Soft Computing (2021) 25:2433–2452
https://doi.org/10.1007/s00500-020-05312-w

Multi-objective learning backtracking search algorithm for economic emission dispatch problem

Xinlin Xu[1] · Zhongbo Hu[2] · Qinghua Su[2] · Zenggang Xiong[3] · Mianfang Liu[4]

Published online: 1 October 2020

Abstract
The backtracking search algorithm (BSA) as a novel intelligent optimizer belongs to population-based evolutionary algorithms. In this paper, a multi-objective learning backtracking search algorithm (MOLBSA) is proposed to solve the environmental/economic dispatch (EED) problem. In this algorithm, we design two novel learning strategies: a leader-choosing strategy, which takes a sparse solution from an external archive as leader; a leader-guiding strategy, which updates individuals with the guidance of leader. These two learning strategies have outstanding performance in improving the uniformity and diversity of obtained Pareto front. The extreme solutions, compromise solution and three metrics obtained by MOLBSA are further compared with those of well-known multi-objective optimization algorithms in IEEE 30-bus 6-unit test system and 10-unit test system. Simulation results demonstrate the capability of MOLBSA in generating well-distributed and high-quality approximation of true Pareto front for the EED problem.

Keywords Backtracking search algorithm · Environmental/economic dispatch · Multi-objective optimization

1 Introduction

The backtracking search algorithm (BSA) (Civicioglu 2013), a population-based evolutionary algorithm (EA), was first proposed by Civicioglu in 2013 as a novel approach to solve nonlinear, non-differentiable and multi-modal numerical optimization problems. Compared with some similar evolutionary algorithms, BSA has only one single control parameter ($mixrate$). More particularly, BSA possesses a memory called old population to provide search direction for the mutation, which stores a population randomly selected from the current generation or a previous generation. Over the past few years, there have been some successful applications of BSA in various fields (Wang et al. 2019; Pare et al. 2018; Mohd Zain et al. 2018; Hannan et al. 2018; Abdolrasol et al. 2018). It is widely used in engineering fields, such as power system (Modiri-Delshad and Rahim 2014; Ali 2015; Modiri-Delshad et al. 2016; Pal et al. 2016; Modiri-Delshad et al. 2016; Dubey et al. 2016; Bhattacharjee et al. 2015; Modiri-Delshad and Rahim 2016; Kılıç 2015; Ayan and Kılıç 2016; Chaib et al. 2016; Ishak et al. 2014; El-Fergany 2015a, b; Khamis et al. 2015a, b; Gupta et al. 2015; El-Fergany 2016; Khamis and Tai 2017; Khamis et al. 2015; Shahriar et al. 2015; Shafiullah et al. 2015; Niamul Islam et al. 2016; Islam et al. 2017; Shafiullah et al. 2017; De Souza et al. 2016; Najibi et al. 2016), control engineering (Ali et al. 2015, 2016), digital image processing (Eskandari and Toygar 2015; Atasever et al. 2014), antenna array (Guney et al. 2014; Guney and Durmus 2015, 2016) and machine layout and measurement (Vitayasak et al. 2016).

Communicated by V. Loia.

✉ Zhongbo Hu
huzbdd@yangtzeu.edu.cn

Xinlin Xu
xuxlyangtze@163.com

Qinghua Su
suqhdd@yangtzeu.edu.cn

Zenggang Xiong
jkxxzg2003@163.com

Mianfang Liu
mfliu106@163.com

[1] School of Computer Science and Technology, Wuhan University of Technology, Wuhan, Hubei, China

[2] School of Information and Mathematics, Yangtze University, Jingzhou, Hubei, China

[3] School of Computer and Information Science, Hubei Engineering University, Xiaogan, Hubei, China

[4] School of Software and Information Engineering, Hunan Software Vocational Institute, Xiangtan, Hunan, China

As a young intelligence optimization algorithm, BSA has been widely researched and applied in power system since it was proposed in 2013. At present, we can find the references on the application of BSA in power system no less than 30. The number of these references is the largest number of BSA publications in the engineering field. These literatures can be classified as Table 1. The following three conclusions can be given from the table.

- Only one literature (Ali 2015) proposes the improved BSA to solve power system problem and the remaining ones adopt the basic BSA.
- Only one literature (Modiri-Delshad and Rahim 2016) presents the multi-objective BSA with non-dominated approach to handle multi-objective problems and the remaining ones use the basic BSA with weighted sum method.
- The ED problem and the EED problem developed on the ED problem are the research hotspot of BSA application in power system.

Many researchers have been interested in solving the bi-objective EED problem. Initially, some researchers adopt conventional optimization methods (Talaq et al. 1994; Farag et al. 1995). However, these conventional optimization methods can not achieve ideal results in the complex bi-objective EED problem. Therefore, with the rise of meta-heuristics, using meta-heuristics to solve the bi-objective EED problem has become a research hotspot (Zhang et al. 2012; Jubril et al. 2014; Bhattacharjee et al. 2014). The reported methods in the literatures of BSA application to handle the EED problem can be separated into two groups. The first method is weighting aggregation. Bhattacharjee et al. (2015) used the linear combination of two different objectives as a weighted sum to transform the bi-objective EED problem into a single-objective problem. The second method is multi-objective BSA. Modiri-Delshad and Rahim (2016) used multi-objective BSA to handle the both objectives (fuel cost and NO_x emission) in the EED problem simultaneously, and an elitist external archive is adopted to store non-dominated solutions. The first method is easy to be implemented, while its common defect is that it needs to run many times to obtain the trade-off non-dominated solution set by varying different weights. The second method can not find the solutions with a uniform distribution.

There are many other natural heuristics applied to EED problem except BSA, and nature-inspired algorithm is the most successful and suitable method. Qu et al. (2017) have reviewed the application of multi-objective evolutionary algorithms (MOEAs) in solving bi-objective EED problem, mainly including GA-based approaches, PSO-based approaches, DE-based approaches and some hybrid algorithms. Although the classical EED problem has a lot of publications, the research of MOEAs on EED problem is still in its infancy. Moreover, there are few studies on the improvement of multi-objective BSA. The existing research does not provide a unified comprehensive evaluation of the accuracy, stability and time efficiency of algorithm.

In this paper, an enhanced multi-objective learning backtracking search algorithm (MOLBSA) is proposed to solve the bi-objective EED problem. This algorithm use the information of the solution selected in elitist external archive to enhance the diversity and uniformity of obtained Pareto front. The primary contributions of MOLBSA are that it uses two different strategies:

- **Leader-choosing strategy** The solutions with the maximum distance between non-dominated solutions are used in the course of searching to enhance the uniformity performance of approximate solutions.
- **Leader-guiding strategy** Some individuals are replaced by the leaders' neighbors, while others are updated by learning from leader.

The remainder of this paper is organized as follows: Section 2 formulates the mathematical model of EED problem. Section 3 briefly reviews the basic BSA. Section 4 presents MOLBSA. Section 5 describes the search behavior of two strategies. Section 6 describes how to implement MOLBSA in the EED problem. Section 7 provides the comparison of several existing optimization methods. Finally, Section 8 gives the concluding remarks.

2 Mathematical model of the EED problem

As a nonlinear bi-objective optimization problem, EED problem is the simultaneous minimization of the fuel cost and emission while satisfying a real power balance equality constraint and several boundary constraints. The objective functions and constraints are described as follows. The parameter setting and function meaning of the EED problem are introduced in more detail in Saadat (1979).

2.1 Objective functions

2.1.1 Fuel cost minimization

Considering the valve point loading effects, the total fuel cost $f_c(P_u)$(\$/h) of D generators can be represented by adding a sine component to a quadratic function. The expression of the $f_c(P_u)$ can be as Eq. (1):

Table 1 Summary of applications of BSA to power system

Name of Sub Areas	Explanation of Sub Areas	Types of DE Applied and References
Economic dispatch (ED) (Modiri-Delshad and Rahim 2014; Ali 2015; Modiri-Delshad et al. 2016; Pal et al. 2016; Modiri-Delshad et al. 2016; Dubey et al. 2016)	A fundamental economic operation problem to meet electrical demand minimum cost by allocating the generating unit outputs	BSA with considering constraint in objective function (Modiri-Delshad and Rahim 2014), Memetic BSA (Ali 2015), BSA with valve point effects (VPE) and multiple fuel options (MFO) (Modiri-Delshad et al. 2016), BSA with ramp rate limit (RRL) and prohibited operating zones (POZ) (Pal et al. 2016), BSA in a microgrid system (Modiri-Delshad et al. 2016), BSA in multi-area economic dispatch (MAED) (Dubey et al. 2016)
Environmental/economic dispatch (EED) (Bhattacharjee et al. 2015; Modiri-Delshad and Rahim 2016)	A bi-objective problem with two conflicting objectives (fuel cost and emission) to minimize simultaneously under satisfying constraints	BSA with weighting aggregation (Bhattacharjee et al. 2015), multi-objective BSA with an external archive (Modiri-Delshad and Rahim 2016)
Optimal power flow (OPF) (Kılıç 2015; Ayan and Kılıç 2016; Chaib et al. 2016)	Adjusting the optimal control variables by satisfying all operation constraints and minimizing a certain performance index (such as generation cost or transmission power losses)	BSA with VPE and POZ (Kılıç 2015), BSA in two-terminal high-voltage direct current (HVDC) systems (Ayan and Kılıç 2016), BSA considering fuel cost, MFO and VPE objectives (Chaib et al. 2016)
Distribution systems' network reconfiguration (Ishak et al. 2014; El-Fergany 2015a, b; Khamis et al. 2015a, b; Gupta et al. 2015; El-Fergany 2016; Khamis and Tai 2017; Khamis et al. 2015)	Getting optimal the location and size of distributed generation (DG) based on maximum power stability index while minimization of network loss and DG cost	BSA with voltage stability index (Ishak et al. 2014), BSA with fuzzy expert rules (El-Fergany 2015a), BSA in 69-bus and 136-bus test systems (El-Fergany 2015b), BSA in load shedding scheme (Khamis et al. 2015a), BSA in islanding detection (Khamis et al. 2015b), BSA considering different load models (Gupta et al. 2015), BSA with weighting aggregation (El-Fergany 2016), BSA considering linear static voltage stability margin (VSM) and amount of load (active and reactive) curtailment (Khamis and Tai 2017; Khamis et al. 2015)
Power system stabilizers (PSS) (Shahriar et al. 2015; Shafiullah et al. 2015; Niamul Islam et al. 2016; Islam et al. 2017; Shafiullah et al. 2017)	Simultaneous stabilization of multi-machine power system networks by tuning the parameters of PSS	BSA employed in design of unified power flow controller (UPFC) (Shahriar et al. 2015), BSA in damping out the oscillations (Shafiullah et al. 2015), BSA in coordination design of PSS and thyristor-controlled series compensation (TCSC) damping controller (Niamul Islam et al. 2016), BSA in 2-AREA and 5-AREA (Islam et al. 2017), BSA in single machine infinite bus (SMIB) system (Shafiullah et al. 2017)
Others (De Souza et al. 2016; Najibi et al. 2016)		BSA in topology optimization of transmission line tower (De Souza et al. 2016), BSA in micro-grids (MG) problem (Najibi et al. 2016)

$$f_c(P_u) = \sum_{j=1}^{D}(a_j + b_j P_{uj} + c_j P_{uj}^2)$$
$$+|d_j \times sin(e_i(P_{uj}^{\min} - P_{uj}))| \quad (1)$$

where j=1,2,...,D. The vector P_u=(P_{u1},P_{u2},...,P_{uD}) is the output power of generators. P_{uj} is the output power of the jth generator. a_j, b_j, c_j, d_j and e_j are the cost function coefficients for the jth generator. $P_{uj}^{\min}$ is the minimum output power of the jth generator.

2.1.2 Pollution emission minimization

The pollution emission objective function can be expressed as the sum of all kinds of the pollutant emission. But in this study, we only consider the emission of nitrogen oxides NO_x caused by the generators. The total pollution emission $f_c(P_u)$(ton/h) of the NO_x is expressed as the sum of some quadratic and exponential functions, as shown in Eq. (2).

$$f_e(P_u) = \sum_{j=1}^{D}((\alpha_j + \beta_j P_{uj} + \gamma_j P_{uj}^2)10^{-2} + \zeta_j e^{\lambda_j} P_{uj}) \quad (2)$$

where α_j, β_j, γ_j, ζ_j and λ_j are the NO_x emission function coefficients for the jth generator.

2.2 Constraints

2.2.1 Generation capacity constraints

To ensure the stable operation of the power system, the actual output power of each generator should be limited by the upper and lower limits. The formula is as follows.

$$P_{uj}^{\min} \leq P_{uj} \leq P_{uj}^{\max}, \ j = 1, 2, \cdots, D. \quad (3)$$

where $P_{uj}^{\min}$ and $P_{uj}^{\max}$ are the minimum and maximum limits of the jth generator, respectively.

2.2.2 Real Power balance constraint

The total real power generation must be the sum of the transmission power loss PL and the total power demand PD, namely,

$$\sum_{j=1}^{D} P_{uj} = PL + PD \quad (4)$$

Where PL is calculated by Kron's loss formula (Saadat 1979). The calculation formula of PL is as follows.

$$PL = \sum_{j=1}^{D}\sum_{k=1}^{D} P_{uj} B_{jk} P_{uj} + \sum_{j=1}^{D} B_{0j} P_{uj} + B_{00} \quad (5)$$

where B, B_0 and B_{00} are the transmission network power loss coefficients.

2.3 Problem statement

The EED problem is composed of some constraint functions and two nonlinear objective functions which need to be minimized simultaneously. It is mathematically formulated as

$$\begin{aligned} \min \quad & (f_c(P_u), f_e(P_u)) \\ st. \quad & \begin{cases} h(P_u) = 0 \\ g(P_u) \leq 0 \end{cases} \end{aligned} \quad (6)$$

3 Backtracking search algorithm

BSA is a novel population-based EA and is designed for solving numerical optimization problems. It mainly consists of five parts: initialization, selection-I, mutation, crossover and selection-II. Its unique feature is that it uses two populations i.e., current and historical populations. As a memory of BSA, the historical population is randomly selected from the previous generation to calculate the search-direction matrix. BSA uses two random crossover operations to ensure that each generation will produce new trial individuals and retain certain information from the previous generation. The five parts of BSA are described below.

3.1 Initialization

The beginning of BSA is to generate new individuals randomly within a specified range to form a population P and a historical population $oldP$. The mode of population generation is shown in Eq. (7).

$$\begin{cases} P_{i,j} \sim U(P_j^{\min}, P_j^{\max}) \\ oldP_{i,j} \sim U(P_j^{\min}, P_j^{\max}) \end{cases} \quad (7)$$

where $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, D$, N and D are the population size and the dimension of the population, respectively. $P_{i,j}$ and $oldP_{i,j}$ are the jth element of the ith individual in the population P and the historical population $oldP$, respectively. U is a random uniform distribution. $P_j^{\min}$ and $P_j^{\max}$ are the minimum and maximum limits of the jth dimension of individual, respectively.

3.2 Selection-I

This stage is the beginning of each iteration, which is to update and disorder the historical population $oldP$ according to Eq. (8):

$$oldP = \begin{cases} oldP & \text{if } r1 < 0.5 \\ P & \text{otherwise} \end{cases} \tag{8}$$

where $r1$ is a random number from 0 to 1. After assigning the $oldP$, a key operation is to shuffle the $oldP$ randomly.

3.3 Mutation

Unlike other EA, the BSA uses the difference of population P and historical population $oldP$ for its mutation operators. The generation of the initial trial population Mut is shown in Eq. (9).

$$Mut = P + F \cdot (oldP - P), \quad F = 3 \cdot randn \tag{9}$$

where F denotes the control factor to control the amplitude of the search direction, $randn$ is a random number which obeys the standard normal distribution. As shown in Eq. (9), BSA generates the trial population Mut by using the experience of previous generations.

3.4 Crossover

In this process, one or more population dimensions need to be selected. Then, the same dimension elements of individual in P and individual in Mut are exchanged to generate new individual. A matrix Map with the size $N * D$ composed of zero and one is adopted to record this operation. The Map is represented in Eq. (10).

$$\begin{cases} Map_{i,u(1:\lceil D \cdot rand(0,1) \cdot mixrate \rceil)} = 0 & \text{if } r2 < 0.5 \\ Map_{i,randi(D)} = 0 & \text{otherwise} \end{cases} \tag{10}$$

where $u = permuting([1, 2, \cdots, D])$, $permuting(\cdot)$ is a random shuffling function, $mixrate = 1$, $randi(D)$ generates a uniformly distributed random integer in [1, D], $\lceil \cdot \rceil$ rounds the elements to the nearest integers toward infinity, $r2$ is a random number from 0 to 1.

The generation of the final trial population T is calculated by in Eq. (11):

$$T_{i,j} = \begin{cases} P_{i,j} & \text{if} Map_{i,j} = 1 \\ V_{i,j} & \text{otherwise} \end{cases} \tag{11}$$

If some dimensions of individuals in T exceed the feasible region, they are randomly generated within the allowed range.

Eq. (9) in the mutation and Eq. (11) in the crossover are built together in the following formula.

$$T = P + Map \cdot 3 \cdot randn \cdot (oldP - P) \tag{12}$$

3.5 Selection-II

In BSA, this process is to select the individuals with high fitness from the P and T to enter the next generation. The selection mechanism is greedy selection. When the fitness of T_i is better than that of P_i, T_i is used to replace P_i. Otherwise P_i is retained. The selection strategy formula is shown in Eq. (13):

$$P_i^{new} = \begin{cases} T_i & \text{if} fitness(T_i) < fitness(P_i) \\ P_i & \text{otherwise} \end{cases} \tag{13}$$

The best individual of the population is also updated in this process.

4 Multi-objective learning backtracking search algorithm(MOLBSA)

BSA has the characteristics of simple principle and good optimization ability, and has been successfully applied to various single-objective optimization problems. Moreover, BSA's ability to learn from population is weak during the evolution process, which leads to slow convergence speed. One reason is that BSA uses historical information to update all individuals, but doesn't make good use of the best information in current population. Hence, an extended BSA, multi-objective learning backtracking search algorithm (MOLBSA), is designed to solve multi-objective optimization problem. It selects leaders to guide individual updating according to the maximum distance of Pareto optimal solution. MOLBSA's operation strategy will be given in detail below.

4.1 Leader-choosing strategy

A multi-objective optimization problem has two or more conflicting objective functions. However, it is difficult for the basic BSA to solve conflicting multi-objective problem. Because the convergence speed of basic BSA is slow, a leader-choosing strategy is designed, which selects a leader Xg from the external archive to guide individual mutation. In this strategy, a new concept of maximum distance is designed. The maximum distance is used to measure the sparsity of each solution in the external archive, so that the sparse solution can be chosen as a leader according to the sparsity. The md_t of tth non-dominated solution (Ar_t) in Ar is calculated

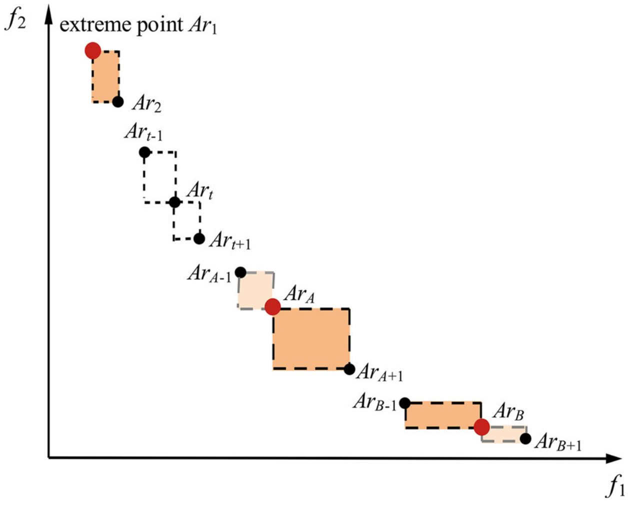


Fig. 1 Maximum distance of MOLBSA

as follows.

$$md_t = \max(ud_t, ld_t) \quad t = 1, \ldots, Nt \tag{14}$$

where Nt is non-dominated solutions number in Ar. The formulas for calculating the distance of Ar_t from the upper point Ar_{t-1} to the lower point Ar_{t+1} are shown in Eq. (15) and (16), respectively.

$$ud_t = \sum_{m=1}^{M} \left| \frac{f_m(Ar_t) - f_m(Ar_{t-1})}{f_m^{\max} - f_m^{\min}} \right| \tag{15}$$

$$ld_t = \sum_{m=1}^{M} \left| \frac{f_m(Ar_t) - f_m(Ar_{t+1})}{f_m^{\max} - f_m^{\min}} \right| \tag{16}$$

where M indicates the objective functions number. $f_m(Ar_t)$ is the mth objective function value of the tth non-dominated solution Ar_t. $f_m^{\min}$ and $f_m^{\max}$ are the minimum and maximum values of the tth objective function, respectively.

In Fig. 1, the solid points represent the non-dominated solutions in Ar. md_t is the relative maximum of the length and width sums of its two adjacent rectangles. The two extreme points of Ar have only one adjacent point, so the md of two extreme points is the distance between the point and the only adjacent point. Fig. 1 shows that the md_1 of the extremum point Ar_1 is the relative value of side-length sum of the matrix composed of Ar_1 and Ar_2.

MOLBSA selects leaders from Ar by using a roulette wheel selection. In **Algorithm 1**, the leader-choosing strategy is described in detail in pseudocode.

In the line 4 of **Algorithm 1**, the calculation formula of P_t is given. P_t is the selection probability of the tth solution in Ar. In **Roulette_wheel_selection**, the input and output are an array of probabilities $p_1, p_2, \cdots, p_{Nt}$ and an index $index_i$, respectively.

Algorithm 1 Leader_choosing

```
Input: $Ar$
Output: $Xg$   % $Xg$: leader solution
1: $Nt = |Ar|$   % $Nt$: non-dominated solutions number in $Ar$
2: for $t = 1\ to\ Nt$ do
3:     calculate $md_t$
4:     $p_t = \frac{md_t}{\sum_{t=1}^{Nt} md_t}$   % $p_t$ is the probability that $Ar_t$ is selected as a leader
5: for $i = 1\ to\ N$ do
6:     % Use roulette wheel selection to select a $Ar_{index}$ according to the probabilities $p_1, p_2, \cdots, p_{Nt}$
7:     $index_i$ = Roulette_wheel_selection($p_1, p_2, \cdots, p_{Nt}$)
8:     $Xg_i = Ar_{index_i}$
```

4.2 Leader-guiding strategy

As is well known, some evolutionary algorithms have shown that learning from the best individual is an effective method for improving the convergence speed of EA. However, only the historical information is used to generate a new individual in the basic BSA. To enhance the learning ability of MOLBSA, leaders' guidance is designed in the process of updating population. The new learning method of MOLBSA, leader-guiding strategy, contains two parts i.e., a wise random walk strategy based on non-dominated solution and a guidance strategy using the leaders and the $oldP$ to generate search direction. First, each individual is replaced by its corresponding leader's neighbor. Second, each individual learns knowledge from the historical population ($oldP$) and the corresponding leader (Xg). The two parts have the same probability to be selected in the following updating equation. The formula for the generation of trial individual T_i is as Eq. 17.

$$T_i = \begin{cases} Xg_i + rand \cdot (Ar_{index_i + l_{index}} - Xg_i), & if\ r3 < 0.5 | r3 \sim U(0,1) \\ P_i + Map_i \cdot (rand \cdot (oldP_i - P_i) & \\ \quad + rand \cdot (Xg_i - P_i)), & otherwise \end{cases} \tag{17}$$

where $i = 1, 2, \cdots, N$. l_{index} is the sparse direction of $index$th solution of the archive.

Eq. 18 gives the design formula of the sparse distance l_t for tth solution in Ar. l_t is recorded by comparing ud_t and ld_t. If ld_t is greater than ud_t, then $l_t = 1$. Otherwise, $l_t = -1$.

$$l_t = \begin{cases} 1, & ld_t > ud_t \\ -1, & ld_t \le ud_t \end{cases}, \quad t = 1, 2, \cdots, Nt \tag{18}$$

Figure 2 shows the sparse directions of Ar_1, Ar_A and Ar_B. In this figure, ld_A is greater than ud_A for Ar_A, so $l_A = 1$. For the Ar_B, ud_B is greater than ld_B, hence $l_B = -1$. For the first extreme point Ar_1, its neighbor has only Ar_2, so $l_1 = 1$.

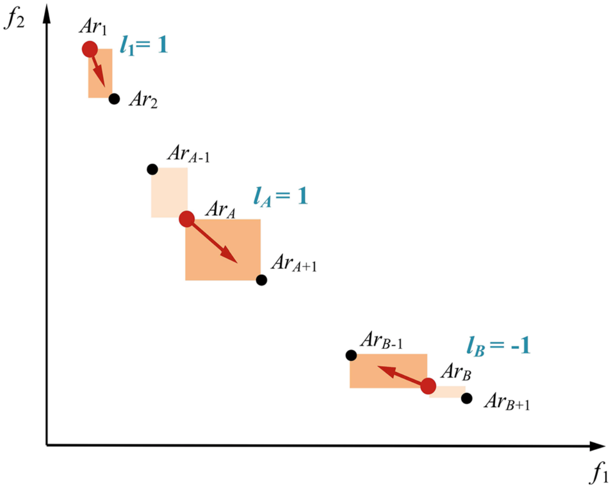


Fig. 2 Sparse direction of MOLBSA

On the one hand, the strategy makes the population P move around Xg, thus generates a new population T, which can make the Pareto front more uniform. In Fig. 2, the red arrow indicates the direction of the non-dominated solution. Ar_A will move to its neighbor Ar_{A+1} because there is a big gap between them.

On the other, the strategy brings the population P nearer to the population $oldP$ and the current archive Ar.

4.3 Update and maintenance of archive

Different from the single objective algorithms, the multi-objective algorithms need to output a non-dominated solution set at the end after cycle termination. It is essential to maintain the quantity and quality of non-dominated solutions in search process. Since Goldberg (1989) put forward a selection strategy with non-dominated sorting in niched pareto genetic algorithm (NSGA) in 1989. Srinivas and Deb (1994) described and used the non-dominated sorting. In 1999, Zitzler and Thiele (1999) introduced formally strength pareto evolutionary algorithm (SPEA) with elitist reservation mechanism. Deb et al. (2002) designed a fast non-dominated sorting, which is more efficient non-dominated sorting technique.

This paper uses an external archive with a specific volume to save non-dominated solutions in each iteration. Initially, the non-dominated solutions are stored into this empty external archive. There are three rules when a trial vector X compares with the current external archive Ar as follows.

(i) if $\exists a, a \in Ar, a \prec X$, then X is rejected into Ar;
(ii) if $\exists a, a \in Ar, X \prec a$, then $Ar = Ar/a \wedge Ar = Ar \cup X$;
(iii) if $\forall a, a \in Ar, X \nprec a \wedge a \nprec X$, then $Ar = Ar \cup X$

When the non-dominated solutions number in the external archive reaches its own capacity, the crowding distance designed by Deb et al. (2002) in NSGA-II is used to remove redundant members with small crowding distance, so as to ensure that the solutions number in the external archive will not exceed the capacity. In NSGA-II, the crowding distance of the external archive is calculated only once in each iteration, which may lead to the sparsity of the congested area on the Pareto front after one iteration. Therefore, a cyclic crowding sorting technique (Luo et al. 2010) is generated to enhance the uniformity of Pareto front. The cyclic crowded sorting algorithm is illustrated as the following **Algorithm 2**.

Algorithm 2 Cyclic_crowding_sorting

```
Input: $Ar$, $Na$   % $Ar$: external archive, $Na$: maximum capacity of $Ar$
Output: $newAr$   % The number of solutions of $newAr$ is equal to the maximum capacity of $Ar$
1: $Nt = |Ar|$   % $Nt$: non-dominated solutions number in $Ar$
2: while $Nt > Na$ do   % Check whether the number of solutions in $AR$ exceeds
3:     for $t = 1$ to $Nt$ do   % Initialize the crowding distances
4:         $Ar_t.distance = 0$   % $Ar_t$ is the $t$th non-dominated solution in $Ar$
5:     for $m = 1$ to $M$ do   % $M$: objective functions number
6:         $Ar := sort(Ar, m)$   % Sort $Ar$ according to $m$th objective function value
7:         $Ar_1.distance = Inf$; $Ar_{Nt}.distance = Inf$ % Set the crowding distances of extreme points to infinity
8:         for $t = 2$ to $Nt - 1$ do
9:             $Ar_t.distance = \frac{Ar_{t+1}.distance - Ar_{t-1}.distance}{Ar_{Nt}.distance - Ar_1.distance} + Ar_t.distance$
10:    $k = min\_Ar.distance$   % $k$: the sequence number of solution with minimum crowding distance
11:    $Ar_k = [\ ]$   % Remove $Ar_k$ from $Ar$
12:    $Nt = Nt - 1$   % Update $Nt$
13: $newAr = Ar$
```

In **Algorithm 2**, we first check to see if the number of solutions in Ar has exceeded the maximum capacity (Na) of Ar. If it exceeds, the crowding distance of each solution is calculated. The formula of crowding distance is presented in the line 9 of **Algorithm 2**, and the crowding degree of extremum solutions are set as infinite. It is necessary to find out index of the solution with the minimum crowding distance and eliminate the solution in Ar. Again, we check whether the number of residual solutions in Ar exceeds the maximum capacity of Na. If it exceeds, the algorithm enters the loop again, otherwise, the algorithm stops the loop. Finally, we get an external archive Ar whose number of solutions is equal to the maximum capacity Na.

4.4 Algorithm procedures

Algorithm 3 shows the pseudocode for MOLBSA. In **Algorithm 3**, the two novel learning strategies are shown on lines 14, 16 and 23, respectively, and are shown in bold and underline. The update and maintenance of Ar in lines 20-32. Figure 3 is a flowchart of the algorithm.

Algorithm 3 Multi-objective learning backtracking search algorithm

```
Input: input parameters $N, D, Na, T^{max}, mixrate, P^{min}_{1:D}, P^{max}_{1:D}$   % $T^{max}$: maximum number of iterations
Output: $Op$
1:  //Initialization
2:  for $i = 1$ to $N$ do
3:      for $j = 1$ to $D$ do
4:          $P_{i,j} \sim U(P^{min}_j, P^{max}_j), oldP_{i,j} \sim U(P^{min}_j, P^{max}_j)$   % Initialize $P$ and $oldP$
5:  $fitnessP = ObjFun(P)$   % Evaluate the fitness of $P$
6:  $Ar \leftarrow Non\_dominated(P)$   % Find the non-dominated solutions of $P$ into $Ar$
7:  while $tt \le T^{max}$ do
8:      //selection-I
9:      if $rand < 0.5$ then $oldP := P$   % $rand$: generate a random number from [0, 1]
10:     $oldP := permuting(oldP)$   % $permuting$: disrupt the order of individuals in oldP
11:     $Map_{1:N,1:D} = ones(N, D)$   % Generate an N-by-D matrix of ones
12:     //Mutation
13:     for $i = 1$ $to$ $N$ do
14:         $\underline{Xg_i = \textbf{Leader\_choosing}(Ar)}$   % Find leader $Xg$ of each individual
15:         if $rand < 0.5$ then
16:             $\underline{T_i = Xg_i + rand \cdot (Ar_{index_i + l_{index}} - Xg_i)}$   % Directed search $Xg_i$ neighbor to update $P_i$
17:         else
18:             //Crossover
19:             if $rand < 0.5$ then   % Cross matrix (Map) generating
20:                 $Map_{i,u(1:\lceil D \cdot rand(0,1) \cdot mixrate \rceil)} = 0$
21:             else
22:                 $Map_{i,randi(D)} = 0$
23:             $\underline{T_i = P_i + Map_i \cdot (rand \cdot (oldP_i - P_i) + rand \cdot (Xg_i - P_i))}$
24:     //Selection-II
25:     $fitnessT = ObjFun(T)$   % Calculate the fitness value
26:     for $i = 1$ to $N$ do
27:         if $T_i \prec P_i$ then
28:             $P_i = T_i$
29:             $fitnessP_i = fitnessT_i$
30:     $Ar \leftarrow Non\_dominated(P \bigcup Ar)$   % Update the external archive $Ar$
31:     if $|Ar| > Na$ then
32:         $\underline{Ar = \textbf{Cyclic\_crowding\_sorting}(Ar)}$   % Maintain the archive
33:     $tt = tt + 1$
34: $Op \leftarrow Ar$
```

5 Analysis of the search behavior of MOLBSA

As mentioned earlier, the proposed algorithm MOLBSA has two core strategies. The first strategy (strategy1) is the leader-choosing strategy. It calculates the sparsity of non-dominated solutions by Eq. (14), and chooses the sparse non-dominated solution as the leader to serve for the second strategy. The second strategy (strategy2), leader-guiding strategy, operates through Eq. (17) which randomly selects one of two mutation operators to mutate individuals. The first operator is to make the Pareto front more uniform. The second operator is to preserve the global search ability of the population.

In order to illustrate the search behavior of MOLBSA and the effectiveness of its two strategies, three algorithms are designed and compared as follows.

(i) **multi-objective BSA** (origin MOBSA for short).
(ii) **Origin MOBSA + strategy2**: being build by adding the leader-guiding strategy to origin MOBSA
(iii) **Origin MOBSA+strategy1+strategy2**(MOLBSA): being formulated by adding the leader-choosing strategy and the leader-guiding strategy to origin MOBSA.

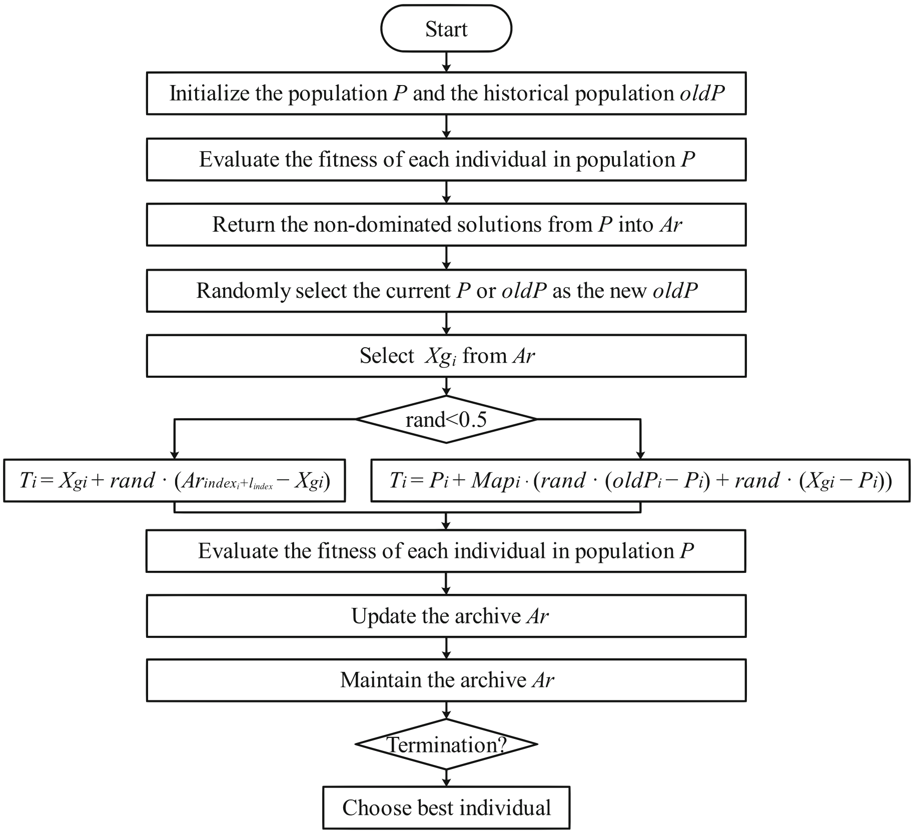
In this little test, three algorithms are applied to a simple bi-objective problem, which is shown in the following Eq. (19). The parameters including $N = 50$,$D = 2$,$Na = 30$,$T^{max} = 40$ are provided.

$$\begin{cases} \min f_1 = 4x_1^2 + 4x_2^2 \\ \min f_2 = (x_1 - 5)^2 + 4(x_2 - 5)^2 \\ st.\ 0 < x_1 < 5,\ \ 0 < x_2 < 3 \end{cases} \tag{19}$$

Figure 4 shows the Pareto front obtained by the three algorithms. The following two results can be obtained from the figure:

(i) From Fig. 4, the Pareto front obtained by the origin MOBSA + strategy2 has better uniformity than origin MOBSA. This illustrates that the strategy2 (leader-guiding strategy) enhances the uniformity of Pareto front. The Pareto front obtained by the origin MOBSA + strategy1 + strategy2 has slightly better uniformity than that of the origin MOBSA + strategy2. This is because the strategy1 (the leader-choosing strategy) plays a significant role in the origin MOBSA + strategy1 + strategy2. The strategy1 selects the sparse solution of external archive as the leader. Under the action of the strategy2, more solutions are generated near the sparse solution, thus the distribution of the obtained Pareto front is more uniform.
(ii) In the figure, the extreme points of the origin MOBSA + strategy2 (represented by two green diamonds) has slightly better coverage than those of the origin

Fig. 3 The flowchart of MOLBSA


Start
Initialize the population P and the historical population $oldP$
Evaluate the fitness of each individual in population P
Return the non-dominated solutions from P into Ar
Randomly select the current P or $oldP$ as the new $oldP$
Select Xg_i from Ar
rand<0.5
$T_i = Xg_i + rand \cdot (Ar_{index_i+l_{index}} - Xg_i)$
$T_i = P_i + Map_i \cdot (rand \cdot (oldP_i - P_i) + rand \cdot (Xg_i - P_i))$
Evaluate the fitness of each individual in population P
Update the archive Ar
Maintain the archive Ar
Termination?
Choose best individual

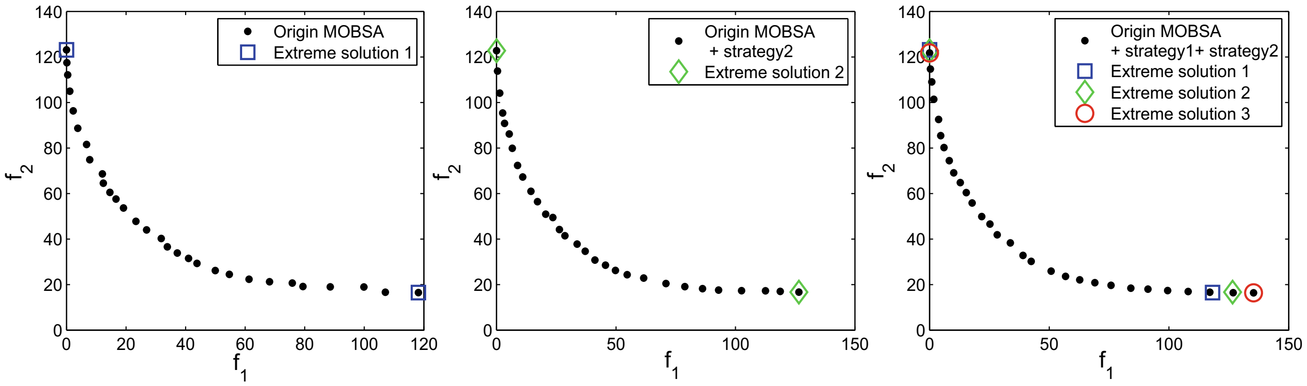


Fig. 4 Comparisons of adding strategies in MOBSA

MOBSA represented by two blue squares. The extreme points obtained by the origin MOBSA + strategy1 + strategy2 (represented by two red circles) has slightly better coverage than those of the origin MOBSA + strategy2. Therefore, it can be concluded that the leader-choosing strategy and leader-guiding strategy can improve the diversity of Pareto front.

6 Implementation of MOLBSA

In subsection 2.2, we have introduced the constraint of EED problem, which contains one power balance equality constraint. Therefore, a constraint handling mechanism is needed to move infeasible solutions to feasible regions. In addition, a fuzzy theory to select a best compromise solution has been frequently adopted to simulate the preference of decision maker. In this final subsection, the parameter setting of MOLBSA is introduced.

6.1 Constraint handling

In this section, a constraint handling strategy is introduced to solve the power balance equality constraint in the EED problem. To guarantee all solutions can satisfy the equality constraint, scholars adopt a rejecting strategy to handle the power balance equality constraint in the EED problem (Wang and Singh 2007, 2008; Cai et al. 2009). But this method is time-consuming to deal with equality constraint.

In MOLBSA, a straightforward constraint handling approach is provided to deal with the power balance equality constraint. Algorithm 4 gives the detailed operation of **Constrint_Handling**. In Algorithm 4, we set $\sigma = 1e - 12$.

Algorithm 4 Constraint_handling

```
Input: $P_i$, $P^{min}$, $P^{max}$
Output: $P_i$
$e1 = PL + PD - \sum_{j=1}^{D} P_i$   % Calculate error between $PL + PD$ and the sum of $P_i$
while $|e1| > \sigma$ do
    $k = randi(1, D)$   % Select a random integer in [1, D]
    $P_{i,k} = P_{i,k} \cdot (PD + \frac{PL}{\sum_{j=1}^{D} P_i})$   % Adjust $X_i$ to satisfy the constraint
    if $P_{i,k} > P_k^{max}$ or $P_{i,k} < P_k^{min}$ then
        $P_{i,k} = P_k^{min} + rand \cdot (P_k^{max} - P_k^{min})$
    $e1 = PL + PD - \sum_{j=1}^{D} P_i$   % Calculate the error again
```

6.2 Select of compromise solution

After getting Pareto optimal solutions, one solution called compromise solution is selected to satisfy the different goals. The difficulties of trade-off decision is summarized that the different objectives function values are measured in different physical units. Moreover, there are no specific compromise rules between different objectives. In the published literatures (Modiri-Delshad and Rahim 2016; Sivasubramani and Swarup 2011; Wu et al. 2010), a fuzzy-based approach (Sakawa et al. 1987) is usually adopted to simulate preference of a decision maker.

This work calculates the satisfactory degrees of Pareto optimal solutions by using a simple linear membership function. The objective functions of different physical units are scaled by 0-1 metric. The satisfactory degree $\mu_{t,j}$ of Ar_t for f_j is herein defined as

$$\mu_{t,m} = \begin{cases} 1, & f_m(Ar_t) \le f_m^{\min} \\ \frac{f_m^{\max} - f_m(Ar_t)}{f_m^{\max} - f_m^{\min}}, & f_m^{\min} \le f_m(Ar_t) \le f_m^{\max} \\ 0, & f_m(Ar_t) \ge f_m^{\max} \end{cases} \quad (20)$$

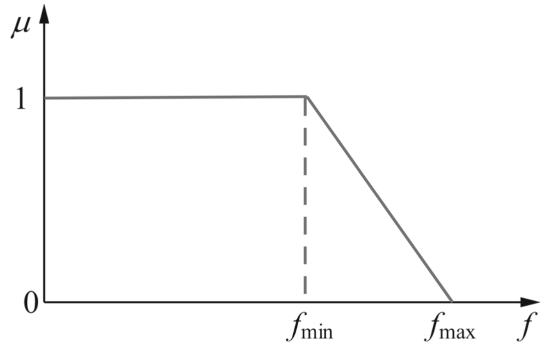


Fig. 5 The fuzzy-based membership function

where $t = 1, 2, \cdots, Nt$, $m = 1, 2, \cdots, M$, Nt and M are the number of solutions in Ar and the number of objective functions, respectively. The M of the bi-objective EED problem is 2. $f_m(Ar_t)$ is the mth objective function of the tth solution in Ar. $\mu_{t,m} = 1$ indicates complete satisfaction, and $\mu_{t,m} = 0$ expresses dissatisfaction. The normalized membership function is defined as follows. The fuzzy membership function is described in Fig.5.

$$\mu_k = \frac{\sum_{m=1}^{M} \mu_{k,m}}{\sum_{t=1}^{Nt} \sum_{m=1}^{M} \mu_{t,m}} \quad (21)$$

The non-dominated solution with maximum μ_k is considered as a compromise solution which represents the decision makers' decision.

6.3 Parameter setting

The proposed MOLBSA is simulated on IEEE 30-bus 6-unit system with the total power demand 2.834 p.u and 10-unit system with the total power demand 2000 MW. Tables 2 and 3 provide the data of fuel cost and NO_x emission coefficients referenced in Basu (2011). The transmission power loss formula coefficients of 6-unit system and 10-unit system are exhibited in Tables 4 and 5.

The parameters for all simulation runs are set as follows. Both the maximum capacity of Ar (Na) and the population size (N) and are set to 50. The number of fitness function evaluations is restricted to 10000 for all experiments as terminating condition. All algorithms are run independently for 30 times. To demonstrate the optimization effectiveness of MOLBSA, the simulations are carried out for two different cases as follows.

- Case1: not consider the transmission power losses in power balance constraint.

Table 2 Fuel cost and NO_x emission coefficients in 6-unit system

Unit	$P^{\min}$	$P^{\max}$	a	b	c	α	β	γ	ζ	λ
u_1	0.05	0.5	10	200	100	4.091	−5.554	6.490	2.0e−4	2.857
u_2	0.05	0.6	10	150	120	2.543	−6.047	5.638	5.0e−4	3.333
u_3	0.05	1.0	20	180	40	4.258	−5.094	4.586	1.0e−6	8.000
u_4	0.05	1.2	10	100	60	5.326	−3.550	3.380	2.0e−3	2.000
u_5	0.05	1.0	20	180	40	4.258	−5.094	4.586	1.0e−6	8.000
u_6	0.05	0.6	10	150	100	6.131	−5.555	5.151	1.0e−5	6.667

Table 3 Fuel cost and NO_x emission coefficients in 10-unit system

Unit	$P^{\min}$	$P^{\max}$	a	b	c	d	e	α	β	γ	ζ	λ
u_1	10	55	1000.403	40.5407	0.12951	33	0.0174	360.0012	− 3.9864	0.04702	0.25475	0.01234
u_2	20	80	950.606	39.5804	0.10908	25	0.0178	350.0056	− 3.9524	0.04652	0.25475	0.01234
u_3	47	120	900.705	36.5104	0.12511	32	0.0162	330.0056	− 3.9023	0.04652	0.25163	0.01215
u_4	20	130	800.705	39.5104	0.12111	30	0.0168	330.0056	− 3.9023	0.04652	0.25163	0.01215
u_5	50	160	756.799	38.5390	0.15247	30	0.0148	13.8593	0.3277	0.00420	0.24970	0.01200
u_6	70	240	451.325	46.1592	0.10587	20	0.0163	13.8593	0.3277	0.00420	0.24970	0.01200
u_7	60	300	1243.531	38.3055	0.03546	20	0.0152	40.2669	− 0.5455	0.00680	0.24800	0.01290
u_8	70	340	1049.998	40.3965	0.02803	30	0.0128	40.2669	− 0.5455	0.00680	0.24990	0.01203
u_9	135	470	1658.569	36.3278	0.02111	60	0.0136	42.8955	− 0.5112	0.00460	0.25470	0.01234
u_{10}	150	470	1356.659	38.2704	0.01799	40	0.0141	42.8955	− 0.5112	0.00460	0.25470	0.01234

Table 4 Transmission power loss formula coefficients in 6-unit system

$B =$	0.1382	− 0.0299	0.0044	− 0.0022	− 0.0010	− 0.0008	$B_{00} = 0.00098573$
	− 0.0299	0.0487	− 0.0025	0.0004	0.0016	0.0041	
	0.0044	− 0.0025	0.0182	− 0.0070	− 0.0066	− 0.0066	
	− 0.0022	0.0004	− 0.0070	0.0137	0.0050	0.0033	
	− 0.0010	0.0016	− 0.0066	0.0050	0.0109	0.0005	
	− 0.0008	0.0041	− 0.0066	0.0033	0.0005	0.0244	
$B_0 =$	− 0.0107	0.0060	− 0.0017	0.0009	0.0002	0.0030	

Table 5 Transmission power loss formula coefficients in 10-unit system

$B =$	4.9e−5	1.4e−5	1.5e−5	1.5e−5	1.6e−5	1.7e−5	1.7e−5	1.8e−5	1.9e−5	2.0e−5
	1.4e−5	4.5e−5	1.6e−5	1.6e−5	1.7e−5	1.5e−5	1.5e−5	1.6e−5	1.8e−5	1.8e−5
	1.5e−5	1.6e−5	3.9e−5	1.0e−5	1.2e−5	1.2e−5	1.4e−5	1.4e−5	1.6e−5	1.6e−5
	1.5e−5	1.6e−5	1.0e−5	4.0e−5	1.4e−5	1.0e−5	1.1e−5	1.2e−5	1.4e−5	1.5e−5
	1.6e−5	1.7e−5	1.2e−5	1.4e−5	3.5e−5	1.1e−5	1.3e−5	1.3e−5	1.5e−5	1.6e−5
	1.7e−5	1.5e−5	1.2e−5	1.0e−5	1.1e−5	3.6e−5	1.2e−5	1.2e−5	1.4e−5	1.5e−5
	1.7e−5	1.5e−5	1.4e−5	1.1e−5	1.3e−5	1.2e−5	3.8e−5	1.6e−5	1.6e−5	1.8e−5
	1.8e−5	1.6e−5	1.4e−5	1.2e−5	1.3e−5	1.2e−5	1.6e−5	4.0e−5	1.5e−5	1.6e−5
	1.9e−5	1.8e−5	1.6e−5	1.4e−5	1.5e−5	1.4e−5	1.6e−5	1.5e−5	4.2e−5	1.9e−5
	2.0e−5	1.8e−5	1.6e−5	1.5e−5	1.6e−5	1.5e−5	1.8e−5	1.6e−5	1.9e−5	4.4e−5
$B_0 = 0$	$B_{00} = 0$									

Table 6 Best solutions for f_c and f_e minimized individually in 6-unit system for two cases

	Case1		Case2	
	Best f_c	Best f_e	Best f_c	Best f_e
u_1	0.109504	0.406044	0.121142	0.410922
u_2	0.299658	0.458929	0.286268	0.463659
u_3	0.524296	0.537951	0.583369	0.544396
u_4	1.016239	0.383080	0.992870	0.390390
u_5	0.524428	0.537847	0.524124	0.544455
u_6	0.359874	0.510149	0.351791	0.515508
f_c(\$/h)	**600.1114**	638.2602	**605.9983**	646.2059
f_e(ton/h)	0.222156	**0.194203**	0.220726	**0.194179**

- Case2: consider the transmission power losses in power balance constraint.

7 Computational results

Experiments in this paper are designed to estimate the quality of the Pareto optimal solution obtained by comparing the simulation results of extreme solutions, compromise solutions, spacing metric(SP), hypervolume (HV) and c-metric (C) in all cases for MOLBSA and other algorithms.

7.1 Comparisons of compromise solutions and extreme solutions in Pareto front

In this part, an experiment is designed to study the performance of MOLBSA by comparisons of the extreme points and compromise solutions obtained by different algorithms. Firstly, the fuel cost and NO_x emissions are independently optimized as a single-objective function. Table 6 shows the results of best solution for 6-unit system in two cases. From this table, when the fuel cost is minimized as a unique objective function, the optimal values obtained by BSA are 600.1114(\$/h) and 605.9983(\$/h) in two cases. When the NO_x emission is only minimized, BSA reaches the optimal values of 0.194203(ton/h) and 0.194179(ton/h) in Case1 and Case2, respectively.

Next, we discuss the multi-objective extreme solutions and compromise solutions obtained by MOLBSA for 6-unit system—Case1. Tables 7 and 8 compare the extreme points of Pareto fronts obtained by MOLBSA , MOBSA, multi-objective stochastic search technique (MOSST) (Das and Patvardhan 1998), non-dominated sorting genetic algorithm (NSGA) (Abido 2003a), niched pareto genetic algorithm (NPGA) (Abido 2003b), strength pareto evolutionary algorithm (SPEA) (Abido 2003c), non-dominated sorting genetic algorithm-II (NSGA-II) (Ah King et al. 2005),and fuzzy clustering-based PSO (FCPSO) (Agrawal et al. 2008). The best results are bold in all tables.

As is revealed in Table 7, the best fuel cost of MOLBSA equals 600.120251(\$/h) when using the minimum 10,000 times function evaluation times (FEs), which is superior to the results of other seven algorithms compared. The error of equality constraint obtained by MOLBSA is 0. This value is better than those of MOSST, NSGA and SPEA, which is the same as those of MOBSA, NPGA, NSGA-II and FCPSO. In Table 8, the best solution for the emission obtained by MOLBSA equals 0.194200(ton/h) when only using the minimum 10,000 times FEs. Although this value is not the smallest of the eight algorithms listed in Table 8, it is also idea with ranking second. The minimum emission value obtained by MOSST is 0.19418(ton/h), but MOSST's the error of equality constraint is not ideal equal to 0.027000. However, the error of equality constraint by using MOLBSA is 0. In Tables 7and 8, MOLBSA only use 10,000 times FEs. This FEs value is the same as those of MOBSA and NSGA-II, and is smaller than those of other comparative algorithms. From the discussion, it can be concluded that MOLBSA is more effective than almost all other seven algorithms.

Table 9 shows the compromise solutions of six algorithms. Since these compromise solutions have no dominance relationship, it is difficult to compare them directly. The method of decision-maker's average satisfactory degree (ASD) mentioned in Zhang et al. (2012) is adopted to estimate the quality of compromise solutions. The ASD of compromise solution is calculated by

$$\bar{\mu}^{com} = \frac{1}{M}\sum_{m=1}^{M}\mu_m^{com} \tag{22}$$

In Table 9, the $\bar{\mu}^{com}$ value obtained by MOLBSA is 0.757962, which is larger than those of the other five algorithms. This proves that the satisfaction of compromise solution obtained by MOLBSA is higher than those of other five algorithms to some extent.

Fig.6 depicts the approximations of the true Pareto front obtained by MOLBSA and MOBSA. From Fig.6, the distribution of Pareto front obtained by MOLBSA is more uniform than that of MOBSA in 6-unit system—Case1.

Tables 10 and 11 show the numerical results of two extreme points in Pareto front (minimum cost and minimum emission) for seven algorithms as shown in 6-unit system—Case2. The numerical results of MOLBSA are compared with those of well-known MOBSA, NSGA (Abido 2003a), NPGA (Abido 2003b), SPEA (Abido 2003c), NSGA-II (Ah King et al. 2005) and FCPSO (Agrawal et al. 2008).

From Table 10, MOLBSA gains the minimum fuel cost equal to 606.00818(\$/h) with fewer FEs. Not only the minimum fuel cost of MOLBSA is lower than those of the

Table 7 Best solutions for f_c in 6-unit system—Case1

	MOLBSA	MOBSA	MOSST (Das and Patvardhan 1998)	NSGA (Abido 2003a)	NPGA (Abido 2003b)	SPEA (Abido 2003c)	NSGA-II (Ah King et al. 2005)	FCPSO (Agrawal et al. 2008)
u_1	0.105975	0.106963	0.112500	0.156700	0.108000	0.106200	0.105900	0.107000
u_2	0.296339	0.289092	0.302000	0.287000	0.328400	0.289700	0.317700	0.289700
u_3	0.523213	0.543369	0.531100	0.467100	0.538600	0.528900	0.521600	0.525000
u_4	1.018500	1.019590	1.020800	1.046700	1.006700	1.002500	1.014600	1.015000
u_5	0.534317	0.516086	0.531100	0.503700	0.494900	0.540200	0.515900	0.530000
u_6	0.355654	0.358900	0.362500	0.372900	0.357400	0.366400	0.358300	0.367300
f_e(ton/h)	0.222578	0.222727	0.222200	0.222820	0.221160	0.221500	0.221880	0.222300
f_c(\$/h)	**600.1202**	600.1438	605.8890	600.5720	600.2590	600.1500	600.1550	600.1315
$\lvert\sum_{j=1}^{D} P_{uj} - PD\rvert$	**0**	**0**	0.202600	1e-04	**0**	1e-04	**0**	**0**
FEs	**10,000**	**10,000**	-	100,000	100,000	100,000	**10,000**	20,000

Table 8 Best solutions for f_e in 6-unit system—Case1

	MOLBSA	MOBSA	MOSST (Das and Patvardhan 1998)	NSGA (Abido 2003a)	NPGA (Abido 2003b)	SPEA (Abido 2003c)	NSGA-II (Ah King et al. 2005)	FCPSO (Agrawal et al. 2008)
u_1	0.407923	0.400065	0.409500	0.439400	0.400200	0.411600	0.407400	0.409700
u_2	0.459555	0.462015	0.462600	0.451100	0.447400	0.453200	0.457700	0.455000
u_3	0.540294	0.547272	0.542600	0.510500	0.516600	0.532900	0.538900	0.536300
u_4	0.374542	0.385022	0.388400	0.387100	0.368800	0.383200	0.383700	0.384200
u_5	0.537124	0.522121	0.542700	0.555300	0.575100	0.538300	0.535200	0.534800
u_6	0.514562	0.517504	0.515200	0.490500	0.525900	0.514800	0.511000	0.514000
f_c(\$/h)	639.1857	638.1144	644.1120	639.2310	639.1820	638.5100	638.2690	638.3577
f_e(ton/h)	0.194200	0.194225	**0.194180**	0.194360	0.194330	0.194200	0.194200	0.194200
$\lvert\sum_{j=1}^{D} P_{uj} - PD\rvert$	**0**	**0**	0.027000	1e-04	**0**	**0**	1e-04	**0**
FEs	**10,000**	**10,000**	-	100,000	100,000	100,000	**10,000**	20,000

Table 9 Best compromise solutions in 6-unit system—Case1

	u_1	u_2	u_3	u_4	u_5	u_6	f_c(\$/h)	f_e(ton/h)	$\bar{\mu}^{com}$
MOLBSA	0.248627	0.380983	0.553120	0.683372	0.552311	0.441996	616.081773	0.200552	**0.757962**
MOBSA	0.261512	0.378118	0.565218	0.675094	0.552174	0.428139	616.526846	0.200241	0.777384
NSGA (Abido 2003a)	0.257100	0.377400	0.538100	0.687200	0.540400	0.433700	610.067000	0.200600	0.755100
NPGA (Abido 2003b)	0.269600	0.367300	0.559400	0.649600	0.539600	0.448600	612.127000	0.199410	0.749100
SPEA (Abido 2003c)	0.278500	0.376400	0.530000	0.693100	0.540600	0.415300	610.254000	0.200550	0.752700
FCPSO (Agrawal et al. 2008)	0.319300	0.393400	0.535900	0.592100	0.545700	0.447000	619.998000	0.197150	0.726700

other six algorithms, but also its FEs is the smallest. As is demonstrated in Table 11, the NO_x emissions of MOLBSA is 0.194196(tons/h). Its value is only 0.0006(tons/h) larger than that of NSGA-II, but smaller than those of the other five methods. Therefore, the above results show that MOLBSA can obtain more extensive Pareto front compared with the other six algorithms.

7.2 Comparison of solution quality

Three performance evaluation criteria are generally introduced to evaluate the performance of Pareto optimal solutions obtained by MOEAs (Zitzler 1999).

- **Uniformity** The distribution of the obtained Pareto front is as uniform as possible.

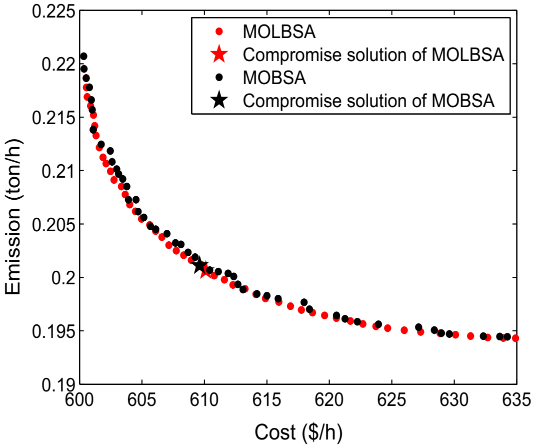


Fig. 6 Pareto fronts obtained by MOLBSA and MOBSA in 6-unit system—Case1

- **Diversity** The distribution of the obtained Pareto front is as wide as possible.
- **Convergence** The obtained Pareto optimal set is as close as possible to the true Pareto optimal front.

In order to evaluate the quality of the Pareto optimal solutions, MOBSA, multi-objective PSO (MOPSO) (Bo and Yi-Jia 2005) and NSGA-II (Ah King et al. 2005) are selected to compare with MOLBSA. Table 12 provides the parameter settings (Xu et al. 2018). The four algorithms adopt the same constraint handling strategy, which has been shown in Section 6.1.

7.2.1 Comparison of spacing metric value

To evaluate the uniformity of the Pareto front obtained, a spacing metric (SP) (Schott 1995) is used to measure the uniformity of Pareto front. The formula of SP is as Eq. (23).

$$SP = \sqrt{\frac{1}{|Ar|-1}\sum_{t=1}^{|Ar|}(\bar{d}-d_t)^2}, \quad d_t = \min_{Ar_k \in Ar \wedge Ar_k \neq Ar_t} \sum_{m=1}^{M} |f_m(Ar_t) - f_m(Ar_k)| \tag{23}$$

where d_t indicates the Euclidean distance of two consecutive solutions of Pareto optimal set. $\bar{d}$ is the average of all d_t. If the SP value is very small, it indicates that the distribution of Pareto front is uniform. The SP value is zero, which represents that all solutions of the Pareto front are equidistant.

Tables 13 and 14 show the statistical results of the SP values of the four algorithms on two systems for Case2, respectively. The statistics are the best, median, worst, average and standard deviation (Std) of SP values in 30 independent experiments. From Tables 13 and 14, the statistical results of five statistics obtained by MOLBSA are smaller than those of the other three algorithms. That is to say, the Pareto front obtained by MOLBSA is more uniform than those of the other algorithms on two system for case2. Moreover, MOLBSA also demonstrates the best robustness in terms of uniformity.

Table 10 Best solutions for f_c in 6-unit system—Case2

	u_1	u_2	u_3	u_4	u_5	u_6	f_c($/h)	f_e(ton/h)	FEs
MOLBSA	0.125512	0.288003	0.575201	0.995006	0.519317	0.356745	**606.0081**	0.220542	**10,000**
MOBSA	0.111114	0.297220	0.601200	0.994180	0.505513	0.350254	606.0568	0.221115	**10,000**
NSGA (Abido 2003a)	0.116800	0.316500	0.544100	0.944700	0.549800	0.396400	608.2450	0.216640	100,000
NPGA (Abido 2003b)	0.124500	0.279200	0.628400	1.026400	0.469300	0.399300	608.1470	0.223640	100,000
SPEA (Abido 2003c)	0.108600	0.305600	0.581800	0.984600	0.528800	0.358400	607.8070	0.220150	100,000
NSGA-II (Ah King et al. 2005)	0.118200	0.314800	0.591000	0.971000	0.517200	0.354800	607.8010	0.218910	**10,000**
FCPSO (Agrawal et al. 2008)	0.113000	0.314500	0.582600	0.986000	0.526400	0.345000	607.7862	0.220100	20,000

Table 11 Best solutions for f_e in 6-unit system—Case2

	u_1	u_2	u_3	u_4	u_5	u_6	f_c($/h)	f_e(ton/h)	FEs
MOLBSA	0.403331	0.465857	0.555970	0.396765	0.538258	0.508468	645.015154	0.194196	**10,000**
MOBSA	0.413816	0.475663	0.560749	0.378567	0.526089	0.514714	647.822343	0.194224	**10,000**
NSGA (Abido 2003a)	0.411300	0.459100	0.511700	0.372400	0.581000	0.530400	647.251000	0.194320	100,000
NPGA (Abido 2003b)	0.392300	0.470000	0.556500	0.369500	0.559900	0.516300	645.984000	0.194240	100,000
SPEA (Abido 2003c)	0.404300	0.452500	0.552500	0.407900	0.546800	0.500500	642.603000	0.194220	100,000
NSGA-II (Ah King et al. 2005)	0.414100	0.460200	0.542900	0.401100	0.542200	0.504500	644.133000	**0.194190**	**10,000**
FCPSO (Agrawal et al. 2008)	0.406300	0.428600	0.551000	0.408400	0.543200	0.497400	642.896400	0.194200	20,000

Table 12 Parameter settings for 4 algorithms

	N	Na	FEs	Other prameters
MOLBSA	50	50	10,000	$mixrate = 1$
MOBSA	50	50	10,000	$mixrate = 1$
MOPSO	50	50	10,000	Inertia weight $w = 0.7$,learning coefficient $c1 = 1.4$, $c2 = 1.4$, grids per dimension number $G = 7$
NSGA-II	50	50	10,000	Mutation percentage $pr = 0.7$, crossover percentage $pc = 0.7$,

Table 13 Statistical results of the SP in 6-unit system—Case2

	Best	Median	Worst	Average	Std.
MOLBSA	**0.003408**	**0.00481**	**0.009428**	**0.005008**	**0.001223**
MOBSA	0.010203	0.016751	0.050250	0.019943	0.009611
MOPSO	0.015180	0.018031	0.024172	0.018250	0.001839
NSGA-II	0.016372	0.020987	0.024364	0.020640	0.002010

Table 14 Statistical results of the SP in 10-unit system—Case2

	Best	Median	Worst	Average	Std
MOLBSA	**0.003775**	**0.005917**	**0.008666**	**0.005825**	**0.001259**
MOBSA	0.009281	0.018864	0.036722	0.019143	0.006577
MOPSO	0.013331	0.018276	0.023795	0.018079	0.002462
NSGA-II	0.018399	0.021314	0.025325	0.021402	0.001657

To visually compare the distribution of the Pareto front obtained, Figs. 7 and 8 illustrate the Pareto fronts obtained by MOLBSA, MOBSA, MOPSO and NSGA-II. From these two figures, it can be inferred that the distribution of MOLBSA is more uniform than those of the other three algorithms on two systems for Case2. Combined with the information of graphs and tables, it can be concluded that the Pareto front obtained by MOLBSA is more uniform than those of the other three algorithms.

7.2.2 Comparison of hypervolume value

Hypervolume (HV) (Wu et al. 2010) is a hybrid metric proposed by Zitler and Thiele, which can measure the convergence and diversity of Pareto front. HV denotes the volume covered by Pareto front in the target domain, Eq. (24) is the specific calculation formula of HV.

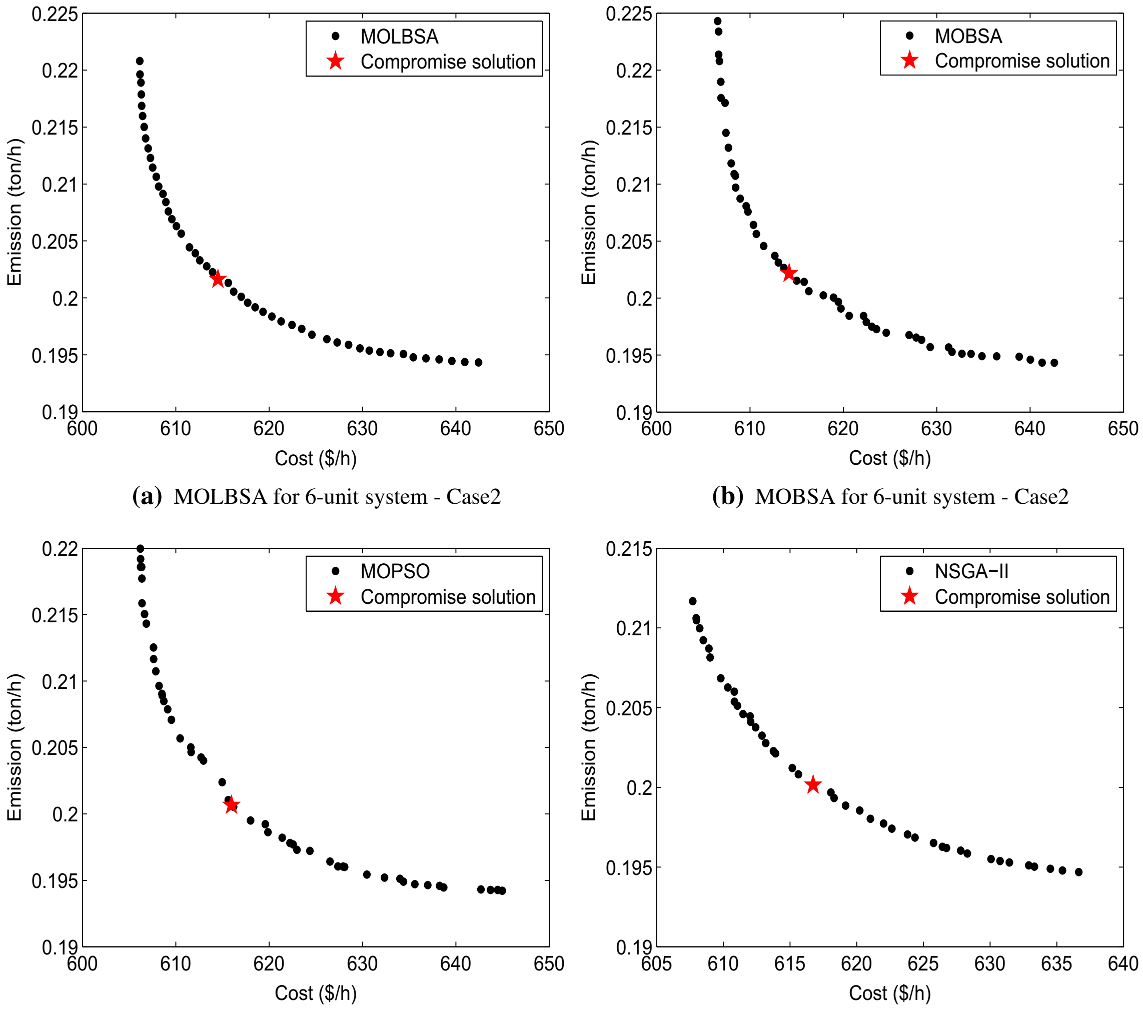
$$HV = \sum_{t=1}^{|Ar|} v_t \tag{24}$$

where v_t is the volume of a hypercube formed by a given fixed reference point w_r and the solution Ar_t. HV can be obtained from the sum of v_t, and varies with w_r taking different points. The larger the HV value, the wider the coverage of obtained Pareto front is. The same reference point w_r is adopted to calculate HV in the four algorithms.

The statistical results of the HV values for the four different algorithms are compared in Tables 15 and 16. From Table 15, the five statistics of the proposed MOLBSA reach the maximum HV values. This implies that the Pareto front obtained by MOLBSA has better the diversity and convergence than those of MOBSA, MOPSO and NSGA-II in 6-unit system. The results in 10-unit system are slightly different from those in 6-unit system. Table 16 shows that the results of MOLBSA are not the best at the four statistics (Best, Worst, Median and Average). However, in terms of standard deviation (Std), the MOLBSA is the best. This shows that the diversity of MOLBSA is no longer an advantage with the increase of variables, but MOLBSA has the best robustness in the HV.

7.2.3 Comparison of c-metric value

C-metric (C) (Zitzler et al. 2000) is adopted to evaluate the quality of the obtained Pareto front for optimization problems



(a) MOLBSA for 6-unit system - Case2

(b) MOBSA for 6-unit system - Case2

(c) MOPSO for 6-unit system - Case2

(d) NSGA-II for 6-unit system - Case2

Fig. 7 Pareto fronts and compromise solutions obtained by 4 algorithms in 6-unit system—Case2

with unknown true Pareto front in advance, and to represent the dominance relationship between two Pareto optimal sets. C formula can be expressed as

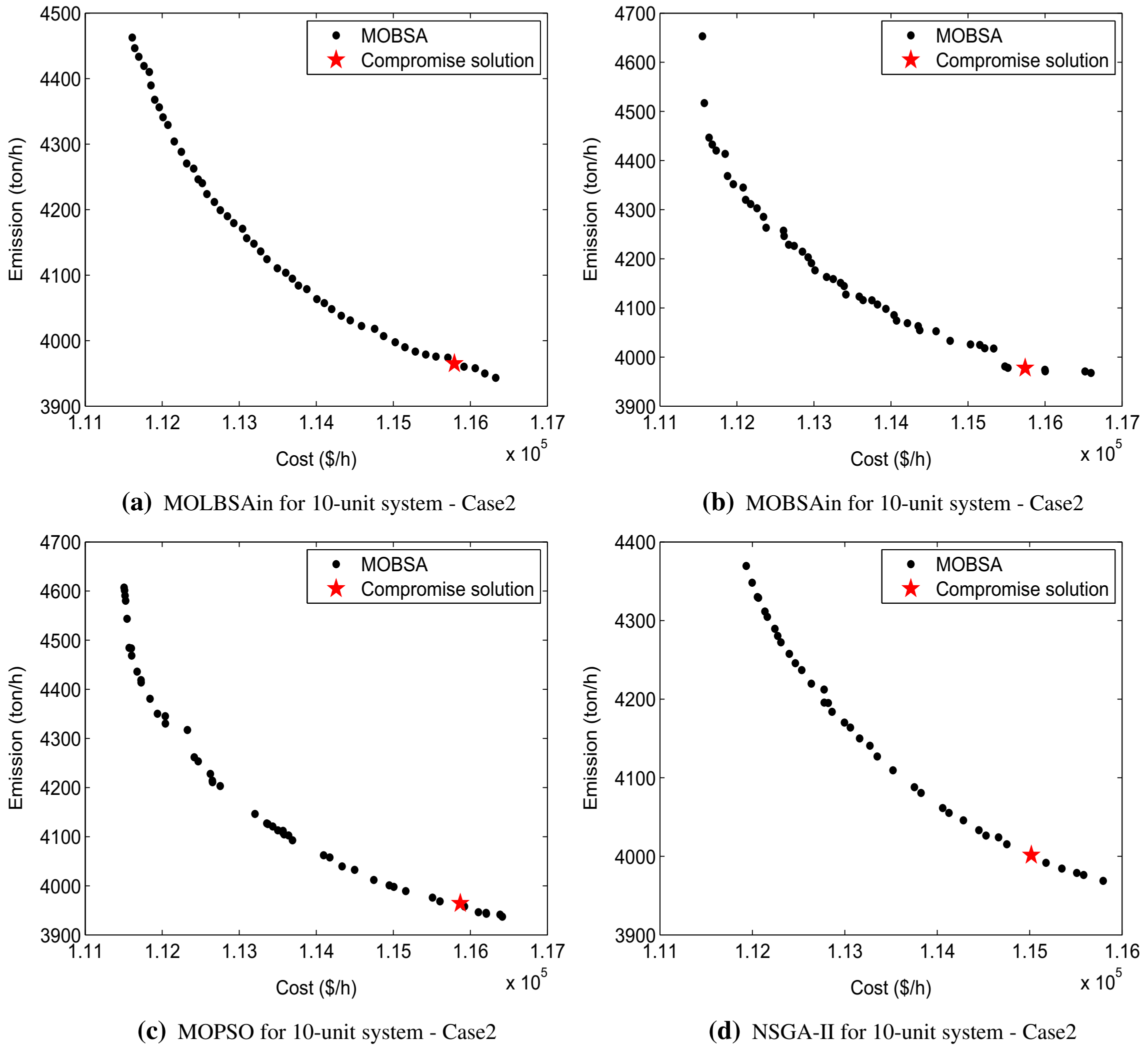
$$C(Ar1, Ar2) = \frac{|\{x_2 \in Ar2, \exists x_1 \in Ar1 : x_1 \prec x_2\}|}{|Ar2|} \qquad (25)$$

where $Ar1$ and $Ar2$ are two Pareto optimal sets obtained by two different algorithms, respectively. $C(Ar1, Ar2) = 1$ represents that $Ar2$ are dominated by $Ar1$. That is to say, $Ar1$ covers $Ar2$. $C(Ar1, Ar2) = 0$ means that none solution in $Ar2$ is dominated by any solution in $Ar1$.

Tables 17 and 18 show the comparison results of the C values calculated from the best Pareto optimal sets of different algorithms. From Table 17, the solutions of MOBSA, MOPSO and NSGA-II dominate 2%, 2% and 6% solutions of MOLBSA, respectively. However, the solutions of MOLBSA dominate 72%, 68% and 24% of the solutions of MOBSA, MOPSO and NSGA-II separately. As shown in Table 18, the solutions of MOBSA, MOPSO and NSGA-II dominate 0%, 24% and 4% solutions of MOLBSA, respectively, and the solutions of MOLBSA dominate 72%, 18% and 48% of the solutions of MOBSA, MOPSO and NSGA-II, respectively. All these data clearly prove the fact that the Pareto front obtained by MOLBSA is closer to the true Pareto optimal front than those obtained by MOBSA, MOPSO and NSGA-II in two system. That is to say, the Pareto front obtained by MOLBSA has better convergence than those of MOBSA, MOPSO and NSGA-II.

8 Conclusions

In this paper, a multi-objective learning backtracking search algorithm (MOLBSA) is successfully presented for solving the bi-objective EED problem with constrains. To enhance the quality of the Pareto optimal solutions obtained by



(a) MOLBSAin for 10-unit system - Case2

(b) MOBSAin for 10-unit system - Case2

(c) MOPSO for 10-unit system - Case2

(d) NSGA-II for 10-unit system - Case2

Fig. 8 Pareto fronts and compromise solutions obtained by the four algorithms in 10-unit system—Case2

Table 15 Statistical results of the HV in 6-unit system—Case2

	Best	Median	Worst	Average	Std
MOLBSA	**1.173061**	**1.170642**	**1.165222**	**1.170423**	**0.001671**
MOBSA	1.159150	1.155928	1.149041	1.155336	0.002623
MOPSO	1.165447	1.161064	1.152740	1.160788	0.002821
NSGA-II	1.163402	1.150145	1.101075	1.146295	0.013986

Table 16 Statistical results of the HV in 10-unit system—Case2

	Best	Median	Worst	Average	Std
MOLBSA	2255222.356006	2217643.293265	2162564.382445	2217627.894755	**22524.012869**
MOBSA	**2871278.841368**	2183943.621607	2156167.509809	**2270772.824151**	182308.755271
MOPSO	2486496.666509	**2226955.778590**	**2209963.182218**	2254816.020503	64768.6043301
NSGA-II	2186724.121689	2071084.782599	1830337.699687	2042967.690990	98767.582055

Table 17 Statistical results of the C in 6-unit system—Case2

	MOLBSA	MOBSA	MOPSO	NSGA-II
C(MOLBSA,*)	–	**0.72**	**0.68**	**0.24**
C(MOBSA,*)	0.02	–	0.26	0.02
C(MOPSO,*)	0.02	0.4	–	0.08
C(NSGA-II,*)	0.06	0.52	0.36	–

Table 18 Statistical results of the C in 10-unit system—Case2

	MOLBSA	MOBSA	MOPSO	NSGA-II
C(MOLBSA,*)	–	**0.72**	**0.18**	**0.48**
C(MOBSA,*)	0	–	0.04	0.06
C(MOPSO,*)	0.24	0.48	–	0.38
C(NSGA-II,*)	0.04	0.38	0.02	–

MOLBSA, two learning strategies are proposed. The first is a leader-choosing strategy, which accords to maximum distance between solutions to select the sparse Pareto optimal solution as the leader. The second is a leader-guiding strategy, which leads individuals to move toward the direction of sparse solution in Pareto optimal set. Moreover, a constraint handling technique is used to settle the power balance equality constraint in the EED problem. This paper studies the effectiveness of MOLBSA by testing 6-unit system and 10-unit system. Numerical results of compromise solutions and extreme solutions using MOLBSA and other seven well-known approaches disclosure that MOLBSA has can obtain a satisfactory compromise solution and highly diverse Pareto optimal set. Compared with other three established methods for three metrics, namely, spacing metric (SP), hypervolume (HV) and c-metric (C), the Pareto front obtained by MOLBSA in EED problem shows superior uniformity. However, as the number of decision variables increases, the diversity of Pareto front obtained by MOLBSA is no longer advantageous. This is because the proposed algorithm puts more efforts on improving the uniformity.

Acknowledgements This work was supported in part by Hubei Provincial Department of Education Outstanding Youth Scientific Innovation Team Support Foundation (T201410), National Natural Science Foundation of China (No. 61370092).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Abdolrasol MGM, Hannan MA, Mohamed A, Amiruldin UAU, Abidin IZ, Uddin M (2018) An Optimal Scheduling Controller for Virtual Power Plant and Microgrid Integration Using the Binary Backtracking Search Algorithm. IEEE Trans Ind Appl 54(3):2834–2844. https://doi.org/10.1109/TIA.2018.2797121

Abido MA (2003) A novel multiobjective evolutionary algorithm for environmental/economic power dispatch. Electr Power Syst Res 65(1):71–81. https://doi.org/10.1016/S0378-7796(02)00221-3

Abido MA (2003) A niched Pareto genetic algorithm for multiobjective environmental/economic dispatch. Int J Electr Power Energy Syst 25(2):97–105. https://doi.org/10.1016/S0142-0615(02)00027-3

Abido MA (2003) Environmental/economic power dispatch using multiobjective evolutionary algorithms. IEEE Trans Power Syst 18(4):1529–1537. https://doi.org/10.1109/TPWRS.2003.818693

Agrawal S, Panigrahi BK, Tiwari MK (2008) Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch. IEEE Trans Evol Comput 12(5):529–541. https://doi.org/10.1109/TEVC.2007.913121

Ah King RTF, Rughooputh HCS, Deb K (2005) Evolutionary multi-objective environmental/economic dispatch: Stochastic versus deterministic approaches. In: International Conference on Evolutionary Multi-Criterion Optimization, pp 677–691. https://doi.org/10.1007/978-3-540-31880-4_47

Ali JA, Hannan MA, Mohamed A (2015) Backtracking search algorithm approach to improve indirect field-oriented control for induction motor drive. In: IEEE 3rd International Conference on Smart Instrumentation, Measurement and Applications. IEEE. https://doi.org/10.1109/ICSIMA.2015.7559016

Ali AF (2015) A memetic backtracking search optimization algorithm for economic dispatch problem. Egypt Comput Sci J 39(2):56–71

Ali JA, Hannan MA, Mohamed A, Maher GM (2016) Fuzzy logic speed controller optimization approach for induction motor drive using backtracking search algorithm. Measurement 78:49–62. https://doi.org/10.1016/j.measurement.2015.09.038

Atasever UH, Civicioglu P, Besdok E, Ozkan C (2014) A new unsupervised change detection approach based on dwt image fusion and backtracking search optimization algorithm for optical remote sensing data. Int Arch Photogramm Remote Sens S 40(7):15–18. https://doi.org/10.5194/isprsarchives-XL-7-15-2014

Ayan K, Kılıç U (2016) Optimal power flow of two-terminal HVDC systems using backtracking search algorithm. Int J Electr Power Energy Syst 78:326–335. https://doi.org/10.1016/j.ijepes.2015.11.071

Basu M (2011) Economic environmental dispatch using multi-objective differential evolution. Appl Soft Comput 11(2):2845–2853. https://doi.org/10.1016/j.asoc.2010.11.014

Bhattacharjee K, Bhattacharya A, nee Dey SH (2014) Solution of economic emission load dispatch problems of power systems by real coded chemical reaction algorithm. Int J Electr Power Energy Syst 59:176–187. https://doi.org/10.1016/j.ijepes.2014.02.006

Bhattacharjee K, Bhattacharya A, Dey SHN (2015) Backtracking search optimization based economic environmental power dispatch problems. Int J Electr Power Energy Syst 73:830–842. https://doi.org/10.1016/j.ijepes.2015.06.018

Bo Z, Yi-Jia C (2005) Multiple objective particle swarm optimization technique for economic load dispatch. J Zhejiang Univ-SCI A 6(5):420–427. https://doi.org/10.1007/BF02839410

Cai J, Ma X, Li Q, Peng H, Li L (2009) A multi-objective chaotic particle swarm optimization for environmental/economic dispatch. Energy Convers Manag 50(5):1318–1325. https://doi.org/10.1016/j.enconman.2009.01.013

Chaib AE, Bouchekara HREH, Mehasni R, Abido MA (2016) Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm. Int J Electr Power Energy Syst 81:64–77. https://doi.org/10.1016/j.ijepes.2016.02.004

Civicioglu P (2013) Backtracking search optimization algorithm for numerical optimization problems. Appl Math Comput 219(15):8121–8144. https://doi.org/10.1016/j.amc.2013.02.017

Das DB, Patvardhan C (1998) New multi-objective stochastic search technique for economic load dispatch. IEE Proc Gener Trans Distrib 145(6):747–752. https://doi.org/10.1049/ip-gtd:19982367

De Souza RR, Miguel LFF, Lopez RH, Fadel Miguel LF, Torii AJ (2016) A procedure for the size, shape and topology optimization of transmission line tower structures. Eng Struct 111:162–184. https://doi.org/10.1016/j.engstruct.2015.12.005

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017

Dubey HM, Pandit M, Tyagi N, Panigrahi BK (2016) Wind integrated multi area economic dispatch using backtracking search algorithm. In: 2016 IEEE 6th International Conference on Power Systems. IEEE. https://doi.org/10.1109/ICPES.2016.7584188

El-Fergany A (2015) Optimal allocation of multi-type distributed generators using backtracking search optimization algorithm. Int J Electr Power Energy Syst 64:1197–1205. https://doi.org/10.1016/j.ijepes.2014.09.020

El-Fergany A (2015) Study impact of various load models on DG placement and sizing using backtracking search algorithm. Appl Soft Comput 30:803–811. https://doi.org/10.1016/j.asoc.2015.02.028

El-Fergany A (2016) Multi-objective allocation of multi-type distributed generators along distribution networks using backtracking search algorithm and fuzzy expert rules. Electr Power Compon Syst 44(3):252–267. https://doi.org/10.1080/15325008.2015.1102989

Eskandari M, Toygar Ö (2015) Selection of optimized features and weights on face-iris fusion using distance images. Comput Vis Image Underst 137(C):63–75. https://doi.org/10.1016/j.cviu.2015.02.011

Farag A, Al-Baiyat S, Cheng TC (1995) Economic load dispatch multiobjective optimization procedures using linear programming techniques. IEEE Trans Power Syst 10(2):731–738. https://doi.org/10.1109/59.387910

Goldberg DE (1989) Genetic algorithm in search optimization and machine learning. Addison Wesley xiii(7):2104–2116. https://doi.org/10.1111/j.1365-2486.2009.02080.x

Guney K, Durmus A (2015) Pattern nulling of linear antenna arrays using backtracking search optimization algorithm. Int J Antennas Propag. https://doi.org/10.1155/2015/713080

Guney K, Durmus A (2016) Elliptical antenna array synthesis using backtracking search optimisation algorithm. Def Sci J 66(3):272–277

Guney K, Durmus A, Basbug S (2014) Backtracking search optimization algorithm for synthesis of concentric circular antenna arrays. Int J Antennas Propag. https://doi.org/10.1155/2014/250841

Gupta V, Donepudi SR, Subrahmanyam N (2015) Optimal placement of distributed generators in distribution system using backtracking search optimization for various load models. In: 2015 International Conference on Recent Developments in Control, Automation and Power Engineering. IEEE. https://doi.org/10.1109/rdcape.2015.7281423

Hannan MA, Lipu MSH, Hussain A, Saad MH, Ayob A (2018) Neural network approach for estimating state of charge of lithium-ion battery using backtracking search algorithm. IEEE Access 6:10069–10079. https://doi.org/10.1109/ACCESS.2018.2797976

Ishak R, Mohamed A, Abdalla AN, Che Wanik MZ (2014) Optimal DG Placement and Sizing for Voltage Stability Improvement Using Backtracking Search Algorithm. In: IEEE International Conference on Power and Energy. IEEE. https://doi.org/10.1109/PECON.2014.7062432

Islam NN, Hannan MA, Shareef H, Mohamed A (2017) An application of backtracking search algorithm in designing power system stabilizers for large multi-machine system. Neurocomputing 237:175–184. https://doi.org/10.1016/j.neucom.2016.10.022

Jubril AM, Olaniyan OA, Komolafe OA, Ogunbona PO (2014) Economic-emission dispatch problem: A semi-definite programming approach. Appl Energy 134:446–455. https://doi.org/10.1016/j.apenergy.2014.08.024

Khamis A, Shareef H, Mohamed A, Dong ZY (2015) A load shedding scheme for dg integrated islanded power system utilizing backtracking search algorithm. Ain Shams Eng J. https://doi.org/10.1016/j.asej.2015.10.001

Khamis A, Shareef H, Mohamed A (2015) Islanding detection and load shedding scheme for radial distribution systems integrated with dispersed generations. IET Gener Transm Distrib 9(15):2261–2275. https://doi.org/10.1049/iet-gtd.2015.0263

Khamis A, Shareef H, Mohamed A, Dong ZY (2015) A load shedding scheme for DG integrated islanded power system utilizing backtracking search algorithm. Ain Shams Eng J 9:161–172. https://doi.org/10.1016/j.asej.2015.10.001

Khamis A, Tai OB (2017) Optimal load curtailment for DG integrated islanded power system utilizing backtracking search algorithm. In: Tencon IEEE Region 10 Conference. IEEE, pp 1708–1714. https://doi.org/10.1109/TENCON.2017.8228134

Kılıç U (2015) Backtracking search algorithm-based optimal power flow with valve point effect and prohibited zones. Electr Eng 97(2):101–110. https://doi.org/10.1007/s00202-014-0315-0

Luo CY, Chen MY, Zhang CY (2010) Improved NSGA-II algorithm with circular crowded sorting. Control Decis 25(2):227–231 (in chinese)

Modiri-Delshad M, Rahim NA (2014) Solving non-convex economic dispatch problem via backtracking search algorithm. Energy 77:372–381. https://doi.org/10.1016/j.energy.2014.09.009

Modiri-Delshad M, Rahim NA (2016) Multi-objective backtracking search algorithm for economic emission dispatch problem. Appl Soft Comput 40:479–494. https://doi.org/10.1016/j.asoc.2015.11.020

Modiri-Delshad M, Kaboli SHA, Taslimi-Renani E, Abd Rahim N (2016) Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options. Energy 116:637–649. https://doi.org/10.1016/j.energy.2016.09.140

Modiri-Delshad M, Barati M, Abd Rahim N (2016) Economic power dispatch in microgrids through backtracking search algorithm. In: 4th IET Clean Energy and Technology Conference. https://doi.org/10.1049/cp.2016.1326

Mohd Zain MZB, Kanesan J, Kendall G, Chuah JH (2018) Optimization of fed-batch fermentation processes using the backtracking search algorithm. Expert Syst Appl 91:286–297. https://doi.org/10.1016/j.eswa.2017.07.034

Najibi F, Niknam T, Kavousi-Fard A (2016) Optimal stochastic management of renewable MG (micro-grids) considering electro-thermal model of PV (photovoltaic). Energy 97:444–459. https://doi.org/10.1016/j.energy.2015.12.122

Niamul Islam N, Hannan MA, Mohamed A, Shareef H (2016) Improved power system stability using backtracking search algorithm for coordination design of PSS and TCSC damping controller. PloS one 11(1):e0146277. https://doi.org/10.1371/journal.pone.0146277

Pal A, Dasgupta K, Banerjee S, Chanda CK (2016) An analysis of Economic Load Dispatch with Ramp-rate limit constraints using BSA. In: 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science. https://doi.org/10.1109/sceecs.2016.7509271

Pare S, Bhandari AK, Kumar A, Bajaj V (2018) Backtracking search algorithm for color image multilevel thresholding. Signal Image Video Process 12(2):385–392. https://doi.org/10.1007/s11760-017-1170-z

Qu BY, Zhu YS, Jiao YC, Wu MY, Suganthan PN, Liang JJ (2017) A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2017.06.002

Saadat H (1979) Power system analysis. Wiley, New Jersey. https://doi.org/10.2991/978-94-6239-064-5_6

Sakawa M, Yano H, Yumine T (1987) An interactive fuzzy satisficing method for multiobjective linear-programming problems and its application. IEEE Trans Syst Man Cybern 17(4):654–661. https://doi.org/10.1109/TSMC.1987.289356

Schott JR (1995) Fault tolerant design using single and multicriteria genetic algorithm optimization. M.S. Thesis Cambridge MA

Shafiullah M, Abido MA, Coelho LS (2015) Design of robust PSS in multimachine power systems using backtracking search algorithm. In: 18th International Conference on Intelligent System Application to Power Systems. IEEE, pp 1–6. https://doi.org/10.1109/ISAP.2015.7325528

Shafiullah M, Rana MJ, Coelho LS, Abido MA (2017) Power system stability enhancement by designing optimal PSS employing backtracking search algorithm In: 6th International Conference on Clean Electrical Power. IEEE, pp 712–719. https://doi.org/10.1109/ICCEP.2017.8004769

Shahriar MS, Shafiullah M, Asif MA, Hasan MM, Rafiuzzaman M (2015) Design of multi-objective UPFC employing backtracking search algorithm for enhancement of power system stability. In: 18th International Conference on Computer and Information Technology. IEEE, pp 323–328. https://doi.org/10.1109/ICCITechn.2015.7488090

Sivasubramani S, Swarup KS (2011) Environmental/economic dispatch using multi-objective harmony search algorithm. Electr Power Syst Res 81(9):1778–1785. https://doi.org/10.1016/j.epsr.2011.04.007

Srinivas N, Deb K (1994) Muiltiobjective optimization using nondominated sorting in genetic algorithms. Evol Comput 2(3):221–248. https://doi.org/10.1162/evco.1994.2.3.221

Talaq JH, El-Hawary F, El-Hawary ME (1994) A summary of environmental/economic dispatch algorithms. IEEE Trans Power Syst 9(3):1508–1516. https://doi.org/10.1109/59.336110

Vitayasak S, Pongcharoen P, Hicks C (2016) A tool for solving stochastic dynamic facility layout problems with stochastic demand using either a genetic algorithm or modified backtracking search algorithm. Int J Prod Econ 190(C):146–157. https://doi.org/10.1016/j.ijpe.2016.03.019

Wang L, Singh C (2007) Environmental/economic power dispatch using a fuzzified multi-objective particle swarm optimization algorithm. Electr Power Syst Res 77(12):1654–1664. https://doi.org/10.1016/j.epsr.2006.11.012

Wang L, Singh C (2008) Balancing risk and cost in fuzzy economic dispatch including wind power penetration based on particle swarm optimization. Electr Power Syst Res 78(8):1361–1368. https://doi.org/10.1016/j.epsr.2007.12.005

Wang HL, Hu ZB, Sun YQ, Su QH, Xia XW (2019) A novel modified BSA inspired by species evolution rule and simulated annealing principle for constrained engineering optimization problems. Neural Comput Appl 31(8):4157–4184. https://doi.org/10.1007/s00521-017-3329-5

Wu LH, Wang YN, Yuan XF, Zhou SW (2010) Environmental/economic power dispatch problem using multi-objective differential evolution algorithm. Electr Power Syst Res 80(9):1171–1181. https://doi.org/10.1016/j.epsr.2010.03.010

Xu XL, Hu ZB, Su QH, Xiong ZG (2018) Multiobjective collective decision optimization algorithm for economic emission dispatch problem. Complexity 2018:1–20. https://doi.org/10.1155/2018/1027193

Zhang Y, Gong DW, Ding Z (2012) A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. Inf Sci 192:213–227. https://doi.org/10.1016/j.ins.2011.06.004

Zitzler E (1999) Evolutionary algorithms for multiobjective optimization: methods and applications. Shaker, Ithaca

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans Evol Comput 3(4):257–271. https://doi.org/10.1109/4235.797969

Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195. https://doi.org/10.1162/106365600568202