infeasible solutions (i.e., B, C, D, E, H, I, and J). Thus, $P_{fea} = 0.3$. Table I summarizes their original objective values and constraint violations and Fig. 3(a) shows their original positions in the objective space. Obviously, B, C, D, and E share the same neighbouring feasible solutions and local feasible nadir, and H, I, and J share the same neighbouring feasible solutions and local feasible nadir. From Fig. 3(a), C and I are more

TABLE I
SIMULATION OF THE SHIFT AND PENALTY MEASURES WITH $P_{fea} = 0.3$ AND $c = 2$

Solution	Original Objectives (f_1, f_2)	Local Feasible Nadir (z_1, z_2)	Shift Amount $\langle 0.6(z_1^* - f_1), 0.6(z_2^* - f_2)\rangle$	Penalty Amount $\langle CV, CV\rangle$	Modified Objectives (f_1', f_2')
A	(0.172, 1.224)	–	–	–	–
B	(0.299, 1.125)	(0.733, 1.224)	$\langle 0.261, 0.059\rangle$	$\langle 0.108, 0.108\rangle$	(0.668, 1.293)
C	(0.362, 0.878)	(0.733, 1.224)	$\langle 0.223, 0.207\rangle$	$\langle 0.045, 0.045\rangle$	(0.630, 1.130)
D	(0.528, 0.875)	(0.733, 1.224)	$\langle 0.123, 0.209\rangle$	$\langle 0.187, 0.187\rangle$	(0.838, 1.271)
E	(0.632, 0.753)	(0.733, 1.224)	$\langle 0.061, 0.283\rangle$	$\langle 0.134, 0.134\rangle$	(0.827, 1.170)
F	(0.733, 0.651)	–	–	–	–
G	(0.819, 0.558)	–	–	–	–
H	(0.937, 0.407)	(1.351, 0.558)	$\langle 0.248, 0.090\rangle$	$\langle 0.129, 0.129\rangle$	(1.314, 0.626)
I	(1.014, 0.041)	(1.351, 0.558)	$\langle 0.202, 0.310\rangle$	$\langle 0.101, 0.101\rangle$	(1.317, 0.452)
J	(1.351, 0.128)	(1.351, 0.558)	$\langle 0.000, 0.258\rangle$	$\langle 0.245, 0.245\rangle$	(1.596, 0.631)

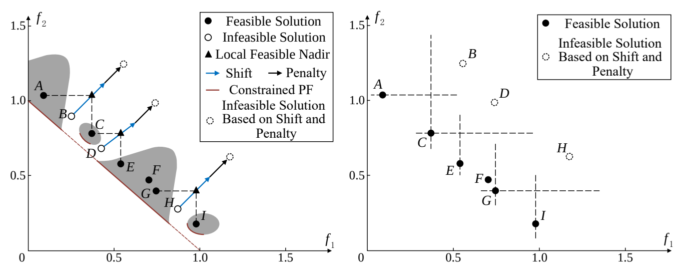
promising than other infeasible solutions: C has the potential to enter a disjoint feasible part, and I is very close to the Pareto optimal solutions on the boundary of a disjoint feasible part. On the contrary, B, E, and H are less valuable for exploration because they are close to the current feasible solutions, and D and J are far away from the feasible region. Based on P_{fea} and c, Table I also gives the shift amount, penalty amount, and modified objectives of each infeasible solution, and Fig. 3(a) also shows the modified positions of the ten individuals by ShiP in the objective space, which are further depicted in Fig. 3(b) for clarity. As can be seen from Fig. 3(a), B, D, E, H, and J are outside the areas defined by their neighbouring feasible solutions and local feasible nadirs after shift and penalty. It is because the shifted B, E, and H are close to the boundaries of the areas defined by their neighbouring feasible solutions and local feasible nadirs. Thus, after penalty (the penalty angle is 45° since the penalty amount in each objective is the same), the shifted B, E, and H are outside. In addition, the reason why the shifted D and J are outside after penalty is mainly due to their high constraint violations. In contrast, C and I are inside, which can be attributed to the fact that they have low constraint violations and are far away from the current feasible solutions. From Fig. 3(a), based on Pareto dominance, the seven infeasible solutions are nondominated with the three feasible solutions, in terms of the original objectives. However, from Fig. 3(b), only C and I are nondominated with the three feasible solutions, and the other infeasible solutions are dominated by A, F, or G, in terms of the modified objectives. Thus, as promising infeasible solutions, C and I are very likely to survive during the evolution. By making use of such promising infeasible solutions, the population can gradually approach the boundary of the feasible region or enter the feasible region while maintaining good diversity, thus achieving both diversity and feasibility.

3) When $P_{fea} \geq 1/c$, after shift, all infeasible solutions are dominated by feasible solutions as analyzed in Case 2 of Section III-B. Further, after penalty, all infeasible solutions are definitely outside the areas defined by their neighbouring feasible solutions and local feasible nadirs since their constraint violations are bigger than 0. An example is given in Fig. 4. In this example, there are ten hypothetical solutions. Among them, three are infeasible solution and seven are feasible solutions. Thus, $P_{fea} = 0.7$. In principle, ShiP with $P_{fea} \geq 1/c$



(a) (b)

Fig. 3. Distribution of ten hypothetical solutions A–I in the case that $0 < P_{fea} < 1/c$. (a) Their original positions in the objective space. (b) Their modified positions by ShiP in the objective space.



(a) (b)

Fig. 4. Distribution of ten hypothetical solutions A–I in the case that $P_{fea} \geq 1/c$. (a) Their original positions in the objective space. (b) Their modified positions by ShiP in the objective space.

is equivalent to CDP, therefore giving more opportunities to feasible solutions to evolve. As a result, based on the diverse feasible solutions provided by $0 < P_{fea} < 1/c$, ShiP can motivate them toward the Pareto optimal solutions, thus achieving both diversity and convergence.

From the above analysis, ShiP is capable of achieving both diversity and feasibility in the early stage of evolution, and both diversity and convergence in the later stage of evolution. In addition, ShiP achieves an adaptive transition from "diversity & feasibility" to "diversity & convergence" by using P_{fea}. Therefore, ShiP provides an effective way to address the three major challenges mentioned previously. Moreover, the implementation of ShiP is simple.

The above principle analysis of ShiP is based on Pareto dominance. Under the framework of MOEA/D, for a certain subproblem, a dominated solution can be identified by the aggregation function (e.g., Tchebycheff approach). In addition, IBEA with $I_{\epsilon+}$ indicator has been proven to be compliant with Pareto dominance. Therefore, ShiP is also applicable to the decomposition and indicator-based MOEA frameworks. A brief analysis can prove that the computational time complexity of ShiP is $O(mN^2)$, where N is the population size. Thus, ShiP does not bring extra computational burden to existing MOEAs.

Remark 1: Despite both ShiP and Woldesenbet *et al.*'s method [12] have two components in the modified objectives and utilize the feasibility proportion, they are different in nature. In ShiP, the shift measure is based on neighbouring feasible solutions and local feasible nadirs; thus, the local information of population distribution is utilized. Therefore, ShiP can adapt more complicated fitness landscapes. Overall, ShiP is able to balance feasibility, diversity, and convergence during the evolution. However, in Woldesenbet *et al.*'s method, if the feasibility proportion is small, priority is given to the search for feasible individuals, and if the feasibility proportion is large, priority is given to find individuals with better objective values. As a result, it is hard to keep the diversity of the population during the evolution due to the fact that this method does not make use of the local information of population distribution, especially when there are several disjoint feasible parts.

IV. Experimental Study

This section experimentally studies the performance of ShiP by embedding it into three well-known MOEA frameworks: NSGA-II [13], MOEA/D [15], and IBEA [16]. From (13), ShiP revises each objective and transforms a CMOP into an unconstrained MOP. Therefore, ShiP can be directly implemented under the framework of NSGA-II, MOEA/D, and IBEA to solve CMOPs. In our experiments, MW [37] was adopted as the test suite. In MW, CMOPs are classified into four types: the constrained PF is the same with the unconstrained PF (type I), the constrained PF is a part of the unconstrained PF (type II), the constrained PF consists of a part of the unconstrained PF and a part of the boundary of the feasible region (type III), and the unconstrained PF is entirely located outside the feasible region and the constrained PF is composed of a part of the boundary of the feasible region (type IV). For the sake of clarity, the discussions about the results in this section were based on the above classification. Under each MOEA framework, we compared ShiP with some representative CHTs:

- Under the framework of NSGA-II, the compared CHTs included CDP [13], self-adaptive penalty function (SP) [12], the multiobjective optimization-based method (MO) proposed in IDEA [14], and adaptive tradeoff model (ATM) [36]. Their corresponding constrained MOEAs (CMOEAs) were denoted as CDP-NSGA-II, SP-NSGA-II, MO-NSGA-II, and ATM-NSGA-II, respectively.
- Under the framework of MOEA/D, in addition to CDP [32] and SP [20], we selected stochastic ranking (SR) [32] and an improved ϵ constrained method proposed in [29]. Their corresponding CMOEAs were denoted as CDP-MOEA/D, SP-MOEA/D, SR-MOEA/D, and ϵ-MOEA/D, respectively.
- Currently, very few research works have focused on CHTs under the framework of IBEA or other indicator-based MOEAs. We considered a combination of SR and IBEA presented in [38], denoted as SR-IBEA.

A. Performance Metrics

To assess the performance of the compared CMOEAs, a commonly used metrics was chosen in this paper: inverted generational distance (IGD) [39]. Note that, IGD is able to measure the diversity and convergence of an obtained approximation $\mathcal{P}$. Assuming that $\mathcal{P}^*$ is a set of samples evenly distributed on the constrained PF, the IGD metric is calculated as:

$$IGD = \frac{\sum_{\vec{a}^* \in \mathcal{P}^*} d(\vec{a}^*, \mathcal{P})}{|\mathcal{P}^*|} \tag{14}$$

where $|\mathcal{P}^*|$ is the cardinality of $\mathcal{P}^*$, and $d(\vec{a}^*, \mathcal{P})$ is the minimal Euclidean distance between $\vec{a}^*$ and all members in $\mathcal{P}$. A smaller IGD value indicates a better approximation of the constrained PF.

B. Parameter Settings

Our experiments were conducted under the following parameter settings:

- Number of independent runs: 100;
- Maximum generation number: $G_{max} = 600$;
- Population size: $N = 100$;
- Number of decision variables: $n = 15$;
- Number of objectives for the test problems with scalable numbers of objectives: $m = 3$.

All the compared CMOEAs used simulated binary crossover (SBX) [40] and polynomial mutation (PM) [13] to produce the offspring:

- SBX: the crossover probability $p_c = 0.9$ and the distribution index $\eta_c = 20$;
- PM: the mutation probability $p_m = 1/n$ and the distribution index $\eta_m = 20$.

In addition, on each test problem, the results included the average and standard deviation over 100 independent runs. To detect the statistical significance, the Wilcoxon's rank sum test at a 0.05 significance level was implemented between ShiP and other CHTs.

C. Comparison under the Framework of NSGA-II

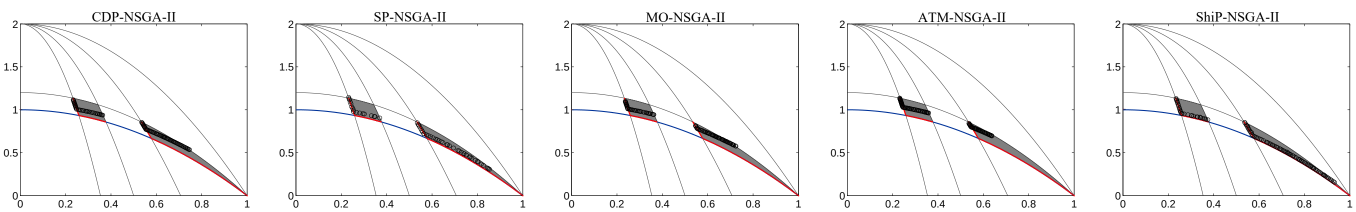
First, ShiP was compared with four other CHTs, i.e., CDP, SP, MO, and ATM, on MW test suite under the NSGA-II framework. Table II show the results in terms of IGD.

It can be seen from Table II that ShiP-NSGA-II performs the best on eight cases, i.e., MW1, MW5–MW7, MW10, and MW12–MW14, which are mainly type-II and type-III CMOPs. For the rest of cases, ATM-NSGA-II is better than others on MW2 and MW4 which are type-I CMOPs; MO-NSGA-II achieves more promising results than others on MW8, MW9, and MW11 which mainly belong to type IV; and SP-NSGA-II wins on MW3. The Wilcoxon's rank-sum test shows that ShiP is significantly better than the four competitors on seven, six,

TABLE II

STATISTICS OF THE IGD METRIC OBTAINED BY FIVE CHTS UNDER THE FRAMEWORK OF NSGA-II, WHERE "AVG" AND "STD DEV" ARE THE AVERAGE AND STANDARD DEVIATION OF THE IGD VALUES OVER 100 INDEPENDENT RUNS, RESPECTIVELY. THE BEST RESULT FOR EACH TEST PROBLEM AMONG THE FIVE COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED, WHERE "+", "≈", AND "−" DENOTE THAT SHIP PERFORMS BETTER THAN, SIMILAR TO, AND WORSE THAN ITS COMPETITORS, RESPECTIVELY.

Type	Prob.	CDP-NSGA-II Avg(±Std Dev)	SP-NSGA-II Avg (±Std Dev)	MO-NSGA-II Avg (±Std Dev)	ATM-NSGA-II Avg (±Std Dev)	ShiP-NSGA-II Avg (±Std Dev)
I	MW2	2.841E-02 (±1.32E-02)≈	2.482E-02 (±1.28E-02)≈	2.619E-02 (±1.02E-02)≈	**2.467E-02 (±1.47E-02)**≈	2.667E-02 (±1.47E-02)
	MW4	5.644E-02 (±2.75E-03)−	6.220E-02 (±5.69E-02)+	5.659E-02 (±2.29E-03)−	**5.639E-02 (±2.62E-03)**−	5.823E-02 (±2.63E-03)
	MW14	1.388E-01 (±1.76E-02)≈	1.386E-01 (±2.31E-02)≈	1.388E-01 (±2.60E-02)+	1.387E-01 (±8.28E-03)+	**1.386E-01 (±1.11E-02)**
II	MW1	3.011E-02 (±8.32E-02)≈	1.781E-02 (±4.29E-02)≈	4.145E-02 (±8.92E-02)+	1.937E-02 (±4.63E-02)≈	**8.055E-03 (±1.11E-02)**
	MW5	2.878E-01 (±3.03E-01)+	2.153E-01 (±2.95E-01)+	5.687E-02 (±7.00E-02)≈	2.639E-01 (±2.90E-01)+	**4.724E-02 (±3.88E-02)**
	MW6	6.432E-02 (±1.06E-01)+	4.552E-02 (±4.43E-02)≈	8.393E-02 (±1.29E-01)+	8.883E-02 (±1.17E-01)+	**4.494E-02 (±9.67E-02)**
	MW8	6.221E-02 (±1.85E-02)≈	6.294E-02 (±2.15E-02)≈	**5.976E-02 (±8.97E-03)**−	6.036E-02 (±5.73E-03)≈	6.027E-02 (±5.53E-03)
III	MW3	1.101E-02 (±2.14E-02)≈	**9.817E-03 (±1.98E-02)**≈	6.432E-02 (±2.00E-01)+	1.747E-02 (±2.13E-02)≈	1.636E-02 (±3.01E-02)
	MW7	3.771E-02 (±1.03E-01)+	8.673E-03 (±1.44E-02)≈	8.769E-03 (±1.67E-02)≈	7.394E-02 (±3.76E-02)+	**8.152E-03 (±1.32E-02)**
	MW10	1.327E-01 (±1.08E-01)+	1.291E-01 (±1.42E-01)+	1.516E-01 (±1.70E-01)+	1.485E-01 (±1.59E-01)+	**5.362E-02 (±6.41E-02)**
	MW13	1.982E-01 (±1.79E-01)+	1.778E-01 (±2.78E-01)+	2.455E-01 (±1.14E-01)+	2.214E-01 (±1.22E-01)+	**1.522E-01 (±8.41E-02)**
IV	MW9	1.256E-01 (±2.19E-01)−	1.630E-01 (±2.63E-01)−	**2.608E-02 (±8.31E-02)**−	1.793E-01 (±2.84E-01)−	4.210E-01 (±1.15E-01)
	MW11	5.185E-01 (±1.75E-01)+	4.615E-01 (±2.29E-01)+	**1.721E-01 (±1.91E-01)**≈	5.750E-01 (±1.89E-01)+	1.733E-01 (±2.04E-01)
	MW12	1.501E-01 (±2.49E-01)+	7.048E-02 (±1.90E-01)+	4.014E-02 (±1.33E-01)+	8.735E-02 (±2.37E-01)+	**2.421E-02 (±8.84E-02)**
$+/\approx/-$		7/5/2	6/7/1	7/4/3	8/4/2	/



(a) (b) (c) (d) (e)

Fig. 5. The solution sets obtained by the five compared algorithms with the median IGD value among 100 independent runs on MW10. The blue curve denotes the unconstrained PF and the red curves denote the constrained PF. (a) CDP-NSGA-II. (b) SP-NSGA-II. (c) MO-NSGA-II. (d) ATM-NSGA-II. (e) ShiP-NSGA-II

seven, and eight cases, respectively; while ShiP is worse than them on only two, one, three, and two cases, respectively.

It is clear that ShiP has advantages on type-II and type-III CMOPs and is also competitive on type IV. The following two reasons support the above observations: 1) on types III and IV, ShiP can keep some valuable infeasible solutions close to the boundary of the feasible region; and 2) on type II, when $P_{fea} \geq 1/c$, ShiP encourages diverse feasible solutions to approximate the constrained PF and eliminates infeasible solutions. Fig. 5 plots the feasible solutions resulting from the five compared algorithms with the median IGD value among 100 independent runs on MW10.

D. Comparison under the Framework of MOEA/D

Subsequently, ShiP-MOEA/D was compared with four other CMOEAs (i.e., CDP-MOEA/D, SP-MOEA/D, SR-MOEA/D, and ϵ-MOEA/D) on MW test suite. Table III presents the results in terms of IGD.

It can be observed from Table III that ShiP-MOEA/D is the best algorithm on seven test problems (i.e., MW2, MW5–MW8, MW13, and MW14) that cover types I–III. In addition, CDP-MOEA/D performs better than other algorithms on MW4, MW9, and MW10; ϵ-MOEA/D achieves better approximations on MW1, MW3, and MW11; and SP-MOEA/D beats others on MW12. The Wilcoxon's rank-sum test confirms that ShiP-MOEA/D is better than the four competitors on eight, nine, ten, and five test problems, respectively. However, ShiP-MOEA/D is worse than them on only four, three, one, four test problems, respectively.

In summary, under the framework of MOEA/D, ShiP presents quite encouraging results on types I–III. However, ShiP seems to be slightly worse than CDP and SP on type-IV cases (especially, on MW9 and MW12). The reason can be attributed to the fact that MW9 and MW12 have connected feasible regions. The weights adopted in MOEA/D can well cover these feasible regions as the evolution proceeds. Under this condition, CDP and SP, which highly bias toward feasibility, will motivate the population to promptly extend along the connected feasible region and approximate the constrained PF from the feasible side. Fig. 6 plots the feasible solutions resulting from the five compared algorithms with the median IGD value among 100 independent runs on MW7.

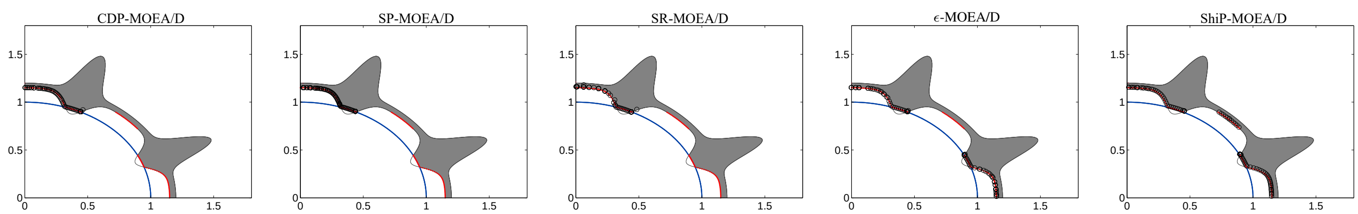
E. Comparison under the Framework of IBEA

Finally, we compared ShiP with SR on MW test suite by embedding them into IBEA. Table IV provides the results in terms of IGD obtained by SR-IBEA and ShiP-IBEA. It can be seen from Table IV that ShiP-IBEA wins on MW1, MW3, MW5, MW7, and MW9–MW12 which are mainly types II–IV; while on the remaining test problems, SR-IBEA performs better. The Wilcoxon's rank-sum test also reveals that ShiP-

TABLE III

STATISTICS OF THE IGD METRIC OBTAINED BY FIVE CHTS UNDER THE FRAMEWORK OF MOEA/D, WHERE "AVG" AND "STD DEV" ARE THE AVERAGE AND STANDARD DEVIATION OF THE IGD VALUES OVER 100 INDEPENDENT RUNS, RESPECTIVELY. THE BEST RESULT FOR EACH TEST PROBLEM AMONG THE FIVE COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED, WHERE "+", "≈", AND "−" DENOTE THAT SHIP PERFORMS BETTER THAN, SIMILAR TO, AND WORSE THAN ITS COMPETITORS, RESPECTIVELY.

Type	Prob.	CDP-MOEA/D Avg(±Std Dev)	SP-MOEA/D Avg (±Std Dev)	SR-MOEA/D Avg (±Std Dev)	ϵ-MOEA/D Avg (±Std Dev)	ShiP-MOEA/D Avg (±Std Dev)
I	MW2	1.091E-01 (±8.89E-02)≈	1.094E-01 (±9.92E-02)≈	1.086E-01 (±7.58E-02)≈	1.122E-01 (±8.66E-02)≈	**1.031E-01 (±7.50E-02)**
	MW4	**4.915E-02 (±5.91E-04)**−	4.964E-02 (±5.97E-04)−	5.075E-02 (±1.21E-03)+	4.927E-02 (±8.30E-04)−	5.047E-02 (±5.15E-03)
	MW14	3.478E-01 (±1.14E-01)≈	3.832E-01 (±1.26E-01)+	3.702E-01 (±1.03E-01)+	3.335E-01 (±9.66E-02)≈	**3.327E-01 (±1.06E-01)**
II	MW1	3.864E-02 (±7.02E-02)+	3.477E-02 (±7.31E-02)+	2.837E-02 (±9.03E-02)≈	**1.293E-02 (±3.27E-02)**−	1.430E-02 (±4.09E-02)
	MW5	6.018E-01 (±2.89E-01)+	6.429E-01 (±2.05E-01)+	6.433E-01 (±2.10E-01)+	1.662E-01 (±1.45E-01)+	**1.481E-01 (±2.13E-01)**
	MW6	3.606E-01 (±2.54E-01)+	3.318E-01 (±2.11E-01)+	3.462E-01 (±2.46E-01)+	3.146E-01 (±2.09E-01)≈	**2.829E-01 (±2.21E-01)**
	MW8	1.316E-01 (±8.57E-02)+	1.526E-01 (±1.10E-01)+	1.545E-01 (±1.00E-01)+	1.295E-01 (±5.52E-02)+	**1.082E-01 (±5.11E-02)**
III	MW3	3.598E-01 (±4.30E-01)+	4.226E-01 (±4.57E-01)+	9.009E-02 (±1.92E-01)−	**1.076E-02 (±1.90E-03)**−	1.475E-01 (±3.26E-01)
	MW7	2.797E-01 (±1.97E-01)+	2.692E-01 (±1.87E-01)+	3.165E-01 (±1.50E-01)+	8.750E-02 (±9.89E-02)+	**8.327E-02 (±1.43E-01)**
	MW10	**2.201E-01 (±1.92E-01)**−	2.560E-01 (±1.91E-01)≈	2.719E-01 (±1.83E-01)≈	2.464E-01 (±2.04E-01)≈	2.651E-01 (±2.24E-01)
	MW13	2.756E-01 (±3.53E-01)+	2.362E-01 (±3.17E-01)+	4.996E-01 (±1.37E-01)+	2.800E-01 (±1.13E-01)+	**1.670E-01 (±5.01E-02)**
IV	MW9	**1.216E-02 (±3.50E-03)**−	1.277E-01 (±1.78E-02)−	2.396E-01 (±1.07E-01)+	7.065E-02 (±7.46E-02)−	1.301E-01 (±1.41E-01)
	MW11	3.929E-01 (±1.85E-01)+	4.283E-01 (±2.55E-01)+	5.799E-01 (±3.23E-01)+	**1.378E-01 (±2.04E-01)**≈	1.991E-01 (±1.88E-01)
	MW12	5.521E-03 (±3.64E-04)−	**5.398E-03 (±3.54E-04)**−	1.389E-01 (±6.72E-02)+	1.357E-02 (±8.65E-03)+	8.967E-03 (±1.01E-03)
+/≈/−		8/2/4	9/2/3	10/3/1	5/5/4	/



(a) (b) (c) (d) (e)

Fig. 6. The solution sets obtained by the five compared algorithms with the median IGD value among 100 independent runs on MW7. The blue curve denotes the unconstrained PF and the red curves denote the constrained PF. (a) CDP-MOEA/D. (b) SP-MOEA/D. (c) SR-MOEA/D. (d) ε-MOEA/D. (e) ShiP-MOEA/D.

TABLE IV

STATISTICS OF THE IGD METRIC OBTAINED BY SR AND SHIP UNDER THE FRAMEWORK OF IBEA, WHERE "AVG" AND "STD DEV" ARE THE AVERAGE AND STANDARD DEVIATION OF THE IGD VALUES OVER 100 INDEPENDENT RUNS, RESPECTIVELY. THE BETTER RESULT FOR EACH TEST PROBLEM BETWEEN THE TWO COMPARED ALGORITHMS IS HIGHLIGHTED IN BOLDFACE. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PRESENTED, WHERE "+", "≈", AND "−" DENOTE THAT SHIP PERFORMS BETTER THAN, SIMILAR TO, AND WORSE THAN SR, RESPECTIVELY.

Type	Prob.	SR-IBEA Avg (±Std Dev)	ShiP-IBEA Avg (±Std Dev)
I	MW2	**2.354E-02 (±1.01E-02)**≈	2.359E-02 (±7.87E-03)
	MW4	**5.013E-02 (±1.54E-03)**≈	6.515E-02 (±3.31E-03)
	MW14	**1.884E-01 (±5.38E-03)**−	2.977E-01 (±2.09E-02)
II	MW1	2.373E-02 (±4.40E-02)≈	**1.665E-02 (±2.29E-02)**
	MW5	3.091E-01 (±3.46E-01)+	**2.009E-01 (±1.15E-01)**
	MW6	**8.795E-02 (±1.11E-01)**−	1.828E-01 (±1.43E-01)
	MW8	**1.016E-01 (±1.05E-02)**−	1.957E-01 (±3.35E-02)
III	MW3	5.713E-02 (±1.75E-02)≈	**5.430E-02 (±1.81E-02)**
	MW7	1.310E-01 (±8.02E-02)+	**7.419E-02 (±2.10E-02)**
	MW10	1.462E-01 (±1.20E-01)+	**5.430E-02 (±1.41E-02)**
	MW13	**2.515E-01 (±1.29E-01)**−	4.027E-01 (±1.66E-01)
IV	MW9	1.903E-01 (±1.92E-01)+	**1.253E-01 (±9.36E-02)**
	MW11	6.588E-01 (±2.28E-01)+	**4.475E-01 (±2.19E-01)**
	MW12	1.222E-01 (±2.05E-01)+	**8.036E-02 (±1.33E-01)**
+/≈/−		6/4/4	/

Algorithm 1: Framework of ShiP$^+$

Initialize population P and archive A: P and A include N randomly chosen individuals from the decision space, respectively;
$W = \{\mathbf{w}^1, ..., \mathbf{w}^N\}$ is a set of uniformly distributed weight vectors;
while *the stopping criterion is not met* **do**
    M = MatingSelection(P, A, W);
    Produce offspring population Q by executing the reproduction operators on M;
    // *the reproduction operators are the same as in NSGA-II*
    A = ArchiveUpdate(P, Q, A, W);
    P = Environmental Selection(P, Q);
    // *the environmental selection is the same as in ShiP-NSGA-II*
    if *A has at lease one feasible solution* **then**
        P = PopulationUpdate(P, A);
Return P;

IBEA has competitive performance. It is better than and similar to SR-IBEA on six and four test problems, respectively.

Based on the above discussion, ShiP-IBEA is more suitable for MW test problems with types II–IV. However, SR-IBEA is better than ShiP-IBEA on type-I cases, whose constrained PFs are partly in narrow feasible regions. It is because SR always chooses some infeasible solutions with a certain probability,

TABLE V

COMPARISON OF THREE ARCHIVE-ASSISTED METHODS (I.E., PPS, DAE, AND SHIP$^+$) IN TERMS OF THE IGD METRIC ON LIR-CMOP AND DAS-CMOP TEST SUITES. "AVG" AND "STD DEV" ARE THE AVERAGE AND STANDARD DEVIATION OF THE IGD VALUES OVER 50 INDEPENDENT RUNS, RESPECTIVELY. THE BEST RESULT FOR EACH TEST PROBLEM AMONG THE THREE METHODS IS HIGHLIGHTED IN BOLDFACE. THE WILCOXON'S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PRESENTED, WHERE "$+$", "$\approx$", AND "$-$" DENOTE THAT SHIP$^+$ PERFORMS BETTER THAN, SIMILAR TO, AND WORSE THAN ITS COMPETITORS, RESPECTIVELY.

Problem	PPS	DAE	ShiP$^+$
	Avg ($\pm$Std Dev)	Avg ($\pm$Std Dev)	Avg ($\pm$Std Dev)
LIR-CMOP1	1.4884e-1 ($\pm$5.08e-2) $+$	2.1442e-1 ($\pm$1.15e-1) $+$	**1.3177e-2 ($\pm$8.66e-4)**
LIR-CMOP2	1.3699e-1 ($\pm$2.96e-2) $+$	1.8735e-1 ($\pm$1.03e-1) $+$	**1.3456e-2 ($\pm$8.04e-4)**
LIR-CMOP3	2.0167e-1 ($\pm$6.56e-2) $+$	2.8120e-1 ($\pm$1.45e-1) $+$	**2.1669e-2 ($\pm$9.13e-3)**
LIR-CMOP4	1.8786e-1 ($\pm$3.09e-2) $+$	1.9257e-1 ($\pm$5.05e-2) $+$	**2.7904e-2 ($\pm$1.28e-2)**
LIR-CMOP5	2.9991e-1 ($\pm$3.55e-2) $\approx$	3.6495e-1 ($\pm$4.54e-2) $+$	**2.9746e-1 ($\pm$3.26e-2)**
LIR-CMOP6	3.6765e-1 ($\pm$9.38e-2) $\approx$	4.1716e-1 ($\pm$1.10e-1) $+$	**3.4702e-1 ($\pm$8.32e-2)**
LIR-CMOP7	1.2423e-1 ($\pm$2.43e-2) $\approx$	1.6883e-1 ($\pm$3.05e-2) $+$	**1.1404e-1 ($\pm$2.31e-2)**
LIR-CMOP8	1.8410e-1 ($\pm$2.62e-2) $+$	1.9303e-1 ($\pm$2.05e-2) $+$	**1.6922e-1 ($\pm$4.37e-2)**
LIR-CMOP9	4.0142e-1 ($\pm$2.26e-1) $+$	**1.3912e-1 ($\pm$2.31e-1)** $-$	3.9336e-1 ($\pm$3.30e-2)
LIR-CMOP10	1.7486e-1 ($\pm$2.10e-1) $-$	**9.6965e-2 ($\pm$2.22e-1)** $-$	3.9537e-1 ($\pm$4.84e-2)
LIR-CMOP11	1.8991e-1 ($\pm$2.73e-1) $-$	**1.3629e-1 ($\pm$3.76e-1)** $-$	1.9238e-1 ($\pm$7.41e-2)
LIR-CMOP12	2.1996e-1 ($\pm$1.70e-1) $\approx$	**5.7498e-2 ($\pm$1.44e-1)** $-$	2.0528e-1 ($\pm$6.50e-2)
LIR-CMOP13	1.2186e-1 ($\pm$3.60e-3) $+$	9.6800e-2 ($\pm$8.86e-4) $+$	**8.9861e-2 ($\pm$9.29e-4)**
LIR-CMOP14	1.1843e-1 ($\pm$2.76e-3) $+$	**9.9132e-2 ($\pm$7.12e-4)** $\approx$	1.0452e-1 ($\pm$1.64e-3)
DAS-CMOP1	6.9915e-1 ($\pm$6.26e-2) $+$	6.2131e-1 ($\pm$1.38e-1) $+$	**2.6580e-3 ($\pm$7.83e-4)**
DAS-CMOP2	2.4212e-1 ($\pm$3.95e-2) $+$	2.0926e-1 ($\pm$3.68e-2) $+$	**4.0653e-2 ($\pm$2.99e-2)**
DAS-CMOP3	3.2762e-1 ($\pm$4.34e-2) $+$	3.2860e-1 ($\pm$1.04e-1) $+$	**1.6394e-1 ($\pm$1.87e-2)**
DAS-CMOP4	1.4958e-3 ($\pm$7.35e-5) $-$	**1.3024e-3 ($\pm$4.31e-5)** $-$	1.7090e-3 ($\pm$2.24e-4)
DAS-CMOP5	4.3636e-3 ($\pm$1.60e-3) $+$	**3.1800e-3 ($\pm$7.31e-5)** $\approx$	3.2259e-3 ($\pm$1.43e-4)
DAS-CMOP6	2.3203e-2 ($\pm$8.16e-3) $+$	2.3209e-2 ($\pm$1.69e-2) $+$	**1.4342e-2 ($\pm$2.98e-3)**
DAS-CMOP7	5.7109e-2 ($\pm$4.61e-3) $+$	**3.6837e-2 ($\pm$1.85e-3)** $-$	5.1091e-2 ($\pm$1.98e-3)
DAS-CMOP8	7.7258e-2 ($\pm$8.66e-3) $+$	**5.0377e-2 ($\pm$4.17e-3)** $-$	6.0765e-2 ($\pm$3.06e-3)
DAS-CMOP9	4.1735e-1 ($\pm$1.42e-1) $+$	2.7994e-1 ($\pm$1.73e-1) $+$	**9.6791e-2 ($\pm$7.73e-2)**
$+/\approx/-$	16/4/3	14/2/7	

which is useful to approach these narrow feasible regions during the evolution.

Remark 2: Benefiting in part from the emphasis on objectives in the early stage of evolution, ShiP can maintain good population diversity as well as escape from local feasible regions. However, it may inevitably encounter some new issues. Due to its preference to objectives, the population runs the risk of getting trapped into the local optima caused by multimodal features of objectives. One effective way to handle this issue is to use external archive that can preserve some promising solutions far away from the local optima. This motivates us to develop archive-assisted ShiP in the next subsection.

F. ShiP$^+$

Using external archive has been verified to be very effective to solve highly complex constrained optimization problems [41], [42]. On the basis of ShiP-NSGA-II, we proposed an archive-assisted CMOEA, called ShiP$^+$. In each iteration, as shown in **Algorithm 1**, ShiP$^+$ includes four main steps: 1) generate mating pool M based on both archive A and population P; 2) update A, after producing offspring population Q; 3) execute the environmental selection which is the same as in ShiP-NSGA-II; and 4) update P with the nondominated feasible solution in A, if A has at least one feasible solution. Due to the space limitation, the details of these components (except the environmental selection) are listed in the supplementary file. In ShiP, A and P cooperatively evolve by exchanging useful information as follows: 1) in Step 5, Q is produced based on M, the construction of which is based on both A and P in Step 4; 2) Q is used to update both A in Step 6 and P in Step 7; and 3) A is used to update P if A has at least one feasible solution in Steps 8 and 9.

To investigate the performance of ShiP$^+$, we compared it with two other state-of-the-art archive-assisted CMOEAs (i.e., PPS [29] and DAE [42]) on LIR-CMOP1–14 in [29] and DAS-CMOP1–9 in [43] with 30 decision variables. To make the comparison fair, all the algorithms used SBX and PM as the reproduction operators.

Table V provides the results in terms of IGD over 50 independent runs. From Table V, ShiP$^+$ achieves the best results on 14 cases, including LIR-CMOP1–8, LIR-CMOP13, DAS-CMOP1–3, DAS-CMOP6, and DAS-CMOP9. On the remaining nine cases, DAE is better than others. The Wilcoxon's rank sum test confirms that ShiP$^+$ surpasses PPS and DAE on 16 and 14 cases, respectively, and performs worse than them on only three and seven cases, respectively.

The superiority of ShiP$^+$ can be explained as follows. A gives higher priority to feasibility, if there is no feasible solution in P (please see Steps 9-12 in Algorithm 2 of the supplementary file); while the priority switches to optimality, when P already has enough feasible solutions (please see Step 20 in Algorithm 2 of the supplementary file). Thus, A works in a way that is complementary to ShiP. Furthermore, the mating selection and the archive update use the angle information of individuals, aiming at enhancing the ability of exploitation and exploration, respectively.

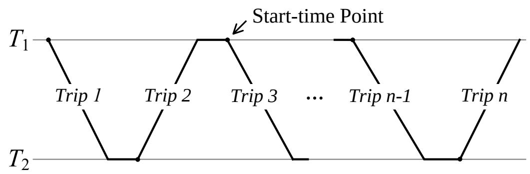


Fig. 7. A block that includes n consecutive trips of a single vehicle.

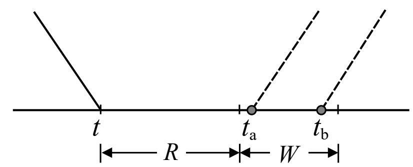


Fig. 8. The start-time points in the range of $[t+R, t+R+W]$ are available for the beginning of the next trip.

G. Parameter Sensitivity Analysis

In (11), the aim of c is to control the transition from "diversity & feasibility" to "diversity & convergence". The sensitivity analysis of c was conducted by setting it to seven different values (i.e., 0.5, 1, 2, 3, 5, 7, and 9) under the framework of NSGA-II. The corresponding variants were denoted as ShiP-1–ShiP-7. Table S-I in the supplementary file presents their results in terms of IGD on MW test suite over 100 independent runs. Note that, as ShiP-3 is equivalent to the original ShiP-NSGA-II, the results of ShiP-3 were directly taken from Table II.

From Table S-I, it can be observed that ShiP-3 (i.e., $c=2$) and ShiP-4 (i.e., $c=3$) perform the best. Compared with them, ShiP-1 (i.e., $c=0.5$) is unable to provide promising results on Types II–IV. This is because ShiP with a small c value fails to put enough emphasis on feasibility. Additionally, ShiP-2 (i.e., $c=1$) show poor performance on type IV, which is further verified by the Wilcoxon's rank-sum test. It can be attributed to two primary reasons: 1) only if P_{fea} is approximately equal to 100%, the feasibility of the population can be guaranteed in ShiP-2; and 2) it is easy for ShiP-2 to generate infeasible solutions on type IV, where the constrained PF is entirely on the boundary of the feasible region. As a result, the final population of ShiP-2 always contains some infeasible solutions. Besides, the above reasons also explain the advantage of ShiP-2 on MW5: in the later stage of evolution, ShiP-2 can keep enough infeasible solutions to approach the discrete constrained PF from the infeasible side.

As the value of c increases, the performance also gradually degenerates. It is because a variant with a greater c value will have less number of generations to utilize valuable infeasible solutions. For ShiP-5 (i.e., $c=5$), ShiP-6 (i.e., $c=7$), and ShiP-7 (i.e., $c=9$), if P_{fea} is greater than 1/5, 1/7, and 1/9, infeasible solutions will lose their advantages in comparison with feasible ones.

Therefore, $c=2$ and $c=3$ are the best choices for general use.

V. Case Study: Vehicle Scheduling of Urban Bus Line

In this section, the effectiveness of ShiP is investigated on a real-world optimization problem: urban bus scheduling problem [44].

The scheduling task considers a single urban bus line with two bus termini (denoted as T_1 and T_2), that is, all the vehicles assigned to the line should commute between T_1 and T_2. Each terminal has a departure timetable that includes nearly 400 start-time points. A trip is a directed route of a vehicle from one terminal to the other, and each trip should begin at a certain start-time point. A block is a set of consecutive trips in the schedule of a single vehicle. Fig. 7 shows a block that includes n trips. The starting time of a block is called the initial start-time point (i.e., the start-time point of a block's first trip). According to the regulations, all blocks should start before 11 a.m. from T_1, and there are 120 initial start-time points in the departure timetable of T_1, denoted as $\boldsymbol{I}=\{I_1,\ldots,I_{120}\}$. The maximum working time of a bus driver is 8 hours, and a short/long block requires one/two drivers, lasting 8/16 hours. According to our investigation, a single trip spends nearly 35 minutes during peak hours (i.e., 6:30–21:00). For other time periods, it is 32 minutes. When a trip is completed, its driver has a 10-minute rest time (R). Thus, the interval between any two adjacent trips in a block should be longer than R.

In practice, to reduce the operation costs and increase their revenue, bus companies usually need to minimize the numbers of vehicles and drivers [44]. Therefore, the urban bus scheduling problem can be formulated as follows:

$$
\begin{aligned}
&\min \begin{cases} f_1 = \sum\limits_{i=1}^{|B|} z_i \\ f_2 = \sum\limits_{i=1}^{|B|} z_i L_i \end{cases} \\
&s.t.\ C_1 : \sum_{i=1}^{|B|} z_i a_{i,j} > 0,\ j=1,\ldots,l \\
&\qquad C_2 : \sum_{i=1}^{|B|} z_i a_{i,j} \le H,\ j=1,\ldots,l \\
&\qquad C_3 : z_i = \{0,1\},\ i=1,\ldots,|B| \\
&\qquad C_4 : a_{i,j} = \{0,1\},\ i=1,\ldots,|B|,\ j=1,\ldots,l \\
&\qquad C_5 : L_i = \{1,2\},\ i=1,\ldots,|B|
\end{aligned}
\tag{15}
$$

where f_1 and f_2 are the numbers of vehicles and drivers required in a scheduling scheme, respectively; B is the candidate set of all blocks; C_1 ensures that each start-time point should be covered; C_2 indicates the number of trips that cover each start-time point should not exceed H; if the ith block is included in the scheduling scheme, $z_i=1$, otherwise $z_i=0$; if the ith block covers the jth start-time point, $a_{i,j}=1$, otherwise $a_{i,j}=0$; if the ith block is a long trip (i.e., requiring two drivers), $L_i=2$, otherwise $L_i=1$; and l is the number of start-time points and equal to 794 in this paper.

In this paper, to solve this scheduling problem, a CMOEA consists of four important components: 1) block initialization; 2) encoding and decoding; 3) genetic operations; and 4) local adjustment procedure. The details are described in the following.

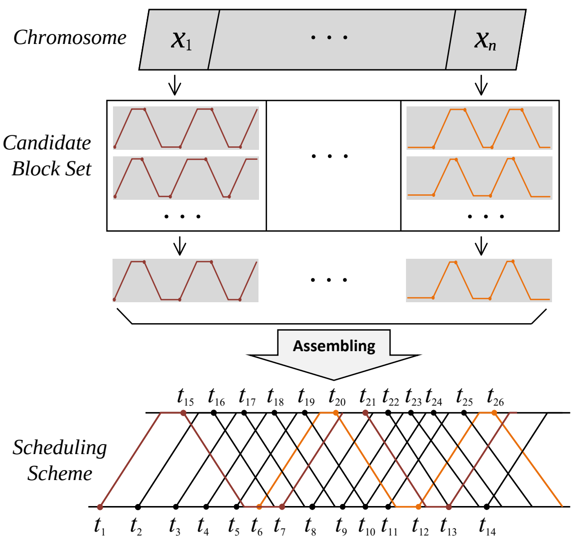


Fig. 9. Decoding process.

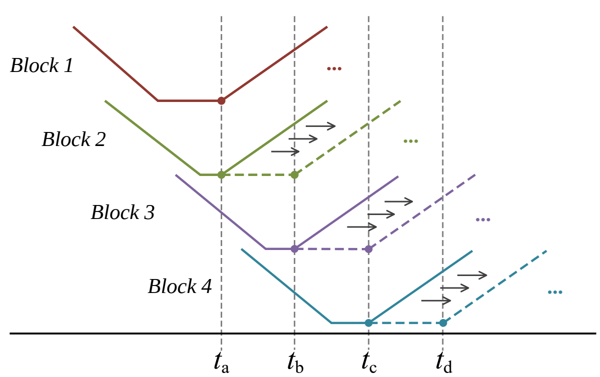


Fig. 10. Illustration about the adjustment of blocks in a local scope.

1) Block Initialization: We first initialize B. Specifically, for each initial start-time point I_i $(i = 1, \ldots, 120)$, we add trips into an empty block one by one, until the block exceeds its predefined length (i.e., 8 or 16 hours). As shown in Fig. 8, assuming that the previous trip is completed at the time point t, we can choose one of the start-time points in the range of $[t + R, t + R + W]$ (i.e., t_a or t_b) as the beginning of the next trip, where the waiting time W is generally set to $1.2(t_b - t_a)$. The above process is repeated until the set of all the feasible blocks that begin at I_i is generated, denoted as B_i. Then, the candidate block set is denoted as $B = \{B_1, \ldots, B_{|\boldsymbol{I}|}\}$.

2) Encoding and Decoding: If we directly use $(z_1, \ldots, z_{|B|})$ as chromosome, (15) will be formulated as a large-scale CMOP. It is because the number of the feasible blocks in B is greater than 10^4. In order to shorten the length of encoding, the chromosome is designed as $(x_1, \ldots, x_{|\boldsymbol{I}|})$. For the ith gene site, x_i takes any integer between 0 and $|B_i|$. If $x_i = j$ $(j \in \{1, \ldots, |B_i|\}$, the jth block in B_i will be selected in the corresponding scheduling scheme; otherwise, no block in B_i will be selected. Fig. 9 shows the decoding process of a chromosome.

TABLE VI

STATISTICS OF THE HV METRIC OBTAINED BY THE FOUR COMPARED ALGORITHMS, WHERE THE BEST RESULTS IN TERMS OF MEDIAN AND IQR ARE HIGHLIGHTED IN BOLDFACE.

Alg.	CDP-NSGA-II*	SP-NSGA-II*	MO-NSGA-II*	ShiP-NSGA-II*
Median	2.650E+02	3.174E+02	2.950E+02	**3.370E+02**
IQR	3.20E+01	1.47E+01	2.07E+01	**1.10E+01**

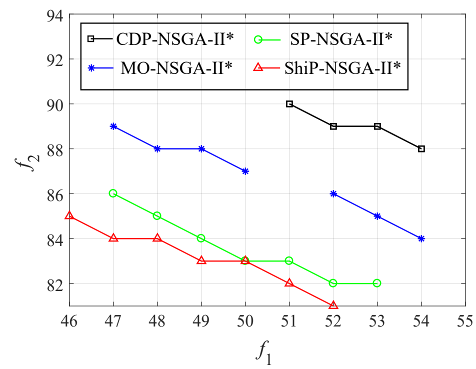


Fig. 11. The nondominated solutions obtained by the four compared algorithms with the best HV values.

3) Genetic Operators: Single-point crossover and bitwise mutation are used for offspring reproduction. When a gene site mutates, it is set to 0 with the probability of 0.2, and set to a random integer in $[1, |B_i|]$ with the probability of 0.8.

4) Local Adjustment Procedure: As the scheduling scheme obtained by directly assembling the selected blocks may not cover all the start-time points, we need to conduct a local adjustment procedure similar to that in [44]. As shown in Fig. 10, t_d is initially missed by four blocks, while t_a is covered by two blocks (i.e., *Block 1* and *Block 2*). Therefore, certain trips in the adjacent blocks will be adjusted to make t_d covered.

In this section, we combined ShiP-NSGA-II with the above components and the resultant algorithm was called ShiP-NSGA-II*. For comparison, three peer algorithms (i.e., CDP-NSGA-II, SP-NSGA-II, and MO-NSGA-II) under the framework of NSGA-II were modified in a similar way. The corresponding algorithms were denoted as CDP-NSGA-II*, SP-NSGA-II*, and MO-NSGA-II*, respectively. Note that, ATM-NSGA-II was absent in the comparison, as it failed to find any encouraging feasible solution. In addition, the parameters were set as follows: the length of chromosome (i.e., n) was 120, which is equal to the number of initial start-time points; the population size was 800; the maximum generation number was 500; and the probabilities of crossover and mutation were set to 0.7 and 0.05, respectively.

Table VI provides the results of the four compared algorithms in terms of median and interquartile range (IQR) of the HV values [45] over 10 independent runs. As show in Table VI, ShiP-NSGA-II* obtains the best results. Besides, Fig. 11 shows the nondominated solutions achieved by the four compared algorithms with the best HV values. It is clear that the PF of ShiP-NSGA-II* has better convergence and spread than the PFs of the three competitors, which indicates that ShiP is also

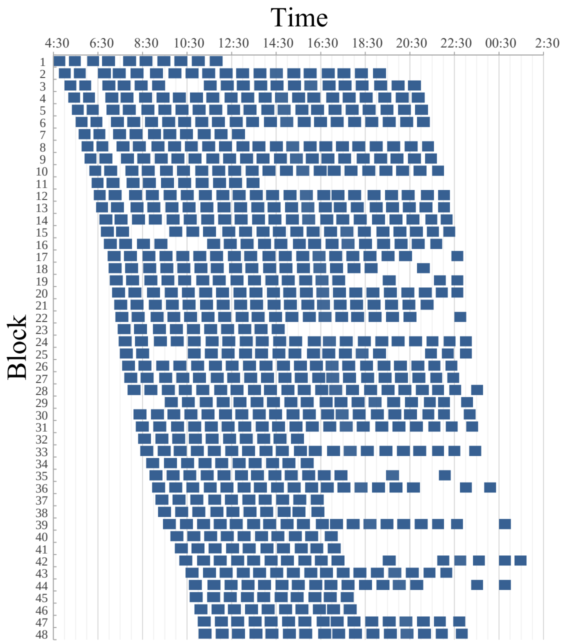


Fig. 12. A scheduling scheme with 48 vehicles and 84 drivers. Each row represents a block, where each blue rectangle is a trip and the interval between any two trips is the waiting time of the corresponding vehicle.

promising for solving real-world CMOPs.

Due to the space limitation, we only plot the scheduling scheme with 48 vehicles and 84 drivers obtained by ShiP-NSGA-II* in Fig. 12. It can be seen that this scheme uses 12 short blocks and 36 long blocks to compete the scheduling task.

VI. Conclusion

In this paper, we considered three major challenges in evolutionary constrained multiobjective optimization: achieving both diversity and feasibility in the early stage of evolution, achieving both diversity and convergence in the later stage of evolution, and providing a transition from "diversity & feasibility" to "diversity & convergence". To address these three challenges, a shift-based penalty function, called ShiP, was proposed in this paper. ShiP constructed new objectives with shift and penalty measures. When there are few feasible solutions in the population, ShiP identified promising infeasible solutions and guided them to enter the feasible region from different directions. With the increase of feasible solutions, ShiP made use of the feasibility proportion of the current parent and offspring populations for an adaptive transition to both diversity and convergence, i.e., motivating the already obtained diverse feasible solutions to converge toward the Pareto optimal solutions.

ShiP was implemented in three well-known MOEA frameworks, which demonstrated its simplicity, flexibility, and generalization. The results on MW test suite verified its effectiveness to solve CMOPs. Moreover, ShiP was extended by an archive-assisted strategy to solve highly complex CMOPs. Finally, ShiP was applied to the vehicle scheduling of urban bus line with promising results.

The source code of this paper can be downloaded from: https://intleo.csu.edu.cn/publication.html

Acknowledgment

The authors would like to sincerely thank Prof. X. Zuo and Dr. Y. Liu for providing the data of the urban bus scheduling problem.

References

[1] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[2] K. Deb, A. Pratap, and T. Meyarivan, "Constrained test problems for multi-objective evolutionary optimization," in *Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, 2001, pp. 284–298.

[3] J. P. Li, Y. Wang, S. Yang, and Z. Cai, "A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 4175–4182.

[4] J. Wang, G. Liang, and J. Zhang, "Cooperative differential evolution framework for constrained multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2060–2072, 2019.

[5] R. Tanabe, A. Oyama, R. Tanabe, and A. Oyama, "A note on constrained multi-objective optimization benchmark problems," in *2017 IEEE Congress on Evolutionary Computation (CEC)*, 2017, pp. 1127–1134.

[6] W. Lakhdhar, R. Mzid, M. Khalgui, Z. Li, G. Frey, and A. Al-Ahmari, "Multiobjective optimization approach for a portable development of reconfigurable real-time systems: From specification to implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 3, pp. 623–637, 2019.

[7] B. Tan, H. Ma, Y. Mei, and M. Zhang, "Evolutionary multi-objective optimization for web service location allocation problem," *IEEE Transactions on Services Computing*, 2018, in press, DOI 10.1109/TSC.2018.2793266.

[8] H. Wang and Y. Jin, "A random forest assisted evolutionary algorithm for data-driven constrained multi-objective combinatorial optimization of trauma systems," *IEEE Transactions on Cybernetics*, vol. 50, no. 2, pp. 536–549, 2020.

[9] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, and S. Chai, "Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm," *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1630–1643, 2020.

[10] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: past, present and future," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[11] Z.-Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 870–884, 2019.

[12] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[14] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," in *Constraint-Handling in Evolutionary Optimization*, Springer, Berlin, Heidelberg, 2009, pp. 145–165.

[15] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[16] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *International Conference on Parallel Problem Solving from Nature*, pp. 832–842, Springer, 2004.

[17] A. Zhou, B. Qu, H. Li, S. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 23–49, 2011.

[18] B. Tessema and G. G. Yen, "A self adaptive penalty function based algorithm for constrained optimization," in *2006 IEEE Congress on Evolutionary Computation (CEC)*, 2006, pp. 246–253.

[19] L. Jiao, J. Luo, R. Shang, and F. Liu, "A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems," *Applied Soft Computing*, vol. 14, pp. 363–380, 2014.

[20] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *2010 UK Workshop on Computational Intelligence (UKCI)*, 2010, pp. 1–6.

[21] M. A. Jan, N. Tairan, and R. A. Khanum, "Threshold based dynamic and adaptive penalty functions for constrained multiobjective optimization," in *1st International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, 2013, pp. 49–54.

[22] J. Wei and Y. Wang, "An infeasible elitist based particle swarm optimization for constrained multiobjecitve optimization and its convergence," *International Journal of Pattern Recognition & Artificial Intelligence*, vol. 24, no. 3, pp. 381–400, 2010.

[23] H. L. Liu and D. Wang, "A constrained multiobjective evolutionary algorithm based decomposition and temporary register," in *2013 IEEE Congress on Evolutionary Computation (CEC)*, 2013, pp. 3058–3063.

[24] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. D. Goodman, "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems," *Applied Soft Computing*, vol. 74, pp. 621–633, 2019.

[25] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites," in *2006 IEEE Congress on Evolutionary Computation (CEC)*, 2006, pp. 1–8.

[26] F. Qian, B. Xu, R. Qi, and H. Tianfield, "Self-adaptive differential evolution algorithm with α-constrained-domination principle for constrained multi-objective optimization," *Soft Computing*, vol. 16, no. 8, pp. 1353–1372, 2012.

[27] Z. Yang, X. Cai, and Z. Fan, "Epsilon constrained method for constrained multiobjective optimization problems: Some preliminary results," in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 1181–1186.

[28] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions," *Soft Computing*, vol. 23, no. 23, pp. 12491–12510, 2019.

[29] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. D. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.

[30] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.

[31] H. Geng, M. Zhang, L. Huang, and X. Wang, "Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization," in *International Conference on Simulated Evolution and Learning*, pp. 336–344, Springer, Berlin, Heidelberg, 2006.

[32] M. A. Jan and R. A. Khanum, "A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D," *Applied Soft Computing*, vol. 13, no. 1, pp. 128–148, 2013.

[33] A. Isaacs, T. Ray, and W. Smith, "Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems," in *2008 IEEE Congress on Evolutionary Computation (CEC)*, 2008, pp. 2780–2787.

[34] D. Vieira, R. Adriano, J. Vasconcelos, and L. Krähenbühl, "Treating constraints as objectives in multiobjective optimization problems using niched pareto genetic algorithm," *IEEE Transactions on Magnetics*, vol. 40, no. 2, pp. 1188–1191, 2004.

[35] Q. Long, "A constraint handling technique for constrained multi-objective genetic algorithm," *Swarm and Evolutionary Computation*, vol. 15, pp. 66–79, 2014.

[36] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, 2008.

[37] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 972–986, 2019.

[38] G. Fileccia Scimemi and S. Rizzo, "An hypervolume based constraint handling technique for multi-objective optimization problems," in *20th Congress of the Italian Association of Theoretical and Applied Mechanics*, 2011.

[39] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[40] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 115–148, 1994.

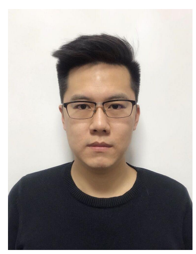
[41] Y. Wang, B. Wang, H. Li and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2938-2952, 2016.

[42] Q. Zhu, Q. Zhang and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 5, pp. 938–947, 2020.

[43] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb and E. Goodman, "Difficulty adjustable and scalable constrained multiobjective test problem toolkit," *Evolutionary Computation*, vol. 28, no. 3, pp. 339–378, 2020.

[44] X. Zuo, C. Chen, W. Tan, and M. Zhou, "Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1030–1041, 2015.

[45] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.



Zhongwei Ma received the Ph.D. degree in computer science and technology from the Central South University, Changsha, China, in 2020. He is researching on the motion planning of Autonomous Driving with Allride.ai, Shanghai, China. His current research interests include constrained optimization, behavioral decision making, trajectory optimization, and vehicle platoon formation.

Yong Wang (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include intelligent learning and optimization and their interdisciplinary applications.

Dr. Wang is an Associate Editor of the *IEEE Transactions on Evolutionary Computation* and the *Swarm and Evolutionary Computation*. He was a recipient of Cheung Kong Young Scholar by the Ministry of Education, China, in 2018, and a Web of Science highly cited researcher in Computer Science in 2017 and 2018.

Supplementary File for "Shift-based Penalty for Evolutionary Constrained Multiobjective Optimization and Its Application"

S-I. Details of Three Components of SHiP^{+}

This section presents the detailed implementation of three components of ShiP^{+} . **Algorithm 1**, **Algorithm 2**, and **Algorithm 3** are the mating selection, the updating of archive, and the updating of population, respectively.

Algorithm 1: MatingSelection(P, A, W)

1 Set $B = P \bigcup A$;
2 Set a temporary pool $\overline{M} = \emptyset$;
3 Associate each solution in B with its nearest weight vector in W;
4 Δ^i $(i = 1, ..., N)$ is the set of solutions associated with $\mathbf{w}^i$;
5 **for** $i = 1$ *to* N **do**
6   **if** $\Delta^i = \emptyset$ **then**
7   Randomly select a solution in B, denoted as $\vec{x}^*$;
8   **else**
9   Randomly select a solution in Δ^i, denoted as $\vec{x}^*$;
10     $\overline{M} = \overline{M} \bigcup \vec{x}^*$;
11 Set the mating pool $M = \emptyset$;
12 **for** $i = 1$ *to* N **do**
13   **if** $rand < 0.9$ **then**
14   Let $\vec{x}^i$ be the ith solution in $\overline{M}$;
15   Compute the angle between $\vec{x}^i$ and each solution in $\overline{M} \backslash \{\vec{x}^i\}$;
16   Find the solution that has the $(0.2N)$-th smallest angle in $\overline{M} \backslash \{\vec{x}^i\}$, denoted as $\vec{x}^{**}$;
17   **else**
18   Randomly select a solution in $\overline{M}$, denoted as $\vec{x}^{**}$;
19     $M = M \bigcup \vec{x}^i \bigcup \vec{x}^{**}$;
20 **return** M;

The aim of Steps 5-9 is to select diverse individuals for mating. Similar to MOEA/D, we select neighboring individuals for mating to enhance the ability of exploitation (Steps 13-16) and also randomly select individuals for mating to enhance the ability of exploration (Steps 17 and 18). M includes $2N$ individuals. Pair-wise individuals are selected to implement SBX. Afterward, two offspring will be produced and one of them is randomly chosen to undergo PM.

Algorithm 2: `ArchiveUpdate`(P, Q, A, W)

1 Set $B = Q \cup A$;
2 Associate each solution in B with its nearest weight vector in W;
3 Δ^i $(i = 1, ..., N)$ is the set of solutions associated with $\mathbf{w}^i$;
4 Count the number of feasible solutions in P: N_f;
5 **if** $N_f = 0$ **then**
6   **while** $|B| > N$ **do**
7   Find the subset that has the most solutions, denoted as Δ^j;
8   Find the two solutions that share the smallest angle in Δ^j, denoted as $\vec{x}^a$ and $\vec{x}^b$;
9   **if** $CV(\vec{x}^a) > CV(\vec{x}^b)$ **then**
10             $B = B \backslash \{\vec{x}^a\}$;
11   **else**
12             $B = B \backslash \{\vec{x}^b\}$;
13     $A = B$;
14 **else**
15     $A = \emptyset$;
16   **for** $i = 1$ *to* N **do**
17   **if** $|\Delta^i| = 1$ **then**
18             $A = A \bigcup \Delta^i$;
19   **if** $|\Delta^i| > 1$ **then**
20   Find the best solution in Δ^i based on the comparison criteria in (1), denoted as $\vec{x}^*$;
21             $A = A \bigcup \vec{x}^*$;
22 **return** A;

Note that, in Step 20, the solutions associated with one weight vector (e.g., $\mathbf{w}^*$) are compared based on the following criteria:

$$\vec{x}^a \text{ is better than } \vec{x}^b \Leftrightarrow \begin{cases} g^{te}(\vec{x}^a, \mathbf{w}^*) < g^{te}(\vec{x}^b, \mathbf{w}^*), & \text{if both } \vec{x}^a \text{ and } \vec{x}^b \text{ are feasible } \textbf{or} \text{ if there exists a feasible solution} \\ & \text{associated with } \mathbf{w}^* \text{ in } P \text{ and } P_{fea} > 0.5, \\ CV(\vec{x}^a) < CV(\vec{x}^b), & \text{otherwise} \end{cases} \tag{1}$$

where $g^{te}(\bullet, \mathbf{w}^*)$ is the value of Tchebycheff aggregation function.

The aim of Steps 6-12 is to remove crowding individuals one by one. In Step 8, the angle information is used to add the diversity and enhance the ability of exploration. In Steps 9-12, A gives higher priority to feasibility, if there is no feasible solution in P; while in Step 20, the priority switches to optimality, when P already has enough feasible solutions. Thus, A works in a way that is complementary to ShiP.

Algorithm 3: `PopulationUpdate`(P, A)

1 Compute the set of nondominated feasible solutions in A, denoted as S_a;
2 Set $P = P \bigcup S_a$;
3 Count the number of feasible solutions in P: N_f;
4 **if** $N_f = N$ **then**
5   Set $P = \{\vec{x} \mid \vec{x} \in P \text{ and } CV(\vec{x}) = 0\}$;
6 **else if** $N_f > N$ **then**
7   Set $S_f = \{\vec{x} \mid \vec{x} \in P \text{ and } CV(\vec{x}) = 0\}$;
8   Divide S_f into different nondominated levels by using nondominated sorting: $L = \{L_1, L_2, ...\}$;
9   Set $P = \emptyset$;
10   Fill P with the nondominated levels in turn, until $|P| > N$;
11   Let L_j be the last nondominated level put into P;
12   **while** $|P| > N$ **do**
13   Find the two solutions that share the smallest angle in L_j, denoted as $\vec{x}^a$ and $\vec{x}^b$;
14   **if** $rand < 0.5$ **then**
15             $P = P \backslash \{\vec{x}^a\}$;
16   **else**
17             $P = P \backslash \{\vec{x}^b\}$;
18 **else**
19   Sort P based on the degree of constraint violation in the ascending order, and delete the last $(|P| - N)$ individuals;
20 **return** P;

In Algorithm 3, P is updated with the nondominated feasible solutions in A, if A has at least one feasible solution. In Steps 6-17, if $N_f > N$, nondominated sorting is first implemented to promote the convergence toward the constrained PF, and then the angle information is utilized to preserve the diversity of the population.

TABLE S-I

COMPARISON OF SIX VARIANTS (I.E., SHIP-1–SHIP-7) IN TERMS OF THE IGD METRIC ON MW TEST SUITE. “AVG” AND “STD DEV” ARE THE AVERAGE AND STANDARD DEVIATION OF THE IGD VALUES OVER 100 INDEPENDENT RUNS, RESPECTIVELY. THE BEST RESULT FOR EACH TEST PROBLEM AMONG THE SIX VARIANTS IS HIGHLIGHTED IN BOLDFACE. THE WILCOXON’S RANK-SUM TEST AT A 0.05 SIGNIFICANCE LEVEL IS PERFORMED, WHERE “+”, “$\approx$”, AND “-” DENOTE THAT SHIP-3 (I.E., THE ORIGINAL SHIP-NSGA-II) PERFORMS BETTER THAN, SIMILAR TO, AND WORSE THAN ITS COMPETITORS, RESPECTIVELY.

Type	Prob.	Statistics	ShiP-1	ShiP-2	ShiP-3	ShiP-4	ShiP-5	ShiP-6	ShiP-7
I	MW2	Avg	2.812E-02 $\approx$	2.891E-02 $\approx$	2.667E-02	**2.658E-02** $\approx$	2.717E-02 $\approx$	2.747E-02 $\approx$	2.893E-02 $\approx$
		(±Std Dev)	(±1.39E-02)	(±1.54E-02)	(±1.47E-02)	**(±1.44E-02)**	(±1.40E-02)	(±1.49E-02)	(±1.70E-02)
	MW4	Avg	5.742E-02 $\approx$	5.814E-02 $\approx$	5.823E-02	5.975E-02 $\approx$	**5.727E-02** $\approx$	5.959E-02 $\approx$	5.934E-02 $\approx$
		(±Std Dev)	(±3.85E-03)	(±2.61E-03)	(±2.63E-03)	(±2.86E-03)	**(±2.74E-03)**	(±2.85E-03)	(±2.84E-03)
	MW14	Avg	1.404E-01 $\approx$	1.392E-01 $\approx$	**1.386E-01**	1.402E-01 $\approx$	1.435E-01 $\approx$	1.458E-01 $\approx$	1.480E-01 $\approx$
		(±Std Dev)	(±2.32E-02)	(±1.35E-02)	**(±1.11E-02)**	(±3.246E-02)	(±2.68E-02)	(±3.51E-02)	(±3.95E-02)
II	MW1	Avg	1.629E-02 +	7.961E-03 $\approx$	8.055E-03	**7.889E-03** $\approx$	7.999E-03 $\approx$	9.214E-03 +	1.158E-02 +
		(±Std Dev)	(±1.54E-02)	(±9.84E-03)	(±1.11E-02)	**(±9.07E-03)**	(±1.08E-02)	(±1.12E-02)	(±1.14E-02)
	MW5	Avg	1.615E-01 +	**4.588E-02** −	4.724E-02	4.658E-02 $\approx$	4.625E-02 $\approx$	4.967E-02 +	5.325E-02 +
		(±Std Dev)	(±2.71E-01)	**(±3.69E-02)**	(±3.88E-02)	(±3.84E-02)	(±3.75E-02)	(±4.65E-02)	(±4.44E-02)
	MW6	Avg	6.119E-02 +	4.591E-02 $\approx$	**4.494E-02**	5.061E-02 +	5.203E-02 +	7.855E-02 +	9.361E-02 +
		(±Std Dev)	(±1.22E-01)	(±8.52E-01)	**(±9.67E-02)**	(±9.47E-02)	(±1.18E-01)	(±1.25E-01)	(±1.34E-01)
	MW8	Avg	6.327E-02 $\approx$	6.299E-02 $\approx$	**6.027E-02**	6.414E-02 $\approx$	6.312E-02 $\approx$	6.905E-02 $\approx$	7.884E-02 +
		(±Std Dev)	(±6.65E-03)	(±4.10E-03)	**(±5.53E-03)**	(±1.93E-02)	(±1.84E-02)	(±1.20E-02)	(±3.11E-02)
III	MW3	Avg	1.589E-02 $\approx$	1.545E-02 $\approx$	1.636E-02	**1.405E-02** $\approx$	1.596E-02 $\approx$	4.015E-02 +	6.822E-02 +
		(±Std Dev)	(±3.12E-02)	(±2.78E-02)	(±3.01E-02)	**(±2.75E-02)**	(±3.04E-02)	(±1.32E-01)	(±2.19E-01)
	MW7	Avg	7.959E-02 $\approx$	8.696E-02 $\approx$	**8.152E-03**	8.550E-03 $\approx$	1.006E-02 +	1.397E-02 +	1.604E-02 +
		(±Std Dev)	(±1.13E-02)	(±1.19E-02)	**(±1.32E-02)**	(±1.22E-02)	(±2.11E-02)	(±2.69E-02)	(±6.16E-02)
	MW10	Avg	1.390E-01 +	5.557E-02 $\approx$	5.362E-02	4.732E-02 −	**4.192E-02** −	5.426E-02 $\approx$	6.834E-02 +
		(±Std Dev)	(±1.51E-01)	(±3.73E-02)	(±6.41E-02)	(±2.54E-02)	**(±2.16E-02)**	(±2.76E-02)	(±2.82E-02)
	MW13	Avg	2.074E-01 +	**1.510E-01** $\approx$	1.522E-01	1.549E-01 $\approx$	1.881E-01 +	2.031E-01 +	2.072E-01 +
		(±Std Dev)	(±1.06E-01)	**(±8.17E-02)**	(±8.41E-02)	(±1.16E-01)	(±1.05E-01)	(±1.19E-01)	(±1.94E-01)
IV	MW9	Avg	6.091E-01 +	6.066E-01 +	4.210E-01	**9.169E-02** −	9.577E-02 −	1.056E-01 −	1.750E-01 −
		(±Std Dev)	(±1.44E-01)	(±1.58E-01)	(±1.15E-01)	**(±7.35E-02)**	(±8.33E-02)	(±1.17E-01)	(±2.05E-01)
	MW11	Avg	8.190E-01 +	1.909E-01 +	1.733E-01	**1.720E-01** $\approx$	1.887E-01 +	2.020E-01 +	2.540E-01 +
		(±Std Dev)	(±1.06E-01)	(±1.95E-01)	(±2.04E-01)	**(±1.64E-01)**	(±1.86E-01)	(±2.01E-01)	(±2.18E-01)
	MW12	Avg	3.114E-01 +	6.358E-02 +	**2.421E-02**	2.934E-02 $\approx$	3.176E-02 +	3.943E-02 +	6.680E-02 +
		(±Std Dev)	(±2.31E-01)	(±1.99E-01)	**(±8.84E-02)**	(±1.16E-01)	(±1.55E-01)	(±1.43E-01)	(±1.69E-01)
$+/\approx/-$			8/6/0	3/10/1	/	1/11/2	5/7/2	8/5/1	10/3/1

S-II. Additional Experiments

In ShiP, the shift measure uses P_{fea} to adaptively control the shift degree during the evolution. To verify the effectiveness of this design, we investigated the performance of ShiP without P_{fea}.

The results in terms of IGD are presented in Table S-II, where "Avg" and "Std Dev" are the average and standard deviation of the IGD values over 100 independent runs, respectively. In addition, to detect the statistical differences, the Wilcoxon's rank-sum test at a 0.05 significance level was conducted between the results obtained by ShiP without P_{fea} and the other two algorithms on each test problem. Note that, the results of CDP and the original ShiP were directly taken from Table II.

As shown in Table S-II, in terms of IGD, the original ShiP is still the best algorithm and achieves the best results on 11 cases. Among the remaining three cases (i.e., MW3, MW4, and MW9), ShiP without P_{fea} and CDP outperform others on two cases and one case, respectively. This demonstrates that ShiP without P_{fea} is significantly worse than the original one, but is similar to CDP. Additionally, the Wilcoxon's rank-sum test supports the above observations.

It can be concluded from the comparisons that using P_{fea} is indeed effective for ShiP. Due to $c = 2$, ShiP without P_{fea} always moves infeasible solutions to the areas that are dominated by the neighboring feasible solutions. Thus, it is equivalent to CDP.

TABLE S-II

Statistics of the IGD Metric Obtained by CDP, ShiP, and ShiP without P_{fea} under the Framework of NSGA-II, Where "Avg" and "Std Dev" are the Average and Standard Deviation of the IGD Values over 100 Independent Runs, Respectively. The Best Result for Each Test Problem among the three Compared Algorithms is Highlighted in Boldface. The Wilcoxon's Rank-Sum Test at a 0.05 Significance Level is Performed, Where "+", "≈", and "−" Denote that ShiP without P_{fea} Performs Better than, Similar to, and Worse than Its Competitors, Respectively.

Type	Prob.	CDP Avg(±Std Dev)	ShiP Avg (±Std Dev)	ShiP without P_{fea} Avg (±Std Dev)
I	MW2	2.841E-02 (±1.32E-02)≈	**2.667E-02 (±1.47E-02)**≈	2.797E-02 (±1.40E-02)
	MW4	5.644E-02 (±2.75E-03)≈	5.823E-02 (±2.63E-03)+	**5.642E-02 (±2.07E-02)**
	MW14	1.388E-01 (±1.76E-02)≈	**1.386E-01 (±1.11E-02)**≈	1.392E-01 (±3.32E-02)
II	MW1	3.011E-02 (±8.32E-02)≈	**8.055E-03 (±1.11E-02)**≈	2.853E-02 (±8.00E-02)
	MW5	2.878E-01 (±3.03E-01)+	**4.724E-02 (±3.88E-02)**−	1.724E-01 (±2.77E-01)
	MW6	6.432E-02 (±1.06E-01)≈	**4.494E-02 (±9.67E-02)**−	7.159E-02 (±1.64E-01)
	MW8	6.221E-02 (±1.85E-02)≈	**6.027E-02 (±5.53E-03)**≈	6.271E-02 (±1.45E-02)
III	MW3	1.101E-02 (±2.14E-02)≈	1.636E-02 (±3.01E-02)≈	**1.093E-02 (±2.49E-02)**
	MW7	3.771E-02 (±1.03E-01)≈	**8.152E-03 (±1.32E-02)**−	3.973E-03 (±1.50E-01)
	MW10	1.327E-01 (±1.08E-01)≈	**5.362E-02 (±6.41E-02)**−	1.293E-01 (±1.45E-01)
	MW13	1.982E-01 (±1.79E-01)≈	**1.522E-01 (±8.41E-02)**−	1.793E-01 (±1.49E-01)
IV	MW9	**1.256E-01 (±2.19E-01)**−	4.210E-01 (±1.15E-01)+	1.733E-01 (±2.44E-01)
	MW11	5.185E-01 (±1.75E-01)≈	**1.733E-01 (±2.04E-01)**−	5.240E-01 (±1.53E-01)
	MW12	1.501E-01 (±2.49E-01)≈	**2.421E-02 (±8.84E-02)**−	1.614E-01 (±2.59E-01)
+/ ≈ /−		1/12/1	2/5/7	/