Hindawi
Advances in Operations Research
Volume 2017, Article ID 9094514, 16 pages
https://doi.org/10.1155/2017/9094514

Research Article

Evolutionary Multiobjective Optimization including Practically Desirable Solutions

Miyako Sagawa,[1] Natsuki Kusuno,[1] Hernán Aguirre,[1] Kiyoshi Tanaka,[1] and Masataka Koishi[2]

[1] *Faculty of Engineering, Shinshu University, Nagano, Japan*
[2] *R&D Center, The Yokohama Rubber Co., Ltd., Tokyo, Japan*

Correspondence should be addressed to Miyako Sagawa; 15st204e@shinshu-u.ac.jp

Received 19 May 2017; Accepted 25 September 2017; Published 8 November 2017

Academic Editor: Yi-Kuei Lin

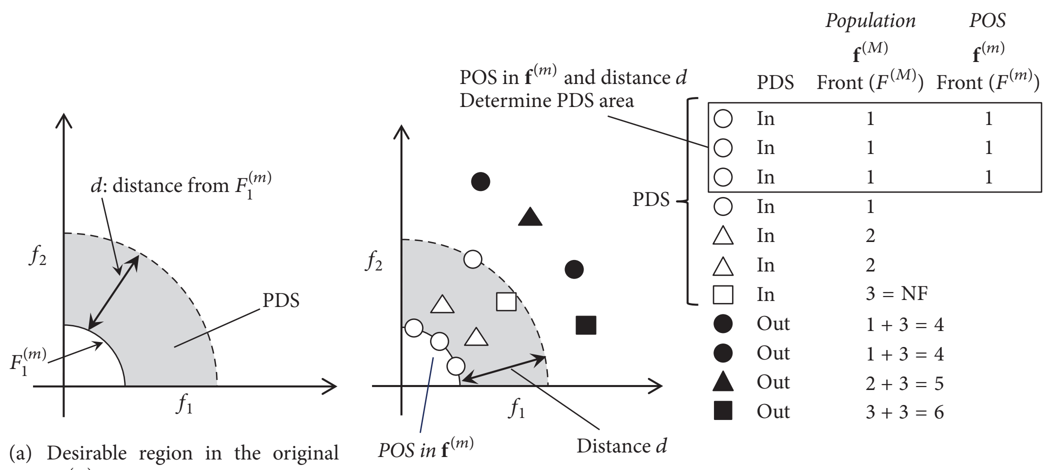
In many practical situations the decision-maker has to pay special attention to decision space to determine the constructability of a potential solution, in addition to its optimality in objective space. Practically desirable solutions are those around preferred values in decision space and within a distance from optimality. This work investigates two methods to find simultaneously optimal and practically desirable solutions. The methods expand the objective space by adding fitness functions that favor preferred values for some variables. In addition, the methods incorporate a ranking mechanism that takes into account Pareto dominance in objective space and desirability in decision space. One method searches with one population in the expanded space, whereas the other one uses two populations to search concurrently in the original and expanded space. Our experimental results on benchmark and real world problems show that the proposed method can effectively find optimal and practically desirable solutions.

1. Introduction

Evolutionary multiobjective algorithms [1, 2] optimize simultaneously two or more objective functions that are usually in conflict with each other. The aim of the algorithm is to find an approximation of the set of Pareto optimal solutions that capture the trade-offs among objective functions. In the presence of several optimal solutions, a decision-maker often considers preferences in objective space and can choose one or few candidate solutions for implementation [3]. Several optimization methods that combine preferences with multiobjective evolutionary algorithms have been proposed; see, for example, [4–19]. Preferences can be determined a priori, during the search, or a posteriori. Once preferred solutions are found, the exact values of solutions in decision space are implicitly determined. This approach is valid when there is no concern about the buildability of candidate solutions.

In many practical situations the decision-maker has to pay special attention to decision space in order to determine the constructability of a potential solution. In manufacturing applications, for example, preferences for particular values of decision variables could appear due to unexpected operational constraints, such as the availability or lack of materials with particular specifications, or simply because physical processes that determine a particular value for a decision variable have become easier to perform than those required to determine another value. Also, it may be necessary to introduce new equipment depending on the combination of decision variables. When these situations arise the decision-maker is interested in knowing how far these possible solutions are from optimality. Furthermore, in design optimization and innovation related applications the extraction of useful design knowledge is extremely relevant. In these cases, analysis of what-if scenarios to understand trade-offs in decision space, without losing sight of optimality, is important.

A way of emphasizing preferred values in decision space is to modify the range of variables, so the search could focus on the regions of interest. A drawback of this approach is that the preferred regions in decision space may not contain optimal solutions. Thus, we could obtain solutions around the preferred values in decision space, but we could lose the trade-off information between the original fitness functions. Another way is to add objective functions for decision


d: distance from $F_1^{(m)}$
f_2
PDS
$F_1^{(m)}$
f_1
f_2
f_1
POS in $\mathbf{f}^{(m)}$
Distance d
POS in $\mathbf{f}^{(m)}$ and distance d
Determine PDS area
PDS
Population $\mathbf{f}^{(M)}$ Front ($F^{(M)}$)
POS $\mathbf{f}^{(m)}$ Front ($F^{(m)}$)
○ In 1 1
○ In 1 1
○ In 1 1
○ In 1
△ In 2
△ In 2
□ In 3 = NF
● Out 1 + 3 = 4
● Out 1 + 3 = 4
▲ Out 2 + 3 = 5
■ Out 3 + 3 = 6

(a) Desirable region in the original space $\mathbf{f}^{(m)}$

(b) Sorting by front number and desirability

Figure 1: Region of practically desirable solutions with preferred values in variable space located within at most a distance d from the Pareto optimal set computed in the original objective space $\mathbf{f}^{(m)}$. Sorting by desirability with respect to the original space $\mathbf{f}^{(m)}$ and front number in the extended space $\mathbf{f}^{(M)}$.

variables, hoping that the search will render optimal as well as preferred solution in decision space. However, it is well known that multiobjective evolutionary algorithms can lose their effectiveness on problems with more than 3 objectives. Thus, the obtained solutions could be suboptimal and an analysis of preferred solutions with respect to these obtained solutions would be misleading.

From this standpoint, in this work, we investigate ways to enhance approaches that incorporate additional fitness functions associated with particular decision variables, aiming to find solutions around preferred values of the chosen variables while searching for optimal solutions in the original objective space. We aim to obtain optimal solutions as well as solutions with preferred settings on decision space that are close to the Pareto front.

In addition to expanding the objective space, we also constraint the distance that solutions could be away from the instantaneous Pareto nondominated set computed in the original space. We call these solutions as practically desirable solutions. We put forward two methods. One method uses two populations to search concurrently in the original and extended spaces, ranking solutions by Pareto dominance and practical desirability. The other method uses just one population to search in the extended space but ranks solutions by Pareto dominance and practical desirability. We compare with an algorithm that simply restricts the range of decision variables around the preferred values and an algorithm that expands the space without constraining the distance from optimality. We test the algorithms using DTLZ functions with two and three objectives in the original space and two additional objectives for the expanded space. We also use these approaches in real world design optimization problems. Our results show that the proposed method can effectively find practically desirable solutions that are valuable to establish trade-offs in decision space and extract relevant design knowledge.

2. Proposed Method

2.1. Concept. We pursue approaches that incorporate additional fitness functions associated with particular decision variables, aiming to find solutions around preferred values of the chosen variables while searching for optimal solutions in the original objective space.

Let us define the original objective space $\mathbf{f}^{(m)}$ as the vector of functions

$$\mathbf{f}^{(m)}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\right), \tag{1}$$

where $\mathbf{x}$ is a vector of variables and $m \geq 2$ the number of functions. The extended objective space $\mathbf{f}^{(M)}$ with $M > m$ objectives is given by

$$\begin{aligned}\mathbf{f}^{(M)}&(\mathbf{x})\\ &= \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}), f_{m+1}(\mathbf{x}), \ldots, f_M(\mathbf{x})\right),\end{aligned} \tag{2}$$

where $f_{m+1}(\mathbf{x}), \ldots, f_M(\mathbf{x})$ are the additional $M-m$ functions used to evaluate solutions with preferred values in one or more decision variables.

The aim of extending the objective space is bias selection to include solutions with particular desired values for some decision variables. However, it is also expected that evolution in an expanded objective space would substantially increase diversity of solutions, which could jeopardize convergence of the algorithm in the original space and the expanded space as well. Thus, in addition to an expanded space, we also constraint the distance that solutions could be from the instantaneous set of Pareto nondominated solutions computed in the original space, as illustrated in Figure 1. We call these solutions as practically desirable solutions. In the following we describe two methods that implement the concept outlined above.

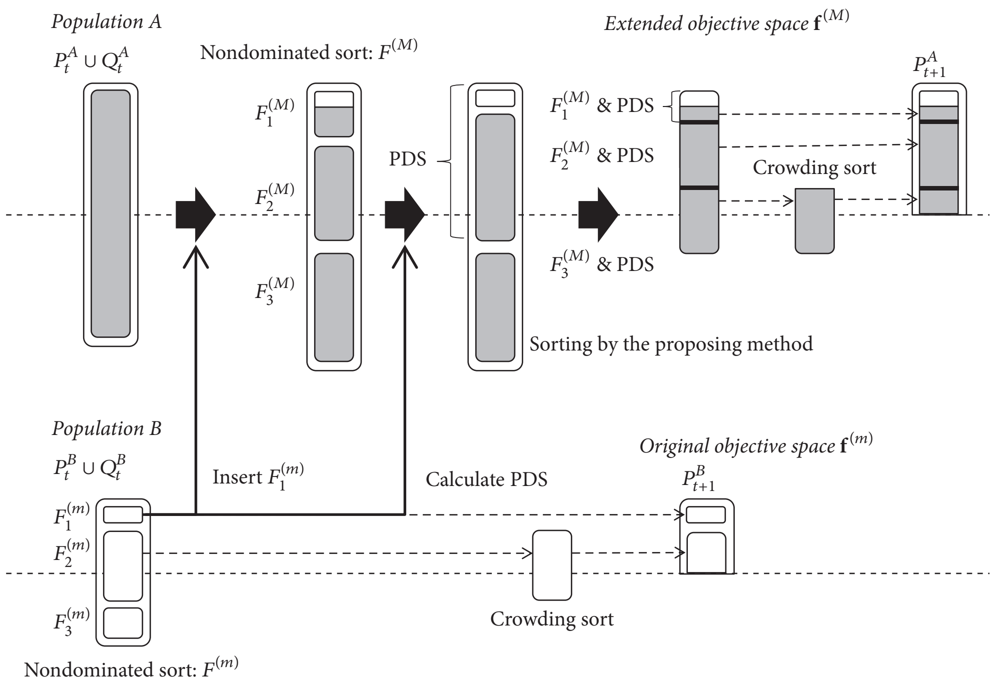


Figure 2: Two evolving populations, concurrent search method.

2.2. Two-Population Concurrent Evolution. This method evolves concurrently two populations in different objective spaces as illustrated in Figure 2. Population A evolves in the extended objective space $\mathbf{f}^{(M)}$ using an enhanced ranking of solutions that prefers practically desirable solutions for survival selection and parent selection. On the other hand, Population B evolves in the original objective space $\mathbf{f}^{(m)}$. The instantaneous set of Pareto nondominated solutions computed in $\mathbf{f}^{(m)}$ from the Population B is incorporated into Population A and used as a reference to establish the desirability of solutions in Population A. Ranking for Population A is enhanced by making it dependant on both front number in the extended space $\mathbf{f}^{(M)}$ and desirability with respect to the original space $\mathbf{f}^{(m)}$. This new ranking is used for survival selection and parent selection as well. In this method, since Population B evolves independently, a good convergence is expected in the original space, which implies a reference for desirability closer to the true Pareto front of the problem. In addition, since the set of Pareto solutions in Population B is copied to Population A, a high pressure towards the Pareto optimal front of the extended space is also expected.

In this work, Population A evolves using NSGA-II with the enhanced ranking and survival selection, whereas Population B evolves using conventional NSGA-II [20]. In the following we explain survival selection and ranking procedure used to evolve Population A, illustrated in Figure 2.

Step 1. Get a copy of the set of nondominated solutions from Population B that evolves in the original space $\mathbf{f}^{(m)}$. Let us call this set $F_1^{(m)}$.

Step 2. Apply nondominated sorting to $R_t^A \cup F_1^{(m)}$ in the space $F^{(M)}$, where $R_t^A = P_t^A \cup Q_t^A$ is the combined population of parents P_t^A and offspring Q_t^A evolving in the expanded space $\mathbf{f}^{(M)}$. Classify solutions into fronts $F_i^{(M)}$ and rank solutions according to the ith front they belong to, where $i = 1, 2, \ldots, \mathrm{NF}$. Note that solutions in $F_1^{(m)}$ will be part of $F_1^{(M)}$.

Step 3. Calculate the Euclidean distance, in the original objective space $\mathbf{f}^{(m)}$, between solutions in the fronts $F_i^{(M)}$ and the set $F_1^{(m)}$. The distance from solution $\mathbf{x} \in F_i^{(M)}$ to $F_1^{(m)}$ is given by $\delta(\mathbf{x}) = \min \|\mathbf{f}^{(m)}(\mathbf{x}) - \mathbf{f}^{(m)}(\mathbf{y})\|$, $\mathbf{y} \in F_1^{(m)}$. If the distance $\delta(\mathbf{x})$ is smaller than a threshold distance d then solution $\mathbf{x}$ is marked as *desirable*. Otherwise, it is marked as *undesirable*.

Step 4. Sort solutions by front rank and *desirability*. The front number (rank) of desirable solutions remains the same, while the front number of an undesirable solution initially classified in front i is modified to $i + \mathrm{NF}$, where NF is the number of fronts initially obtained by nondominated sorting. That is, undesirable solutions are penalized so that no undesirable solution is assigned better rank than a desirable one, while still differentiating among undesirable ones. Sorting by front number and desirability is illustrated in Figure 1(b).

Step 5. Form the population P_{t+1}^A for the next generation by copying to it fronts $F_i^{(M)}$ in ascending order, starting with front $F_1^{(M)}$. If all solutions in $F_i^{(M)}$ do not fit in P_{t+1}^A ($|P_{t+1}^A| = |R_t^A|/2$), select the required number according to their crowding distance (less crowded is better). Since undesirable solutions are penalized, as explained above, desirable solutions are

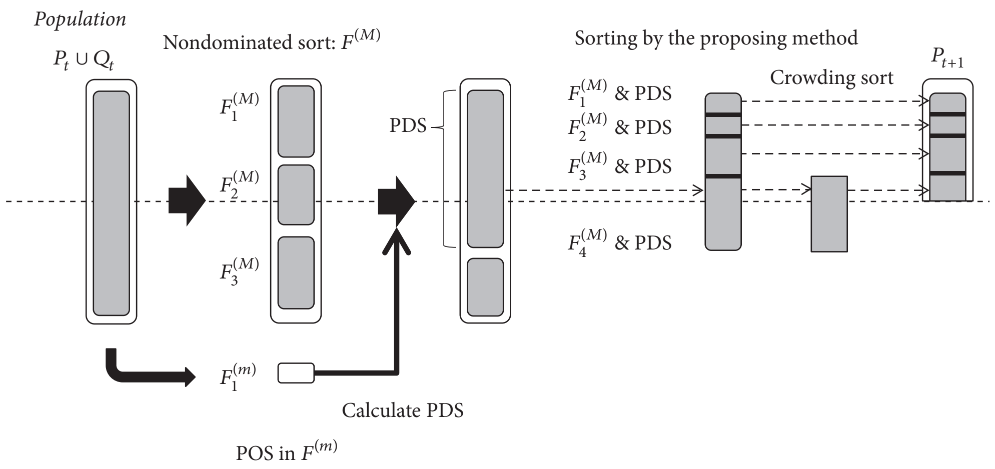


FIGURE 3: Single population method.

given priority for survival and reproduction as well (better rank than undesirable solutions).

2.3. Single Population. This method evolves a single population in the extended objective space $\mathbf{f}^{(M)}$ as illustrated in Figure 3. At each generation, it classifies solutions as desirable if they are within a distance d of the instantaneous set of Pareto optimal solutions computed from the same population in the original space $\mathbf{f}^{(m)}$ and enhances ranking of solutions by making it dependant on both front number in the extended space $\mathbf{f}^{(M)}$ and desirability with respect to the original space $\mathbf{f}^{(m)}$. This new ranking is used for survival selection and parent selection as well.

In this work we evolve the population using NSGA-II [20] with the enhanced ranking and survival selection illustrated in Figure 3 and described as follows.

Step 1. Apply nondominated sorting to the combined population R_t of parents P_t and offspring Q_t, $R_t = P_t \cup Q_t$, calculating dominance among solutions in the extended space $\mathbf{f}^{(M)}$, classify solutions into fronts $F_i^{(M)}$, and rank solutions according to the ith front they belong to, where $i = 1, 2, \ldots, \mathrm{NF}$.

Step 2. Obtain the set of nondominated solutions in the original space $\mathbf{f}^{(m)}$ from the combined population R_t. Let us call this set $F_1^{(m)}$.

Step 3. Calculate the Euclidean distance between solutions in the fronts $F_i^{(M)}$ and the set $F_1^{(m)}$ and mark solutions as desirable or undesirable as described in Step 3 of the previous section. If the distance $\delta(\mathbf{x})$ is smaller than a threshold distance d then solution $\mathbf{x}$ is marked as *desirable*. Otherwise, it is marked as *undesirable*.

Step 4. Sort solutions by their *desirability* as described in Step 4 of the previous section.

Step 5. Form the population P_{t+1} for the next generation as described in Step 5 of the previous section.

3. Test Problems, Performance Indicators, and Experimental Setup

We study the performance of the algorithms in continuous DTLZ2 and DTLZ3 functions [21]. These functions are scalable in the number of objectives and variables and have a nonconvex Pareto optimal surface that lies inside the first quadrant of the unit hypersphere, with Pareto-local fronts constructed parallel to it. DTLZ3 is a variation of DTLZ2 that introduces a large number of local Pareto optimal fronts located far away from the true Pareto optimal set, which increases the difficulty to converge towards the true Pareto optimal set. Functions in DTLZ2 are unimodal, whereas functions in DTLZ3 are multimodal.

In our experiments with the DTLZ problems we set the number of objectives to $m = \{2, 3\}$ varying the number of variables $n = \{5, 10, 15\}$. Thus, the original objective space is given by $\mathbf{f}^{(m=2)} = (f_1, f_2)$ and $\mathbf{f}^{(m=3)} = (f_1, f_2, f_3)$, respectively. The original objective space is extended by adding two functions to form $\mathbf{f}^{(M)}$, where $M = m + 2$. The two additional functions are as follows:

$$\begin{aligned} f_{m+1} &= \left|x_5 - 0.3\right|, \\ f_{m+2} &= \left|x_5 - 0.4\right|. \end{aligned} \tag{3}$$

Here, the assumed desirable values for variable x_5 are 0.3 and 0.4. Also, in this problem it is known that the optimal value for x_5 is 0.5. We set the threshold distance $d = 10$ to determine the desirability of solutions with respect to Pareto optimal solutions in $\mathbf{f}^{(m)}$.

In addition to DTLZ functions, we also test the algorithms on two formulations of a real world problem for tire design optimization. Details about the problem are included in the experimental section.

To evaluate convergence of solutions obtained by the algorithms we use the generational distance (GD) [22], which measures the distance of the obtained solutions to the true Pareto front using (4), where P denotes a set solutions found by the algorithm and $\mathbf{x}$ a solution in the set. Smaller values of GD indicate that the set P is closer to the Pareto optimal front. That is, smaller values of GD mean better convergence of solutions.

$$\mathrm{GD} = \underset{\mathbf{x}\in P}{\mathrm{average}} \left\{ \left[\sum_{i=1}^{m} \left(f_i(\mathbf{x})\right)^2 \right]^{1/2} - 1 \right\}. \quad (4)$$

We also use the $\mathscr{C}$-metric [22] to compare sets of Pareto nondominated solutions obtained by MOEAs and provide complementary information on convergence. Let us denote $\mathscr{A}$ and $\mathscr{B}$ to be the set of nondominated solutions found by two algorithms. $\mathscr{C}(\mathscr{A},\mathscr{B})$ gives the fraction of solutions in $\mathscr{B}$ that are dominated at least by one solution in $\mathscr{A}$. More formally,

$$\mathscr{C}(\mathscr{A},\mathscr{B}) = \frac{|\{\mathbf{b}\in\mathscr{B} \mid \exists\mathbf{a}\in\mathscr{A} : \mathbf{a}\succeq\mathbf{b}\}|}{|\mathscr{B}|}, \quad (5)$$

where $\mathbf{a} \succeq \mathbf{b}$ indicates that $\mathbf{a}$ dominates $\mathbf{b}$. $\mathscr{C}(\mathscr{A},\mathscr{B}) = 1.0$ indicates that all solutions in $\mathscr{B}$ are dominated by solutions in $\mathscr{A}$, whereas $\mathscr{C}(\mathscr{A},\mathscr{B}) = 0.0$ indicates that no solution in $\mathscr{B}$ is dominated by solutions in $\mathscr{A}$. Since usually $\mathscr{C}(\mathscr{A},\mathscr{B}) + \mathscr{C}(\mathscr{B},\mathscr{A}) \neq 1.0$, both $\mathscr{C}(\mathscr{A},\mathscr{B})$ and $\mathscr{C}(\mathscr{B},\mathscr{A})$ are required to understand the degree to which solutions of one set dominate solutions of the other set.

We study three algorithms, a conventional NSGA-II and the two proposed methods explained in Section 2. We run the algorithms 30 times and present average results, unless stated otherwise. We use a different random seed in each run, but all algorithms use the same seeds. The number of generations is set to 1000 generations, parent, and offspring population size $|P_t| = |Q_t| = 2500$. In case of the proposed method that evolves two populations concurrently, $|P_t^A| = |Q_t^A| = 2250$ for the search on the expanded space and $|P_t^B| = |Q_t^B| = 250$ for the search on the original space. These settings are chosen for comparison. A discussion on population size is included in Section 4.5. As variation operators, the algorithms use SBX crossover and polynomial mutation, setting their distribution exponents to $\eta_c = 15$ and $\eta_m = 20$, respectively. Crossover rate is $pc = 1.0$, crossover rate per variable is $pcv = 0.5$, and mutation rate per variable is $pm = 1/n$, where n is the number of variables of the problem.

4. Simulation Results and Discussion

4.1. Results by Conventional NSGA-II. First, we run a conventional NSGA-II to optimize the original space $\mathbf{f}^{(m)} = (f_1, f_2)$ modifying the range of variable x_5 to $[0.29, 0.41]$ from its original range $[0.0, 1.0]$, so that the search could focus on a subregion that includes the practically desirable values 0.3 and 0.4 established by the designer for variable x_5. Results for DTLZ3 are shown in Figure 5 for $m = 2$ objectives and $n = 5$ variables. Note that the algorithms are able to find solutions around $x_5 = 0.4$, but not for $x_5 < 0.4$.

This is because solutions around $x_5 = 0.4$ completely dominate solutions $x_5 < 0.4$. In addition, since the reduced range of variable x_5 does not include the value 0.5 no optimal solutions are found. Thus, simply restricting the range of the variables is not an appropriate option to induce practically desirable solutions.

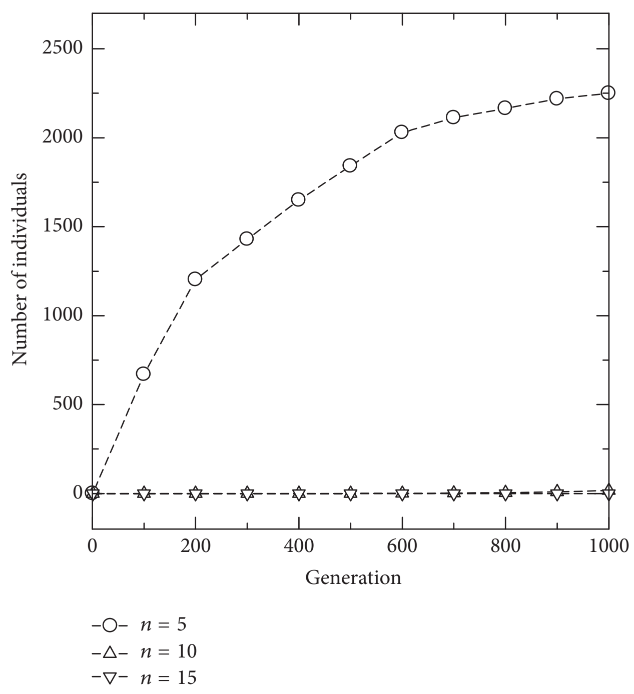
Then second, we run a conventional NSGA-II to optimize DTLZ3 expanding its objective space from $\mathbf{f}^{(m)}$ to $\mathbf{f}^{(M)}$ in order to investigate whether the simple addition of objectives f_{m+1} and f_{m+2} that try to favor a preferred region of variable space is effective or not. Figure 6 shows results at the final generation by conventional NSGA-II for DTLZ3 problem with $m = 2$ objectives in the original objective space $\mathbf{f}^{(m)} = (f_1, f_2)$ and $M = 4$ objectives in the expanded space $\mathbf{f}^{(M)}$. Note that a large number of solutions can be found in the range $x_5 = [0.3, 0.4]$ and some around $x_5 = 0.5$, as shown in Figure 6(a). This shows that objectives f_{m+1} and f_{m+2} introduce trade-offs and work effectively to generate solutions in the range that include the desirable values 0.3 and 0.4 for x_5. However, these solutions are far away from the Pareto optimal front as shown in Figure 6(b). This can be seen more precisely in Figures 6(c) and 6(d) that zoom in the region $f_1 \leq 20$ and $f_2 \leq 20$. Note that the Pareto optimal front in this problem is located in the first quadrant of the circle of radius one. In summary, no desirable solution close to the optimal front could be found by conventional NSGA-II just by including the additional functions to bias the search towards a preferred region of variable space. In fact, no solution, preferred or not, close to the Pareto optimal front, could be found.

4.2. Results by Proposed Method Evolving Solutions Concurrently on the Original and Expanded Space. Figure 7 shows results for DTLZ3 by the proposed method searching concurrently on the original space $\mathbf{f}^{(m)} = (f_1, f_2)$, $m = 2$, $n = 5$, and on the expanded space $\mathbf{f}^{(M)}$, $M = 4$, ranking solutions by their desirability to bias survival and parent selection. From Figure 7(a) it can be seen that the proposed method effectively finds solutions around the two preferred values $x_5 = 0.3$ and $x_5 = 0.4$. In addition it also finds solutions around $x_5 = 0.5$, the value at which solutions become Pareto optimal in this problem. Also, from Figure 7(b) note that the solutions found are within the threshold distance $d = 10$ established as a condition for solutions desirability.

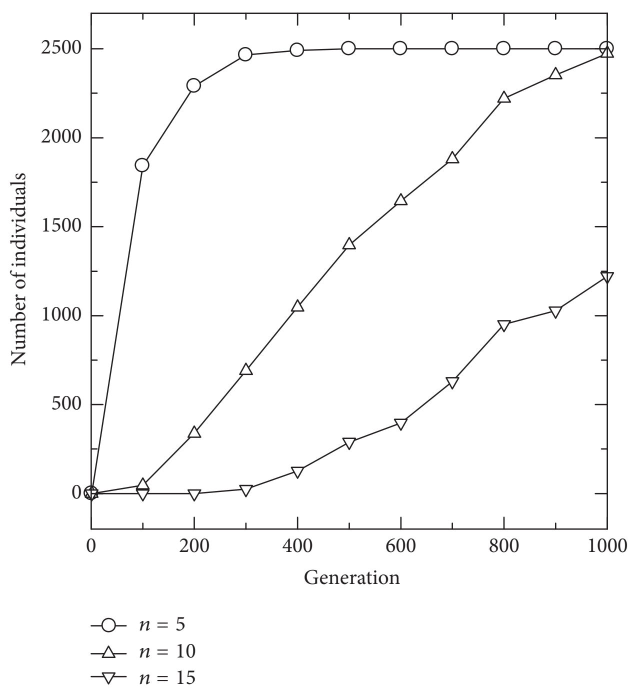
These solutions are valuable for the designer to analyze alternatives that include practical manufacturing desirable features in addition to optimality.

4.3. Comparison between Methods Incorporating Desirability Sort. In this section we compare the two methods presented in Section 2. These methods search on the extended space $\mathbf{f}^{(M)}$ incorporating two fitness functions $f_{(m+1)}$ and $f_{(m+2)}$ to induce preferred solutions in variable space and a desirability sort to favor solutions close to the Pareto optimal set in the original space $\mathbf{f}^{(m)}$. The difference between these methods is that one of them evolves a single population in the extended space, whereas the other one evolves concurrently an additional population in the original space.

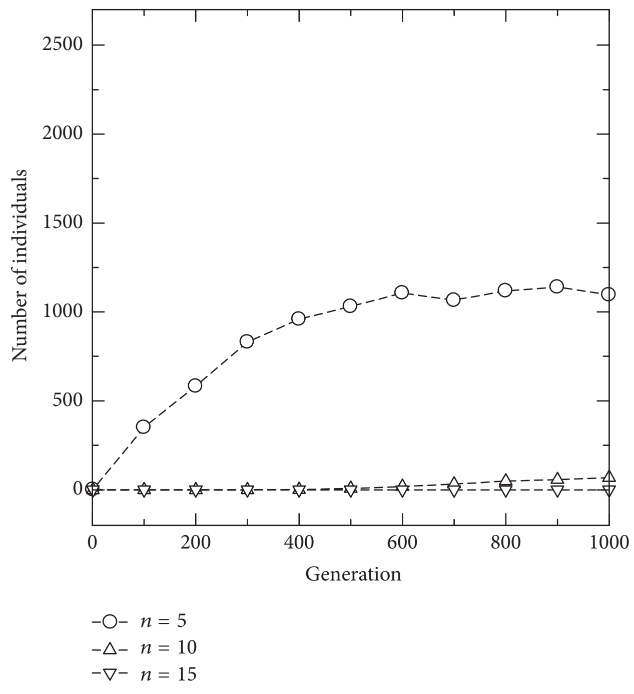
Figure 4 shows the number of solutions that fall within the desirable area at various generations of the evolutionary process that is solutions located within a distance $d = 10$ of the instantaneous set of Pareto nondominated solutions in $\mathbf{f}^{(m)}$. Results are shown for DTLZ3 problem with $m = \{2, 3\}$ original objectives varying the number of variables



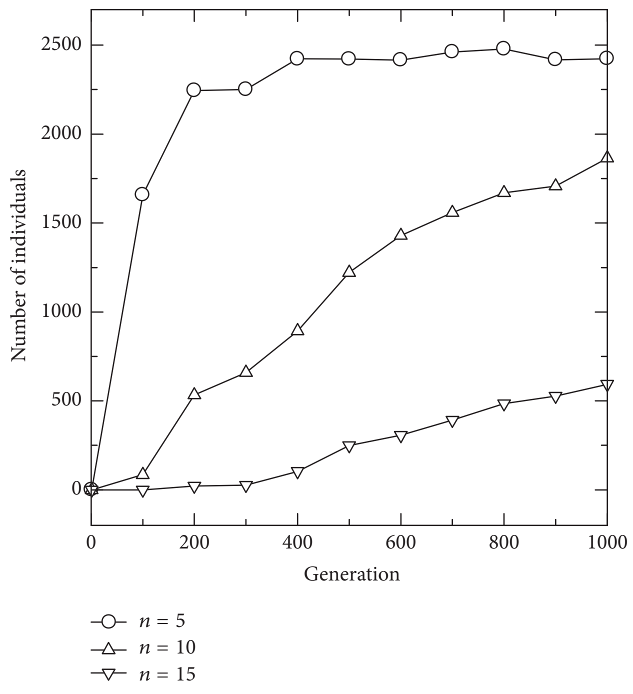
(a) Single population $\mathbf{f}^{(M=4)}$, $\mathbf{f}^{(m=2)}$



(b) Concurrent search $\mathbf{f}^{(M=4)}$, $\mathbf{f}^{(m=2)}$



(c) Single population $\mathbf{f}^{(M=5)}$, $\mathbf{f}^{(m=3)}$

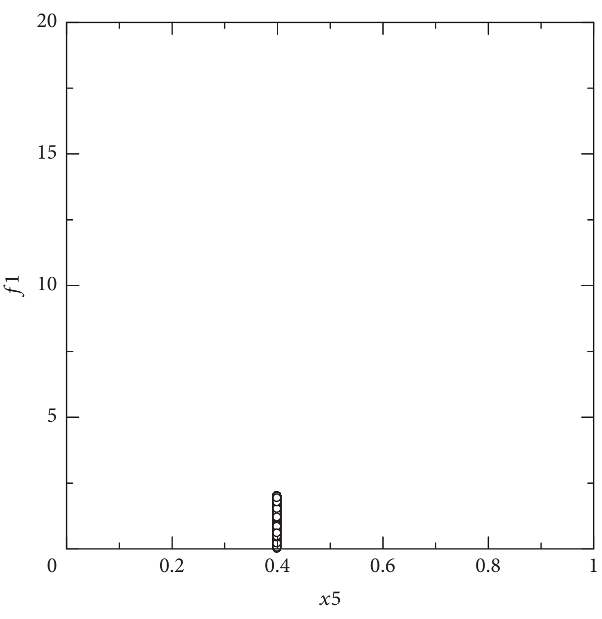


(d) Concurrent search $\mathbf{f}^{(M=5)}$, $\mathbf{f}^{(m=3)}$

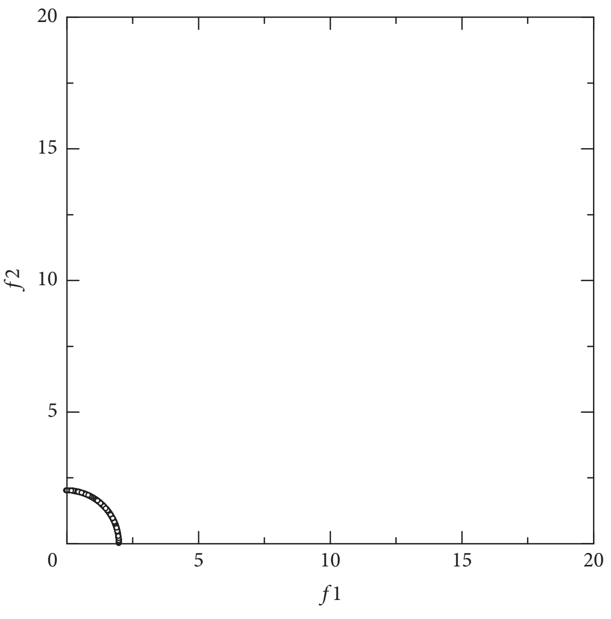
Figure 4: Number of solutions within the desirable area over the generations, that is, solutions located within a distance $d = 10$ of the Pareto optimal solutions in $\mathbf{f}^{(m)}$ found by the proposed method. DTLZ3 problem with $m = 2$ and $m = 3$ original objectives and two additional fitness functions.

$n = \{5, 10, 15\}$. Note that the method that evolves a single population is able to find a considerable number of solutions for two and three objective problems for $n = 5$ variables, but it cannot do it for $n = 10$ and $n = 15$ variables. On the other hand, the method that evolves concurrently a population in the extended space and a population in the original space can effectively find a large number of solutions for any number of variables.

Figure 8 shows the generational distance (GD) over the generations by the single population method and the



(a) f_1 and variable x_5



(b) $f^{(m)} = (f_1, f_2)$

FIGURE 5: Conventional NSGA-II searching on original space $f^{(m)} = (f_1, f_2)$, $n = 5$, restricting variable x_5 to the range $[0.29, 0.41]$.

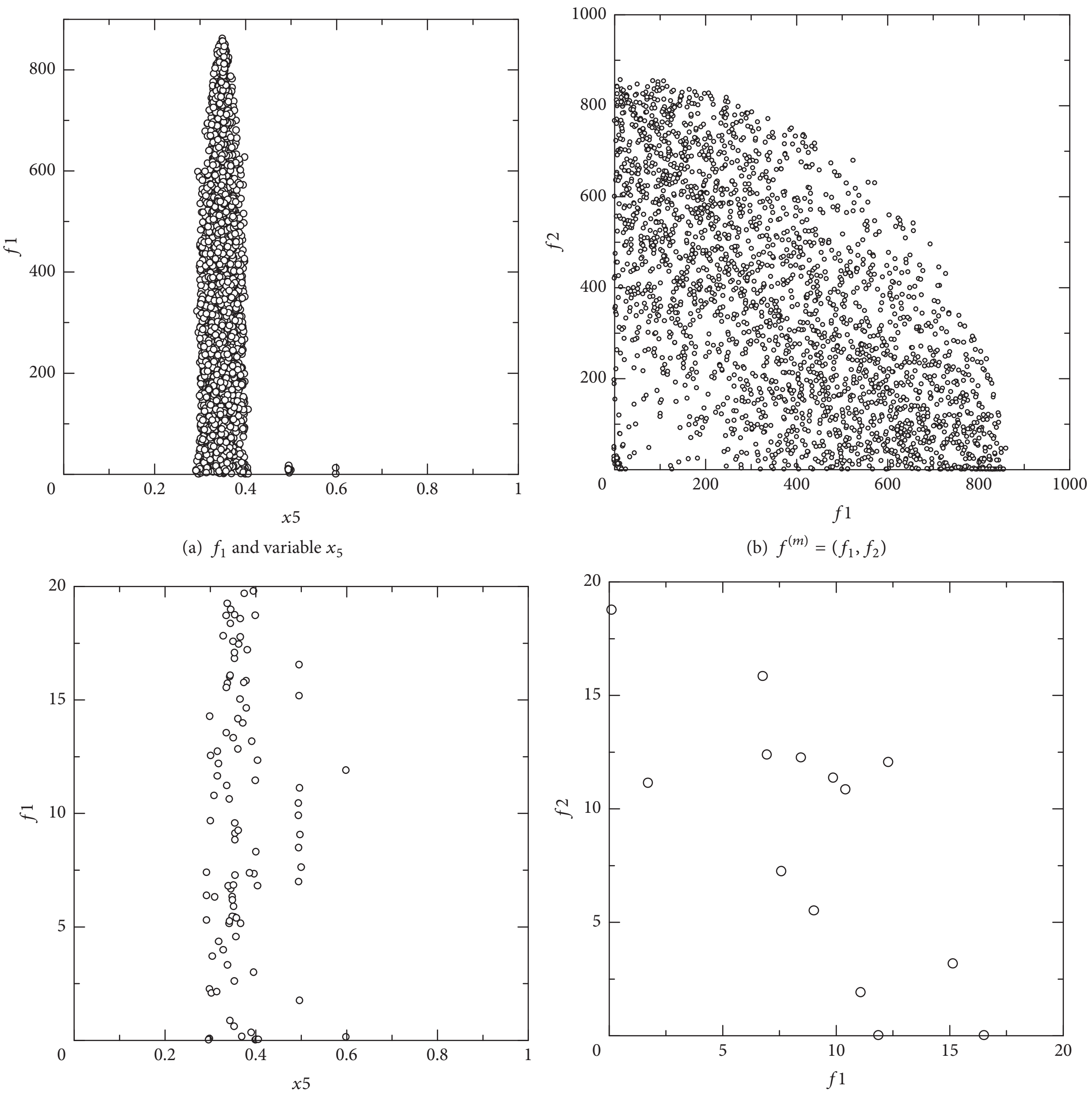
proposed concurrent search method. Results are shown for DTLZ3, $m = 2$ original objectives, $M = 4$ objectives in the extended space, and $n = \{5, 10, 15\}$ variables. GD is calculated separately grouping solutions around the preferred values $x_5 = 0.3$ and $x_5 = 0.4$ and optimal value $x_5 = 0.5$. Solutions are considered within a group if the value of x_5 is in the range $[x_5 - 0.005, x_5 + 0.005]$. Similarly, Figure 9 shows results for $m = 3$ original objectives and $M = 5$ objectives in the extended space. From these figures, note that for the three groups of solutions the method that searches concurrently in the original space and in the extended space overall achieves better (smaller) GD than the method that searches only in the extended space using a single population. This clearly shows that the concurrent search on the original space pulls the population closer to the Pareto optimal front and achieves better convergence in addition to finding solutions around the preferred values in variable space.

4.4. Threshold Distance for Desirability. The threshold distance d used to determine desirability of solutions is a parameter set by the user. With this parameter the user establishes how much he is willing to trade optimality for constructability of solutions. The designer often has some idea of how to set this parameter. However, it can be used to explore different scenarios by the designer and learn more about the problem. To illustrate this, Figure 10 shows results on a DTLZ2 by NSGA-II and by the proposed method evolving concurrently in the original and extended space set with two values of d. The DTLZ2 problem used here has $n = 5$ variables, $m = 2$ objectives in the original space, and $M = 4$ in the extended space. The objective functions and preferred values are the same as those set for DTLZ3.

From Figures 10(a) and 10(d) note that NSGA-II evolving in the extended space is able to find solutions around the desired values $x_5 = 0.3$ and $x_5 = 0.4$, but many of those solutions are too far away from the Pareto optimal front. By setting $d = 0.25$, the proposed algorithm finds desirable solutions around the preferred values in variable space closer to optimality than NSGA-II. However, when $d = 0.025$ is used solutions around 0.4 are found, but no solution around 0.3 can be found. This tells the designer that solutions very close to optimality can be implemented if he is willing to build his solutions around $x_5 = 0.4$. But he must trade more optimality if he wants to build the solution around $x_5 = 0.3$. Tables 1 and 2 show the C-metric values comparing solutions obtained by NSGA-II (algorithm B) with solutions obtained by the proposed algorithm set with distance $d = 0.25$ (algorithm A_1) and $d = 0.025$ (algorithm A_2). From these tables note that more than 93% of solutions found by NSGA-II are dominated by solutions found by the proposed method.

TABLE 1: C-metric every hundred generations by proposed method with $d = 0.25$ (algorithm A_1) and NSGA-II (algorithm B).

T	$C(B, A_1)$	$C(A_1, B)$
100	0.66	0.93
200	0.65	0.94
300	0.66	0.94
400	0.67	0.93
500	0.66	0.93
600	0.66	0.94
700	0.66	0.94
800	0.66	0.94
900	0.67	0.93
1000	0.67	0.93



(a) f_1 and variable x_5

(b) $f^{(m)} = (f_1, f_2)$

(c) Zoom $f_1 \leq 20$ and x_5

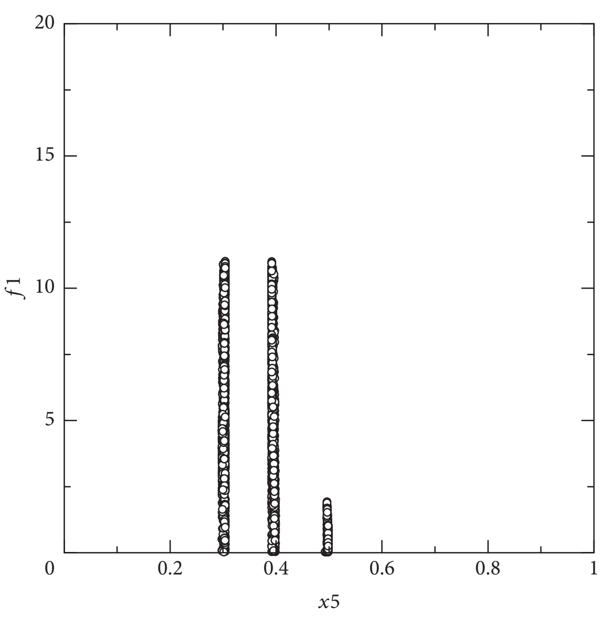
(d) Zoom $f_1 \leq 20$ and $f_2 \leq 20$

Figure 6: Conventional NSGA-II searching on expanded space $\mathbf{f}^{(M)}$, $M = 4$. Original space $\mathbf{f}^{(m)} = (f_1, f_2)$, $m = 2$, $n = 5$. Final generation, DTLZ3 problem.

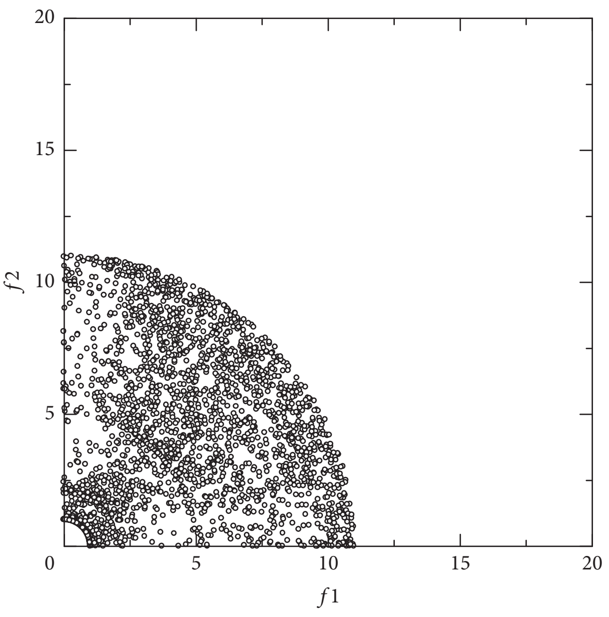
4.5. Population Size and Iterations. We choose the population size 2500 for the benchmark problems because this setting allows the single population approach to find a large number of practically desirable solutions (PDS) on DTLZ3 with $n = 5$ variables, around 90% for $M = 4$ and 44% for $M = 5$, although this population size is not enough to find PDS on problems with a larger number of variables. For the concurrent search approach we set the overall population size 2500 to compare with the single population approach using similar settings. As shown in Tables 3–7, the proposed concurrent approach scales up to problems with a larger number of variables and can use smaller populations.

We use a large number of iterations in order to compare both algorithms after they have approached convergence. However, the number of iterations can be reduced in the concurrent approach and still achieve acceptable good performance.

To clarify this, Table 3 shows the percentage of PDS solutions in the final population for the single population approach on $M = 4$ objectives (2 objectives in the original space and 2 additional objectives in the extended space).



(a) f_1 and variable x_5



(b) $f^{(m)} = (f_1, f_2)$

Figure 7: Proposed method searching concurrently on the original space $\mathbf{f}^{(m)} = (f_1, f_2)$, $m = 2$, $n = 5$, and expanded space $\mathbf{f}^{(M)}$, $M = 4$. Final generation, DTLZ3 problem.

Table 2: C-metric every hundred generations by proposed method with $d = 0.025$ (algorithm A_2) and NSGA-II (algorithm B).

T	$C(B, A_2)$	$C(A_2, B)$
100	0.57	0.94
200	0.53	0.95
300	0.53	0.94
400	0.53	0.94
500	0.54	0.94
600	0.54	0.94
700	0.54	0.94
800	0.53	0.94
900	0.54	0.94
1000	0.53	0.94

Table 3: Percentage of PDS in the final population for the single population approach on $M = 4$.

Pop size	$n = 5$	$n = 10$	$n = 15$
500	59.7	0	0
1000	64.1	0	0
1500	66.5	0	0
2000	68.0	0	0
2500	89.9	0.64	0

Table 4: Percentage of PDS in the final population for the two-population concurrent search approach on $M = 4$.

Pop size	$n = 5$	$n = 10$	$n = 15$
500 (450, 50)	100	93.8	31.6
1000 (900, 100)	100	96.2	57.1
1500 (1350, 150)	100	99.6	75.3
2000 (1800, 200)	100	99.9	85.1
2500 (2250, 250)	100	100	85.2

Table 5: Percentage of PDS in the final population for the two-population concurrent search approach on $M = 5$.

Pop size	$n = 10$	$n = 15$
1500 (1350, 150)	60.4	17.2
2000 (1800, 200)	68.4	21.7
2500 (2250, 250)	64.9	26.8

Table 6: Percentage of PDS in the final population for the concurrent search approach changing the population ratio between extended and original space. $M = 4$, overall population size $|P| = 1000$.

Percentage of population on original space	$n = 5$	$n = 10$	$n = 15$
5% (950, 50)	100	85.9	24.8
10% (900, 100)	100	96.2	57.1
15% (850, 150)	100	99.9	71.6

Results are shown for population sizes 500, 1000, 1500, 2000, and 2500 on DTLZ3 problems with $n = 5$, 10, and 15 variables. Similarly, Table 4 shows results for the two-population concurrent search approach. In general, a reduction in population size or an increase in number of variables leads to a reduction in number of PDS the algorithm finds. Note that the single population approach finds PDS

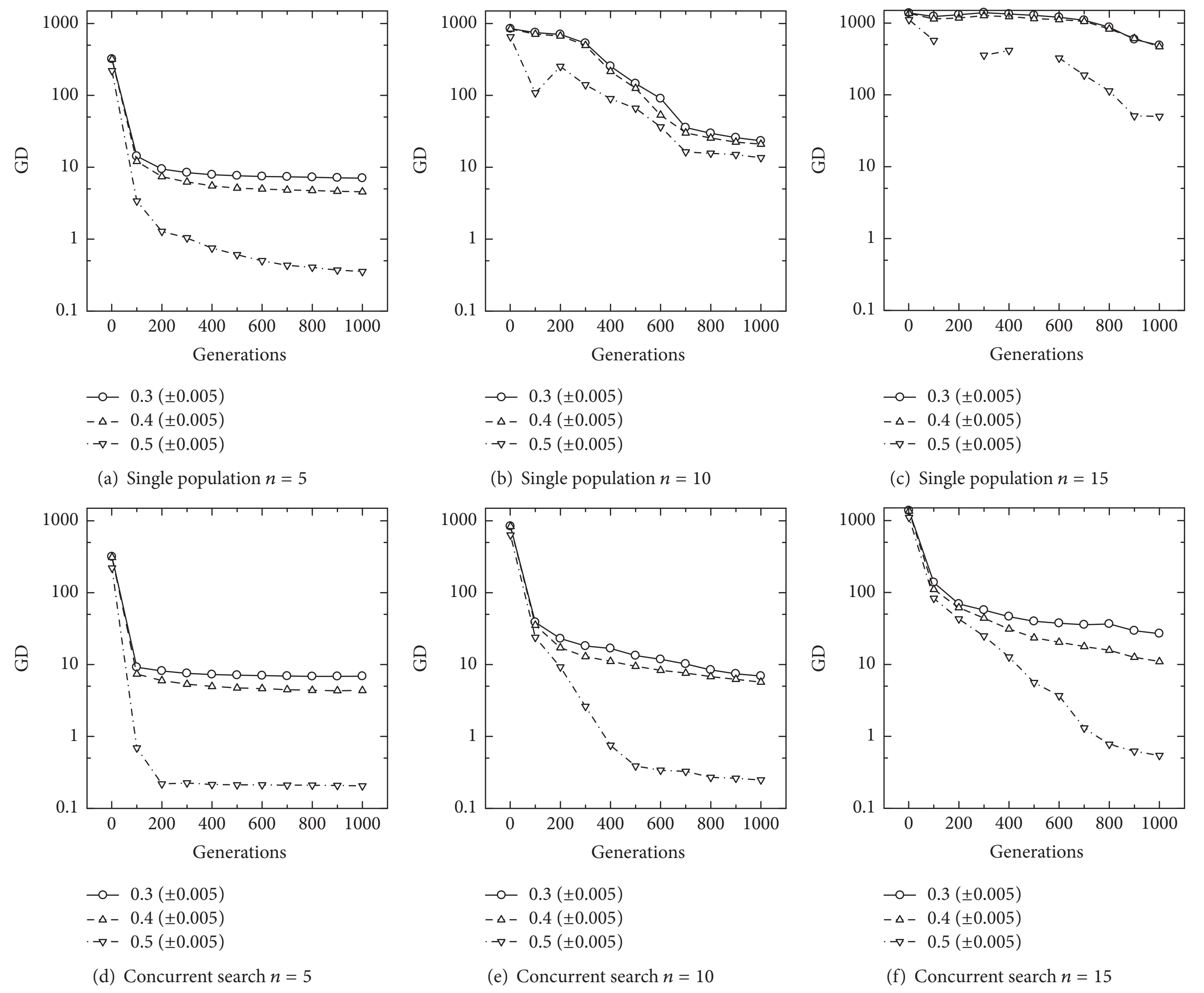


Figure 8: GD over the generations by single population method and proposed concurrent search method. DTLZ3, $m = 2$ original objectives, $M = 4$ objectives in the extended space, and $n = \{5, 10, 15\}$ variables. GD is calculated separately grouping solutions into three groups according the preferred values $x_5 = 0.3$, $x_5 = 0.4$ and optimal value $x_5 = 0.5$. Solutions are considered within a group if $x_5 = 0.3 \pm 0.005$, $x_5 = 0.4 \pm 0.005$, or $x_5 = 0.5 \pm 0.005$, respectively.

Table 7: Percentage of PDS in the final population for the concurrent search approach changing the population ratio between extended and original space. $M = 4$, overall population size $|P| = 2500$.

Percentage of population on original space	$n = 5$	$n = 10$	$n = 15$
5% (2375, 125)	100	100	56.9
10% (2250, 250)	100	100	85.2
15% (2125, 375)	100	100	99.9

solutions for $n = 5$ only when population size is 2500. For $n = 10$ and $n = 15$, not even a population size 2500 is enough to find PDS. On the other hand, the two-population concurrent approach can find PDS even in $n = 15$ with a small population. Note that more than 30% of solutions are PDS for $n = 15$ with an overall population size of 500.

Table 5 shows the percentage of PDS solutions in the final population for the two-population concurrent approach on $M = 5$ (3 objectives in the original space and 2 additional objectives in the extended space). Results are shown for population sizes 1500, 2000, and 2500 on DTLZ3 problems with $n = 10$ and 15 variables. Note that increasing the number of objectives and variables makes it harder for the algorithm to find PDS solutions. This is because the underlying NSGA-II algorithm is less effective in larger dimensional spaces. Nonetheless, the concurrent approach still can find 17.2% PDS for population size 1500 and $n = 15$ variables. In order to get more PDS population size in the original space should be increased, as explained in Table 5. The single population approach cannot find PDS on $M = 5$ objectives.

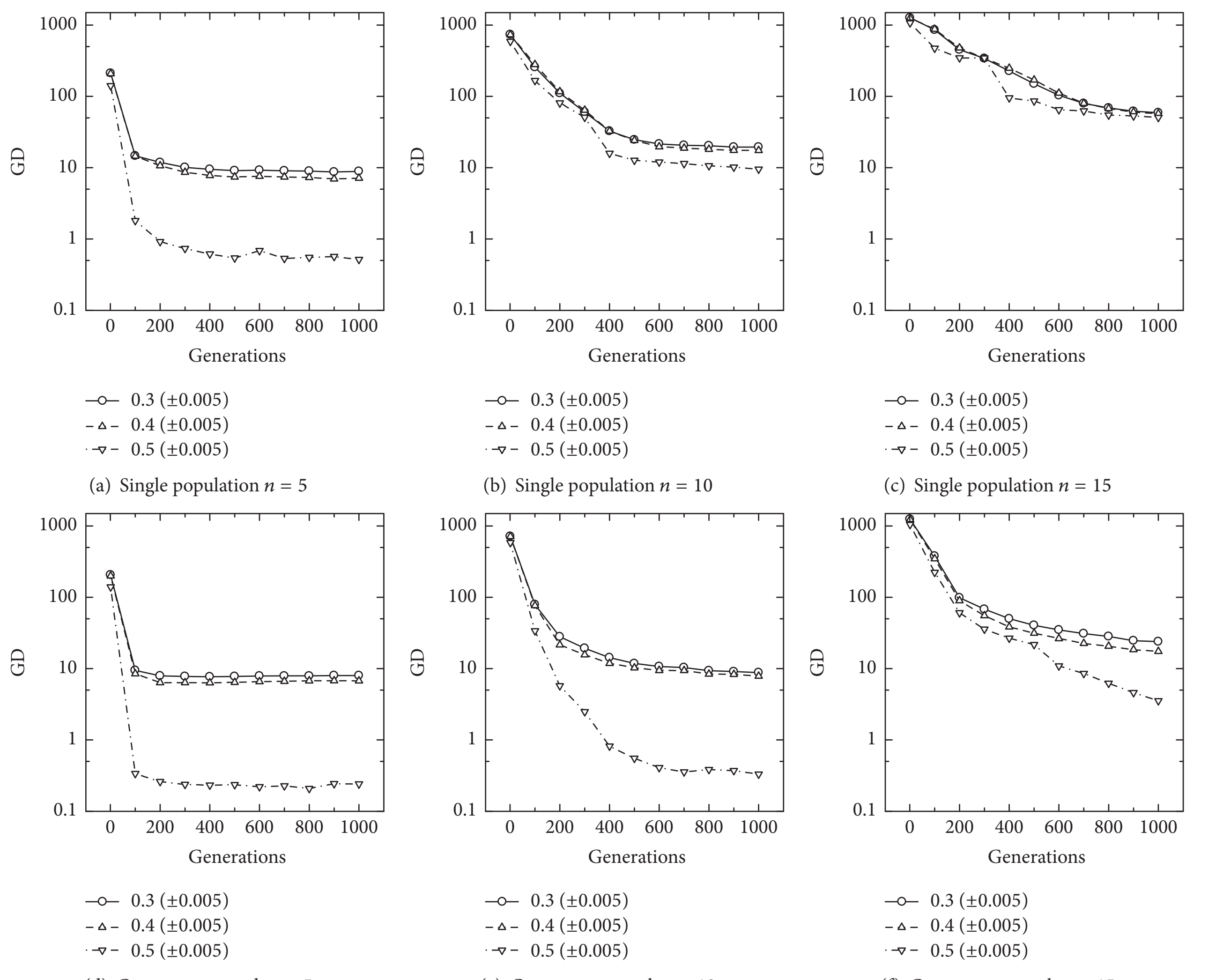
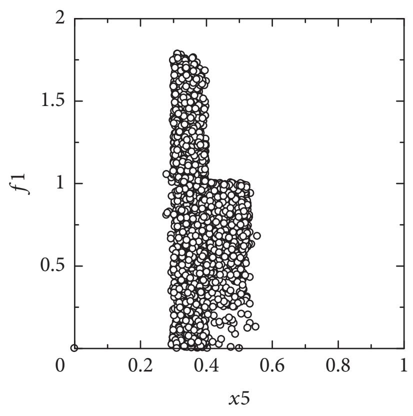


Figure 9: GD over the generations by single population method and proposed concurrent search method. DTLZ3, $m = 3$ original objectives, $M = 5$ objectives in the extended space, and $n = \{5, 10, 15\}$ variables. GD is calculated separately grouping solutions around the preferred values $x_5 = 0.3$ and $x_5 = 0.4$ and optimal value $x_5 = 0.5$. Solutions are considered within a group if $x_5 = 0.3 \pm 0.005$, $x_5 = 0.4 \pm 0.005$, or $x_5 = 0.5 \pm 0.005$, respectively.

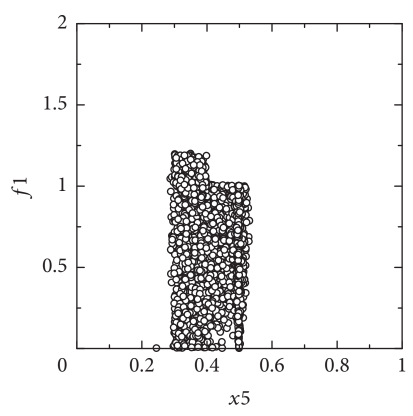
As mentioned above, we set the overall population size to 2500 to compare with the single population approach. For two and three objectives a population of 200 or 300 is commonly used when we search for the Pareto optimal set. Thus, we take this as reference and set the population size to 250 to search in the original space and assign the rest of the overall population to the search in the extended space (2250).

To investigate whether smaller populations in the original and extended space could work, Tables 6 and 7 show the percentage of PDS solutions in the final population for the two-population concurrent approach varying the population ratio between the extended and original space on $M = 4$ objectives and $n = \{5, 10, 15\}$ variables. Results are shown for overall population sizes of 1000 and 2500, respectively. From these tables note that it is crucial to increase the population in the original space to an appropriate size in order to find more PDS solutions rather than increasing the population in the extended space. In general, population size in the extended space should be larger than the population in the original space because of the higher dimensionality of the extended space. However, as shown here, it does not need to be too large as the value 2500 used for comparison with the single population approach.

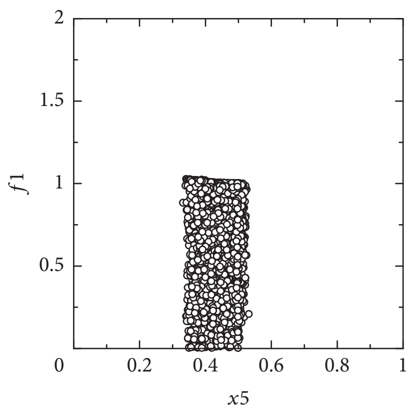
The poor performance of the single population algorithm in the extended spaces of 4 and 5 objectives can be explained from the lack of scalability of the underlying NSGA-II algorithm and the added complexity of finding PDS. Dominance based algorithms, such as NSGA-II, show good optimization performance for multiobjective optimization problems with two or three objectives and are frequently applied to optimize real world problems. However, it is known in the literature that the optimization performance of these kind of algorithms significantly deteriorate as we increase the number of objective functions [23, 24]. A way



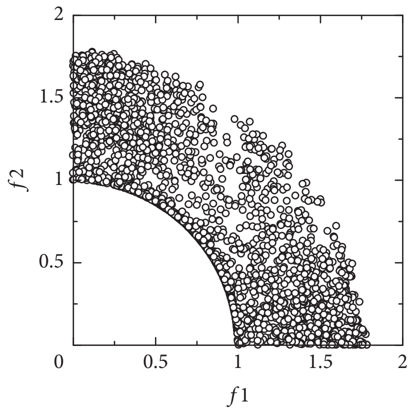
(a) f_1 and x_5 by NSGA-II



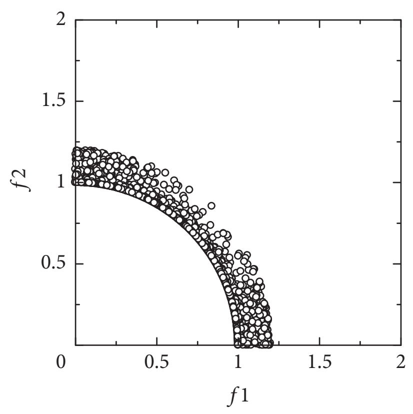
(b) f_1 and x_5 by concurrent search $d = 0.25$



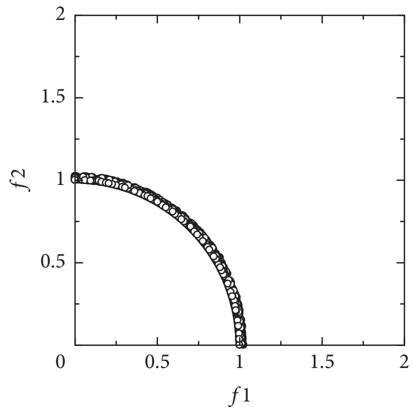
(c) f_1 and x_5 by concurrent search $d = 0.025$



(d) f_1 and f_2 by NSGA-II



(e) f_1 and f_2 by concurrent search $d = 0.25$



(f) f_1 and f_2 by concurrent search $d = 0.025$

Figure 10: Solutions by NSGA-II and proposed method searching concurrently with two populations in the original and extended space on problem. DTLZ2, $m = 2$ original objectives, $M = 4$ objectives in the extended space, and $n = 5$ variables.

to improve performance of these algorithms on problems with 4 or more objectives is to increase the population size [25, 26]. However, the inclusion of the PDS criteria adds to the complexity of the problem and an increase in population size is not enough, as we show in this work.

4.6. Real World Problem. We have also applied the methods discussed above to tire design optimization. The simplest formulation of the problem consists of 2 highly conflicting objectives $\mathbf{f} = (f_1, f_2)$ and 6 real-value design variables $\mathbf{x} = (x_1, x_2, \ldots, x_5, x_6)$, where each variable is defined in the range $x_i = [-r_i, r_i]$. We run a conventional evolutionary multiobjective optimization algorithm for 1000 generations setting its population size to $|P_t| = |Q_t| = 400$ individuals. The Pareto set found by the algorithm (POS^F) is computed from the population at the last generation. The POS^F contains a large number of solutions and the trade-offs in objective space can be clearly seen in Figure 11(a), as expected by the conflicting nature of the objectives. However, analysis of the trade-offs in variable space shows that all solutions in the POS^F take extreme values for variables x_5 and x_6, $x_5 \simeq -3$ and $x_6 \simeq 3$, as shown in Figure 11(b). Tires with these specifications can be constructed; however tires with $x_5 \simeq 0$ and $x_6 \simeq 0$ are preferable because they are known to be physically easier to build.

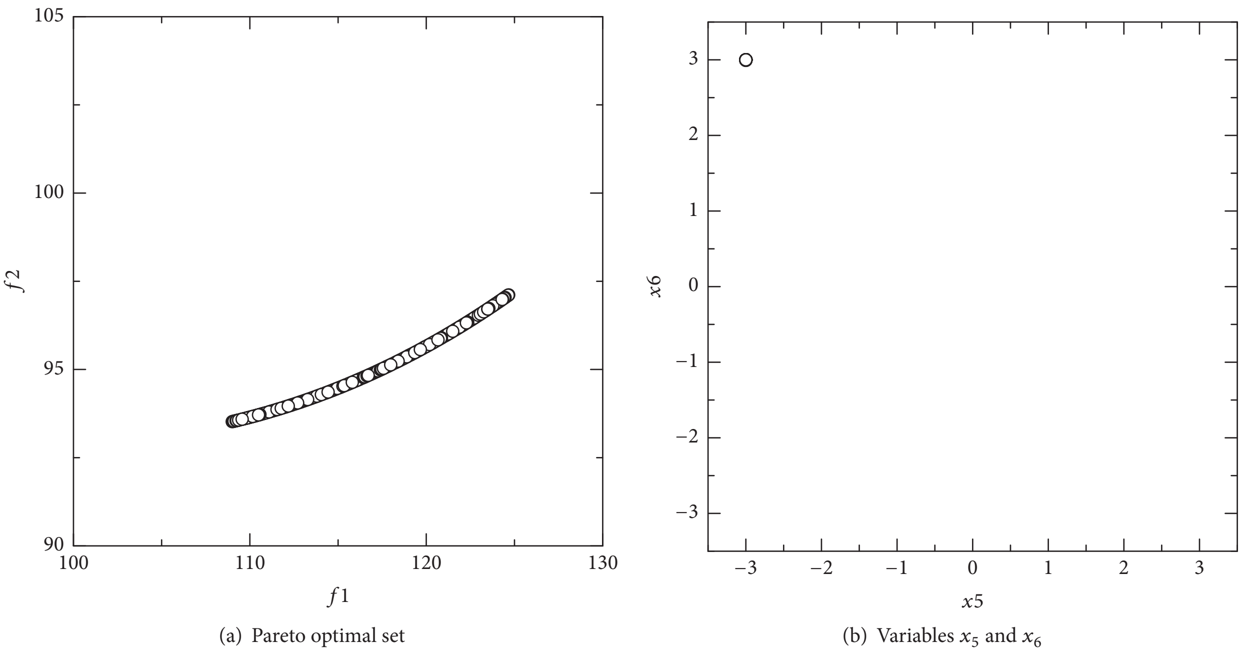
We tried to find solutions around the preferred values by restricting the ranges for variables x_5 and x_6. However, this approach did not work because the solutions obtained are too far from optimality, similar to DTLZ3. Thus, no trade-off design knowledge between tire performance and tire constructability could be extracted.

Similar to benchmark problems, we extended the original objective space $\mathbf{f}^{(m=2)} = (f_1, f_2)$ by adding two functions to form $\mathbf{f}^{(M=4)} = (f_1, f_2, f_3, f_4)$. The two additional functions are as follows:

$$f_3 = |x_5|, \tag{6}$$

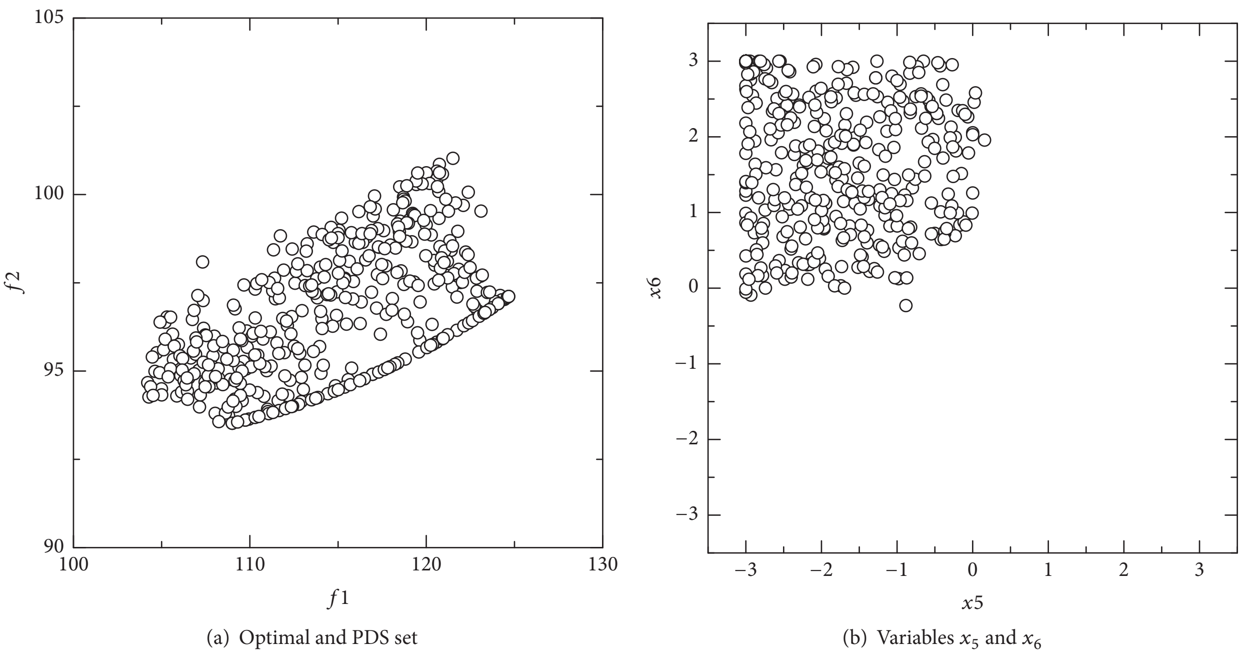
$$f_4 = |x_6|. \tag{7}$$

The threshold distance $d = 5$ used to sort practically desirable solutions is specified by the designer based on knowledge of how much he is prepared to sacrifice in tire performance to favor its constructability. The methods that extend the objective space to include preferred values could find a large number of solutions close to optimality with variables taking values in a broader range, including



(a) Pareto optimal set (b) Variables x_5 and x_6

Figure 11: Results by NSGA-II on real world problem.

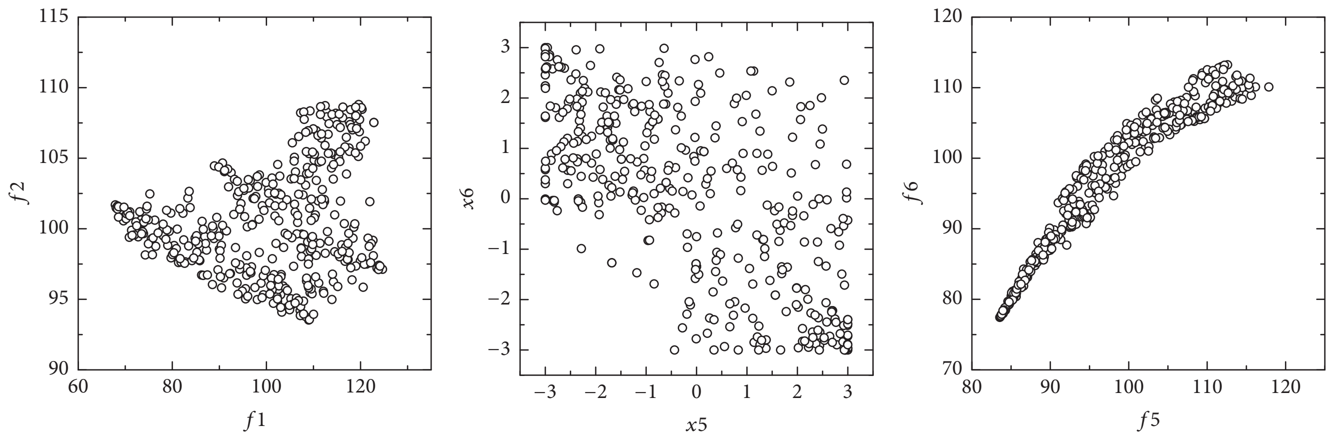


(a) Optimal and PDS set (b) Variables x_5 and x_6

Figure 12: Results by the proposed two-population concurrent evolution method on real world problem, $d = 5$.

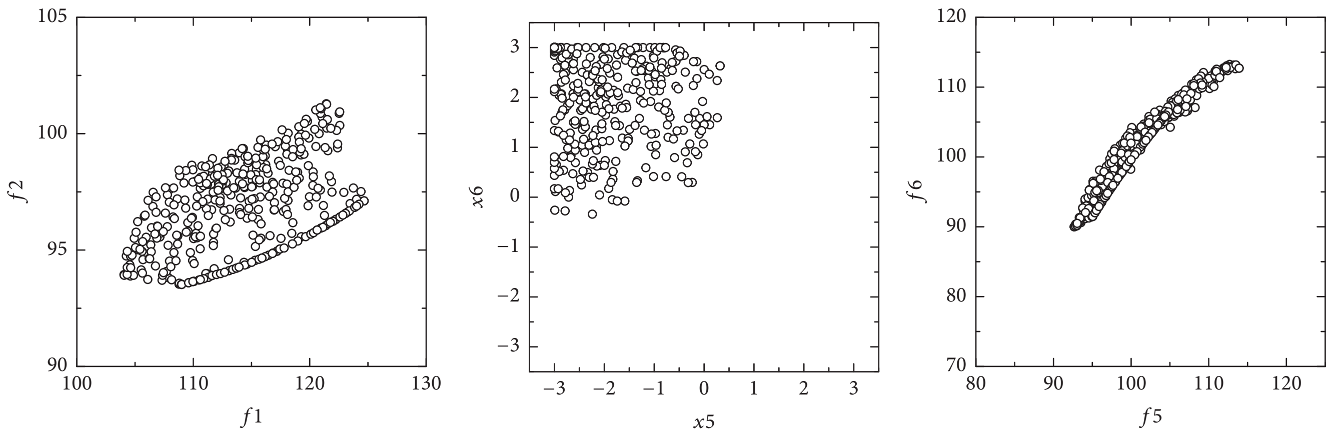
the practically preferred design values. The method that evolves two populations concurrently, set with population sizes $|P_t^A| = |Q_t^A| = 350$ for the extended space and $|P_t^B| = |Q_t^B| = 50$ for the original space, was more successful than the single population approach set with population size $|P_t| = |Q_t| = 400$. Solutions generated by the proposed approach are shown in Figure 12. Note that both optimal and practically desired solutions are obtained by the algorithm, as shown in Figure 12(a), which include a large range of values for x_5 and x_6, including values in the preferred regions around $x_5 = x_6 = 0$, as shown in Figure 12(b). These solutions have proved useful to understand the trade-offs between high-performance and easier to build tires, so that the decision-maker can make an appropriate design decision.

If we color each individual on objective and variable space based on the value of decision variables x_5 and x_6, we can understand visually the trade-off between optimality and constructability by making x_5 and x_6 values approach to 0,



(a) f_1 and f_2 (b) Variables x_5 and x_6 (c) f_5 and f_6

FIGURE 13: Results by NSGA-II on real world problem $\mathbf{f} = (f_1, f_2, f_3, f_4)$.



(a) f_1 and f_2 with PDS (b) Variables x_5 and x_6 (c) f_5 and f_6

FIGURE 14: Results by the proposed two-population concurrent evolution method on real world problem, $\mathbf{f}^{(m=2)} = (f_1, f_2)$, $\mathbf{f}^{(M=6)} = (f_1, f_2, f_3, f_4, f_5, f_6)$, $d = 5$.

that is, the ranges on f_1 and f_2 values that we need to sacrifice in order to ease the implementation or manufacturing of the solution.

We also try the proposed two-population concurrent evolution approach to solve a four-objective formulation of the problem. The preferred plane of objective space is set to $\mathbf{f}^{(m=2)} = (f_1, f_2)$. The extended space is formed by adding four additional fitness functions, so $\mathbf{f}^{(M=6)} = (f_1, f_2, f_3, f_4, f_5, f_6)$, where f_3 and f_4 correspond to the third and fourth objective of the problem definition and f_5 and f_6 are related to preferred variables x_5 and x_6, that is $f_5 = |x_5|$ and $f_6 = |x_6|$. Figure 13 shows results by NSGA-II solving the four-objective formulation $\mathbf{f} = (f_1, f_2, f_3, f_4)$ set with population size $|P_t| = |Q_t| = 400$. Similarly, Figure 14 shows results by the two-population approach optimizing in $\mathbf{f}^{(M=6)} = (f_1, f_2, f_3, f_4, f_5, f_6)$ in population A set with $|P_t^A| = 350$ and $\mathbf{f}^{(m=2)} = (f_1, f_2)$ in population B set with $|P_t^B| = 50$.

From these figures note that the proposed method finds solutions with better converge properties in the objective plane of interest with desirable solutions within the established range. In Figure 13, we obtained nondominated solutions on a 6 objectives space by conventional NSGA-II. Looking at Figure 13(a), note that now we have a suboptimal trade-off in the plane f_1-f_2. In addition, we can see solutions with values of decision variables that are easy to implement, but there are many solutions distributed far from the Pareto front on f_1-f_2 subspace. So, we are not able to use these setting which cannot be used for implementation because the product will be of low quality. Using the two populations approach, we could search the region of interest even if the number of objective functions increases. By considering the optimality of f_5 and f_6, solutions with useful decision variable values for implementation are obtained in the region of interest.

Advanced formulations of this problem include more objectives. In the future we would like to try this approach using a many-objective optimizer [25] instead of the multi-objective optimizer used in this work.

5. Conclusions

In this work we proposed two methods to search practically desirable solutions. The methods are based on an approach that expands the objective space by incorporating additional fitness functions associated with particular decision variables, aiming to find solutions around preferred values of the chosen variables while searching for optimal solutions in the original objective space. The first method evolves concurrently two populations, one in the extended space and the other one in the original space. The population that evolves in the extended space uses an enhanced ranking for survival selection and parent selection that is based on the front number the solution belongs to in the expanded space and its desirability with respect to Pareto optimal solutions computed in the original space. The second method evolves a single population on the expanded space using the enhanced ranking for survival selection and parent selection of the first method.

The proposed methods were compared with an algorithm that simply restricts the range of decision variables around the preferred values and an algorithm that expands the space without constraining the distance from optimality. Our experiments on benchmark problems showed that simply restricting the range of variables is not effective in finding practically desirable solutions. Also, just extending the space without constraining the distance of solutions to the Pareto optimal set in the original space is not effective either. Among the two methods proposed, the one that evolves two populations concurrently can effectively find a large number of practically desirable solutions for 2 and 3 objectives in the original space and 5, 10, and 15 variables. The method that evolves only one population works relatively well just for 5 variables.

We also applied the algorithms discussed in this work to a tire design optimization problem. Similar to the benchmark problem, the method that evolves concurrently a population in the original space and another one in the extended space worked better. Solutions generated by the proposed approach have proved useful to understand the trade-offs between high-performance and easier to build tires, so that the decision-maker can make an appropriate design decision.

In the future we would like to test the proposed approaches on other kinds of problems. Also, we would like to use many-objective optimizers for the search on the extended space, particularly for problem formulations where the original space is already a many-objective optimization problem.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley & Sons, Chichester, UK, 2001.

[2] C. A. Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5 of *Genetic Algorithms and Evolutionary Computation*, Kluwer Academic Publishers, Boston, Mass, USA, 2002.

[3] C. Zopounidis and M. Doumpos, *Multiple Criteria Decision Making: Applications in Management and Engineering*, Springer International Publishing, Berlin, Germany, 2017.

[4] C. A. C. Coello, "Handling preferences in evolutionary multiobjective optimization: a survey," in *Proceedings of the 2000 IEEE Congress on Evolutionary Computation*, vol. 1, pp. 30–37, Piscataway, NJ, USA, July 2000.

[5] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06)*, pp. 635–642, Seattle, DC, USA, July 2006.

[6] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010.

[7] I. Giagkiozis and P. J. Fleming, "Pareto front estimation for decision making," *Evolutionary Computation*, vol. 22, no. 4, pp. 651–678, 2014.

[8] R. Cheng, M. Olhofer, and Y. Jin, "Reference vector based a posteriori preference articulation for evolutionary multiobjective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '15)*, pp. 939–946, Sendai, Japan, May 2015.

[9] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.

[10] G. W. Greenwood, X. Hu, and J. G. D'Ambrosio, "Fitness functions for multiple objective optimization problems: combining preferences with pareto rankings," in *Proceedings of the 4th Workshop on Foundations of Genetic Algorithms*, pp. 437–455, Fitness functions for multiple objective optimization problems, San Diego, Claif, USA, August 1996.

[11] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proceedings of the 7th International Conference on Natural Computation (ICNC '11)*, pp. 1260–1264, Shanghai, China, July 2011.

[12] R. Battiti and A. Passerini, "Brain-computer evolutionary multiobjective optimization: a genetic algorithm adapting to the decision maker," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 671–687, 2010.

[13] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software*, vol. 32, no. 6, pp. 499–507, 2001.

[14] J. W. Fowler, E. S. Gel, M. M. Köksalan, P. Korhonen, J. L. Marquis, and J. Wallenius, "Interactive evolutionary multiobjective optimization for quasi-concave preference functions," *European Journal of Operational Research*, vol. 206, no. 2, pp. 417–425, 2010.

[15] T. Wagner and H. Trautmann, "Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 688–701, 2010.

[16] T. Wagner, H. Trautmann, and D. Brockhoff, "Preference Articulation by Means of the R2 Indicator," in *Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization (EMO '13)*, vol. 7811 of *Lecture Notes in Computer Science*, pp. 81–95, Springer, Sheffield, UK, March 2013.

[17] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Proceedings of the 9th Annual Genetic and Evolutionary Computation Conference (GECCO '07)*, pp. 781–788, London, UK, July 2007.

[18] X. Ma, F. Liu, Y. Qi et al., "MOEA/D with biased weight adjustment inspired by user preference and its application on multi-objective reservoir flood control problem," *Soft Computing*, vol. 20, no. 12, pp. 4999–5023, 2016.

[19] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb, "Integrating user preferences and decomposition methods for many-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '14)*, pp. 421–428, Beijing, China, July 2014.

[20] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," KanGAL report 200001, 2000.

[21] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 825–830, Honolulu, Hawaii, USA, May 2002.

[22] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: methods and applications [Ph.D. thesis]*, Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.

[23] H. E. Aguirre and K. Tanaka, "Working principles, behavior, and performance of MOEAs on MNK-landscapes," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1670–1690, 2007.

[24] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: a short review," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 2419–2426, Hong Kong, China, June 2008.

[25] H. Aguirre, A. Oyama, and K. Tanaka, "Adaptive ε-sampling and ε-hood for evolutionary many-objective optimization," in *Proceedings of the 7th International Conference on Evolutionary Multi-criterion Optimization (EMO '13)*, vol. 7811 of *Lecture Notes in Computer Science*, pp. 322–336, Sheffield, UK, March 2013.

[26] H. Aguirre, A. Liefooghe, S. Verel, and K. Tanaka, "Effects of population size on selection and scalability in evolutionary many-objective optimization," in *Proceedings of the 7th International Conference on Learning and Intelligent Optimization*, vol. 7997 of *Lecture Notes in Computer Science*, pp. 450–454, Catania, Italy, January 2013.