An improved MOEA/D algorithm for bi-objective optimization problems with complex Pareto fronts and its application to structural optimization

Ho-Huu, V.; Hartjes, S.; Visser, H. G.; Curran, R.

DOI
10.1016/j.eswa.2017.09.051

Publication date
2018

Document Version
Accepted author manuscript

Published in
Expert Systems with Applications

Citation (APA)
Ho-Huu, V., Hartjes, S., Visser, H. G., & Curran, R. (2018). An improved MOEA/D algorithm for bi-objective optimization problems with complex Pareto fronts and its application to structural optimization. *Expert Systems with Applications*, *92*, 430-446. https://doi.org/10.1016/j.eswa.2017.09.051

An improved MOEA/D algorithm for bi-objective optimization problems with complex Pareto fronts and its application to structural optimization

V. Ho-Huu*, S. Hartjes, H. G. Visser, R. Curran

Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

E-mails: V.HoHuu@tudelft.nl (V. Ho-Huu), S.Hartjes@tudelft.nl (S. Hartjes), h.g.visser@tudelft.nl (H. G. Visser), R.Curran@tudelft.nl (R. Curran)

Abstract

The multi-objective evolutionary algorithm based on decomposition (MOEA/D) has been recognized as a promising method for solving multi-objective optimization problems (MOPs), receiving a lot of attention from researchers in recent years. However, its performance in handling MOPs with complicated Pareto fronts (PFs) is still limited, especially for real-world applications whose PFs are often complex featuring, e.g., a long tail or a sharp peak. To deal with this problem, an improved MOEA/D (named iMOEA/D) that mainly focuses on bi-objective optimization problems (BOPs) is therefore proposed in this paper. To demonstrate the capabilities of iMOEA/D, it is applied to design optimization problems of truss structures. In iMOEA/D, the set of the weight vectors defined in MOEA/D is numbered and divided into two subsets: one set with odd-weight vectors and the other with even-weight vectors. Then, a two-phase search strategy based on the MOEA/D framework is proposed to optimize their corresponding populations. Furthermore, in order to enhance the total performance of iMOEA/D, some recent developments for MOEA/D, including an adaptive replacement strategy and a stopping criterion, are also incorporated. The reliability, efficiency and applicability of iMOEA/D are investigated through seven existing benchmark test functions with complex PFs and three optimal design problems of truss structures. The obtained results reveal that iMOEA/D generally outperforms MOEA/D and NSGA-II in both benchmark test functions and real-world applications.

Keywords: *multi-objective evolutionary algorithm (MOEA), multi-objective evolutionary algorithm based on decomposition (MOEA/D), complicated Pareto fronts* (*PFs*), *structural optimization, truss structures*.

* Corresponding author: V. Ho-Huu, Email: V.HoHuu@tudelft.nl

1. Introduction

In many real-world engineering applications, for example, structural optimization (Cai & Aref, 2015; Vo-Duy et al., 2017) aircraft trajectory optimization (Hartjes & Visser, 2016; Hartjes, Visser, & Hebly, 2010; Visser & Hartjes, 2014), optimal design problems often have multiple conflicting objectives, and are known as multi-objective optimization problems (MPOs) (Kalyanmoy Deb, 2001). By solving these problems, a set of trade-off solutions between objectives can be found. From this set, decision makers can select the most suitable solutions, which may help them save much time and/or money. Solving real-world MOPs, however, is usually a challenging task for the decision makers because of the complexities of MOPs such as high nonlinearity, non-convexity and discontinuity (Grandhi, 1993). Therefore, the development of efficient optimization methods for coping with these problems becomes more important and attracts much attention from researchers (Trivedi et al., 2016).

Among different approaches, multi-objective evolutionary algorithms (MOEAs) have been recognized as well-suited methods for solving such MOPs since they are capable of approximating multiple non-dominated solutions in a single run (Kalyanmoy Deb, 2001; K. Li et al., 2015). One of the recent effective methods is the multi-objective evolutionary algorithm based on decomposition (MOEA/D) (Zhang & Li, 2007). In MOEA/D, an MOP is decomposed into a set of scalar optimization sub-problems, and these sub-problems are solved simultaneously in a collaborative manner. Owing to the diversity maintenance of sub-problems and the information sharing between individuals dwelling in a neighborhood, MOEA/D may acquire well-distributed solutions over a Pareto front (PF). In a comparative study with the non-dominated sorting genetic algorithm (NSGA-II) (H. Li & Zhang, 2009), the obtained results showed that MOEA/D outperforms NSGA-II in terms of both the quality of solutions and convergence rate. In addition, the efficiency of MOEA/D is also proven through real-world applications such as wireless sensor networks (Konstantinidis & Yang, 2012), route planning (Waldock & Corne, 2011), and economic emission dispatch (Zhu, Wang, & Qu, 2014). Nevertheless, recent research in (Jiang & Yang, 2016; Qi et al., 2013; Wang et al., 2017; Yang, Jiang, & Jiang, 2016) indicated that MOEA/D often only effectively handles MOPs with simple PFs, while it is not good at solving MOPs with complex PFs exhibiting features such as a long tail or a sharp peak.

To enhance the capability of MOEA/D in solving MOPs with complicated PFs, some different approaches have been proposed in recent literature. Qi et al., 2013 proposed an improved MOEA/D with an adaptive weight adjustment. In this version, a new weight vector initialization method is introduced for generating the set of initial weight vectors, and an

adaptive weight vector adjustment strategy is developed to detect the overcrowded solutions on a PF and create new candidates to replace them. Yang et al., 2016 investigated the impact of penalty factors in the penalty boundary intersection (PBI) on the spread of a PF, and then proposed a new variant of MOEA/D with two different penalty schemes. Jiang & Yang, 2016 developed an improved MOEA/D with a new search strategy, namely MOEA/D-TPN, where the optimization procedure is divided into two phases. The solving process of an MOP is started with the first phase, and after this phase terminates, the crowded information of obtained solutions is evaluated to decide whether or not the second phase is continued. Wang et al., 2017 examined, for the first time, the influence of the ideal point and the nadir point in the Tchebycheff function on the distribution of optimal solutions over a PF and then developed a new improved MOEA/D, in which both these points are integrated into the Tchebycheff function to decompose an MOP into a number of scalar optimization sub-problems. Through the evaluations on benchmark test functions in (Jiang & Yang, 2016; Qi et al., 2013; Wang et al., 2017; Yang et al., 2016), it was shown that most of these developed methods outperform MOEA/D and some other available methods. They, however, often require more computational procedures or have more control parameters than MOEA/D, which may lead to certain limitations for engineering designers in applying them to real-world applications. For example, the method in (Qi et al., 2013) has an extra algorithm for detecting and replacing overcrowded sub-problems during the optimization process, while the method in (Jiang & Yang, 2016) requires a pre-defined reasonable number of evaluations for the distribution of computational resources for both Phases.

From the review of the above methods and motivated by the desire to solve complex real-world problems, e.g. structural optimization problems, an improved MOEA/D (iMOEA/D) which only focuses on bi-objective optimization problems (BOPs) is proposed in this paper. The iMOEA/D algorithm is developed based on the studies in (Jiang & Yang, 2016; Wang et al., 2017), where the advantages of using the ideal point and the nadir point, and the benefits of solving an MOP based on two phases are exploited. In iMOEA/D, the solving process of a bi-objective optimization problem (BOP) is separated into two parts, and for each part the general framework of MOEA/D is applied. Firstly, the set of weight vectors of MOEA/D is numbered and divided into two subsets: one set with odd-weight vectors and the other with even-weight vectors. Then, a simple two-phase search strategy is developed to optimize the corresponding populations of each set. In the proposed search strategy, the optimization process of the two phases is almost the same except for the use of the Tchebycheff function type (Miettinen, 1999) in which either the ideal point or the nadir point is utilized. The search

of iMOEA/D is started at the first phase with the set of odd-weight vectors. In this phase, the Tchebycheff function with the ideal point is utilized. After that, the second phase is continued with the set of even-weight vectors, and the Tchebycheff function with the nadir point $\mathbf{z}^{nad}$ is used.

Furthermore, to enhance the overall performance of the algorithm, some recent developments related to MOEA/D consisting of an adaptive replacement strategy (Wang et al., 2016) and a stopping criterion (Abdul Kadhar & Baskar, 2016) are also integrated into iMOEA/D. To estimate the performance of the proposed algorithm, seven existing benchmark test functions with complicated PFs are tested first, and then three structural optimization problems of truss structures are solved. The efficiency and reliability of iMOEA/D are also compared with those of MOEA/D, MOEA/D-TPN and NSGA-II.

The rest of the paper is structured as follows. Section 2 provides some basic backgrounds of an MOP and the Tchebycheff decomposition method. Section 3 presents a general framework of the MOEA/D algorithm, in which some recent developments are also included. The iMOEA/D algorithm is described in Section 4. Experimental studies are presented in Section 5, and some conclusions are drawn in Section 6.

2. Backgrounds

2.1. Basic definitions

A multi-objective optimization problem (MOP) can be stated as follows:

$$\begin{aligned} \min \quad & F(\mathbf{x}) = (f_1(\mathbf{x}),..., f_m(\mathbf{x}))^{\mathrm{T}} \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1,..., x_n)^{\mathrm{T}} \in \mathbf{R}^n$ is the vector of design variables, Ω is the feasible search domain, $f_j(\mathbf{x})$ is the *j*th objective function, and *m* is the number of objective functions.

In multi-objective optimization, some basic definitions in the context of minimization problems are given as follows:

- Let $\mathbf{x}_1, \mathbf{x}_2 \in \Omega$ be two solutions of an MOP, $\mathbf{x}_1$ is said to *dominate* $\mathbf{x}_2$ (denoted $\mathbf{x}_1 \prec \mathbf{x}_2$), if and only if $f_j(\mathbf{x}_1) \le f_j(\mathbf{x}_2), \forall j \in \{1,...,m\}$, and $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ for at least one index $j \in \{1,...,m\}$.
- Let $\mathbf{x}^* \in \Omega$ be called *Pareto optimal* if there is no other solution in Ω which dominates $\mathbf{x}^*$.

- The set of Pareto optimal solutions in Ω are called the *Pareto set* (PS), which is determined by $PS = \{\mathbf{x}^* \mid \neg\exists \mathbf{x} \in \Omega, \mathbf{x} \prec \mathbf{x}^*\}$. The corresponding objective vectors of the solutions in *PS* is called the *Pareto front* (PF) and defined as $PF = \{F(\mathbf{x}) \mid \mathbf{x} \in PS\}$.
- A point $\mathbf{z}^* = (z_1^*, ..., z_m^*)^{\mathrm{T}}$ is called the ideal point if $z_j^* < \min\{f_j(\mathbf{x}) \mid \mathbf{x} \in \Omega, j = 1, ..., m\}$.
- A point $\mathbf{z}^{\mathrm{nad}} = (z_1^{\mathrm{nad}}, ..., z_m^{\mathrm{nad}})^{\mathrm{T}}$ is called the nadir point if $z_j^{\mathrm{nad}} > \max\{f_j(\mathbf{x}) \mid \mathbf{x} \in \Omega, j = 1, ..., m\}$.

2.2. Tchebycheff decomposition approach

Over the past decades, many approaches have been proposed for decomposing an MOP into a set of scalar optimization sub-problems, and can be found in Refs. (Das & Dennis, 1998; Messac, Ismail-Yahaya, & Mattson, 2003). Among these approaches, the weighted Tchebycheff approach is the most widely utilized because of its capability of handling multi-objective optimization problems with non-convex Pareto fronts (Zhang & Li, 2007). A scalar optimization subproblem based on the weighted Tchebycheff approach with the ideal point is determined by

$$\begin{aligned} \min \quad & g^{te}(\mathbf{x} \mid \mathbf{w}, \mathbf{z}^*) = \max_{1 \le j \le m} \left\{ w_j \left| f_j(\mathbf{x}) - \mathbf{z}_j^* \right| \right\} \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{2}$$

where $\mathbf{w} = (w_1, ..., w_m)^{\mathrm{T}}, (w_j \ge 0, \sum_{j=1}^{m} w_j = 1, j = 1, ..., m)$ is the weight vector of the scalar optimization subproblem, and $\mathbf{z}^*$ is the ideal point.

The Tchebycheff approach in Eq. (2) is often only suitable for MOPs with normalized objective functions. Thus, when the ranges of the objectives are on very different scales, the Tchebycheff function is defined as follows (Zhang & Li, 2007):

$$\begin{aligned} \min \quad & g^{te}(\mathbf{x} \mid \mathbf{w}, \mathbf{z}^*) = \max_{1 \le j \le m} \left\{ w_j \left| \frac{f_j(\mathbf{x}) - \mathbf{z}_j^*}{\mathbf{z}_j^{\mathrm{nad}} - \mathbf{z}_j^*} \right| \right\} \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{3}$$

According to Jiang & Yang, 2016, the scalar optimization subproblem may also be formulated by using the nadir point as follows:

$$\begin{aligned} \max \quad & g^{te}(\mathbf{x} \mid \mathbf{w}, \mathbf{z}^{\mathrm{nad}}) = \min_{1 \le j \le m} \left\{ w_j \left(\mathbf{z}_j^{\mathrm{nad}} - f_j(\mathbf{x}) \right) \right\} \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{4}$$

Similarly, to deal with MOPs whose objectives are on very different scales, the sub-problem in Eq. (4) is defined in the following form:

$$\begin{aligned} \max \quad & g^{te}(\mathbf{x} \mid \mathbf{w}, \mathbf{z}^{\text{nad}}) = \min_{1 \le j \le m} \left\{ w_j \left(\frac{\mathbf{z}_j^{\text{nad}} - f_j(\mathbf{x})}{\mathbf{z}_j^{\text{nad}} - \mathbf{z}^*} \right) \right\} \\ \text{s.t.} \quad & \mathbf{x} \in \Omega \end{aligned} \tag{5}$$

3. MOEA/D algorithm

The multi-objective evolutionary algorithm based on decomposition (MOEA/D), was firstly developed by Zhang & Li, 2007, and has been recognized as one of the most popular multi-objective evolutionary algorithms to date (Trivedi et al., 2016). In MOEA/D, MOPs are decomposed into a number of scalar optimization sub-problems by applying decomposition approaches, and these sub-problems are optimized concurrently by mean of using evolutionary algorithms. By employing different decomposition methods and different evolutionary algorithms, various versions of MOEA/D have been developed in recent years such as MOEA/D-DE (H. Li & Zhang, 2009), MOEA/D-DRA (Zhang, Liu, & Li, 2009), MOEA/D-XBS (Zhang & Li, 2007), and MOEA/D-GR (Wang et al., 2016). Although different variants of MOEA/D are available in the literature, there is no single MOEA/D version that combines the many distinct advantages of the various versions. With the aim of developing an efficient version of MOEA/D for real-life problems, a MOEA/D version is therefore developed in this study which is a combination of MOEA/D-DE (H. Li & Zhang, 2009) with a an adaptive replacement strategy (Wang et al., 2016), and a stopping condition criterion (Abdul Kadhar & Baskar, 2016). The general framework of MOEA/D is presented in Algorithm 1.

Algorithm 1. Pseudo-code of MOEA/D algorithm

Input:

- A multi-objective optimization problem as Eq. (1);
- A stopping criterion;
- N: number of sub-problems;
- $\mathbf{w}^i = (w_1^i, ..., w_m^i)^{\mathrm{T}}, i = 1, ..., N$ **:** a set of N weight vectors;
- T_{m}: size of mating neighborhood;
- $T_{r\max}$: maximum size of replacement neighborhood;
- δ: the probability that mating parents are selected from the neighborhood;
- *MaxIter*: maximum iteration;
- *FEs* = 0: the number of function evaluations;

Step 1. Initialization

1.1. Find the T_{m} closest weight vectors to each weight vector based on the Euclidean distances of any two weight vectors. For each sub-problem i = 1,…, N set $\mathbf{B}^i = \{i_1, ..., i_{T_{\mathrm{m}}}\}$ where $\mathbf{w}^{i_1}, ..., \mathbf{w}^{i_{T_{\mathrm{m}}}}$ are the closest weight vectors to $\mathbf{w}^i$;

1.2. Create an initial population $\mathbf{P} = \{\mathbf{x}^1, ..., \mathbf{x}^N\}$ by uniformly randomly sampling from Ω. Evaluate the fitness value FV^i of each solution $\mathbf{x}^i$, i.e. $FV^i = (f_1(\mathbf{x}^i), ..., f_m(\mathbf{x}^i))$ and set $\mathbf{FV} = \{FV^1(\mathbf{x}^1), ..., FV^N(\mathbf{x}^N)\}$;

1.3. Initialize ideal point $\mathbf{z}^* = (z_1^*, ..., z_m^*)^{\mathrm{T}}$ by setting $z_j^* = \min\{f_j(\mathbf{x}) \mid \mathbf{x} \in \Omega, j = 1, ..., m\}$ and nadir point $\mathbf{z}^{\mathrm{nad}} = (z_1^{\mathrm{nad}}, ..., z_m^{\mathrm{nad}})^{\mathrm{T}}$ by setting $z_j^{\mathrm{nad}} = \max\{f_j(\mathbf{x}) \mid \mathbf{x} \in \Omega, j = 1, ..., m\}$;

1.4. Set *FEs* = *FEs* + *N*, and generation: *gen* = 1;

Step 2. Update

while (the stopping condition is not satisfied)

for *i* = 1,…, *N*; do

2.1. Selection of mating/update range

Set $\mathbf{B}_m = \begin{cases} \mathbf{B}^i & \text{if } rand < \delta \\ \{1, ..., N\} & \text{otherwise} \end{cases}$,

where *rand* is a uniformly distributed random number in [0,1];

2.2. Reproduction: randomly select three parent individuals r_1, r_2, r_3 ($r_1 \neq r_2 \neq r_3 \neq i$) from $\mathbf{B}_m$ and generate a solution $\bar{\mathbf{y}}$ by applying "DE/rand/1" operator, and then perform a mutation operator on $\bar{\mathbf{y}}$ to create a new solution $\mathbf{y}$;

2.3. Repair: if any element of $\mathbf{y}$ is out of Ω, its value will be randomly regenerated inside Ω;

2.4. Evaluate the fitness value of new solution $\mathbf{y}$;

2.5. Update of $\mathbf{z}^*, \mathbf{z}^{\mathrm{nad}}$: for each j = 1,…, m if $z_j^* \leq f_j(\mathbf{x}^i)$ then set $z_j^* = f_j(\mathbf{x}^i)$, and if $z_j^{\mathrm{nad}} \geq f_j(\mathbf{x}^i)$ then set $z_j^{\mathrm{nad}} = f_j(\mathbf{x}^i)$;

2.6. Update of solution: use an adaptive replacement strategy in (Wang et al., 2016):

2.6.1. Find the most suitable sub-problem k for the new solution $\mathbf{y}$ by $k = \arg\min_{1 \leq t \leq N}\{g_t^{te}(\mathbf{y})\}$;

2.6.2. Define the maximum numbers of solutions which may be replaced by the new solution $\mathbf{y}$ in the neighborhood set of the sub-problem k by

$T_r = \left\lceil \frac{T_{r\max}}{1 + \exp\left(-20 \times \left(\frac{gen}{MaxIter} - \gamma\right)\right)} \right\rceil$, where $\lceil . \rceil$ is the ceiling function, γ is the control parameter in [0, 1);

2.6.3. Set $\mathbf{B}_r = \mathbf{B}^k\{l\}, l = 1, ..., T_r$ to be the set of solutions in the neighborhood set of the sub-problem k which can be replaced by the new solution $\mathbf{y}$;

2.6.4. For each solution $\mathbf{x}^l$ in $\mathbf{B}_r$, replace $\mathbf{x}^l$ by $\mathbf{y}$ if $g^{te}(\mathbf{y}) \leq g^{te}(\mathbf{x}^l)$;

end for

Set *FEs* = *FEs* + *N*, and *gen* = *gen* + 1;

Step 3. Stopping condition

Use a stopping criterion in (Abdul Kadhar & Baskar, 2016).

if (stopping criterion is satisfied or *MaxIter* is reached)

Stop the algorithm;

end if

end while

Output:

- Pareto set $\mathbf{PS} = \{\mathbf{x}^1, ..., \mathbf{x}^N\}$;
- Pareto front $\mathbf{PF} = \{FV^1(\mathbf{x}^1), ..., FV^N(\mathbf{x}^N)\}$.

In Step 2.2, each element $\bar{y}_k$ of solution $\bar{\mathbf{y}} = (\bar{y}_1, ..., \bar{y}_n)^T$ is generated by using DE/rand/1 operator (Storn & Price, 1997) as follows:

$$\bar{y}_k = \begin{cases} x_k^{r_1} + F(x_k^{r_2} - x_k^{r_3}) & \text{with probability } CR, \\ x_k^{r_1}, & \text{with probability } (1 - CR) \end{cases} \tag{6}$$

where F and CR are two control parameters. The polynomial mutation operator used to create the new solution $\mathbf{y}$ is defined as follows:

$$y_k = \begin{cases} \bar{y}_k + \sigma_k \times (x_{\mathrm{ub}_k} - x_{\mathrm{lb}_k}) & \text{with probability } p_m, \\ \bar{y}_k & \text{with probability } (1 - p_m) \end{cases} \tag{7}$$

where

$$\sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1 & \text{if } rand < 0.5, \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \tag{8}$$

where *rand* is the uniformly distributed random number from [0, 1]. The distribution index η and the mutation rate p_m are two control parameters, and x_{lb} and x_{ub} are the lower and upper bounds of the kth design variable, respectively.

In Step 3, the stopping criterion named maximum Tchebycheff objective error (*MTOE*) (Abdul Kadhar & Baskar, 2016) is utilized. The method uses the information of the Tchebycheff objectives of all sub-problems to set a stopping condition for the algorithm. Firstly, the maximum value of Tchebycheff objective error (*TOE*) is computed based on the absolute difference between the current and previous generation's Tchebycheff objectives. Then, the maximum value in the set of *TOE* (*MTOE*) at the current generation is determined.

$$MTOE_{gen} = \max_{1 \le i \le N} \{TOE_i\} \tag{9}$$

Finally, a χ^2test is applied for statistical evaluation of the variations in the *MTOE* values during previous g generations (Sharma & Rangaiah, 2013; Wagner & Trautmann, 2010). If the variation of *MTOE* is smaller than a pre-defined tolerance value (ε), then the search process of the algorithm is terminated.

$$Chi(MTOE) = \frac{\text{variance}[MTOE_1, MTOE_2, ..., MTOE_\gamma](g-1)}{\varepsilon^2} \tag{10}$$

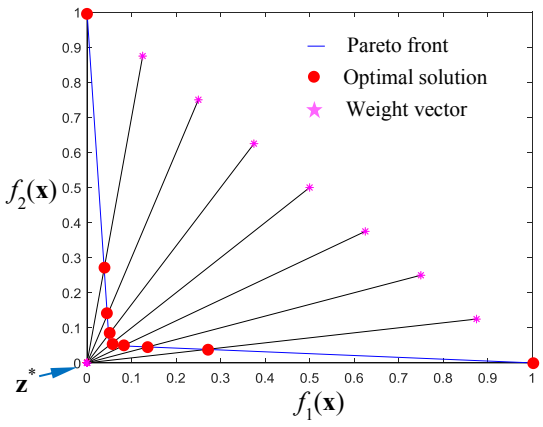
$$p(MTOE) = \chi^2\text{test}[Chi(MTOE), (g-1)] \tag{11}$$

where $MTOE_1, ..., MTOE_\gamma$ are the *MTOE* values of the previous g generations, ε is the expected tolerance value for the standard deviation of *MTOE*, *Chi*(*MTOE*) is the test statistic, and *p*(*MTOE*) is the probability of which the χ^2test supports the hypothesis that the variance

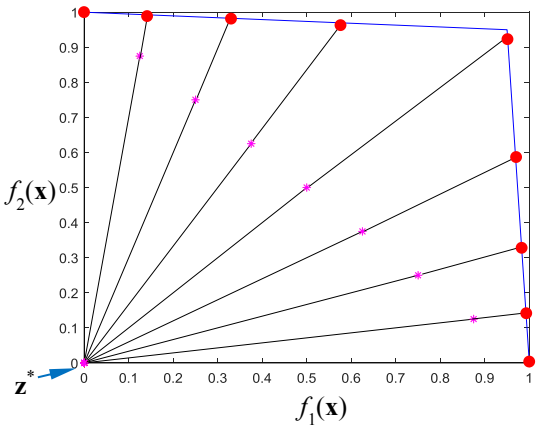
of *MTOE* is lower than the pre-defined tolerance ε. If the probability p is equal to or larger than 99%, the algorithm terminates its search process. The probability p is defined by referring to the lookup table of χ^2 distribution for $(g - 1)$ degrees of freedom, where g is the number of generations. For instance, if the number of generation g is set to be 10, to get a probability p of 99%, then *Chi*(*MTOE*) must be smaller than or equal to 2.088.

4. iMOEA/D algorithm

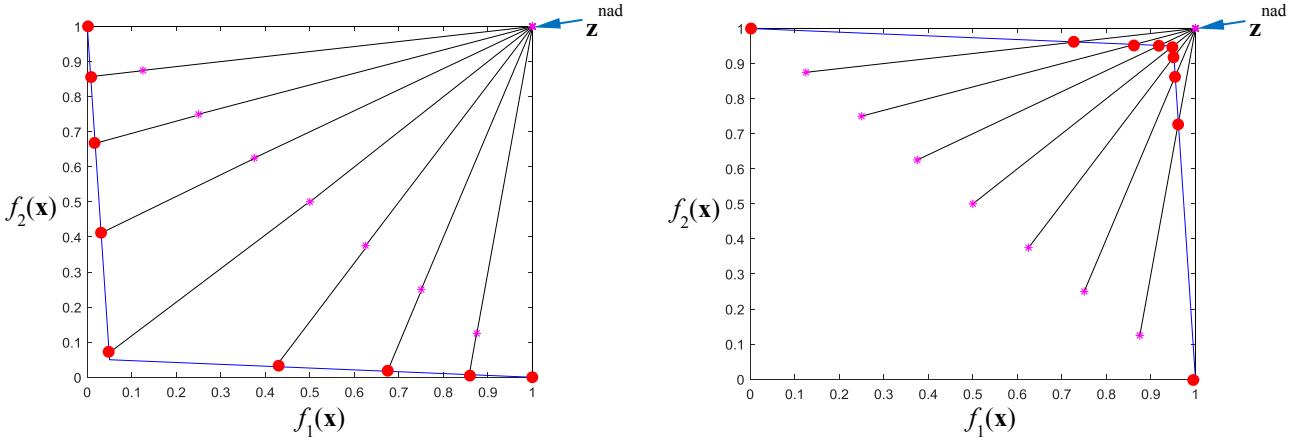
As pointed out by Wang et al., 2017, the use of the ideal point $\mathbf{z}^*$ and the nadir point $\mathbf{z}^{nad}$ in the Tchebycheff function has a significant influence on the distribution of optimal solutions over a PF. Particularly, in the case of using $\mathbf{z}^*$ as the reference point, the optimal solutions of sub-problems for a convex PF and a concave PF are shown in Fig. 1a and b, respectively. From the figures, it is clear that the optimal solution density in the central part of the convex PF is much larger than those of the concave PF, while it is opposite on the boundaries of the PFs. In contrast with the use of $\mathbf{z}^*$, the distribution of optimal solutions on these PFs is in a contrary direction if $\mathbf{z}^{nad}$ is used as the reference point, which is depicted as in Fig. 1c and d, respectively.



a)



b)



c) d)

Fig. 1. The distribution of optimal solutions on Pareto fronts by using $\mathbf{z}^*$ point and $\mathbf{z}^{nad}$ point.

From the obtained results of the investigation in (Wang et al., 2017), it is evident that the idea of combining both the ideal point $\mathbf{z}^*$ and the nadir point $\mathbf{z}^{nad}$ in the Tchebycheff function can be an efficient way to handle MOPs with complicated PFs. To take this advantage, Jiang & Yang, 2016 have also developed a search strategy with two phases for MOEA/D, in which the Tchebycheff function with $\mathbf{z}^*$ is used in the first phase, and the Tchebycheff function with $\mathbf{z}^{nad}$ is employed in the second phase. The search of the algorithm is started with the first phase; and after this phase, the crowded information of obtained solutions is estimated. If the crowded information shows that there is a significant difference between the solutions at the boundary/extreme and the intermediate of the PF, the second phase will be continued which aims to supplement more solutions on the boundary/extreme regions of the PF; otherwise, the second phase is not applied, and the search of the algorithm is completed. Through benchmark test functions with complex PFs, the method has been demonstrated to be able to deal effectively with MOPs with complicated PFs. The method, however, also has two limitations which may be described as follows: 1) the algorithm uses a pre-defined number of evaluations to switch from Phase 1 to Phase 2. It is very difficult to set this in advance without any knowledge about a problem, while it has a significant influence on the performance of the algorithm (Jiang & Yang, 2016); and 2) the computational cost of solving an MOP will increase significantly compared with available versions of MOEA/D if Phase 2 of the algorithm is executed.

Motivated by solving complex real-world problems and in an effort to overcome the limitations of the algorithm in (Jiang & Yang, 2016), this paper attempts to develop a newly improved MOEA/D (iMOEA/D) and applies it to solving bi-objective optimization problems (BOPs) of truss structures. In iMOEA/D, firstly, in order to surmount the first limitation in

(Jiang & Yang, 2016), the stopping criterion recently proposed in (Abdul Kadhar & Baskar, 2016) is applied. This method will help automatically stop the algorithm if there is no considerable improvement on the optimal solutions instead of using the number of evaluations. The details of this approach have been presented in Section 3. Secondly, to improve the distribution of optimum solutions over a complex PF while the computational cost of the algorithm does not increase, a new two-phase search strategy is proposed. The idea behind this approach is depicted in Fig. 2, and it includes the following steps. First, the set of the weight vectors of MOEA/D is numbered and split into two subsets: one set with odd-weight vectors and the other with even-weight vectors. Then, the optimization process is started with the set of odd-weight vectors in the first phase. In this phase, the general framework of MOEA/D as shown in Algorithm 1 is applied, in which the Tchebycheff function with the ideal point $\mathbf{z}^*$ as Eq.(2) is utilized. After this phase has been completed, the nadir point $\mathbf{z}^{nad}$ is determined from the obtained set of solutions, and the Tchebycheff function as Eq. (4) is employed for the second phase with the set of even-weight vectors.

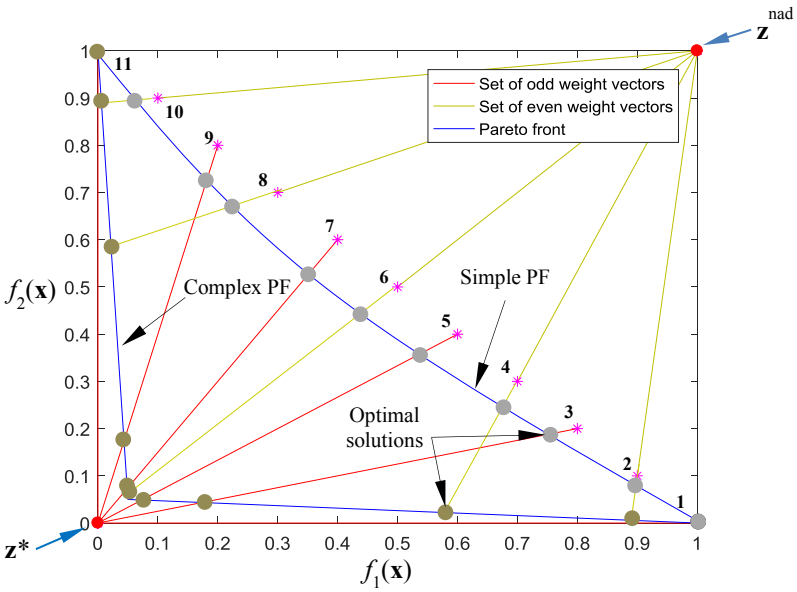


Fig. 2. The distribution of optimal solutions on a complex Pareto front acquired by iMOEA/D.

As shown in Fig. 2, it can be observed that by dividing the set of weight vectors into two subsets and applying different types of the Tchebycheff function, the distribution of optimal points over a complicated PF obtained by the algorithm becomes better. Fig. 2 also shows that this approach can work efficiently for a simple PF. Moreover, the computational cost of the algorithm will not increase because the total population size is equal to the population size of MOEA/D. The framework of iMOEA/D is summarized in Algorithm 2.

Algorithm 2. Pseudo-code of iMOEA/D algorithm

Input:

- A multi-objective optimization problem Eq. (1);
- A stopping criterion;
- N: number of sub-problems;
- $\mathbf{w}^{\text{odd}} = (w_1^i, ..., w_m^i)^{\text{T}}, i = 1,3,..., N$: a set of odd-weight vectors;
- $\mathbf{w}^{\text{even}} = (w_1^i, ..., w_m^i)^{\text{T}}, i = 2,4,..., N-1$: a set of even-weight vectors;
- T_m: size of mating neighborhood;
- $T_{r\max}$: maximum size of replacement neighborhood;
- δ: the probability that mating parents are selected from the neighborhood;
- *MaxIter*: maximum iteration;

Phase 1:

1.1. Apply **Step 1** to **Step 3** in Algorithm 1 for the set of odd-weight vectors $\mathbf{w}^{\text{odd}} = (w_1^i, ..., w_m^i)^{\text{T}}, i = 1,3,..., N$;

1.2. Obtain $\mathbf{PS}_1 = \{\mathbf{x}^1, ..., \mathbf{x}^i ..., \mathbf{x}^N\}, \mathbf{PF}_1 = \{FV^1(\mathbf{x}^1), ..., FV^i(\mathbf{x}^i), ..., FV^N(\mathbf{x}^N)\}, i = 1,3,.., N$;

Phase 2:

2.1. Define $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, ..., z_m^{\text{nad}})^{\text{T}}$ with $z_j^{\text{nad}} = \max\{f_j(\mathbf{x}) \,|\, \mathbf{x} \in \Omega, j = 1, ..., m\}, f_j(\mathbf{x}) \in \mathbf{PF}_1$;

2.2. Set the initial population $\mathbf{P} = \mathbf{PS}_1$ and $\mathbf{FV} = \mathbf{FV}_1$;

2.3. Apply **Step 1** to **Step 3** in Algorithm 1 for the set of even-weight vectors $\mathbf{w}^{\text{even}} = (w_1^i, ..., w_m^i)^{\text{T}}, i = 2,4,..., N-1$, where **Steps 1.2**, **1.3**, **2.5** are ignored; **Step 2.6.4** is changed by "for each current solution $\mathbf{x}^i$ in $\mathbf{B}_r$, replace $\mathbf{x}^i$ by $\mathbf{y}$ if $g^{te}(\mathbf{y}) \geq g^{te}(\mathbf{x}^i)$";

2.4. Obtain $\mathbf{PS}_2 = \{\mathbf{x}^2, ..., \mathbf{x}^i ..., \mathbf{x}^{N-1}\}, \mathbf{PF}_2 = \{FV^2(\mathbf{x}^2), ..., FV^i(\mathbf{x}^i), ..., FV^{N-1}(\mathbf{x}^{N-1})\}, i = 2,4,.., N-1$;

Output:

- Pareto set $\mathbf{PS} = \{\mathbf{PS}_1; \mathbf{PS}_2\}$;
- Pareto front $\mathbf{PF} = \{\mathbf{PF}_1; \mathbf{PF}_2\}$.

In Algorithm 2, it should be noted that the set of odd-weight vectors in the first phase should include the weight vectors of $(1,0)^{\text{T}}$ and $(0,1)^{\text{T}}$. This setting aims to ensure that a nadir point $\mathbf{z}^{\text{nad}}$, which is obtained after finishing Phase 1, can fully cover the range of a Pareto front.

With the new two-phase search strategy and the integration of some recent developments, iMOEA/D possesses some advantages as follows:

1) It is able to handle BOPs with complicated Pareto fronts;
2) It does not require extra computational procedures; and the algorithm structure is almost the same with MOEA/D except for using it for both phases, which is not too difficult for engineering designers to implement.
3) The computational cost for solving BOPs can be reduced significantly, which is quite crucial for solving real engineering applications.

These benefits are verified and evaluated in Section 5 through seven benchmark test functions and three practical applications of the optimal design of truss structures.

5. Experimental study

This section is divided into two parts. The first part is to examine the performance of the proposed method through benchmark test functions with complicated Pareto fronts. The second part is to evaluate the applicability of iMOEA/D for solving structural optimization problems. The performance of iMOEA/D is compared with MOEA/D in Algorithm 1, MOEA/D-TPN (Jiang & Yang, 2016), and NSGA-II (K Deb et al., 2002). To make a fair comparison regarding the computational cost of NSGA-II, a stopping criterion which is proposed for NSGA-II in (Roudenko & Schoenauer, 2004) is also utilized. All algorithms are implemented in Matlab 2016b on a Core i7, 8GB ram laptop.

5.1. Benchmark test functions

In order to evaluate the performance of iMOEA/D, a set of seven test instances as shown in Table 1 is used. In Table 1, F1-F3, F4-F6, and F7 are taken from (Wang et al., 2017), (Yang et al., 2016), and (Jiang & Yang, 2016), respectively. As mentioned in (Jiang & Yang, 2016; Wang et al., 2017; Yang et al., 2016), the Pareto fronts of these problems are very complicated which are very difficult for MOEA/D to obtain a proper distribution of optimal solutions over the Pareto fronts.

5.1.1. Performance metrics

The performance of the algorithms is assessed by two widely-used performance metrics including the Hypervolume indicator (HV) (E. Zitzler & Thiele, 1999) and the inverted generational distance metric (IGD) (Eckart Zitzler et al., 2003). These indicators are defined as follows:

- HV (E. Zitzler & Thiele, 1999): Let $\mathbf{z}^* = (z_1^*, ..., z_m^*)$ be a reference point in the objective space which satisfies $z_j^* \geq \max(f_j)$. Let $\mathbf{P}$ be an approximate set to the PF gained by an algorithm. The HV of $\mathbf{P}$ is the volume of the region dominated by $\mathbf{P}$ and bounded by $\mathbf{z}^*$ and is computed by

$$HV(\mathbf{P}, \mathbf{z}^*) = volume\left(\bigcup [f_1(\mathbf{x}), z_1^*] \times ... [f_m(\mathbf{x}), z_m^*]\right) \quad (12)$$

  where $volume(.)$ is the Lebesgue measure. In this study, $\mathbf{z}^*$ is set to (2.0, 2.0) for all test instances (Jiang & Yang, 2016). It is noted that the method with a larger HV metric is the better.

- IGD (Eckart Zitzler et al., 2003): Let $\mathbf{P}^*$ be a set of uniformly distributed points over the PF in the objective space. Suppose $\mathbf{P}$ be an approximate set to the PF gained by an algorithm. The inverted generational distance from $\mathbf{P}^*$ to $\mathbf{P}$ is defined by

$$IGD(\mathbf{P}^*, \mathbf{P}) = \frac{\sum_{\mathbf{p} \in \mathbf{P}} d(\mathbf{v}, \mathbf{P}^*)}{|\mathbf{P}|} \tag{13}$$

where $d(\mathbf{v}, \mathbf{P}^*)$ is the Euclidean distance between the member $\mathbf{v}$ of $\mathbf{P}$ and the nearest member of $\mathbf{P}^*$. In this paper, 500 representative points created from the true PF are used for all the benchmark test problems. It is noted that the method with a lower IGD metric is the better.

5.1.2. Parameter settings

For all test instances, the parameter settings are the same. The parameters of NSGA-II are set according to (Song, 2011). For MOEA/D, MOEA/D-TPN, and iMOEA/D, their parameters are set as follows:

- *Population size N*: $N = 100$.
- *Reproduction operators*: *CR* and $F = [0.4, 0.6]$, $\delta = 0.9$, and $p_m = 1/n$ (n is the number of decision variables);
- *Neighborhood size*: In MOEA/D, T_m and $T_{rmax} = 0.2N$. In iMOEA/D and MOEA/D-TPN T_m and $T_{rmax} = 0.1N$.
- *Stopping condition*: In MOEA/D, MOEA/D-TPN and iMOEA/D, $\varepsilon = 10^{-6}$, $g = 10$, and *MaxIter* = 1000. The algorithms terminated when their stopping conditions are satisfied or the maximum number of iterations (*MaxIter*) is reached.
- *Number of runs*: Each algorithm is independently run 30 times on each test instance.

It should be noted that the above parameters are set either based on the literature or on the experience obtained by running simulations with different settings. More specifically, N, ε and g are derived from (Yang et al., 2016), and (Abdul Kadhar & Baskar, 2016) respectively, while the others are based on empirical results.

Table 1. Benchmark test functions

Instance	Objective function	Domain	n	PF characteristic
F1	$f_1 = 1-\cos(0.5\pi x_1)+\frac{2}{\lvert I_1\rvert}\sum_{i\in I_1}\lvert g_i\rvert^{0.7}$ $f_2 = 10-10\sin(0.5\pi x_1)+\frac{2}{\lvert I_2\rvert}\sum_{i\in I_2}\lvert g_i\rvert^{0.7}$ $g_i = x_i-0.9\sin\left(\frac{i\pi}{n}\right), i\in 2,...,n$ where $I_1=\{i \mid i \text{ is odd and } 2\le i\le n\}$, and $I_2=\{i \mid i \text{ is even and } 2\le i\le n\}$ Pareto Front : $f_2=(1-\sqrt{f_1})^5$	$[0, 1]\times[-1, 1]^{n-1}$	30	Convex
F2	$f_1 = x_1+\frac{2}{\lvert I_1\rvert}\sum_{i\in I_1}\left\lvert g_i+\frac{\sin(\pi g_i)}{\pi}\right\rvert$ $f_2=\begin{cases}1-19x_1+\frac{2}{\lvert I_2\rvert}\sum_{i\in I_2}\left\lvert g_i+\frac{\sin(\pi g_i)}{\pi}\right\rvert & \text{if } x_1\le 0.005\\ \frac{1}{19}-\frac{x_1}{19}+\frac{2}{\lvert I_2\rvert}\sum_{i\in I_2}\left\lvert g_i+\frac{\sin(\pi g_i)}{\pi}\right\rvert & \text{otherwise}\end{cases}$ $g_i = x_i-0.9\sin\left(\frac{i\pi}{n}\right), i\in 2,...,n$ where $I_1=\{i \mid i \text{ is odd and } 2\le i\le n\}$, and $I_2=\{i \mid i \text{ is even and } 2\le i\le n\}$ Pareto Front : $f_2=\sqrt{(1-f_1^5)}$	$[0, 1]\times[-1, 1]^{n-1}$	30	Piecewise, convex
F3	$f_1 = x_1+\frac{2}{\lvert I_1\rvert}\sum_{i\in I_1}\left(1-e^{-\lvert g_i\rvert}\right)$ $f_2=\begin{cases}1-8x_1^4+\frac{2}{\lvert I_2\rvert}\sum_{i\in I_2}\left(1-e^{-\lvert g_i\rvert}\right) & \text{if } x_1\le 0.5\\ 8(1-x_1)^4+\frac{2}{\lvert I_2\rvert}\sum_{i\in I_2}\left(1-e^{-\lvert g_i\rvert}\right) & \text{otherwise}\end{cases}$ $g_i = x_i-0.9\sin\left(\frac{i\pi}{n}\right), i\in 2,...,n$ where $I_1=\{i \mid i \text{ is odd and } 2\le i\le n\}$, and $I_2=\{i \mid i \text{ is even and } 2\le i\le n\}$ Pareto Front : $f_2=0.5(1-f_1+\sqrt{1-f_1})\cos^2(4\pi(1-f_1))$	$[0, 1]\times[-1, 1]^{n-1}$	30	Separable, convex, concave
F4	$f_1=(1+g)x_1$ $f_2=(1+g)(1-\sqrt{x_1})^3$ $g=2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2-\cos(2\pi y_i)))$ where $y_{i=2:n}=x_i-\sin(0.5\pi x_1)$ Pareto Front : $f_2=(1-\sqrt{f_1})^3$	$[0, 1]^n$	30	Convex
F5	$f_1=(1+g)(x_1+0.05\sin(6\pi x_1))^2$ $f_2=(1+g)(1-x_1+0.05\sin(6\pi x_1))^2$ $g=2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2-\cos(2\pi y_i)))$ where $y_{i=2:n}=x_i-\sin(0.5\pi x_1)$ Pareto Front : $f_1^{0.5}+f_2^{0.5}=1+0.1\sin(3\pi(f_1^{0.5}-f_2^{0.5}+1))$	$[0, 1]^n$	30	Nonconvex

F6	$f_1 = (1+g)(x_1 + 0.05\sin(6\pi x_1))^{0.2}$ $f_2 = (1+g)(1-x_1 + 0.05\sin(6\pi x_1))^{10}$ $g = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2 - \cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_1)$ Pareto Front : $f_1^5 + f_2^{0.1} = 1 + 0.1\sin(3\pi(f_1^5 - f_2^{0.1} + 1))$	$[0, 1]^n$	30	Nonconvex
F7	$f_1 = (1+g)(1-x_1)$ $f_2 = 0.5(1+g)(x_1 + \sqrt{x_1}\cos^2(4\pi x_1))$ $g = 2\sin(0.5\pi x_1)(n-1+\sum_{i=2}^{n}(y_i^2 - \cos(2\pi y_i)))$ where $y_{i=2:n} = x_i - \sin(0.5\pi x_1)$ Pareto Front : $f_2 = 0.5(1 - f_1 + \sqrt{1-f_1}\cos^2(4\pi(1-f_1)))$	$[0, 1]^n$	30	Nonconvex, disconnected

5.1.3. Evaluation of the new improvements

In order to demonstrate the ability of iMOEA/D to solve BOPs with the two-phase search strategy and to establish its advantages compared to the method (denoted as MOEA/D-TPN) in (Jiang & Yang, 2016), the benchmark tests F5 and F6 are used for investigation.

The performance of iMOEA/D in solving BOPs with complex PFs is evaluated by F5 and illustrated in Fig. 3. As shown in Fig. 3, the optimal solutions are densely populated in the central part of the PF and sparsely near the boundary of the PF in Phase 1 (Fig. 3a), and vice versa in Phase 2 (Fig. 3b). By combining the obtained results in both phases, the distribution of the solutions on the final PF has significantly improved as shown in Fig. 3c.

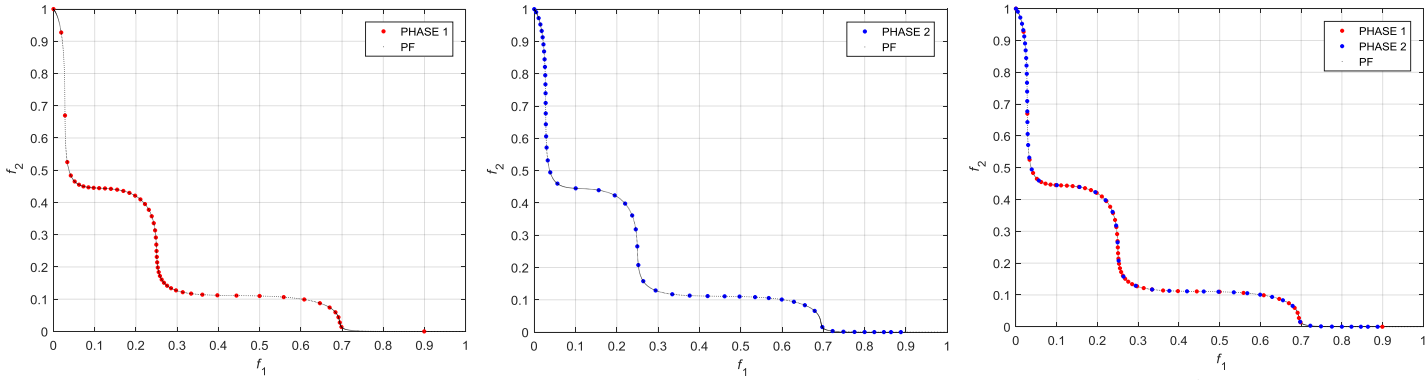


Fig. 3. Illustration of the Pareto front obtained by iMOEA/D for F5 relative to Phases 1, 2 and their combination.

As mentioned in Section 4, iMOEA/D is developed based on MOEA/D-TP (Jiang & Yang, 2016), where the two improvements are integrated: 1) the incorporation of two recently developed features (i.e. an adaptive replacement strategy and a stopping criterion), and 2) the division of the initial weight vectors into two subsets. For the first improvement, it is easy to recognize that the computational cost of MOEA/D-TPN will be significantly larger than that

of iMOEA/D if this improvement is not integrated into MOEA/D-TPN. This is because MOEA/D-TPN uses a conventional replacement scheme which has been demonstrated to be more costly than the adaptive replacement strategy (Wang et al., 2016), and a maximum number of iterations which do not allow to stop the algorithm until this number is reached. Therefore, the remaining comparison between iMOEA/D and MOEA/D-TPN will focus on the second improvement. For this comparison, the first improvement is also incorporated into MOEA/D-TPN. In this version, however, the niche scheme introduced in (Jiang & Yang, 2016) is not included because it is not employed in iMOEA/D. It should also be noted that the HV-metric values can be better if the obtained results have more Pareto solutions, which depend on the initial population size (N). Thus, if the same N is applied for both the methods, the HV-metric values of MOEA/D-TPN may be better than those of iMOEA/D. This is because MOEA/D-TPN will use the same N in both phases if the crowded information of obtained solutions in Phase 1 is satisfied, and Phase II is executed, while iMOEA/D always uses half of N for both phases. For this evaluation, therefore, two different sizes of N (N = 50, 100) are investigated for MOEA/D-TPN, and only one size of N (N = 100) is used for iMOEA/D.

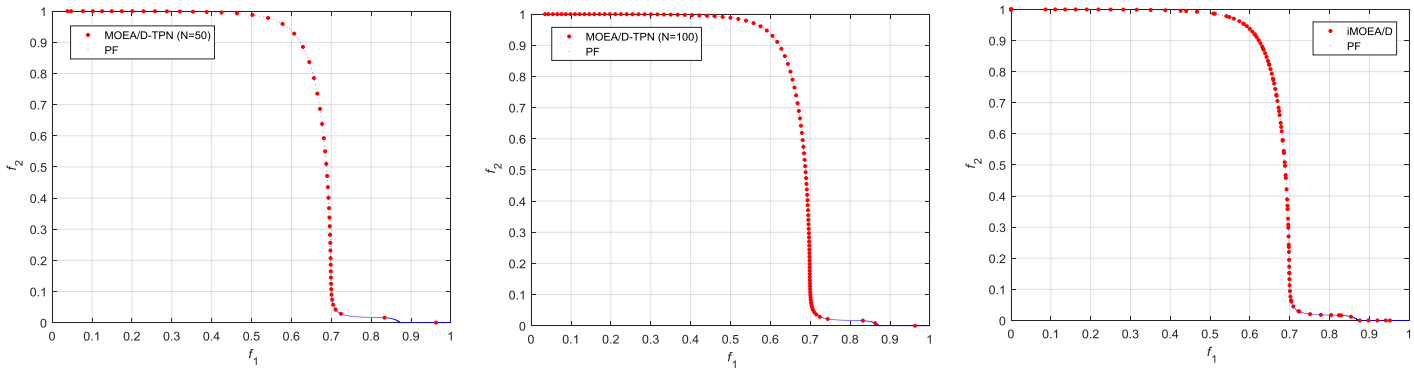
The statistical results of F5 and F6 gained by iMOEA/D and MOEA/D-TPN are provided in Table 2. A comparison of results obtained by iMOEA/D and MOEA/D-TPN with N = 50 indicates that iMOEA/D outperforms MOEA/D-TPN. Specifically, all the comparative quantities obtained by iMOEA/D are better than those of MOEA/D-TPN, except for the computational cost in F6. However, the reason for the reduction of the computation cost in F6 is because the crowded information of obtained results in Phase 1 is unsatisfied, and Phase 2 of MOEA/D-TPN is not executed. This explains why its HV-metric values of F6 is much worse than those of iMOEA/D. This is also reflected in Fig. 4a. Moreover, the diversity of weight vectors with N = 50 is often less than that with N = 100, which can lead to the reduction of the quality of solutions obtained by MOEA/D-TPN.

In contrast to MOEA/D-TPN with N = 50, for the test F5, the HV-metric values obtained by MOEA/D-TPN with N = 100 are much better than those of iMOEA/D, which is as an obvious result of the increased number of Pareto solutions. Nevertheless, the computational cost is mostly doubled because of the double growth of the population when both phases are executed. Like the case of N = 50, for the test F6, MOEA/D-TPN with N = 100 also terminates at Phase 1. Thus, its HV-metric values are still worse than those of iMOEA/D. In addition, the computational cost in this case is still higher compared to those of iMOEA/D. Also, for the test F6, although the diversity of weight vectors is the same, the quality of IGD-

metric values of MOEA/D-TPN is still bigger than those of iMOEA/D. This may be due to the efficiency of using the nadir point $\mathbf{z}^{nad}$ in the second phase of iMOEA/D.

Table 2. Comparison of statistical results of iMOEA/D and MOEA/D-TPN for F5 and F6.

Metric	Method		F5	F6
HV	MOEA/D-TPN (N = 50)	Mean	3.8259	3.3190
		Worst	3.8259	3.3188
		Best	3.8260	3.3191
		Std.	0.0000	0.0001
	MOEA/D-TPN (N = 100)	Mean	3.8271	3.3221
		Worst	3.8271	3.3219
		Best	3.8272	3.3223
		Std.	0.0000	0.0001
	iMOEA/D	Mean	3.8261	3.3233
		Worst	3.8260	3.3232
		Best	3.8261	3.3233
		Std.	0.0000	0.0000
IGD	MOEA/D-TPN (N = 50)	Mean	0.0010	0.0212
		Worst	0.0011	0.0221
		Best	0.0009	0.0207
		Std.	0.0001	0.0004
	MOEA/D-TPN (N = 100)	Mean	0.0009	0.0216
		Worst	0.0010	0.0220
		Best	0.0009	0.0201
		Std.	0.0000	0.0004
	iMOEA/D	Mean	0.0009	0.0118
		Worst	0.0010	0.0125
		Best	0.0008	0.0111
		Std.	0.0000	0.0004
Computational cost	MOEA/D-TPN (N = 50)	FEs	22337	13667
		Time (min)	0.3382	0.1697
	MOEA/D-TPN (N = 100)	FEs	39524	21512
		Time (min)	0.6755	0.2671
	iMOEA/D	FEs	19067	19537
		Time (min)	0.2641	0.2426



a) b) c)

Fig. 4. Illustration of the Pareto fronts obtained by MOEA/D-TPN (N = 50,100) and iMOEA/D for F6.

From the above evaluation results, it can be concluded that the incorporation of the two recent features (i.e. an adaptive replacement strategy and a stopping criterion), and the idea of using two different subsets of weight vectors are helpful and meaningful compared to

MOEA/D-TPN. To further evaluate the performance of iMOEA/D, MOEA/D-TPN is also used to solve the remaining test instances together with NSGA-II and MOEA/D, and for a consistent comparison, the initial populations (N) of all the methods are set to be the same.

5.1.4. Experimental results

In this part, iMOEA/D is tested on all the remaining functions, and the obtained statistical results are presented in Table 3 in comparison with those gained by NSGA-II, MOEA/D and MOEA/D-TPN. By taking a close look at the HV-metric values in Table 3, it can be seen that MOEA/D-TPN is the best method, except for the F6 metrics. However, it should be noted that for F1-F5 and F7, both phases of MOEA/D-TPN are carried out with the same N. Thus, the obtained optimal solutions are mostly doubled, and as a result the HV-metric values are obviously much better. The HV-metric values for F6 are almost the same with those of MOEA/D and worse than those of iMOEA/D and NSGA-II, which is because only the first phase of MOEA/D-TPN is executed. The comparison between iMOEA/D, MOEA/D and NSGA-II indicates that iMOEA/D outperforms NSGA-II and MOEA/D on F1-F6, while NSGA-II and MOEA/D are better on F7. From the comparison of IGD-metric values, it can be seen that iMOEA/D performs slightly better than MOEA/D-TPN and both somewhat worse than MOEA/D, but all three methods are significantly better than NSGA-II. Regarding the computational cost (in terms of time in minute (min) and the number of function evaluations (FEs)), however, it can be observed that iMOEA/D is the best method with the smallest amount of FEs for all the tests. The total FEs of iMOEA/D for F1-F7 is 312,980 which is almost 46% less than those of MOEA/D-TPN (579,357), nearly 25% less than those of MOEA/D (413,003), approximately 35% less than those of NSGA-II (477,163). By looking at the standard deviation (Std.), it is observed that for all the investigated benchmark tests the standard deviations of iMOEA/D, MOEA/D-TPN and MOEA/D are quite small, while those of NSGA-II are often larger.

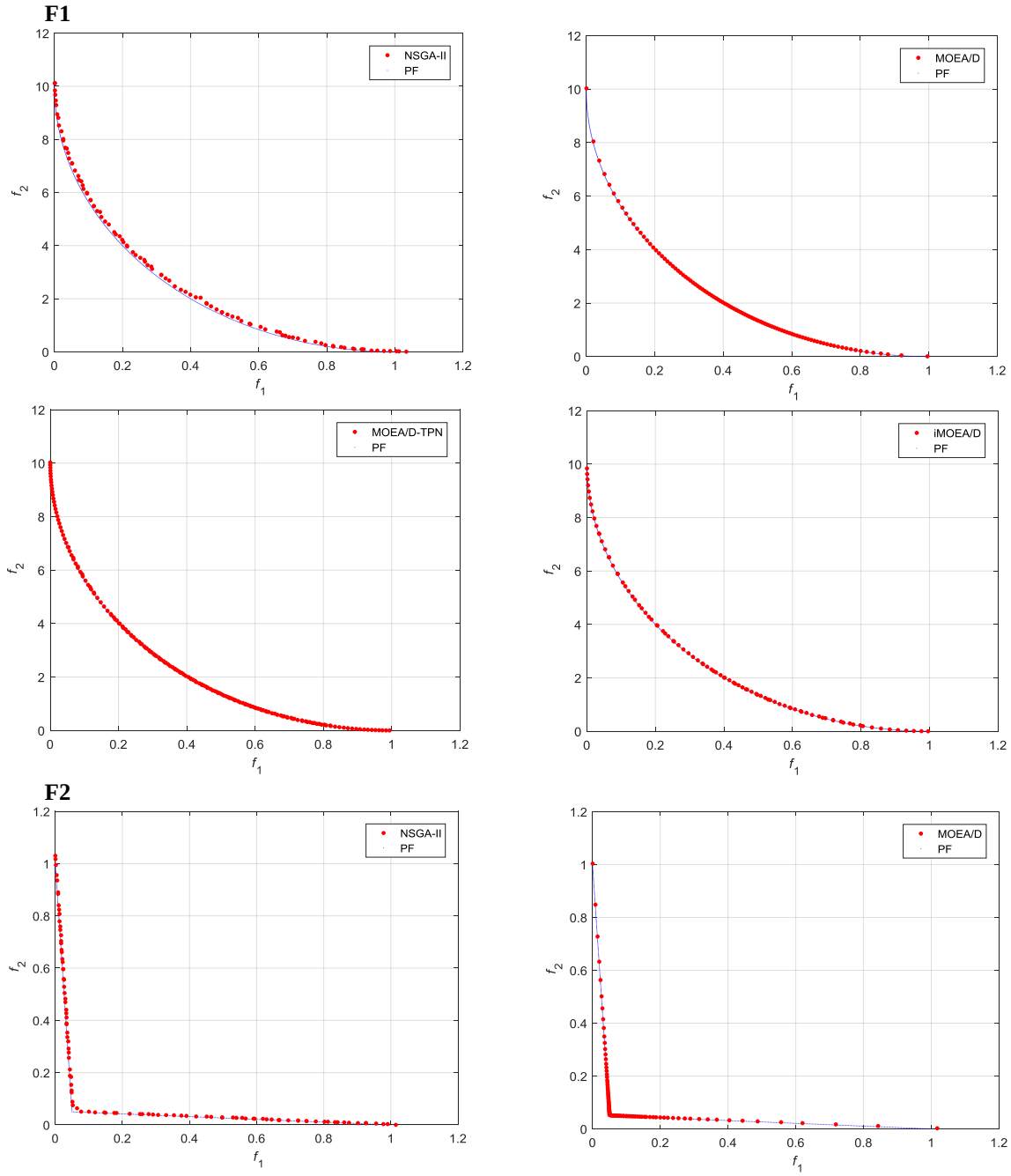
Based on the obtained statistical results, it can be concluded that iMOEA/D is more effective than NSGA-II and MOEA/D in terms of the distribution of optimal solutions over the PFs and the computational cost. Compared with MOEA/D-TPN, iMOEA/D performs worse with respect to HV-metric values, but significantly better in terms of the computational cost and slightly better concerning IGD-metric values. From the results, it can also be recognized that iMOEA/D is a proper method that can balance effectively between the quality of solutions and the computational cost.

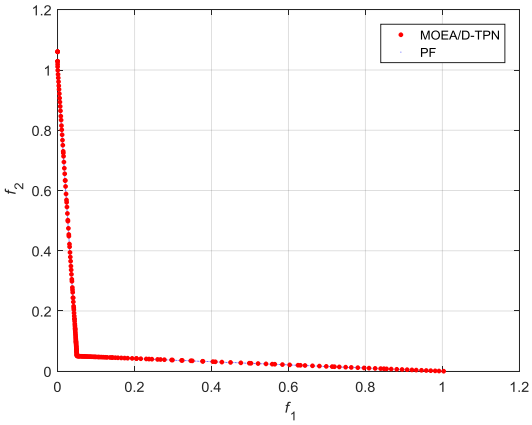
Table 3. Statistical results for the benchmark tests.

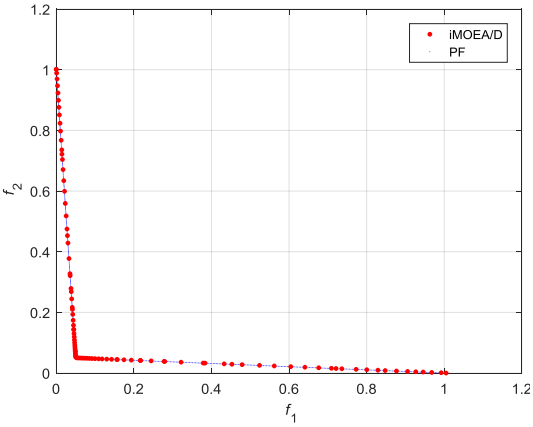
Metric	Method		F1	F2	F3	F4	F5	F6	F7
HV	NSGA-II	Mean	1.6317	3.9422	3.4820	3.8957	3.8250	3.3226	3.6787
		Worst	1.5441	3.9334	3.4739	3.8950	3.8245	3.3223	3.6783
		Best	1.6833	3.9446	3.4855	3.8962	3.8254	3.3230	3.6789
		Std.	0.0388	0.0027	0.0035	0.0003	0.0003	0.0002	0.0001
	MOEA/D	Mean	1.6123	3.9462	3.4925	3.8949	3.8247	3.3221	3.6787
		Worst	1.6076	3.9458	3.4906	3.8948	3.8246	3.3221	3.6786
		Best	1.6199	3.9464	3.4933	3.8949	3.8249	3.3222	3.6787
		Std.	0.0029	0.0002	0.0007	0.0000	0.0001	0.0000	0.0000
	MOEA/D-TPN	Mean	1.8055	3.9486	3.4958	3.8985	3.8271	3.3221	3.6798
		Worst	1.7919	3.9484	3.4937	3.8985	3.8271	3.3219	3.6798
		Best	1.8111	3.9488	3.4966	3.8986	3.8272	3.3223	3.6799
		Std.	0.0041	0.0001	0.0008	0.0000	0.0000	0.0001	0.0000
	iMOEA/D	Mean	1.7793	3.9480	3.4926	3.8969	3.8261	3.3233	3.6784
		Worst	1.7720	3.9475	3.4901	3.8968	3.8260	3.3232	3.6783
		Best	1.7877	3.9481	3.4934	3.8970	3.8261	3.3233	3.6784
		Std.	0.0039	0.0001	0.0009	0.0000	0.0000	0.0000	0.0000
IGD	NSGA-II	Mean	0.0308	0.0069	0.0105	0.0042	0.0012	0.0126	0.0007
		Worst	0.0446	0.0114	0.0157	0.0053	0.0014	0.0177	0.0009
		Best	0.0201	0.0041	0.0075	0.0027	0.0009	0.0034	0.0005
		Std.	0.0064	0.0023	0.0023	0.0005	0.0001	0.0037	0.0001
	MOEA/D	Mean	0.0047	0.0011	0.0018	0.0011	0.0008	0.0217	0.0005
		Worst	0.0056	0.0016	0.0022	0.0012	0.0009	0.0222	0.0006
		Best	0.0041	0.0010	0.0015	0.0011	0.0007	0.0213	0.0004
		Std.	0.0004	0.0002	0.0002	0.0000	0.0000	0.0002	0.0000
	MOEA/D-TPN	Mean	0.0065	0.0014	0.0016	0.0035	0.0009	0.0216	0.0006
		Worst	0.0077	0.0027	0.0020	0.0035	0.0010	0.0220	0.0006
		Best	0.0056	0.0010	0.0014	0.0034	0.0009	0.0201	0.0005
		Std.	0.0005	0.0004	0.0001	0.0000	0.0000	0.0004	0.0000
	iMOEA/D	Mean	0.0065	0.0010	0.0020	0.0035	0.0009	0.0118	0.0005
		Worst	0.0073	0.0012	0.0033	0.0037	0.0010	0.0125	0.0006
		Best	0.0057	0.0010	0.0017	0.0034	0.0008	0.0111	0.0004
		Std.	0.0004	0.0001	0.0003	0.0001	0.0000	0.0004	0.0000
Computat-ional cost	NSGA-II	FEs	71417	88443	83253	44483	75247	57287	57033
		Time (min)	0.8427	1.2515	0.8603	0.4196	0.7851	0.8631	0.5931
	MOEA/D	FEs	101000	101000	101000	32151	24997	27421	25434
		Time (min)	1.2167	1.2667	1.0667	0.3342	0.2887	0.3394	0.2728
	MOEA/D-TPN	FEs	170191	88845	162596	42621	39524	21512	54068
		Time (min)	2.2804	1.1034	1.7003	0.6545	0.6755	0.2671	0.5831
	iMOEA/D	FEs	82597	63950	82700	20557	19067	19537	24572
		Time (min)	1.1067	0.7942	0.8648	0.3157	0.2641	0.2426	0.2650

Corresponding to the results given in Table 3, the PFs acquired by the methods are plotted in Fig. 5. From the figures, it can be seen that NSGA-II, MOEA/D-TPN and iMOEA/D have the ability to give a better distribution of solutions on complicated PFs compared to MOEA/D. However, when taking a closer look at the figures, it can be observed that the quality of the solutions on the PFs obtained by MOEA/D, MOEA/D-TPN and iMOEA/D is much better than those obtained by NSGA-II. Here, it is also recognized that MOEA/D-TPN has more optimal solutions on the PFs at F1-F5 and F7, while those of F6 are almost the same

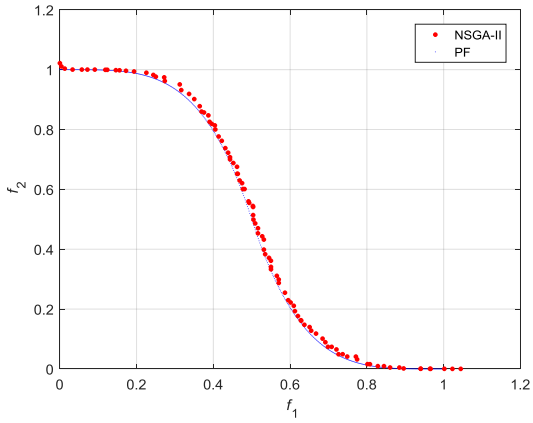
with those of MOEA/D. These illustrations again reflect the HV- and IGD-metric values provided in Table 3.

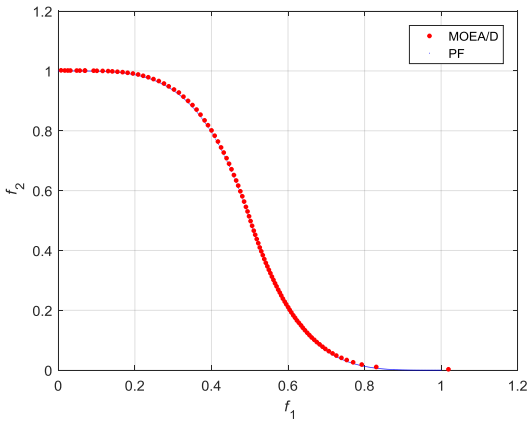


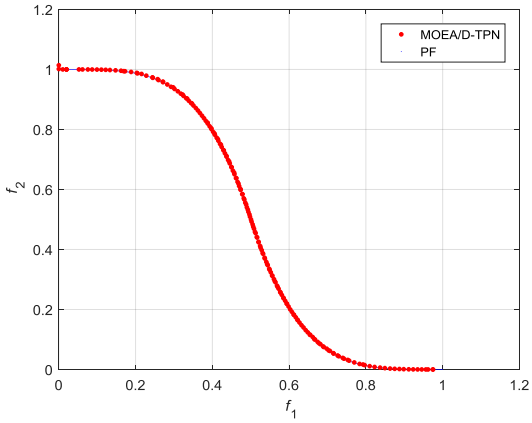

MOEA/D-TPN
PF
f_1
f_2

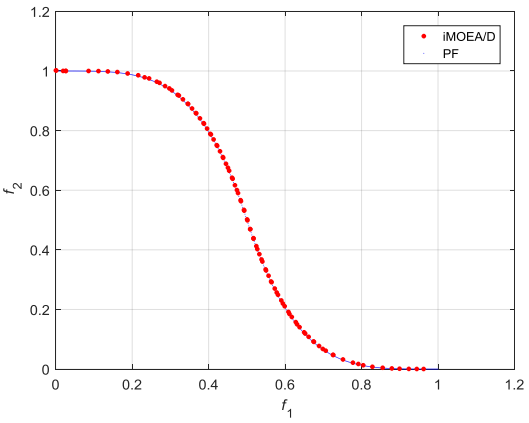

iMOEA/D
PF
f_1
f_2

F3

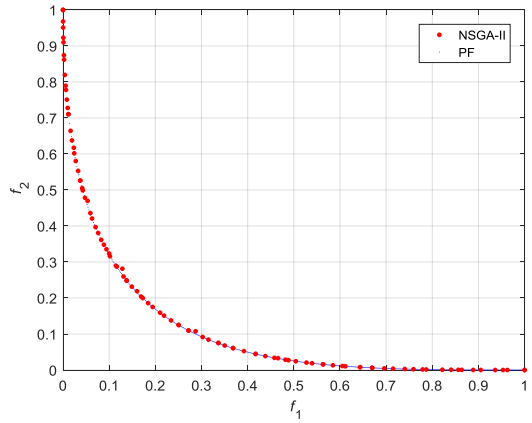

NSGA-II
PF
f_1
f_2

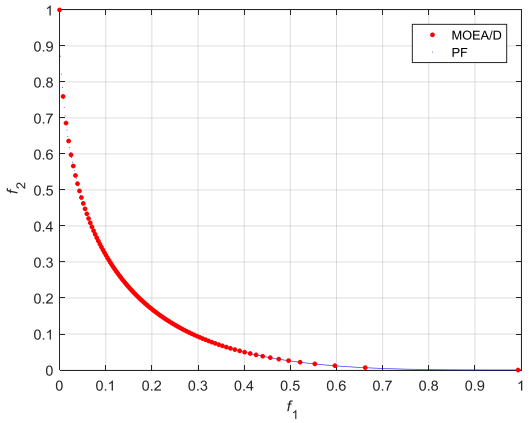

MOEA/D
PF
f_1
f_2

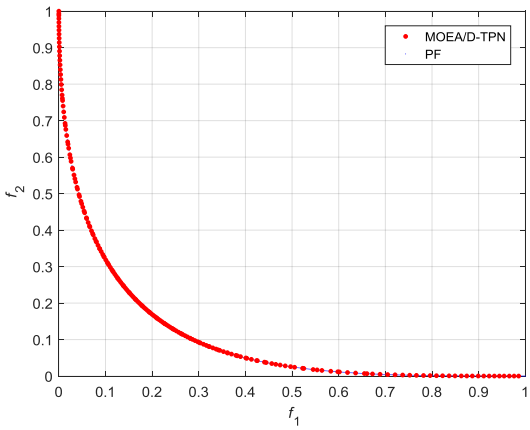

MOEA/D-TPN
PF
f_1
f_2

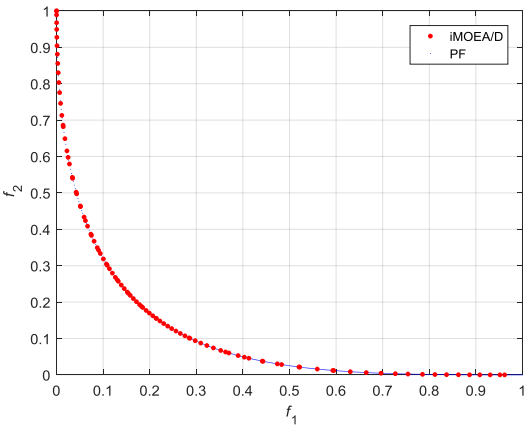

iMOEA/D
PF
f_1
f_2

F4

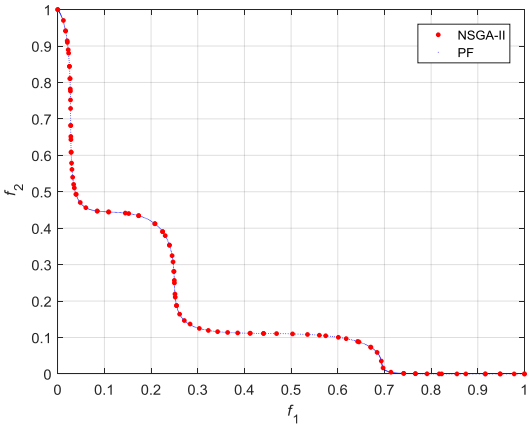

NSGA-II
PF
f_1
f_2

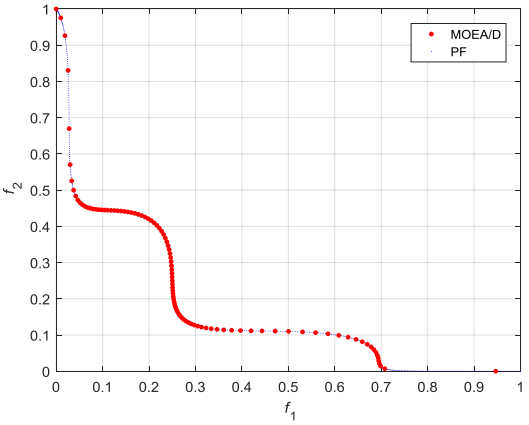

MOEA/D
PF
f_1
f_2

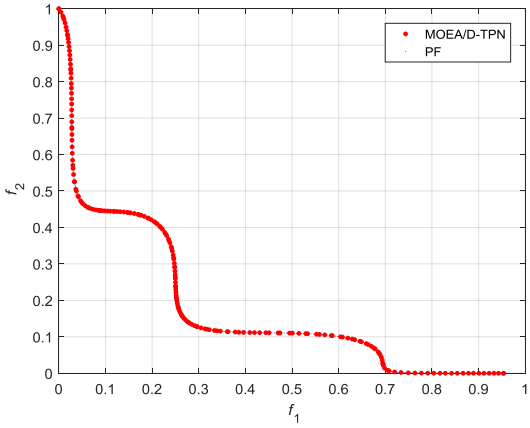

MOEA/D-TPN
PF
f_1
f_2

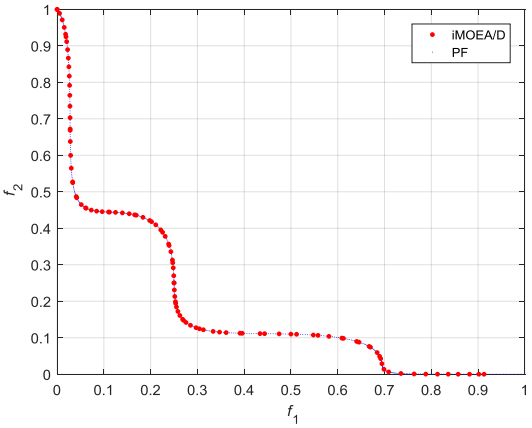

iMOEA/D
PF
f_1
f_2

F5

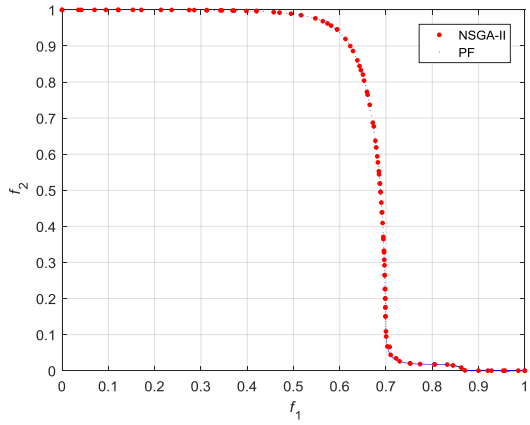

NSGA-II
PF
f_1
f_2

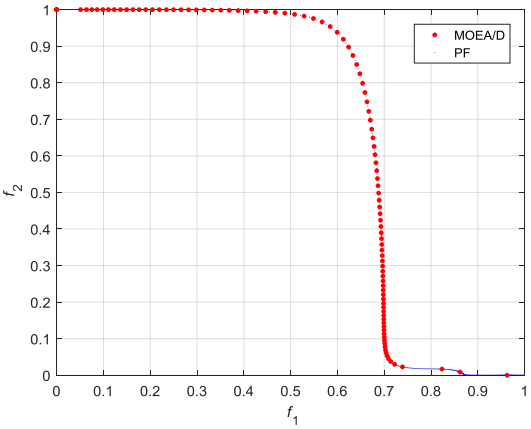

MOEA/D
PF
f_1
f_2


MOEA/D-TPN
PF
f_1
f_2


iMOEA/D
PF
f_1
f_2

F6


NSGA-II
PF
f_1
f_2


MOEA/D
PF
f_1
f_2

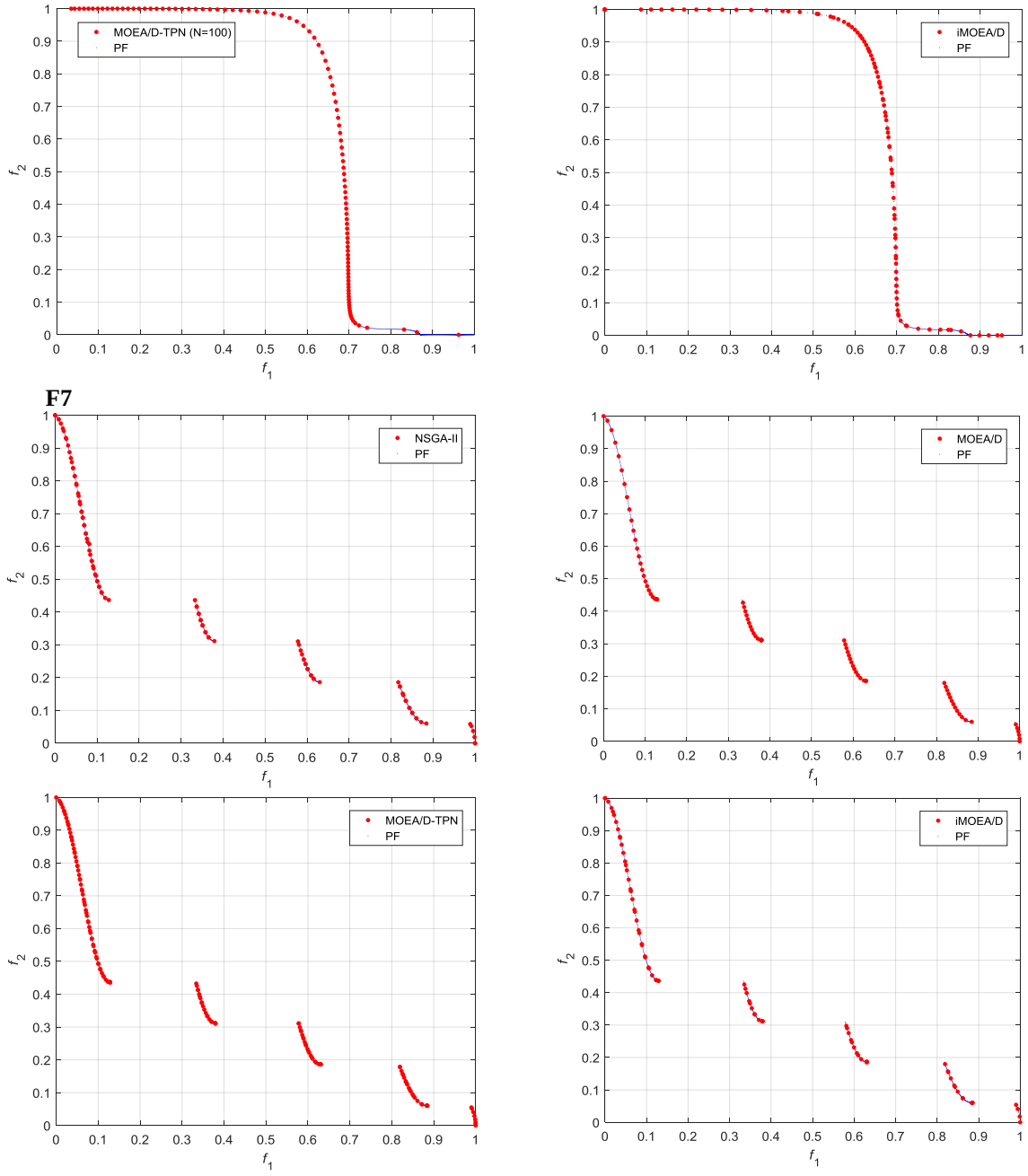


Fig. 5. Pareto fronts obtained by NSGA-II, MOEA/D, MOEA/D-TPN and iMOEA/D for F1-F7.

5.2. Structural optimization problems

In this section, iMOEA/D is applied to deal with three optimal design problems of truss structures. Since the computational cost of MOEA/D-TPN has been demonstrated to be huge in the previous section, which is a large restriction for real-word engineering applications, only MOEA/D and NSGA-II are applied to solve these problems for comparison purposes. The problems include a 15-bar planar truss (Tang, Tong, & Gu, 2005), a 72-bar space truss (Wu & Chow, 1995), and a 160-bar space truss (Groenwold & Stander, 1997) as shown in Fig. 6, Fig. 7, and Fig. 8, respectively. These are structural optimization problems which are

widely used to measure the applicability of single-objective optimization methods in the literature.

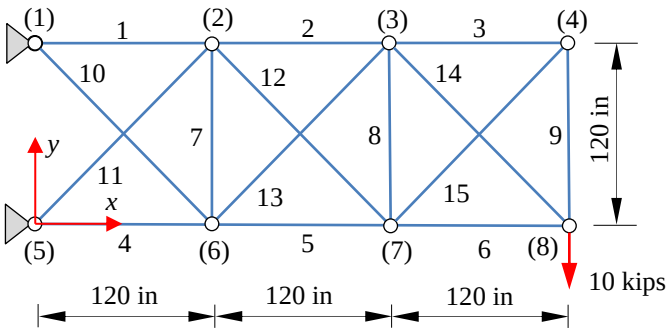


Fig. 6. The 15-bar planar truss.

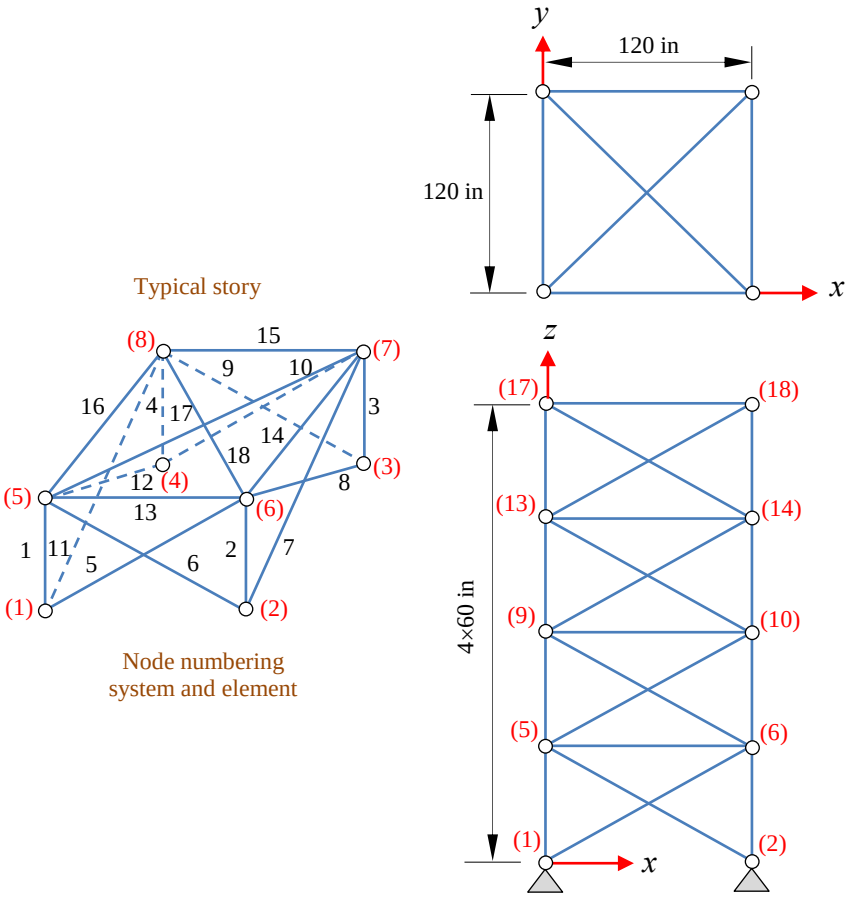


Fig. 7. The 72-bar space truss.

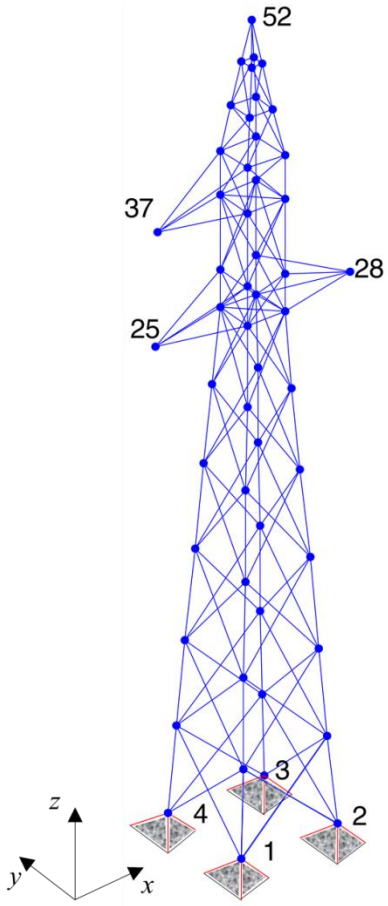


Fig. 8. The 160-bar space truss.

In this study, the above problems are reformulated as multi-objective design optimization problems. For all of the problem-cases considered, the aim of the objective functions is to minimize the overall weight of the structures and the maximum displacement at the truss nodes. The design variables and the constraints for each problem are described as follows:

- The 15-bar planar truss: The design problem has 8 continuous design variables of node coordinates and 15 discrete design variables of cross-section areas. All members are subjected to the stress limitation of ±25 (ksi). The details of the input data for this problem can be found in (Ho-Huu et al., 2015; Rahami et al., 2008).
- The 72-bar space truss: The structure has 72 members that are divided into 16 groups corresponding to 16 design variables. The design variables are discrete values and selected from an available set. This structure was designed for two separate loading conditions and subjected to design constraints which consist of the stress limitations of ±25000 psi and a restriction for all nodal displacements of ±0.25 in. The input data for the problem is available in (Ho-Huu et al., 2016; Kaveh & Mahdavi, 2014).

- The 160-bar space truss: The 160 bars of the truss are connected to 38 independent discrete design variables. The structure is designed for the eight independent load cases, and the design constraints are considered for compression members. The details for these input data are given in (Groenwold & Stander, 1997; Ho-Huu et al., 2016).

In order to handle constraints and discrete design variables for the problems in this section, a constraint-handling technique recently proposed in (Jan & Khanum, 2013), and a rounding technique in (Kaveh & Mahdavi, 2014) are utilized, respectively. All three problems are run 20 independent times with a population size of 80. Due to the exact solutions for these problems not being available, a set of all non-dominate solutions obtained by three methods after 20 independent runs is used as an approximate PF for evaluating the IGD metric. To validate the reliability of the used methods, some obtained solutions on PFs are also compared to those acquired by single-objective optimization methods in the literature.

Table 4 provides the statistical results obtained by the methods. At first glance, it can be seen that the acquired results show the same trend for all the problems. Specifically, in a comparison of the HV-metric values, it is observed that iMOEA/D is the best method in all statistical indices including the worst, best and standard deviation (Std.) values, and NSGA-II is the second one. This implies that iMOEA/D shows a considerable improvement on the distribution of solutions over a PF compared with MOEA/D and NSGA-II. Nevertheless, this order is slightly changed in terms of the IGD-metric values, where the order rank is MOEA/D, iMOEA/D and NSGA-II for all the statistical indexes. This means that the quality of solutions gained by MOEA/D is often better than those of iMOEA/D and NSGA-II. However, the difference of IGD-metric values between MOEA/D and iMOEA/D is small, while it is quite large when comparing with those of NSGA-II.

In terms of computational cost, again iMOEA/D is the best method with the lowest values of time and FEs, while MOEA/D is better than NSGA-II. The FEs of iMOEA/D is around half to three-quarters of those of NSGA-II, and around three-quarters of those of MOEA/D.

From the above results, it can be concluded that with the new features on MOEA/D, iMOEA/D has a significant improvement on the performance of the algorithm, particularly in the distribution of solutions over PFs and the computational cost.

Table 4. Statistical results for the structural optimization problems.

Metric	Method		15-bar truss	72-bar truss	160-bar truss
HV	NSGA-II	Mean	1.3909	0.6413	5.2517
		Worst	1.3880	0.6405	5.2352
		Best	1.3929	0.6419	5.2623
		Std.	0.0012	0.0004	0.0066
	MOEA/D	Mean	1.3901	0.6399	5.2434

		Worst	1.3875	0.6380	5.2109
		Best	1.3920	0.6411	5.2664
		Std.	0.0015	0.0009	0.0186
	iMOEA/D	Mean	1.3981	0.6442	5.3039
		Worst	1.3970	0.6439	5.2914
		Best	1.3989	0.6445	5.3115
		Std.	0.0006	0.0002	0.0054
	NSGA-II	Mean	4.5106	14.4666	12.2144
		Worst	14.6534	20.6656	17.8192
		Best	1.8665	9.2638	8.3395
		Std.	3.1474	2.1842	3.0719
IGD	MOEA/D	Mean	0.3284	2.5892	1.2162
		Worst	0.6924	4.4179	2.0974
		Best	0.2259	1.3245	0.3566
		Std.	0.1097	0.9183	0.5282
	iMOEA/D	Mean	1.2766	6.1325	5.1365
		Worst	2.1820	12.3974	10.1472
		Best	0.3817	2.5065	1.5691
		Std.	0.4866	2.4370	2.2344
	NSGA-II	FEs	80000	80000	80000
		Time (min)	1.6533	4.5233	65.5117
Computational cost	MOEA/D	FEs	68351	57371	72300
		Time (min)	1.1636	3.5033	38.1137
	iMOEA/D	FEs	40035	40803	64397
		Time (min)	0.8434	2.2558	32.3817

The PFs obtained by the methods for all considered problems are illustrated in Fig. 9, Fig. 10, and Fig. 11, respectively. As shown in the figures, the design problems have complicated PFs with a long tail at both ends, and the distributions of solutions acquired by the methods are quite different. The PFs of NSGA-II are often wider than those of MOEA/D and iMOEA/D. Their solution quality, however, is not as good as those of MOEA/D and iMOEA/D. A comparison between MOEA/D and iMOEA/D shows that iMOEA/D always offers a better enhancement on the spread of solutions over the PFs.

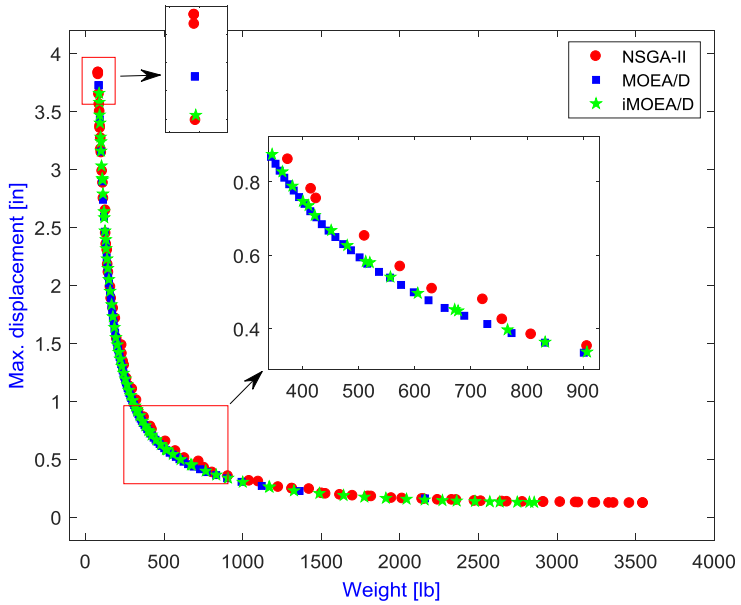


Fig. 9. Pareto fronts with the largest HV among 20 runs for the 15-bar planar truss.

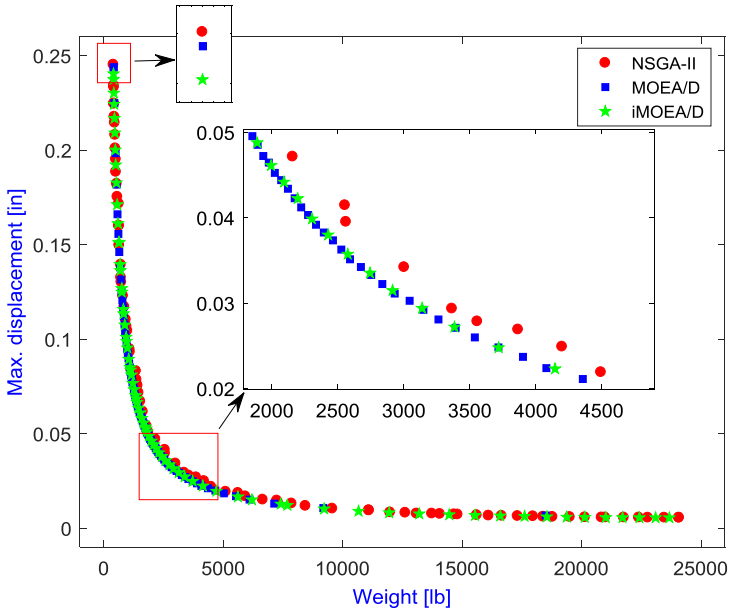


Fig. 10. Pareto fronts with the largest HV among 20 runs for the 72-bar space truss.

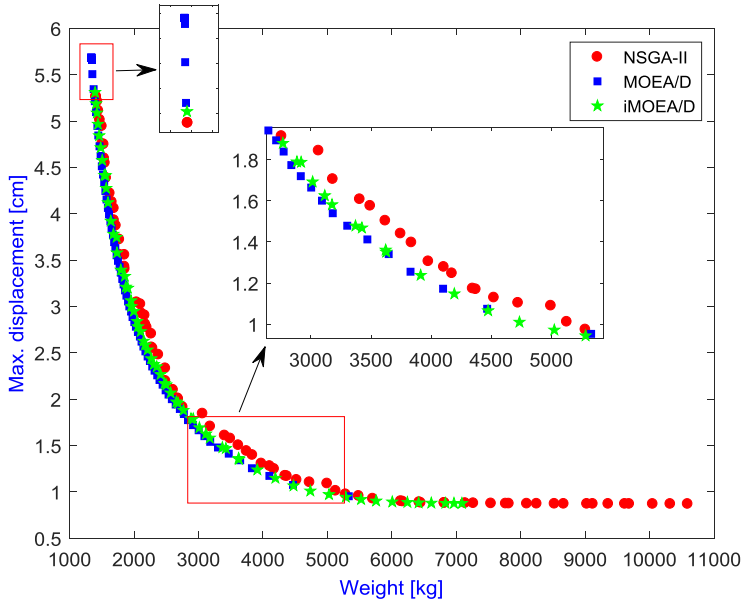


Fig. 11. Pareto fronts with the largest HV among 20 runs for the 160-bar space truss.

The comparison of the obtained solutions with those of single-objective optimization approaches in the literature is presented in Table 5, where the results of iMOEA/D, MOEA/D, and NSGA-II are extracted from the end of the second objective, i.e. maximum displacement as shown in Fig. 9, Fig. 10, and Fig. 11, respectively. From the table, the results indicate that the obtained solutions are reasonable and reliable. Although there is a difference between solutions, this can be due to the different purposes of the employed optimization methods. In fact, single-objective optimization methods only focus on one objective, while multi-objective optimizations methods concentrate on the trade-off between two objectives. This is shown by particular values of the objectives in the table. For example, in the 15-bar truss problem, the method in (Ho-Huu et al., 2015) gives a weight of 74.6818 (lb) with a corresponding maximum displacement of 4.2044 (in), whilst iMOEA/D gives a larger weight of 87.0909 (lb), but with a smaller maximum displacement of 3.6568 (in).

Table 5. Comparison with single objective optimization designs.

Problem	Method	Weight		Maximum displacement	
15-bar truss	Tang et al., 2005	79.820	(lb)	4.2314	(in)
	Rahami et al., 2008	76.685	(lb)	4.1161	(in)
	Ho-Huu et al., 2015	74.681	(lb)	4.2044	(in)
	NSGA-II	82.159	(lb)	3.8358	(in)
	MOEA/D	84.163	(lb)	3.7259	(in)
	iMOEA/D	87.090	(lb)	3.6568	(in)
72-bar truss	Wu & Chow, 1995	427.203	(lb)	0.5996	(in)
	Kaveh et al., 2009	393.380	(lb)	0.2501	(in)
	Ho-Huu et al., 2016	389.334	(lb)	0.2496	(in)
	NSGA-II	401.830	(lb)	0.2452	(in)
	MOEA/D	409.254	(lb)	0.2437	(in)
	iMOEA/D	402.486	(lb)	0.2404	(in)

160-bar truss	Groenwold et al., 1997	1359.781 (kg)	5.6092 (cm)
	Capriles et al., 2007	1348.905 (kg)	5.6525 (cm)
	Ho-Huu et al., 2016	1336.634 (kg)	5.6814 (cm)
	NSGA-II	1412.365 (kg)	5.2638 (cm)
	MOEA/D	1336.794 (kg)	5.6814 (cm)
	iMOEA/D	1396.266 (kg)	5.3089 (cm)

6. Conclusion

In this work, a newly improved version of MOEA/D named iMOEA/D is developed for solving BOPs with complicated PFs. In iMOEA/D, the set of the weight vectors of MOEA/D is numbered and partitioned into two subsets: one set with odd-weight vectors and the other with even ones. Then, a two-phase search strategy based on the MOEA/D framework is developed to optimize their corresponding populations. In the first phase, the population of the set of odd-weight vectors is optimized by using the Tchebycheff function with the ideal point $\mathbf{z}^*$. After that, from the set of obtained solutions, the nadir point $\mathbf{z}^{nad}$ is determined, and the Tchebycheff function with this point is applied for the set of even-weight vectors in the second phase. Moreover, the performance of the iMOEA/D is also further improved by the integration of two recent developments consisting of an adaptive replacement strategy and a stopping criterion.

The reliability, efficiency and the applicability of iMOEA/D are evaluated through the seven existing benchmark test functions with complicated PFs and three optimal design problems of truss structures. The obtained results from the benchmark test functions indicate that iMOEA/D is more competitive than MOEA/D, MOEA/D-TPN and NSGA-II. Although MOEA/D-TPN and NSGA-II provide PFs with better spread compared with iMOEA/D and MOEA/D, the computational cost of MOEA/D-TPN is significantly larger than that of iMOEA/D and MOEA/D, while the quality of optimal solutions of NSGA-II is not as good as that of iMOEA/D and MOEA/D. The obtained results from the practical applications show that iMOEA/D outperforms MOEA/D and NSGA-II, and is a good candidate for solving these kinds of problems.

Although iMOEA/D has shown a considerable improvement in the performance of the algorithm, it is currently still limited to bi-objective optimization problems. For future work, therefore, the idea of the division of weight vectors into two subsets (i.e. even and odd set) should be investigated and extended for optimization problems with more than two objectives. However, to implement this, it should be noted that the distribution of the divided sets should be equal over the weight space, which calls for considerable research efforts in the future. Moreover, although iMOEA/D obtains significant achievements in terms of computational

cost compared to the compared methods, its total number of function evaluations is still high. This will be a major restriction when it is extended and applied to different large problems in civil engineering like frames, composite beams and plates, and for various applications in aerospace engineering, such as the optimal design of departure/arrival routes of aircraft and runway allocations, where the computational cost for each function evaluation is quite large. Thus, in the future, approximate models like artificial neural network (ANN), adaptive neuro fuzzy inference system (ANFIS) (Ramasamy & Rajasekaran, 1996; Rodger, 2014a, 2014b) can also be developed and applied for approximating objective functions to assist iMOEA/D in solving large scale optimization problems in reality.

Furthermore, although the multi-objective optimization problems of truss structures have been solved, suitable criteria for selecting a good candidate from PFs are still an open question. Therefore, studies which aim to help engineering designers to pick a reasonable solution from PFs will also be a potential research direction for researchers in the future.

Acknowledgment

The authors would like to thank the anonymous reviewers for their constructive, helpful and valuable comments and suggestions.

Appendix

The Matlab source codes of MOEA/D, iMOEA/D and application examples can be downloaded from the website: https://www.researchgate.net/profile/V_Ho-Huu.

References

Abdul Kadhar, K. M., & Baskar, S. (2016). A stopping criterion for decomposition-based multi-objective evolutionary algorithms. *Soft Computing*, 1–20. https://doi.org/10.1007/s00500-016-2331-7

Cai, H., & Aref, A. J. (2015). A genetic algorithm-based multi-objective optimization for hybrid fiber reinforced polymeric deck and cable system of cable-stayed bridges. *Structural and Multidisciplinary Optimization*, *52*(3), 583–594. https://doi.org/10.1007/s00158-015-1266-4

Capriles, P. V. S. Z., Fonseca, L. G., Barbosa, H. J. C., & Lemonge, A. C. C. (2007). Rank-based ant colony algorithms for truss weight minimization with discrete variables. *Communications in Numerical Methods in Engineering*, *23*(6), 553–575. https://doi.org/10.1002/cnm.912

Das, I., & Dennis, J. (1998). Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, *8*(3), 631–657. https://doi.org/10.1137/S1052623496307510

Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*. Wiley.

Deb, K., Pratab, S., Agarwal, S., & Meyarivan, T. (2002). A Fast and Elitist Multiobjective Genetic Algorithm: NGSA-II. *IEEE Transactions on Evolutionary Computing*, *6*(2), 182–197. https://doi.org/10.1109/4235.996017

Grandhi, R. (1993). Structural optimization with frequency constraints - A review. *AIAA Journal*, *31*(12), 2296–2303. https://doi.org/10.2514/3.11928

Groenwold, A. A., & Stander, N. (1997). Optimal discrete sizing of truss structures subject to buckling constraints. *Structural Optimization*, *14*(2–3), 71–80. https://doi.org/10.1007/BF01812508

Hartjes, S., & Visser, H. (2016). Efficient trajectory parameterization for environmental optimization of departure flight paths using a genetic algorithm. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *0*(0), 1–9. https://doi.org/10.1177/0954410016648980

Hartjes, S., Visser, H. G., & Hebly, S. J. (2010). Optimisation of RNAV noise and emission abatement standard instrument departures. *Aeronautical Journal*, *114*(1162), 757–767. https://doi.org/10.1017/CBO9781107415324.004

Ho-Huu, V., Nguyen-Thoi, T., Nguyen-Thoi, M. H., & Le-Anh, L. (2015). An improved constrained differential evolution using discrete variables (D-ICDE) for layout optimization of truss structures. *Expert Systems with Applications*, *42*(20), 7057–7069. https://doi.org/10.1016/j.eswa.2015.04.072

Ho-Huu, V., Nguyen-Thoi, T., Vo-Duy, T., & Nguyen-Trang, T. (2016). An adaptive elitist differential evolution for truss optimization with discrete variables. *Computer & Structures*, *165*, 59–75. https://doi.org/10.1016/j.compstruc.2015.11.014

Jan, M. A., & Khanum, R. A. (2013). A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D. *Applied Soft Computing Journal*, *13*(1), 128–148. https://doi.org/10.1016/j.asoc.2012.07.027

Jiang, S., & Yang, S. (2016). An Improved Multiobjective Optimization Evolutionary Algorithm Based on Decomposition for Complex Pareto Fronts. *IEEE Transactions on Cybernetics*, *46*(2), 421–437. https://doi.org/10.1109/TCYB.2015.2403131

Kaveh, A., & Mahdavi, V. R. (2014). Colliding Bodies Optimization method for optimum discrete design of truss structures. *Computer & Structures*, *139*, 43–53.

https://doi.org/10.1016/j.advengsoft.2014.01.002

Kaveh, A., & Talatahari, S. (2009). A particle swarm ant colony optimization for truss structures with discrete variables. *Journal of Constructional Steel Research*, *65*(8–9), 1558–1568. https://doi.org/10.1016/j.jcsr.2009.04.021

Konstantinidis, A., & Yang, K. (2012). Multi-objective energy-efficient dense deployment in Wireless Sensor Networks using a hybrid problem-specific MOEA/D. *Applied Soft Computing Journal*, *12*(7), 1847–1864. https://doi.org/10.1016/j.asoc.2012.04.017

Li, H., & Zhang, Q. (2009). Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, *13*(2), 284–302. https://doi.org/10.1109/TEVC.2008.925798

Li, K., Kwong, S., Zhang, Q., & Deb, K. (2015). Interrelationship-based selection for decomposition multiobjective optimization. *IEEE Transactions on Cybernetics*, *45*(10), 2076–2088. https://doi.org/10.1109/TCYB.2014.2365354

Messac, A., Ismail-Yahaya, A., & Mattson, C. A. (2003). The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, *25*(2), 86–98. https://doi.org/10.1007/s00158-002-0276-1

Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer.

Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., & Wu, J. (2013). MOEA/D with AdaptiveWeight Adjustment. *Evolutionary Computation*, *22*(3), 231–264. https://doi.org/10.1162/EVCO

Rahami, H., Kaveh, a., & Gholipour, Y. (2008). Sizing, geometry and topology optimization of trusses via force method and genetic algorithm. *Engineering Structures*, *30*(9), 2360–2369. https://doi.org/10.1016/j.engstruct.2008.01.012

Ramasamy, J. V, & Rajasekaran, S. (1996). Artificial neural network and genetic algorithm for the design optimizaton of industrial roofs —A comparison. *Computers & Structures*, *58*(4), 747–755. https://doi.org/http://dx.doi.org/10.1016/0045-7949(95)00179-K

Rodger, J. A. (2014a). A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings. *Expert Systems With Applications*, *41*(4), 1813–1829. https://doi.org/10.1016/j.eswa.2013.08.080

Rodger, J. A. (2014b). Application of a Fuzzy Feasibility Bayesian Probabilistic Estimation of supply chain backorder aging , unfilled backorders , and customer wait time using stochastic simulation with Markov blankets. *Expert Systems With Applications*, *41*(16), 7005–7022. https://doi.org/10.1016/j.eswa.2014.05.012

Roudenko, O., & Schoenauer, M. (2004). A Steady Performance Stopping Criterion for

Pareto-based Evolutionary Algorithms. *In Proceedings of the 6th International Multi-Objective Programming and Goal Programming Conference, Hammamet, Tunisia.*

Sharma, S., & Rangaiah, G. P. (2013). An improved multi-objective differential evolution with a termination criterion for optimizing chemical processes. *Computers and Chemical Engineering*, *56*, 155–173. https://doi.org/10.1016/j.compchemeng.2013.05.004

Song, L. (2011). NGPM -- A NSGA-II Program in Matlab, College of Astronautics, Northwestern Polytechnica. https://doi.org/Aerospace Structural Dynamics Research Laboratory College of Astronautics, Northwestern Polytechnical University, China

Storn, R., & Price, K. (1997). Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

Tang, W., Tong, L., & Gu, Y. (2005). Improved genetic algorithm for design optimization of truss structures with sizing, shape and topology variables. *International Journal for Numerical Methods in Engineering*, *62*(13), 1737–1762. https://doi.org/10.1002/nme.1244

Trivedi, A., Srinivasan, D., Sanyal, K., & Ghosh, A. (2016). A Survey of Multiobjective Evolutionary Algorithms based on Decomposition. *IEEE Transactions on Evolutionary Computation*, (c), 1–1. https://doi.org/10.1109/TEVC.2016.2608507

Visser, H. G., & Hartjes, S. (2014). Economic and environmental optimization of flight trajectories connecting a city-pair. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *228*(6), 980–993. https://doi.org/10.1177/0954410013485348

Vo-Duy, T., Duong-Gia, D., Ho-Huu, V., Vu-Do, H. C., & Nguyen-Thoi, T. (2017). Multi-objective optimization of laminated composite beam structures using NSGA-II algorithm. *Composite Structures*, *168*. https://doi.org/http://dx.doi.org/10.1016/j.compstruct.2017.02.038

Wagner, T., & Trautmann, H. (2010). Online convergence detection for evolutionary multi-objective algorithms revisited. *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, 3554–3561. https://doi.org/10.1109/CEC.2010.5586474

Waldock, A., & Corne, D. (2011). Multiple Objective Optimisation Applied to Route Planning. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation* (pp. 1827–1834). New York, NY, USA: ACM. https://doi.org/10.1145/2001576.2001821

Wang, Z., Zhang, Q., Li, H., Ishibuchi, H., & Jiao, L. (2017). On the use of two reference

points in decomposition based multiobjective evolutionary algorithms. *Swarm and Evolutionary Computation*, (January), 0–1. https://doi.org/10.1016/j.swevo.2017.01.002

Wang, Z., Zhang, Q., Zhou, A., Gong, M., & Jiao, L. (2016). Adaptive Replacement Strategies for MOEA/D. *IEEE Transactions on Cybernetics*, *46*(2), 474–486. https://doi.org/10.1109/TCYB.2015.2403849

Wu, S.-J., & Chow, P.-T. (1995). Steady-state genetic algorithms for discrete optimization of trusses. *Computers & Structures*, *56*(6), 979–991. https://doi.org/10.1016/0045-7949(94)00551-D

Yang, S., Jiang, S., & Jiang, Y. (2016). Improving the multiobjective evolutionary algorithm based on decomposition with new penalty schemes. *Soft Computing*. https://doi.org/10.1007/s00500-016-2076-3

Zhang, Q., & Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731. https://doi.org/10.1109/TEVC.2007.892759

Zhang, Q., Liu, W., & Li, H. (2009). The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. *2009 IEEE Congress on Evolutionary Computation*. https://doi.org/10.1109/CEC.2009.4982949

Zhu, Y., Wang, J., & Qu, B. (2014). Multi-objective economic emission dispatch considering wind power using evolutionary algorithm based on decomposition. *International Journal of Electrical Power & Energy Systems*, *63*, 434–445. https://doi.org/10.1016/j.ijepes.2014.06.027

Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, *3*(4), 257–271. https://doi.org/10.1109/4235.797969

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117–132. https://doi.org/10.1109/TEVC.2003.810758