Mutation strategies toward Pareto front for multi-objective differential evolution algorithm

Warisa Wisittipanich* and Voratas Kachitvichyanukul

Industrial and Manufacturing Engineering, School of Engineering and Technology, Asian Institute of Technology, P.O. Box 4, Klong Luang, Pathumthani 12120, Thailand
E-mail: warisa_o@hotmail.com
E-mail: voratas@ait.ac.th
E-mail: Vvoratas@ait.ac.th
*Corresponding author

Abstract: This paper presents a multi-objective differential evolution algorithm, called MODE, to search for a set of non-dominated solutions on the Pareto front. During the iterative search process, the non-dominated solutions found are stored as the ‘Elite group’ of solutions. The study focuses on utilising the solutions in the Elite group to guide the movement of the search. Several potential mutation strategies in MODE framework are proposed as the movement guidance in order to obtain the high-quality front. Each mutation strategy possesses distinct search behaviour which directs a vector in the DE population in different ways with the purpose of reaching the Pareto optimal front. The performance of the proposed algorithm is evaluated on a set of well-known benchmark problems and compared with results from other existing approaches. The experimental results demonstrate that the proposed MODE algorithm is a highly competitive approach for solving multi-objective optimisation problems.

Keywords: mutation strategies; multi-objective problems; Pareto front; evolutionary algorithms; differential evolution.


Biographical notes: Warisa Wisittipanich received her Doctoral in Industrial and Manufacturing Engineering from the Asian Institute of Technology, Thailand and her Master in Systems Engineering from George Mason University, Virginia, USA. Her research interests include scheduling and evolutionary method for optimisation.

Voratas Kachitvichyanukul is a Professor of Industrial and Manufacturing Engineering, Asian Institute of Technology, Thailand. His teaching and research interests include modelling and simulation of large scale industrial systems, evolutionary methods for optimisation, supply chain management and applied operations research.
1 Introduction

Multi-objective (MO) optimisation problem has received increasing attention from both practitioners and researchers not only because most real-world problems do contain multiple conflicting objectives, but also because there are still many challenges in this research area. In fact, there is no single solution in MO but rather a set of solutions and the decision about the ‘best’ solution among these solutions mainly depends on the decision maker. Generally, the methods for solving MO problems can be classified into two types as either non-Pareto or Pareto-based approach. Non-Pareto methods are mostly based on an aggregated weighted approach where multiple objectives functions are given weights a priori and combined into a single objective function. However, these methods require a set of weights based on the prior preference of the decision maker, which may be subjective due to their limited knowledge about the range of each objective function. In addition, the fact that only one single solution is found provides the decision maker with very little information about the potential trade-offs. In order to be more objective, the approaches based on a single aggregated objective function needs to be run several times to find sets of solutions based on varying weights, and as a result, these approaches are highly time consuming. On the contrary, weight-free methods such as Pareto-based approaches become more preferable by researchers to solve MO problems because it allows decision makers to simultaneously find the trade-offs or non-dominated solutions on the Pareto front in a single run without prejudice.

Over the past two decades, multi-objective evolutionary algorithms (MOEAs) have received growing interest as solution techniques to search for the Pareto front because of their efficiency to search for high quality solutions in a reasonable time. Differential evolution (DE) algorithm is one of the novel EAs with high efficiency in search ability and computing time. However, only few research works have attempted to apply DE to find solutions for MO problems. This paper presents a general multi-objective differential evolution (MODE) algorithm for finding a set of non-dominated solutions on the Pareto front. The algorithm focuses on developing several potential mutation strategies dictating the movement behaviour of DE population to search for high-quality Pareto front.

The remainder of this paper is organised as follows. The literature review of MOEAs to search for the Pareto front is provided in Section 2. The fundamental background related to MO optimisation problems and the classic DE algorithm is provided in Section 3. Section 4 presents the proposed MODE algorithm and describes the concepts of mutation strategies with their implementations. The performances of the proposed MODE algorithms on test problems are discussed in Section 5. Finally, conclusion and recommendations for further research are given in Section 6.

2 Literature review

As previously mentioned, many research works on MO optimisation with respect to Pareto-based approaches have been devoted to the subject of MOEAs during the past two decades. In 1990s, multi-objective genetic algorithms (MOGAs) adopting Pareto dominance concept for fitness assignment or selection were developed and taken as benchmark approaches. Fonseca and Fleming (1993) introduced MOGA which determines the fitness value based on the number of solutions that dominate it. In contrast, Srinivas and Deb (1994) proposed non-dominated sorting genetic algorithm
Mutation strategies toward Pareto front for MODE algorithm

MOGA and NSGA applied the concept of fitness sharing which intends to degrade the fitness of an individual according to its density in the neighbourhood area. Therefore, a good distribution of solutions can be maintained. Zitzler and Thiele (1999) proposed strength Pareto evolutionary algorithm (SPEA) to emphasise the importance of elitism which emerged to have a significant influence on the design of subsequent developed algorithms. Knowles and Corne (1999) introduced a simple MOEA called the Pareto-archived evolution strategy (PAES) using local search (LS) of a population of one and reference archive of previously found solutions to determine Pareto solutions. Corne et al. (2000) introduced Pareto envelope-based selection algorithm (PESA) to integrate selection and diversity maintenance through a hyper-grid-based scheme. Deb et al. (2002) realised the drawbacks of NSGA which are the high computational complexity and the use of the predetermined sharing parameter and proposed NSGA-II to remedy the problems by adopting an elitism structure and a measurement of the crowdedness. This algorithm had demonstrated outstanding results over other MOEAs such as PAES and SPEA on many test problems. Chang and Chen (2009) proposed a new algorithm, called the subpopulation genetic algorithm II, with information sharing across all subpopulation. When one subpopulation obtains better non-dominated solutions, other subpopulations will directly apply them in their search space. As a result, all individuals in the same population will be guided toward the true Pareto front. Jin and Wong (2010) proposed an MOEA with a technique called an adaptive rectangle archiving (ARA) strategy to converge solutions to well-distributed Pareto solutions without prior knowledge about the Pareto front. ARA complements the existing archiving techniques which is useful to both researchers and practitioners.

In the past decade, particle swarm optimisation (PSO) has also captured interests from researchers to solve MO problems. Several PSO algorithms have been developed to search for the Pareto front. Coello and Lechuga (2002) proposed the idea to store non-dominated particles in an external archive which is then updated by using geographically-based system. In their algorithm, the search area of the objective-space is divided into hypercubes. Each hypercube is assigned a fitness based on the density of particles. Roulette-wheel selection is applied to choose the hypercube and a random particle from that hypercube is selected as the leader or global guidance for a particle in the swarm in each iteration. To improve this work, Fieldsend and Singh (2002) presented a tree data structure to maintain unconstrained archive. Raquel and Naval (2005) implemented the idea of crowding distance (CD) introduced in NSGA-II in their PSO algorithm as a criterion to select a particle leader from the swarm. In this method, when preparing to move, a particle will select its global leader from the top particles in the external archive which are sorted in decreasing order of CD. The same mechanism was also adopted by Li (2003) where two parameter-free niching techniques (niche count and CD) are used to promote solution diversity. Tripathi et al. (2007) applied the self-adaptive control parameter and used the global best particle selected from the non-dominated solutions with a roulette wheel selection in which the density values are used as fitness. In the work of Janson et al. (2008), the particles are clustered into groups. The global best particle of a particle is from its group and a weighted sum of the objectives is used to maintain its local best particle. Liu et al. (2008) introduced a tournament niche method to select the global best particle, and updated the local best particle by the Pareto dominance. Wang and Yang (2009) proposed a generalisation of Pareto dominance, called preference order, to identify the global best particle and rank all
the particles. Elhossini et al. (2010) presented the three hybrid EA-PSO algorithms where the PSO’s operator is inserted into an EA’s main loop.

Recently, Nguyen and Kachitvichyanukul (2010) introduced MOPSO algorithms which mainly focus on the utilisation of the particles in an external archive as the guidance for searching the non-dominated solutions. They proposed several efficient movement strategies of particles with different search behaviours which aim to obtain the high quality Pareto front. The experimental results showed that their algorithms are very competitive compared to other existing algorithms. Zhang et al. (2011) presented a multi-swarm cooperative PSO algorithm which consists of multiple slave swarms and one master swarm. Each slave swarm aims to optimise one objective function in order to find the non-dominated optima of that objective function while the master swarm intends to cover gaps among solutions found by slave swarms. Several improved techniques were also employed to enhance the algorithm performance, but the weakness of this algorithm is that its performance highly depends on a parameter setting. Zhao and Suganthan (2011) proposed a two-local-best (lbest)-based multi-objective particle swarm optimisation (2LB-MOPSO) technique by using two local bests instead of one personal best and one global best to lead each particle. The two local bests are selected to be close to each other in order to enhance the LS ability of the algorithm. The experimental results showed that 2LB-MOPSO outperform the canonical MO PSO in convergence speed and searching ability. Mousa et al. (2012) proposed a hybrid MOEA combining two evolutionary algorithms (EAs); GA and PSO. The algorithm is first initialised by a set of random particles, and the non-dominated solutions are determined during the evolution process. The LS scheme is also implemented to improve the solution quality by exploring the less-crowded area in the current archive to obtain more non-dominated solutions.

For the past decade, a novel EA, called DE, has gradually gained attention from many researchers due to its successful application to solve many optimisation problems. However, only few research works have attempted to apply DE to find solutions for MO problems. Abbass et al. (2001) presented a Pareto-frontier differential evolution (PDE) approach where only non-dominated solutions are allowed to participate in generating new solutions. However, if only few non-dominated solutions are found in the beginning, the chances of discovering new better solutions may be limited. As a result, the solutions may get trapped at local optima. Madavan (2002) introduced the combination of the newly generated population and the existing parent population which results in double population size. The non-dominated solutions are then selected from this combination to allow the global check among both parents and offspring solutions; however, the drawback of this approach is the requirement of additional computing time in the sorting procedure of the combined population. Alatas et al. (2008) proposed an MODE algorithm with mining numeric association rules. In their algorithm, an individual is treated as a real vector and a rounding operator is applied to repair a real component to an integer component. Gong and Cai (2009) presented an improved version of classic DE algorithm which combines several features of previous EAs in a unique manner. In this approach, population initialisation is based on orthogonal design, an archive is updated by the new Pareto-adaptive ε-dominance at each iteration, extreme points are stored to prevent the loss of extreme points in the final archive, and a random selection and an elitist selection are alternatively used to improve its performance. Tan et al. (2012) proposed the uniform design MODE algorithm based on decomposition (UMODE/D) for MO problems. In UMODE/D, the uniform design method is applied to generate the aggregation coefficient vectors so that the decomposed scalar optimisation subproblems are uniformly
distributed. Consequently, the algorithm could explore uniformly across the region of interest from the initial iteration. In addition, the simplified quadratic approximation with three best points is employed to improve the LS ability and the accuracy of the minimum scalar aggregation function value.

3 Background

3.1 Multi-objective optimisation

MO optimisation problem is characterised by the presence of multiple conflicting objectives. The general mathematical formulation for a minimisation problem with multiple objective functions is given as follow:

Minimise

\[ \bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), \ldots, f_K(\bar{x})] \]  

subject to:

\[ g_i(\bar{x}) \leq 0 \quad i = 1, 2, \ldots, m \]  

\[ h_i(\bar{x}) = 0 \quad i = 1, 2, \ldots, l \]  

where \( \bar{x} \) is the vector of decision variables, \( f_i(\bar{x}) \) is a function of \( \bar{x} \), \( K \) is the number of objective function to be minimised, \( g_i(\bar{x}) \) and \( h_i(\bar{x}) \) are the constraint functions of the problem.

As mentioned, this paper focuses on solution approaches based on Pareto optimality concept which aim to search for multiple non-dominated solutions simultaneously. The set of solutions obtained in the Pareto front with two objective functions to be minimised: \( f_1 \) and \( f_2 \), is illustrated in Figure 1.

Figure 1  \( \vec{x}_1 \prec \vec{x}_2 \) and non-dominated solutions in Pareto front
Given two decision variables $x_1$ and $x_2$, $x_1$ is considered to dominate $x_2$ (denote $x_1 \preceq x_2$) if and only if $f_i(x_1) \leq f_i(x_2)$ for all $i = 1, 2, \ldots, k$ and $\exists j = 1, 2, \ldots, k$ such that $f_j(x_1) < f_j(x_2)$. As shown in Figure 1, for the case that neither $x_1 \preceq x_2$ nor $x_2 \preceq x_1$, $x_1$ and $x_2$ are called non-dominated front or ‘trade-off’ solutions.

### 3.2 Differential evolution

DE, proposed by Storn and Price (1995) in 1995, is one of the latest EAs for global optimisation over continuous search space. DE has been widely applied and shown its strengths in many application areas due to its advantage of relatively few control variables but performing well in convergence. As a population-based search method, DE starts with randomly generate initial population of size $N$ of $D$-dimensional vectors. A solution in DE algorithm is represented by $D$-dimensional position of a vector. Each variable’s value in the dimensional space is represented as the real number. The key idea behind DE is a new mechanism for generating trial vectors by mutation and crossover operation. Then, the replacement or selection of an individual occurs only if the trial vector outperforms its corresponding vector. Once each individual in the current population is updated, the population continues to evolve through mutation, crossover, and selection operation until some stopping criteria are met.

Currently, several variants of DE have been proposed. Generally, the mutation is the main operation which makes each variant distinct from one another. The classic version of DE is the simplest and most popular scheme used in literatures. The mutation operator in the classic DE generates a mutant vector by adding a weighted difference between two randomly selected vectors to the third randomly selected vector. The procedures in the classic DE (Price et al., 2005) can be described as follows.

#### 3.2.1 Population initialisation

DE starts with initialising the population of size $N$ of $D$-dimensional vectors. The lower bound, $b_L$, and upper bound, $b_U$, for the value in each dimension $j$ ($j = 0, 1, \ldots, D - 1$) must be specified. At initialisation step ($g = 0$), the $j$th dimension value of the $i$th vector is randomly generated as follows:

$$x_{j,i,g} = u_j \cdot (b_{j,U} - b_{j,L}) + b_{j,L}$$

(4)

where $u_j$ is a uniform random number in the range $[0, 1]$.  

#### 3.2.2 Mutation

Once initialised, DE mutates and combines current vectors to produce mutant vectors. For each target vector $i$, $X_{i,g}$, at generation $g$, the mutant vector, $V_{i,g}$, is generated according to the following equation:

$$V_{i,g} = X_{i,g} + F \left( X_{r_2,g} - X_{r_1,g} \right)$$

(5)
Mutation strategies toward Pareto front for MODE algorithm

It is noted that $X_n, X_m, \text{ and } X_o$ are randomly selected vectors from the population. They are mutually exclusive and different from the $i^{th}$ target vector, $X_{i,g}$. The scale factor $F$ is a positive number which controls the scale of the difference vector between, $X_n, \text{ and } X_o$ added to the base vector, $X_1$.

### 3.2.3 Crossover

DE applies crossover operator on $X_{i,g}$ and $V_{i,g}$ to generate the trial vector $Z_{i,g}$. In the classic DE, the binomial crossover is employed and the trial vector is generated by the following equation.

$$z_{i,g} = \begin{cases} v_{i,j,g}, & \text{if } u_j \leq C_r \text{ or } j = j_u \\ x_{i,j,g}, & \text{otherwise} \end{cases}$$

(6)

where

- $u_j$: a uniformly random number between $[0, 1]$
- $j_u$: a random chosen index, $j_u \in \{0, 1, \ldots, D - 1\}$
- $C_r$: crossover probability in the range $[0, 1]$.

$C_r$ controls the probability of selecting the value in each dimension for a trial vector from a mutant vector.

### 3.2.4 Selection

The selection operation is performed on each target vector, $X_{i,g}$, and its corresponding trial vector, $Z_{i,g}$, to determine the survival vector for the next generation. For minimisation problem, the vector, $X_{i,g+1}$, is selected according to the greedy criteria.

$$X_{i,g+1} = \begin{cases} Z_{i,g}, & \text{if } f(Z_{i,g}) \leq f(X_{i,g}) \\ X_{i,g}, & \text{otherwise} \end{cases}$$

(7)

This classic DE is commonly denoted as $DE/rand/1/bin$, where $DE$ stands for DE, $rand$ is the type of base vector selected to be perturbed, $I$ is the number of difference vector for permutation, and $bin$ stands for binomial distribution of the number of inherited dimension values of mutant vectors. This scheme has proven to be effective in solving many multimodal optimisation problems due to its good exploration capability (Price et al., 2005; Chakraborty, 2008). Some successful applications of DE to combinatorial optimisation problems include job shop scheduling problems by Wisittipanich and Kachitvichyanukul (2012) and multi-mode resource constrained project scheduling problems by Nguyen and Kachitvichyanukul (2012).

To extend the classic DE for solving MO problems, it requires some modifications on the basic algorithm to allow the solutions on the Pareto front to be determined simultaneously in a single run. The procedures of the proposed MODE are described in the following section.
4 Proposed algorithm

The traditional DE algorithm was initially developed for optimising one single objective. The solution is immediately updated to every single move of a vector when the better solution is found. However, as mentioned earlier that when dealing with MO problems, there is no single solution but rather a set of trade-off solutions. Inspired by the work of MOPSO, proposed by Nguyen and Kachitvichyanukul (2010), this study adopts the same concept of particle movement strategies in MOPSO and emphasises on developing several mutation strategies as the guidance of DE population in order to find the high quality non-dominated solutions. The MODE algorithm for solving MO problems is proposed and its framework is shown in Figure 2.

Figure 2  MODE framework
4.1 MODE framework

Similarly to the Elitist structure in NSGA-II, the population experience is stored in an external archive, called Elite group, as a set of non-dominated solutions. In MODE, instead of applying the sorting procedure to every single move of a vector, the sorting is only performed on the set of newly generated trial vectors after all vectors completed one move to identify the group of new non-dominated solutions. The reason is to reduce computational time. This sorting procedure applies to the group of new solutions and current solutions in the external archive and store only non-dominated solutions into an archive for the Elite group. Then, Elite group screens its solutions to eliminate inferior solutions. As a result, the Elite group in the archive contains only the best non-dominated solutions found so far in the searching process of the MODE population.

Another critical decision in MO problems is how to select the candidates among the Elite group as guidance toward the Pareto frontier. This study adopts one of the most common techniques proposed in NSGA-II that the candidate guidance is selected from the non-dominated solutions located in the less crowded areas. To identify the crowdedness, NSGA-II measures the CD of each member in the population. The density of solutions surrounding a specific solution is determined by calculating the average distance of two points on either side of this point along each of the objective (see Deb et al., 2002 for more details).

It is important to note that, in MO cases, the existence of multiple candidates in the archive offers a large number of options on the movement of vectors, and the quality of the final solutions will be strongly influenced by the movement behaviour adopted by the population. This study proposed several mutation strategies in MODE framework as the movement guidance in order to obtain the high quality Pareto front. These mutation strategies are discussed in the next section.

4.2 Mutation strategies

As mentioned in previous section, the external archive called the Elite group is used to store the non-dominated solutions which can be used to guide vectors in the population. In this paper, five mutation strategies to utilise the information provided by the Elite group are discussed.

4.2.1 Ms1: search around solutions located in the less crowded areas

The key idea behind this mutation strategy is to generate mutant vectors around solutions on the Pareto front which are located in the less crowded area as shown in Figure 3. The goals are to discover more non-dominated solutions and improve the distribution of the current front. In order to measure the crowdedness of a vector in the non-dominated front, the CD in NSGA-II is applied. In this mutation strategy, after the Elite group is updated, the CD value of each vector in this group is calculated. Vectors with higher CDs are located in less crowded area and they are considered to be better candidates as perturbed vector in this mutation strategy. The pseudo code for Ms1 is shown in Algorithm A1 where $E$ is the Elite archive.
A1 Algorithm for Ms1
i Calculate_crowding_distance (E)
ii Sort E by decreasing order of CD values
iii Randomly select a vector P from top r% of E.
iv Generate mutant vector (Vi) by equation (8)
\[ V_i = P + F \times (X_n - X_r) \] (8)

Figure 3 Mutant vector generated from Ms1 in bi-objective space

4.2.2 Ms2: pull the current front to the true front

This mutation strategy aims to pull the current solutions found so far by the population as close to the solutions on the true Pareto front. For each move, two vectors are randomly selected from Elite group as reference positions to guide the movement of vectors. The implementation of this strategy in a bi-objective space is illustrated in figure 4 and its pseudo code is shown in Algorithm A2.

A2 Algorithm for Ms2
i Randomly select two vectors, R1 and R2 from Elite group, E.
ii Generate mutant vector (Vi) by equation (9)
\[ V_i = R_1 + F \times (R_2 - X_n) \] (9)
4.2.3 Ms3: fill the gaps of non-dominated front

Inspired by the particle movement strategy in MOPSO, proposed by Nguyen and Kachitvichyanukul (2010), this mutation strategy, Ms3, is proposed with an aim to fill the gaps of the non-dominated front in order to improve the distribution of the solutions in the front. The first step of this strategy is to determine the potential gaps in the Elite group. Then, for each objective function \( f_i(\cdot) \), the solutions in Elite group are sorted in the increasing order of \( f_i(\cdot) \) and the difference of two consecutive vectors is calculated. If the difference is bigger than predetermined \( x\% \) of the current range of the non-dominated solutions in the Elite group, it is considered as a gap and the two corresponding vectors are stored as a pair in an unexplored set, \( U \). Then, a pair of vectors is randomly selected from the unexplored set and implemented in this mutation strategy to guide vectors in the population to new positions. Figure 5 illustrates the direction of the generated mutant vector when the gap is identified and A3 presents the pseudo code for Ms3.

A3 Algorithm for Ms3

i. Identify the gaps in the Elite group, \( E \)
   - For each objective functions \( f_i(\cdot) \)
   - Sort \( E \) in increasing order of objective function \( f_i(\cdot) \)
   - For \( i = 1 \) to \( |E| - 1 \)
     - \( \text{Gap} = f_i(\Theta_{i+1}) - f_i(\Theta_i) \)
     - If \( \text{Gap} > x\% \cdot (f_i^{\max} - f_i^{\min}) \):
       - add pair \((i, i + 1)\) to an unexplored list, \( U \)

ii. Randomly select one pair \((E_1, E_2)\) from \( U \)

iii. Generate mutant vector \( (V_i) \) by equation (10)
\[ V_i = (W + r_1) \ast (E_2 - X_{i_1}) + r_2 \ast (E_1 - E_2) \] (10)

where

- \( r_1, r_2 \) a uniform random number in the range \([0,1]\)
- \( W \) a constant number which controls the scale difference between \( E_2 \) and \( X_{i_1} \)

---

**Figure 5** Mutant vector generated from Ms3 in bi-objective space

**Figure 6** Mutant vector generated from Ms4 in bi-objective space
4.2.4 Ms4: search toward the border of the non-dominated front

Different from Ms1, Ms2, and Ms3, the goal of mutation strategy Ms4, is to discover the solutions along the current non-dominated front specifically at the border in order to increase the spread of non-dominated solutions. In Ms4, the vector $T_k$ with the minimum objective function value respective to objective function $f_k(.)$ is set as the reference position for vector movement direction. Two vectors randomly selected from Elite group $E$ are combined with $T_k$ to form the mutant vectors. The movement direction of the generated mutant vector toward the border of objective function $f_1$ is illustrated in Figure 6. The pseudo code for this algorithm is shown in A4.

A4 Algorithm for Ms4

i Select a vector $T_k$ which has the minimum objective function $f_k(.)$ value

ii Randomly select a vector $R_1$ from $E$

iii Randomly select a vector $R_2$ from $E$

iv Generate mutant vector ($V_i$) by equation (11)

$$V_i = R_i + F \ast (T_k - R_2)$$

(11)

4.2.5 Ms5: explore solution space with multiple groups of vectors

In Ms5, the DE population is divided into multiple groups of vectors. Each group executed a particular search strategy with information sharing across other groups, thus the combination of multiple search strategies is embedded in one population. Since different mutation strategies possess their own strengths and perform the search in different ways in order to reach the Pareto optimal front, this strategy aims to exploit the strengths of various DE mutation strategies and to compensate for the weaknesses of each individual strategy in order to enhance the overall performance and increase the quality of the non-dominated front. In this study, the DE population is divided into four groups with four distinct mutation strategies as described below.

- group 1 consists of vectors that prefer to search at the less crowded areas (Ms1)
- group 2 consists of vectors that aim in the direction of the true Pareto front (Ms2)
- group 3 consists of vectors that try to fill the gaps of non-dominated front (Ms3)
- group 4 consists of vectors that seeks to find non-dominated solution at the border of the Pareto front (Ms4).

In this strategy, the non-dominated solutions found by these four groups of vectors are stored in the common Elite archive which can be accessed by all members. As mentioned above, each group performs different search strategies with own strengths and weaknesses. Since, the mutation strategy Ms1 guide vectors to search at the less crowded areas, the quality on the distribution of the non-dominated front can be increase. However, the movement of vectors in this group will mainly depend on how the solutions are distributed in the current Elite archive. Differently, the vectors in group 2 focus on pulling the current non-dominated front toward the true front. Then again, the
performance of this strategy deteriorates when only few non-dominated solutions are found since the exploration ability of the vectors is limited. It is important to note that the gaps in the non-dominated front are unavoidable even when the multiple mutation strategies are adopted. Thus, the vectors in group 3 (following Ms3) aim to fill these gaps so that more non-dominated solutions are discovered and the final front can have a better distribution. Finally, the task of the vectors in group 4 (following Ms4) is to explore the border in order to increase the spread of non-dominated fronts.

5 Computational experiments

The purpose of this section is to illustrate the effectiveness and performances of the proposed MODE with different mutation strategies. Section 5.1 lists a set of benchmark problems used in this experiment. Two performance measurements used to measure the quality of the obtained solutions are presented in Section 5.2 and MODE parameter setting is given in Section 5.3. Sections 5.4 and 5.5 demonstrate the experimental results of the proposed MODE compared to the results obtained from the state-of-the-art algorithm.

5.1 Test problems

The proposed MODE is tested on a set of benchmark problems (ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6) as introduced by Deb et al. (2002). These test problems, as defined in Table 1, include several various features such as concave and convex, continuous, and discontinuous Pareto optimal front allowing sufficient evaluation on the ability of the proposed algorithm.

<table>
<thead>
<tr>
<th>Problem</th>
<th># Var.</th>
<th>Variable bound</th>
<th>Objective function</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>30</td>
<td>[0, 1]</td>
<td>$f_1(x) = x_i$</td>
<td>Convex</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$f_2(x) = g(x)\left[1 - \sqrt{x_i/g(x)}\right]$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$</td>
<td></td>
</tr>
<tr>
<td>ZDT2</td>
<td>30</td>
<td>[0, 1]</td>
<td>$f_1(x) = x_i$</td>
<td>Non-convex</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$f_2(x) = g(x)\left[1 - (x_i/g(x))^2\right]$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$</td>
<td></td>
</tr>
<tr>
<td>ZDT3</td>
<td>30</td>
<td>[0, 1]</td>
<td>$f_1(x) = x_i$</td>
<td>Convex and disconnected</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$f_2(x) = g(x)\left[1 - \sqrt{x_i/g(x)} - \frac{x_i}{g(x)}\sin(10\pi x_i)\right]$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$</td>
<td></td>
</tr>
</tbody>
</table>

Note: All objectives are to be minimised.
Table 1  ZDT benchmark problems (continued)

<table>
<thead>
<tr>
<th>Problem</th>
<th># Var.</th>
<th>Variable bound</th>
<th>Objective function</th>
<th>Characteristic</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT4</td>
<td>10</td>
<td>( x_i \in [0, 1] )</td>
<td>( f_1(x) = x_i )</td>
<td>Non-convex</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( x_i \in [-5, 5] )</td>
<td>( f_2(x) = g(x)\left[ 1 - \sqrt{x_i/g(x)} \right] )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( i = 2, \ldots, 10 )</td>
<td>( g(x) = 1 + 10(n - 1) + \sum_{i=2}^{10} \left[x_i^2 - 10\cos(4\pi x_i)\right] )</td>
<td></td>
</tr>
<tr>
<td>ZDT6</td>
<td>10</td>
<td>([0, 1])</td>
<td>( f_1(x) = 1 - \exp(-4x_i)\sin^2(6\pi x_i) )</td>
<td>Non-convex and non-uniformly spaced</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( f_2(x) = g(x)\left[ 1 - (f_1(x)/g(x))^2 \right] )</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>( g(x) = 1 + 9\left[ \left( \sum_{i=2}^{10} x_i \right)/(n-1) \right]^{0.25} )</td>
<td></td>
</tr>
</tbody>
</table>

Note: All objectives are to be minimised.

5.2 Performance measurement

To evaluate the performance of the MODE, this study employs two performance metrics; inverted generational distance (IGD) and spread. IGD is used to measure the distance of the obtained non-dominated solutions with respect to those from the true Pareto front as equation (12).

\[
IGD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n}
\]  (12)

where \( d_i \) is the Euclidian distance between the solution points and the nearest member of the true Pareto front and \( n \) is the number of solution points in the true Pareto front. The smaller value of IGD is better since it indicates that the obtained solutions are close to those in the true Pareto front.

Spread is used to measure the distribution of the obtained solutions which is calculated as follow:

\[
Spread = \frac{d_f + d_i + \sum_{i=2}^{n-1} |d_i - \bar{d}|}{d_f + d_i + (n-1)\bar{d}}
\]  (13)

where \( d_f \) and \( d_i \) are the Euclidian distance between the boundary of obtained solutions and the extreme points of the true Pareto front and \( \bar{d} \) is the average of all distances \( d_i \) which is the Euclidian distance between consecutive solutions in the obtained solutions. The zero value of Spread indicates that the obtained non-dominated solutions are equidistantly spaced.
5.3 Parameter setting

The performance of the proposed algorithm is not only sensitive to the selection of mutation strategies, but also to the choice of control parameters. The value of scale factor $F$ is set to random value to allow variation in the scaled difference and thus retains population diversity throughout the search process. After some preliminary experiments, the value of $F$ set to be uniformly randomised between 1.5 and 2 provides generally good solution quality. In addition, this paper uses binomial crossover since it yields better results than exponential crossover. Crossover rate ($C_r$) is linearly increased from 0.1 to 0.5 to maintain the characteristic of generated trial vectors at the beginning of the search. As the search progress, increasing value of $C_r$ yields more deviations for the generated trial vectors and helps the solution to escape from being trapped at local optima. In the proposed algorithm, a fixed Elite archive is employed. If the number of new non-dominated solutions found by the population exceeds the limit of this archive, the solutions with lower CD will be removed. For Ms5, the ratio of numbers of vectors in each group is 1:1:1:1 respectively. The summary of MODE parameters used in this study is shown in Table 2.

![Table 2: Parameters of MODE](image)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>50 vectors</td>
</tr>
<tr>
<td>Scale factor, $F$</td>
<td>Random value between 1.5 and 2</td>
</tr>
<tr>
<td>$W$</td>
<td>1 (Ms3)</td>
</tr>
<tr>
<td>Crossover rate, $C_r$</td>
<td>Linearly increase from 0.1 to 0.5</td>
</tr>
<tr>
<td>Upper limit of Elite group</td>
<td>100 vectors</td>
</tr>
<tr>
<td>% top members</td>
<td>10% (Ms1, Ms5)</td>
</tr>
<tr>
<td>Potential gap</td>
<td>5% (Ms3, Ms5)</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>2,000</td>
</tr>
</tbody>
</table>

5.4 Experimental results

The results of the MODE with different mutation strategies are compared to the results from the state-of-the-art MOPSO algorithm, introduced by Nguyen and Kachitvichyanukul (2010). In MOPSO, several potential particle movement strategies have been introduced and demonstrated their high effectiveness by providing outstanding results over two existing optimisation approaches: NSGA-II (Deb et al., 2002) and the NSGA-II with self-adaptive mechanism (SAM) (Zeng et al., 2010).

For the fair comparison, the number of function evaluations used on in the proposed MODE is set to be as same as that in MOPSO. For each instance, the average and standard deviation of two performance metrics: IGD and Spread are determined from ten independent runs. This study uses the boxplots to graphically illustrate statistical data of performance metrics and observe the convergence and robustness of the algorithms. The comparison results are summarised in Table 3, and the boxplots are shown in Figure 7.

According to the results in Table 3 and Figure 7, it is apparent that the proposed algorithms generally yield very competitive results for all test problems. For the easy test problems; ZDT1, ZDT2, and ZDT3, the MODE algorithms with all DE mutation strategies can easily obtain smaller means for both IGD and Spread with reasonably
small standard deviations compared to MOPSO (Ms*), NSGA-II and NSGAII + SAM. This means that the proposed MODE can find better solutions which are more closely to the true Pareto front and well-distributed compared to other methods. The boxplots of these test problems also visibly confirm the robustness of the proposed algorithms. In the more complex problems like ZDT4 and ZDT6, the proposed MODE with different mutation strategies are still superior to MOPSO (Ms*), NSGA-II and NSGAII + SAM in term of solution quality and robustness by providing relatively lower average value for both $IDG$ and $Spread$ with smaller standard deviations. The performance of the proposed Ms4 in ZDT4 is observed to be not very desirable; however when a group of vectors with Ms4 is combined with other groups as in Ms5, the population is able to shows acceptably good performance.

Table 3 Experimental results for test problems

<table>
<thead>
<tr>
<th>Test problem</th>
<th>Strategy</th>
<th>Mean $IDG$</th>
<th>Standard deviation $IDG$</th>
<th>Mean $Spread$</th>
<th>Standard deviation $Spread$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>MODE-Ms1</td>
<td>1.30E-04</td>
<td>3.68E-06</td>
<td>2.24E-01</td>
<td>2.61E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms2</td>
<td>1.58E-04</td>
<td>3.84E-06</td>
<td>2.61E-01</td>
<td>1.69E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms3</td>
<td>1.51E-04</td>
<td>1.71E-06</td>
<td>2.33E-01</td>
<td>1.50E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms4</td>
<td>1.56E-04</td>
<td>3.01E-06</td>
<td>2.55E-01</td>
<td>1.37E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms5</td>
<td>1.55E-04</td>
<td>3.38E-06</td>
<td>2.36E-01</td>
<td>1.60E-02</td>
</tr>
<tr>
<td></td>
<td>MOPSO (Ms*)</td>
<td>1.68E-04</td>
<td>3.89E-06</td>
<td>3.08E-01</td>
<td>2.21E-02</td>
</tr>
<tr>
<td></td>
<td>NSGA-II + SAM</td>
<td>1.74E-04</td>
<td>5.10E-06</td>
<td>2.92E-01</td>
<td>3.25E-02</td>
</tr>
<tr>
<td></td>
<td>NSGA-II</td>
<td>1.91E-04</td>
<td>1.08E-05</td>
<td>3.83E-01</td>
<td>3.14E-02</td>
</tr>
<tr>
<td>ZDT2</td>
<td>MODE-Ms1</td>
<td>1.58E-04</td>
<td>4.20E-06</td>
<td>2.27E-01</td>
<td>1.86E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms2</td>
<td>1.60E-04</td>
<td>8.08E-06</td>
<td>2.48E-01</td>
<td>3.01E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms3</td>
<td>1.54E-04</td>
<td>3.33E-06</td>
<td>2.10E-01</td>
<td>1.76E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms4</td>
<td>1.66E-04</td>
<td>1.42E-05</td>
<td>2.56E-01</td>
<td>2.97E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms5</td>
<td>1.58E-04</td>
<td>2.74E-06</td>
<td>2.30E-01</td>
<td>1.34E-02</td>
</tr>
<tr>
<td></td>
<td>MOPSO (Ms*)</td>
<td>1.74E-04</td>
<td>4.19E-06</td>
<td>2.96E-01</td>
<td>2.05E-02</td>
</tr>
<tr>
<td></td>
<td>NSGA-II + SAM</td>
<td>1.79E-04</td>
<td>5.64E-06</td>
<td>3.15E-01</td>
<td>2.01E-02</td>
</tr>
<tr>
<td></td>
<td>NSGA-II</td>
<td>1.88E-04</td>
<td>8.36E-06</td>
<td>3.52E-01</td>
<td>7.25E-02</td>
</tr>
<tr>
<td>ZDT3</td>
<td>MODE-Ms1</td>
<td>1.70E-04</td>
<td>4.42E-06</td>
<td>4.63E-01</td>
<td>9.41E-03</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms2</td>
<td>1.75E-04</td>
<td>8.12E-06</td>
<td>4.61E-01</td>
<td>1.20E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms3</td>
<td>1.77E-04</td>
<td>5.41E-06</td>
<td>4.75E-01</td>
<td>1.56E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms4</td>
<td>1.74E-04</td>
<td>4.65E-06</td>
<td>4.69E-01</td>
<td>1.61E-02</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms5</td>
<td>1.74E-04</td>
<td>1.04E-05</td>
<td>4.64E-01</td>
<td>1.67E-02</td>
</tr>
<tr>
<td></td>
<td>MOPSO (Ms*)</td>
<td>1.98E-04</td>
<td>1.16E-05</td>
<td>5.16E-01</td>
<td>1.66E-02</td>
</tr>
<tr>
<td></td>
<td>NSGA-II + SAM</td>
<td>2.46E-04</td>
<td>7.74E-06</td>
<td>7.31E-01</td>
<td>1.20E-02</td>
</tr>
<tr>
<td></td>
<td>NSGA-II</td>
<td>2.59E-04</td>
<td>1.16E-05</td>
<td>7.49E-01</td>
<td>1.49E-02</td>
</tr>
</tbody>
</table>

Notes: The best performance obtained for each test problem is marked in italics. MOPSO (Ms*) are obtained from the mixed particle strategy (MOPSO-Ms*) which generally yields the most robust and outstanding results as reported in Nguyen and Kachitvichyanukul (2010).
Table 3  Experimental results for test problems (continued)

<table>
<thead>
<tr>
<th>Test problem</th>
<th>Strategy</th>
<th>IGD</th>
<th>Spread</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>ZDT4</td>
<td>MODE-Ms1</td>
<td>1.62E-04</td>
<td>8.30E-06</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms2</td>
<td>1.65E-04</td>
<td>5.17E-06</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms3</td>
<td>1.52E-04</td>
<td>3.40E-06</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms4</td>
<td>5.49E-03</td>
<td>2.68E-03</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms5</td>
<td>1.55E-04</td>
<td>3.72E-06</td>
</tr>
<tr>
<td></td>
<td>MOPSO (Ms*)</td>
<td>1.65E-04</td>
<td>6.49E-06</td>
</tr>
<tr>
<td></td>
<td>NSGA-II + SAM</td>
<td>1.67E-04</td>
<td>8.02E-06</td>
</tr>
<tr>
<td></td>
<td>NSGA-II</td>
<td>1.64E-04</td>
<td>8.68E-06</td>
</tr>
<tr>
<td>ZDT6</td>
<td>MODE-Ms1</td>
<td>1.06E-04</td>
<td>8.07E-06</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms2</td>
<td>1.27E-04</td>
<td>1.70E-05</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms3</td>
<td>1.02E-04</td>
<td>3.68E-06</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms4</td>
<td>1.32E-04</td>
<td>1.62E-05</td>
</tr>
<tr>
<td></td>
<td>MODE-Ms5</td>
<td>1.23E-04</td>
<td>1.61E-05</td>
</tr>
<tr>
<td></td>
<td>MOPSO (Ms*)</td>
<td>1.28E-04</td>
<td>2.82E-05</td>
</tr>
<tr>
<td></td>
<td>NSGA-II + SAM</td>
<td>1.51E-04</td>
<td>1.06E-05</td>
</tr>
<tr>
<td></td>
<td>NSGA-II</td>
<td>1.59E-04</td>
<td>1.24E-05</td>
</tr>
</tbody>
</table>

Notes: The best performance obtained for each test problem is marked in italics. MOPSO (Ms*) are obtained from the mixed particle strategy (MOPSO-Ms*) which generally yields the most robust and outstanding results as reported in Nguyen and Kachitvichyanukul (2010).

It is important to notice that mutation strategy Ms1 and Ms3 show very good performance and have the best IGD and Spread in most test problems. These results confirm that Ms1 and Ms3 are more suitable for finding the high quality Pareto front. The vectors following these searching behaviours (Ms1 and Ms3) are able to find well-distributed non-dominated solutions which are closed to the true Pareto optimal front. The vectors following mutation strategies Ms2 and Ms4 generally yield good Pareto fronts, but Ms4 alone is not very effective in the difficult problem like ZDT4.

The mutation strategy Ms5 is expected to provide the best front quality among other strategies since a population contains multiple vectors which perform different potential search behaviours. However, the results obtained from Ms5 are not as good as other mutation strategies. The reasons may be because Ms5 may requires a different parameter settings; for example, the proportion the population size should be increase to provide sufficient vectors in each group to fully search in different ways or with the same population size, the experiment need to be run longer to find a better front. The extension on the experiments of Ms5 is investigated in the following section.
Figure 7  Boxplots of the performance metric of IGD and Spread for test problems (see online version for colours)
5.5 Experiment extension of Ms5

This experiment is designed to determine whether solutions obtained from Ms5 can be improved by changing a parameter setting. Observing from results in Table 3, Ms1 and Ms3 mostly show the best performance while Ms2 and Ms4 generally provide good solutions. For these reasons, in Ms5, the ratio of numbers of vectors in each group is set as 3:2:3:2 respectively so that it allows more vectors to search following Ms1 and Ms3. The population size is set to the same as 50 and run for the same number of iterations. The experiment is investigated on two test problems: ZDT1 and ZDT4, and the results are shown in Table 4.

<table>
<thead>
<tr>
<th>Ms5</th>
<th>IGD Mean</th>
<th>IGD Standard deviation</th>
<th>Spread Mean</th>
<th>Spread Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>1.48E-04</td>
<td>2.63E-06</td>
<td>2.15E-01</td>
<td>1.56E-02</td>
</tr>
<tr>
<td>ZDT4</td>
<td>1.50E-04</td>
<td>2.28E-06</td>
<td>2.05E-01</td>
<td>1.88E-02</td>
</tr>
</tbody>
</table>

It can be seen from Table 4 that by simply changing the proportion of numbers of vectors in each group and allowing more vectors to search following Ms1 and Ms3 produce significant improvement without having to increase number of function evolutions.

As mentioned, the proposed MODE algorithms emphasise on improving the movement behaviour of DE population by utilising the information of the Elite group which is different from other algorithms found in literatures that mainly used traditional selection scheme borrowed from NSGA. The objective of the experiments in this section is to show that the quality of the non-dominated solutions obtained depends significantly on the mutation strategy adopted by the population. The advantages of these proposed mutation strategies can be easily combined in the DE framework by simply changing the guidance selection. In addition, it can be easily adapted to solve more complex problems without significant deterioration in the performance.

6 Conclusions

This paper proposes the novel MODE algorithm which emphasises on the development of several mutation strategies dictating the movement direction of DE population by utilising the information of the Elite group in order to search for non-dominated or Pareto solutions in MO problems. The idea of directing the movement behaviour of DE population in this study is different from other algorithms found in literatures that mainly used traditional selection scheme borrowed from NSGA. Five potential mutation strategies with distinct search behaviour are proposed to obtain the high quality Pareto front. Mutation strategies Ms1 and Ms3 aim to improve the distribution of solutions on the front. Mutation strategy Ms2 intends to pull to the current front toward the true Pareto front, and mutation strategy Ms4 try to explore more solutions around the border to increase the spread of the front. Mutation strategy Ms5 aims to extract the strengths of various DE mutation strategies and to compensate for the weaknesses of each individual strategy in order to enhance the overall performance by combining four groups of vectors with multiple mutation strategies in one population. The performance of the proposed
algorithms is evaluated on a set of benchmark problems. The experimental results show that MODE algorithms developed from these mutation strategies are very competitive when compared to MOPSO, NSGA, and NSGA-II with self adaptive mechanism with respect to convergence, distribution and robustness. To further demonstrate the effectiveness of the proposed algorithm, the future research directions include improving the algorithm performance and investigating its performance on more complex MO optimisation problems.

Acknowledgements

The authors thank the anonymous reviewers for their valuable comments. The prompt handling of the review process by the editor is also acknowledged.

References


