Proceedings of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2014
August 17-20, 2014, Buffalo, New York, USA

DRAFT: A NEW MULTI-OBJECTIVE MIXED-DISCRETE PARTICLE SWARM OPTIMIZATION ALGORITHM

Weiyang Tong*
Syracuse University
Syracuse, NY 13244
Email: wtong@syr.edu

Souma Chowdhury†
Mississippi State University
Starkville, MS, 39759
Email: chowdhury@bagley.msstate.edu

Achille Messac‡
Mississippi State University
Mississippi State, MS, 39762
Email: messac@bagley.msstate.edu

ABSTRACT

Heuristic algorithms are powerful solvers for high dimensional and nonlinear optimization problems. Among them, the Particle Swarm Optimization (PSO) algorithm has gained significant popularity due to its maturity and fast convergence abilities. This paper seeks to translate the unique benefits of PSO from solving typical continuous single objective problems to solving multi-objective mixed-discrete problems which is relatively a new ground for PSO application. The previously developed Mixed-Discrete Particle Swarm Optimization (MDPSO) algorithm, which includes an exclusive diversity preservation technique to prevent premature particle clustering, has been shown to be a powerful single objective solver for highly constrained MINLP problems. In this paper, we make fundamental advancements to the MDPSO algorithm to make it capable of solving multi-objective problems with mixed-discrete design variables, one of the most challenging classes of optimization problems. In the velocity update equation, the explorative term is modified to point towards the non-dominated solution that is the closest to the corresponding particle (at any iteration). The fractional domain in the diversity preservation technique, which was previously defined in terms of the globally best particle, is now formulated as a function of the extreme solutions in the intermediate Pareto front. The multi-objective MDPSO algorithm is tested using a set of benchmark problems and a wind farm layout optimization problem. To establish the advantages of the new multi-objective MDPSO, the results are compared with those given by Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) in the final paper.

KEYWORDS: Crowding distance, Diversity preservation, Mixed-discrete, MOPSO, Multi-objective, Wind Farm Layout Optimization

INTRODUCTION

Owing to the existence of multi-criteria in real-life problems/applications, *Multi-objective Optimization* is desired, where multiple objectives are to be optimized. Instead of searching one single optimum in the objective space, multi-objective optimization problems aim to explore the best trade-off that comprises a set of Pareto solutions, representing the compromises among multiple objectives.

Aggregating approach is one common solution to solve multi-objective optimization problems due to its relatively easy implementation. However, the implementation of this approach relies on the performance of the selected single objective optimizer, and it is difficult to apply without foreknowing the features of objectives being solved. Mostly importantly, it suffers from a critical drawback: only one solution can be obtained per each function evaluation. This inherently increases the computational cost, especially in the context of complex applications where most problems are often nonlinear and high dimensional.

*Doctoral Student, Multidisciplinary Design and Optimization Laboratory, Department of Mechanical and Aerospace Engineering. ASME Student Member

†Assistant Research Professor, Department of Mechanical Engineering, Center for Advanced Vehicular Systems. ASME Professional Member

‡Dean of the Bagley College of Engineering, Professor of Aerospace Engineering, Earnest W. and Mary Ann Deavenport, Jr. Chair. ASME Lifetime Fellow. Corresponding author.

Heuristic algorithms are suitable solvers for nonlinear and high dimensional problems. Among them, Evolutionary Algorithms (EA) and Particle Swarm Optimization (PSO) algorithms have gained the most popularity. EA are well-suited for multi-objective problems mainly attributed to the fact that they are able to find sub-optimal solutions per function evaluation. This fact allows the Multi-Objective EA (MOEA) to search simultaneously for multiple Pareto solutions [1]. Various MOEA have been developed, such as NSGA-II [2], SPEA [3], SPEA2 [4], and PPES [5]

PSO is a Swarm Intelligence method that roughly models the social behavior of swarms [6]. Owing to its maturity and fast convergence, PSO has been shown as a powerful single-objective optimizer when solving unconstrained problems in continuous domain. Comparing to EA, PSO has the following distinguished features [7]:

Many work related to Multi-Objective PSO (MOPSO) have been done over the past few years. Hu and Eberhart [8, 9] developed a dynamic neighborhood PSO (mainly applicable for bi-objective optimization problems), in which only one objective is optimized at a time and an external archive is used to improve the search performance. Parsopoulos et al. [10] proposed a parallel version of the Vector Evaluated PSO (VEPSO) method for multi-objective problems, which is inspired on the VEGA. This method is a multi-swarm variant of PSO, in which the local search is based on the evaluation of each swarm while the global search depends on the information exchange between multiple swarms. Coello et al. [11] developed a MOPSO algorithm, in which an external repository is used to keep the information regarding historical record of the non-dominated solutions found during the search. In addition, adaptive grids (formed by hypercubes) are used to maintain the uniformity of the Pareto solutions, and a mutation operator is added to avoid the premature of the swarm. Abido [12] proposed a MOPSO algorithm with non-dominated solutions stored in local and global sets. Clustering algorithm is applied to manage the sizes of both sets. A recent comprehensive literature survey by Lalwani et al. [13] also showed that MOPSO algorithms have been widely applied in a variety of fields, as shown in Fig. 1. The previously developed Mixed-Discrete Particle Swarm Optimization (MDPSO) algorithm has been shown to be a powerful single objective solver for highly constrained MINLP problems [14]. In this paper, we make fundamental advancements to the MDPSO algorithm to make it capable of solving constrained multi-objective problems that are nonlinear, high dimensional, and multi-modal. Moreover, the MOMDPSO is also capable of solving problem with mixed-discrete design variables, one of the most challenging classes of optimization problems. However, this feature is rarely reported in the literature of MOPSO.

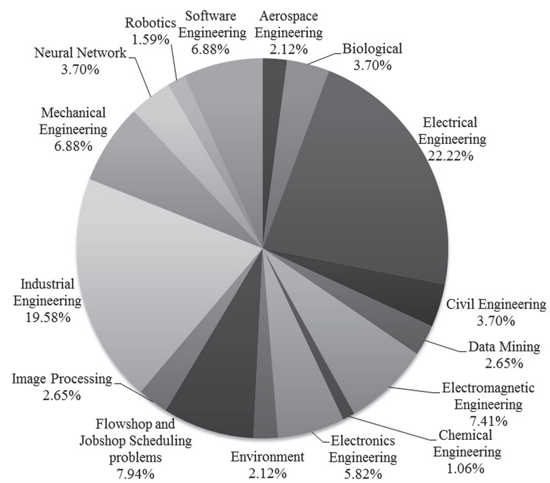


FIGURE 1: Area wise published applications of MOPSO [13]

MULTI-OBJECTIVE MIXED-DISCRETE PARTICLE SWARM OPTIMIZATION ALGORITHM

Basic PSO and Single-objective MDPSO

Particle Swarm Optimization (PSO) is a population-based single-objective optimization algorithm [6]. In basic PSO, candidate solutions are represented by "particles", flying through the search space. The flight is governed by two basic equations: (i) position update and (ii) velocity update. The position of each particle represents one candidate solution in the search space, and the update is based on an assigned velocity vector. This process can be expressed as

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{1}$$

where $\vec{x}_i(t)$ denotes the current position of particle-i at t^{th} generation. The new position at $t+1^{th}$ generation is obtained by adding an updated velocity vector $\vec{v}_i(t+1)$ to the current position. This velocity vector is determined using the equation below:

$$\begin{aligned}\vec{v}_i(t+1) = W\vec{v}_i(t) \; &+ C_1 r_1(\vec{x}_i^{pbest} - \vec{x}_i(t)) \\ &+ C_2 r_2(\vec{x}^{gbest} - \vec{x}_i(t))\end{aligned} \tag{2}$$

where W is the inertia weight that balances the search between exploitation (locally) and exploration (globally); $\vec{x}_i^{pbest}$ is the local best solution that has been found by particle-i till t^{th} generation, which is based on its own experience and that of its neighbors; $\vec{x}^{gbest}$ is the currently the best solution among the entire population (also as known as the global leader), which is determined through a socially information exchange with all the local best solutions;C_1 and C_2 represent cognitive and social parameters, respectively; and r_1 and r_2 are real random numbers between 0 and 1.

However, as a population-based algorithm, PSO suffers from the premature stagnation when solving constrained optimization problems [15]. This primary drawback of PSO is mainly attributed to the loss of diversity during the fast convergence. Chowdhury et al. developed a Mixed-Discrete Particle Swarm Optimization (MDPSO) algorithm to solve mixed-discrete single-objective constrained optimization problem, which is generally formulated as

$$\begin{aligned} & \min_{x_j \in \vec{x}} f(\vec{x}) \\ & \vec{x} = [x_1, x_2, \ldots, x_m, x_{m+1}, x_{m+2}, \ldots, x_n] \\ subject\ to & \\ & g_k(\vec{x}) \leq 0,\ k = 1, 2, \ldots, p \\ & h_l(\vec{x}) = 0,\ l = 1, 2, \ldots, q \end{aligned} \tag{3}$$

where $\vec{x}$ is the design vector that includes m discrete variables and $n-m$ continuous variables; p and q represent the number of inequality ($g_k(\vec{x})$) and equality (($h_l(\vec{x})$) constraints, respectively.

In MDPSO, an explicit diversity preservation term is added to the velocity update equation. Therefore, Eq.(2) can be rewritten as

$$\begin{aligned} \vec{v}_i(t+1) = W\vec{v}_i(t) &+ C_1 r_1 (\vec{x}_i^{pbest} - \vec{x}_i(t)) \\ &+ C_2 r_2 (\vec{x}^{gbest} - \vec{x}_i(t)) \\ &+ \gamma_c r_3 \hat{v}_i(t) \end{aligned} \tag{4}$$

where γ_c is the diversity preservation coefficient for continuous design variables, which is evaluated adaptively as a function of the prevailing diversity in the population at each generation; r_3 is a real random numbers between 0 and 1; and $\hat{v}_i(t)$ is the diverging velocity vector that is defined opposite to the global leader, as given by

$$\hat{v}_i(t) = \vec{x}_i(t) - \vec{x}^{gbest} \tag{5}$$

The mixed-discrete design variables are solved in continuous domain and then approximated using the straightforward Nearest Vertex Approach (NVA). For the j^{th} discrete design variable, its local hypercude H_j^d is defined as

$$\begin{aligned} & H_j^d = \{(x_1^L, x_1^U), (x_2^L, x_2^U), \ldots, (x_m^L, x_m^U)\}, \\ & x_j^L \leq x_j \leq x_j^U,\ \forall j = 1, 2, \ldots, m \end{aligned} \tag{6}$$

where x_j^L and x_j^U are two consecutive feasible values of the j^{th} discrete variable.

As the continuous variable diversity coefficient regulates the particle movement, the diversity preservation scheme for discrete variables is applied within the NVA application, where each discrete candidate solution is approximated to x_j^L or x_j^U based on the evaluated diversity of discrete variables.

In addition, the net constraint violation $f_c(\vec{x})$ is used, which combines the violation values of both inequality and equality constraints. The principle of non-domination is then used to compare solutions and handle constraints [2].

Development of Multi-Objective MDPSO

The MDPSO has well kept the basic features of the original PSO; meantime, it also improves the diversity preservation to avoid the premature stagnation. In this paper, we develop a Multi-Objective MDPSO (MOMDPSO) based on its original structure. In other words, we introduce the distinguished features of MDPSO to a MOPSO algorithm, and make it suitable for solving nonlinear, high dimensional, and highly constrained multi-objective optimization problems with mixed-discrete design variables.

In general, a mixed-discrete multi-objective constrained optimization problem can be formulated as

$$\begin{aligned} & \min_{x_j \in \vec{x}} \vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_{N_{obj}}(\vec{x})] \\ & \vec{x} = [x_1, x_2, \ldots, x_m, x_{m+1}, x_{m+2}, \ldots, x_n] \\ subject\ to & \\ & g_k(\vec{x}) \leq 0,\ k = 1, 2, \ldots, p \\ & h_l(\vec{x}) = 0,\ l = 1, 2, \ldots, q \end{aligned} \tag{7}$$

where $\vec{f}(\vec{x})$ is the vector of objective that includes N_{obj} objectives.

The key to turn the basic PSO into a multi-objective optimizer is the local and global search mechanism. In single-objective PSO algorithm, both the local and global best solutions are represented by a single position is the search space. Analogically, in the MOPSO algorithm we replace the local and global best solutions by a set of non-dominated solutions. Therefore, the position and velocity update equations for MOMDPSO algorithm are given by

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{8}$$

and

$$\begin{aligned} \vec{v}_i(t+1) = W\vec{v}_i(t) &+ C_1 r_1 (\vec{x}_{ir}^{pbest} - \vec{x}_i(t)) \\ &+ C_2 r_2 (\vec{x}_{is}^{gbest} - \vec{x}_i(t)) \\ &+ p_0 \gamma_c r_3 (\vec{x}_i(t) - \vec{x}_{is}^{gbest}) \end{aligned} \tag{9}$$

where $\vec{x}_{ir}^{pbest}$ is particle-i's "local best solution" selected from its non-dominated local set X_i^{pbest};$\vec{x}_{is}^{gbest}$ is particle-i's global leader selected from its non-dominated global set X_i^{gbest}; and p_0 is a user-defined probability: if $r_3 \leq p_0$, γ_c than becomes 0.

The non-dominated local set stores the non-dominated solutions so far have been found by particle-i. It is noted that whether a solution is eligible to enter this local set is determined by the non-domination principle. The stored solution that is the farthest to the current one will be selected as the "local best solution". For the purpose of diversity preservation, the crowding distance is used to regulate the spread of this set in objective space [2].

The non-dominated global set is generated by combining all the non-dominated local sets, and a Pareto filter is then applied to select the non-dominated solutions so far have been found by the entire population. Unlike selecting one solution as the global leader in the basic PSO algorithm, here each particle selects the stored solution that has the shortest distance from the non-dominated global set as its global leader. Therefore, all solutions in the non-dominated global set are possible to be selected as global leaders. Additionally, the capacity of the non-dominated global set is user-defined, and we use a uniformity metric to regulate the size of this global set. The uniformity metric $\Delta'$is given by [16, 17]

$$\Delta' = \sum_{i=1}^{S-1} \frac{|d_i - \bar{d}|}{S-1} \tag{10}$$

where S represents the size of non-dominated global set; d_i is the Euclidean distance between two consecutive solutions in objective space; and $\bar{d}$ is the averaged distance of all the solutions.

Diversity Preservation in MOMDPSO The diversity of the swarm is quantified by the spread of solutions in variable space. For continuous variables, the diversity metric D_c is quantified by the volume of the smallest hypercude enclosing all particles, as given by

$$D_c = \prod_{j=m+1}^{n} \frac{x_j^{max}(t) - x_j^{min}(t)}{x_j^{max} - x_j^{min}} \tag{11}$$

where $x_j^{max}(t)$ and $x_j^{min}(t)$ are respectively the maximum and minimum values that so far have been reached along the j^{th} dimension at the t^{th} generation; and x_j^{max} and x_j^{min} represent the pre-defined upper and lower bounds of the j^{th} design variable, respectively.

For mixed-discrete design variables, the diversity metric is quantified within its local hypercude H_j^d, seeking to avoid the stagnation inside the discrete-space. For the j^{th} discrete design variable, its diversity metric $D_{d,j}$ is given by

$$D_{d,j} = \frac{x_j^{max}(t) - x_j^{min}(t)}{x_j^{max} - x_j^{min}} \tag{12}$$

Here, it is worth to mention that the spread of solutions in variable space is not only defined by the current location of the swarm. The stored positions in all the non-dominated local sets are also taken into account for measuring the diversity metric.

Owing to the impact of outlier particles on the measurement of diversity metric, a fractional domain is applied to modify both D_c and $D_{d,j}$. This fractional domain is defined as a fraction given by

$$F_\lambda = \left(\lambda \frac{N+1}{N_\lambda + 1}\right)^{\frac{1}{n}} \tag{13}$$

where λ is a user-defined parameter that represents the fraction of the volume of the fractional domain to that of the smallest hypercude enclosing all particles (real number between 0 to 1); N is the population size; and N_λ is the number of particles enclosed by the fractional domain. In this paper, the upper and lower bounds of this domain on the j^{th} dimension at the t^{th} generation, $\bar{x}_j^{max}(t)$ and $\bar{x}_j^{min}(t)$, are formulated as

$$\begin{aligned} \bar{x}_j^{max}(t) &= \max \begin{bmatrix} x_j^{min}(t) + \lambda \Delta x_j(t) \\ \min\left(\hat{x}_{extrm}^{gbest} + 0.5\lambda \Delta x_j(t), x_j^{max}(t)\right) \end{bmatrix}, \\ \bar{x}_j^{min}(t) &= \min \begin{bmatrix} x_j^{max}(t) - \lambda \Delta x_j(t) \\ \max\left(\hat{x}_{extrm}^{gbest} - 0.5\lambda \Delta x_j(t), x_j^{min}(t)\right) \end{bmatrix} \end{aligned} \tag{14}$$

where $\Delta x_j(t) = x_j^{max}(t) - x_j^{min}(t)$; $\hat{x}_{extrm}^{gbest}$ represents one of the anchor points so far have been found in the non-dominated global set.

Therefore, the modified diversity metric for both continuous and mixed-discrete variables are expressed as

$$\bar{D}_c = F_\lambda \times D_c \tag{15}$$

$$\bar{D}_{d,j} = F_\lambda \times D_{d,j} \tag{16}$$

The diversity preservation coefficient γ_c in Eq.(9) is then expressed as

$$\begin{aligned}\gamma_c &= \gamma_{c0}\exp\left(\frac{-\bar{D}_c^2}{2\sigma_c^2}\right), \text{and}\\ \sigma_c &= \frac{1}{\sqrt{2\ln 1/\gamma_{min}}}\end{aligned} \tag{17}$$

whereγ_{c0} and γ_{min} are user-defined scale and shape parameters of the diversity metric.

For mixed-discrete design variables, the diversity-based Nearest Vertex Approach (NVA) is applied in this paper as well. The discrete variable diversity preservation coefficient $\gamma_{d,j}$ is expressed as

$$\begin{aligned}\gamma_{d,j} &= \gamma_{d0}\exp\left(\frac{-\bar{D}_{d,j}^2}{2\sigma_c^2}\right), \text{and}\\ \sigma_{d,j} &= \frac{1}{\sqrt{2\ln 1/M_j}}\\ \forall j &= 1,2,\ldots,m\end{aligned} \tag{18}$$

where the scale of the diversity metric is controlled by the size of the j^{th} discrete variable M_j, which represents its total number of feasible values; and γ_{d0} is a user-defined parameter that represents the probability of position update for discrete variables. Based on the value of $\gamma_{d,j}$ and a random real number r_4 between 0 and 1, the position of a particle x_j on the j^{th} dimension will be updated using the following rules:

i If r_4 is less than or equal to $\gamma_{d,j}$, x_j is randomly approximated to either x_j^U or x_j^L (as defined in Eq.(6)).

ii If r_4 is greater than $\gamma_{d,j}$, x_j is approximated to x_j^L if $|x_j - x_j^L| \leq |x_j - x_j^U|$; otherwise, x_j is approximated to x_j^U.

It is important to note that the diversity preservation coefficient for continuous variables is to apply a dynamic repulsion away from the region which the population should converge to; while the one for mixed-discrete variables is to apply stochastic update among all the feasible solutions in the discrete space. In the MDPSO algorithm, the λ-fractional domain reflects how many particles have been entered the region near the global leader. As more particles are closing to the global leader, the modified diversity metric increases. This will result in a reduction of the repulsion. However, for MOPSO algorithm, multiple global leaders are selected. Here, we randomly select one of the anchor points from the non-dominated global set at each generation. In this case, particles tend to expand the spread of the non-dominated global set and capture the full Pareto frontier.

Constraint Handling in MOMDPSO The net constraint violation $f_c(\vec{x})$ is used to calculate the constrain violation, which is given by

$$f_c(\vec{x}) = \sum_{k=1}^{p}\max(\bar{g}_k, 0) + \sum_{l=1}^{q}\max(\bar{k}_l - \varepsilon, 0) \tag{19}$$

where $\bar{g}_k$ and $\bar{h}_l$ represent the normalized inequality and equality constraints, respectively; and ε represents the tolerance for constraints.

The non-domination principle is used for solution comparison to determine if a solution is feasible or not. In single-objective PSO algorithm, if all solutions are infeasible, the particle with smallest violation value will be selected as the global leader. However, for this specific scenario, it is inefficient to only use one global leader for MOPSO algorithm, especially for highly constrained multi-objective optimization problems. In this paper, we developed a constraint handling scheme for constrained multi-objective optimization problems. When all solutions are infeasible, the following scheme will be applied for selecting the global leaders:

```
V = 1
for all particle i ∈ Pop do
    vio = sort(f_{c,i})
    if f_{c,i} < ε̃ then
        V = V + 1
    end if
end for
if V ≥ S then
    s = 1
    while s < S do
        X^{gbest} push_back(V_s)
    end while
else
    s = 1
    while s < L do
        X^{gbest} push_back(V_s)
    end while
end if
```

Here, S is the specified capacity of the non-dominated global set $\vec{X}^{gbest}$, and $\tilde{\varepsilon}$ is a user-defined lower tolerance. Particles with violations smaller than $\tilde{\varepsilon}$ are chosen to enter $\vec{X}^{gbest}$; however, if the population of these particles satisfying $\tilde{\varepsilon}$ is still less than S, a smaller number of particles, L, will be selected based on their violation values ranked by vector *vio*.

NUMERICAL EXPERIMENT

To validate the Multi-Objective MDPSO (MOMDPSO) algorithm, we apply it to three different classes of multi-objective optimization problems: (i) standard unconstrained continuous test functions, (ii) constrained continuous test functions, and (iii) constrained mixed-discrete problems. In the third class, a multi-objective wind farm layout optimization problem is included. These three sets of numerical experiments will be discussed in the following three subsections.

TABLE 1: User-defined parameters for standard unconstrained continuous test functions

Parameter	Value
Population size	$\min(100, 10m)$
Elite set size	40
W	0.5
C_1	1.5
C_2	1.5
λ	0.1
γ_c	1.0
γ_{c0}	$1e-6$
p_0	0.25

Unconstrained Continuous Test Functions

First, the performance of MOMDPSO is tested using a series of well-known unconstrained bi-objective optimization problems, with known analytical solution. Table 2 lists the formulations of the four test functions by Zitzler et al. [18] namely the ZDT test cases, as well as two other popular test cases by Fonseca and Fleming [19] and Coello et al. [20]. In addition, their analytical solutions are also provided. All the six test functions have two objectives that are to be *minimized*. The number of function evaluations for ZDT test cases is $25,000$ times and $2,000$ times for the other two cases.

Two performance metrics introduced by Deb et al. [16] are used to evaluate the performance of MOMDPSO. The first performance metric, γ, measures the convergence towards the actual boundary, which is given by

$$\gamma = \frac{\sum_{i=1}^{|S|} d_i}{|S|}, \qquad d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^{K} \left[\frac{f_k^i - f_k^j}{f_k^{max} - f_k^{min}}\right]^2} \tag{20}$$

where P^* is the reference set of Pareto solutions and S is the obtained set of non-dominated solutions; K is the total number of objectives (in this case, $K = 2$); d_i represents the shortest Euclidean distance between the i^{th} obtained non-dominated solution to P^*; and f_k^{max} and f_k^{min} represent the maximum and the minimum function values of the k^{th} objective in P^*, respectively. In this paper, 500 uniformly distributed solutions are generated in P^* ($|P^*| = 500$) to calculate γ. The second performance metric, Δ, measure the diversity of non-dominated solutions, which is given by

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|S|-1} \frac{|d_i - \bar{d}|}{|S|-1}}{d_f + dl + (|S|-1)\bar{d}} \tag{21}$$

where d_i is the Euclidean distance between two consecutive non-dominated solutions in S; d_f and d_l are the Euclidean distances between the extreme solutions of P^* and those of S, respectively; and $\bar{d}$ is the averaged distance of all $d_i, i \in [1, |S|-1]$.

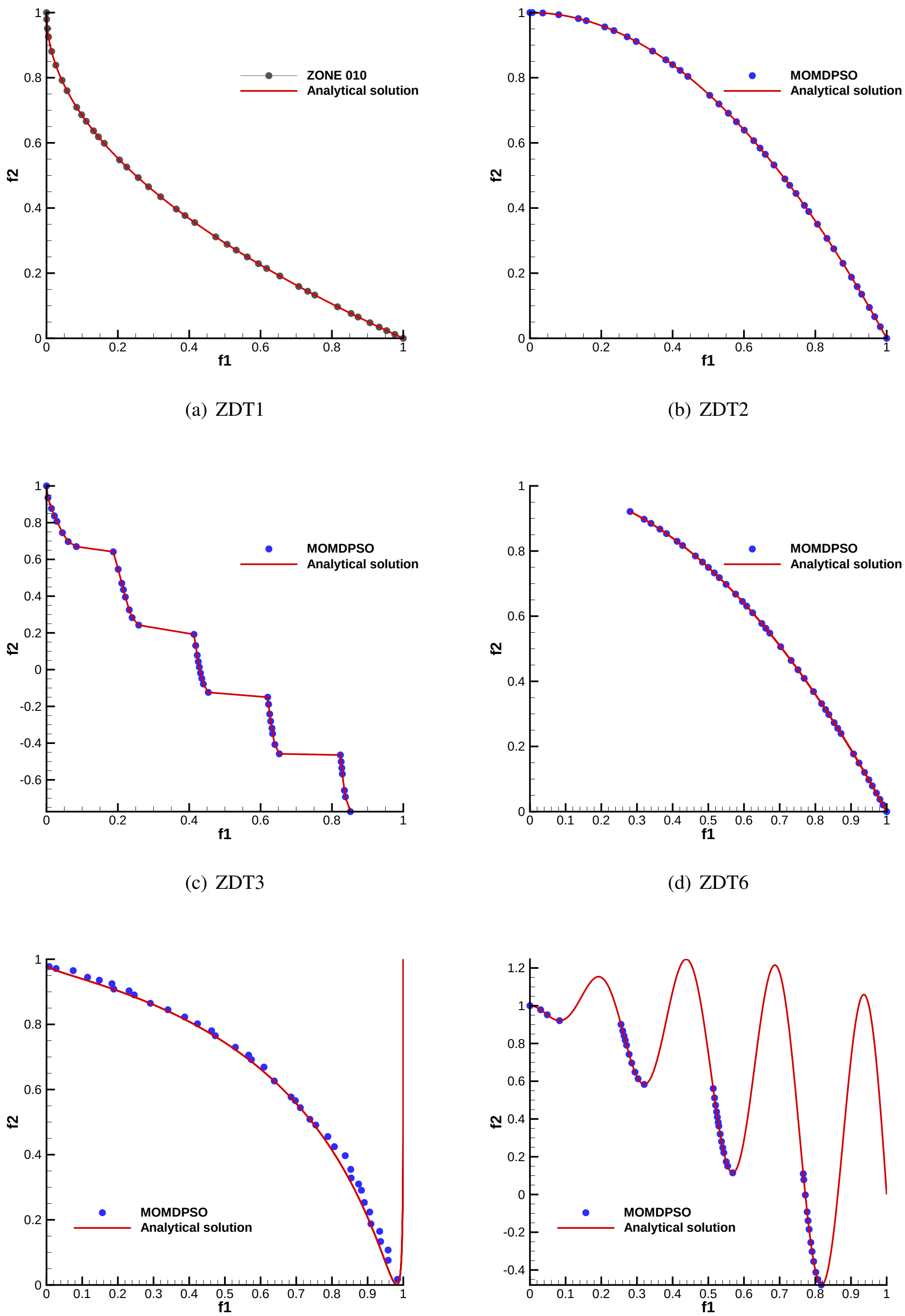
In order to compensate the impact of random parameters, the optimization for each test function is run 10 times. The best result out of 10 runs (in terms of γ and Δ) for each test function is compared with its analytical solution. Fig. 2(a) to 2(f) present the Pareto frontiers obtained by MOMDPSO for all six test cases. From the figures, it is observed that the results for all test functions have strong agreement with the analytical solution, except the Fonseca-Fleming problem, which did not converge to the actual boundary due to the low number of function evaluations. Additionally, it is also observed that the actual extreme solutions for each case are captured by the MOMDPSO.

Constrained Continuous Test Functions

In this subsection, the performance of MOMDPSO is evaluated using five constrained bi-objective test functions. Table 4 lists the formulations of test functions, including the objective functions and constraints. The number of function evaluations for this set of test cases is $30,000$. The algorithm is run 10 times for each case due to the impact of random parameters.Since analytical solutions are not provided for this set of test cases, the best result is determined in terms of Δ' and the Euclidean distance between the extreme solutions in the obtained set S. Fig. 3(a) to 3(e) present the best Pareto sets of all five test cases obtained by MOMDPSO. Additionally, the values of the prescribed MDPSO parameters for this set of test cases can be found in Table 3.

Constrained Mixed-discrete Problems

In this subsection, a constrained bi-objective MINLP problem and a multi-objective wind farm layout optimization problem are used to evaluate the performance of MOMDPSO for solv-

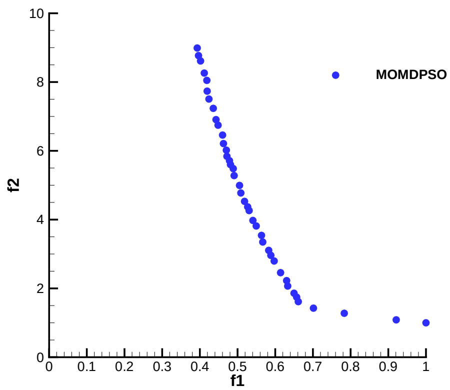


(a) ZDT1 (b) ZDT2

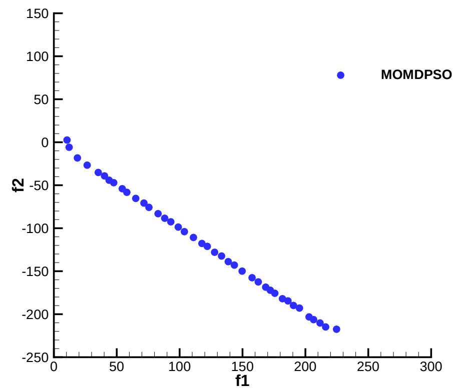
(c) ZDT3 (d) ZDT6

(e) Fonseca-Fleming (f) Coello

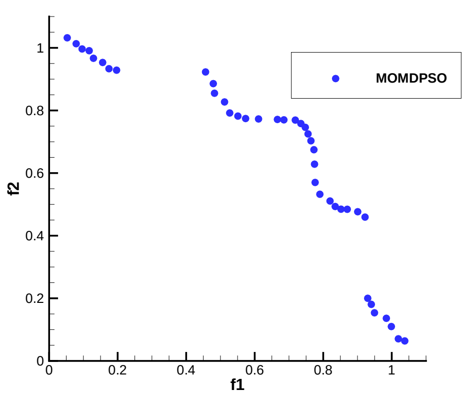
FIGURE 2: MOMDPSO results for standard unconstrained continuous test functions



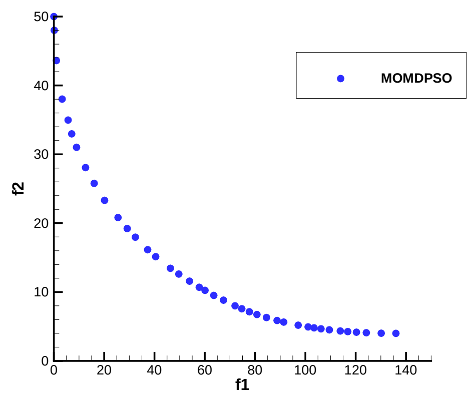
(a) CONSTR



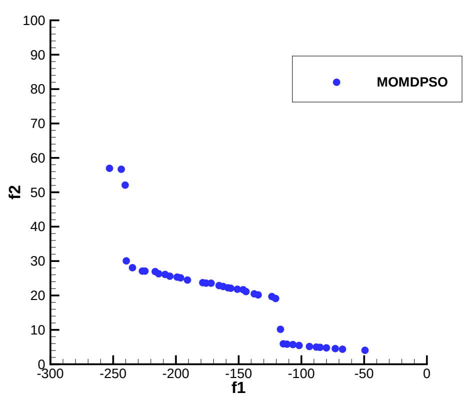
(b) SRN



(c) TNK



(d) Binh-Korn



(e) Osyczka-Kundu

FIGURE 3: MOMDPSO results for standard unconstrained continuous test functions

TABLE 3: User-defined parameters for constrained continuous test functions

Parameter	Value
Population size	100
Elite set size	40
W	0.5
C_1	1.5
C_2	1.5
λ	0.1
γ_c	1.0
γ_{c0}	$1e-5$
p_0	0.25
$\tilde{\varepsilon}$	$1.0e-6$
L	10

ing highly constrained multi-objective mixed-discrete optimization problems. The user-defined parameter for both problems are listed in Table 5.

The constrained bi-objective MINLP problem is adopted from Dimkou et al. [24], which is given by

$$
\begin{aligned}
& \min_{\vec{x},\vec{y}} (f_1(\vec{x},\vec{y}), f_2(\vec{x},\vec{y})) \\
& f_1(\vec{x},\vec{y}) = x_1^2 - x_2 + x_3 + 3y_1 + 2y_2 + y_3 \\
& f_2(\vec{x},\vec{y}) = 2x_1^2 + x_2 - 3x_3 - 2y_1 + y_2 - 2y_3 \\
\textit{subject to} \quad & \\
& g_1(\vec{x},\vec{y}) = 3x_1 - x_2 + x_3 + 2y_1 \le 0 \\
& g_2(\vec{x},\vec{y}) = 4x_1^2 + 2x_1 + x_2 + x_3 - 40 + y_1 + 7y_2 \le 0 \\
& g_3(\vec{x},\vec{y}) = -x_1 - 2x_2 + 3x_3 + 7y_3 \le 0 \\
& g_4(\vec{x},\vec{y}) = -x_1 - 10 + 12y_1 \le 0 \\
& g_5(\vec{x},\vec{y}) = x_1 - 5 - 2y_1 \le 0 \\
& g_6(\vec{x},\vec{y}) = -x_2 - 20 + y_2 \le 0 \\
& g_7(\vec{x},\vec{y}) = x_2 - 40 - y_2 \le 0 \\
& g_8(\vec{x},\vec{y}) = -x_3 - 17 + y_3 \le 0 \\
& g_9(\vec{x},\vec{y}) = x_3 - 25 - y_3 \le 0
\end{aligned}
\tag{22}
$$

where $\vec{x}$ and $\vec{y}$ are continuous and binary design variables, respectively.

For the wind farm layout optimization problem, two objectives are included: (i) minimizing the land area per MW installed (LAMI) and (ii) minimizing the wind farm Capacity Factor (CF).

TABLE 5: User-defined parameters for constrained bi-objective MINLP and multi-objective wind farm layout optimization

Parameter	constrained bi-objective MINLP	multi-objective wind farm layout optimization
Population size	100	$20m$
Elite set size	40	20
Generation	300	500
W	0.5	0.5
C_1	1.5	1.5
C_2	1.5	1.5
λ	0.1	0.1
γ_{c0}	1.0	1.5
γ_{min}	$1e-6$	$1e-8$
γ_{d0}	0.5	0.6
p_0	0.25	0.4
$\tilde{\varepsilon}$	$1.0e-5$	$1.0e-4$
L	10	20

This optimization problem is formulated as

$$
\begin{aligned}
& \min\, [CF(V^{2N},T), A(V^{2N})] \\
& V^{2N} = \{x_1, x_2, \ldots, x_N, y_1, y_2, \ldots, y_N\} \\
& T \in \{1, 2, \ldots, 10\} \\
\textit{subject to} \quad & \\
& g_1(V^{2N},T) \ge D_i^{T_i} + D_j^{T_j} \\
& i, j = 1, 2, \ldots, 10 \\
& g_2[V^{2N}, T, A(V^{2N})] \ge \hat{CF}
\end{aligned}
\tag{23}
$$

where N is the number of turbines, here $N = 25$; V^{2N} represents the design vector of turbine coordinates; and T is an integer design vector representing the turbine type. In this work, 10 candidate turbine types are selected.

This optimization also includes two constraints. $g_1(V^{2N},T)$ represents the inner-turbine spacing constraint, which indicates that the spacing between any two turbines (Turbine-i and Turbine-j) must be no shorter than the sum of their rotor diameters, $D_i^{T_i} + D_j^{T_j}$. Constraint $g_2[V^{2N},T,A(V^{2N})]$ is the predicted CF for a candidate solution of turbine location, turbine type, and land area. For the purpose of practical considerations, the predicted CF must be no lower than a referenced value $\hat{CF}$. The capacity factor of a wind farm is computed using the power generation model incorporated in the Unrestricted Wind Farm Layout Optimization framework [25]. This model quantified the power generation of a wind farm as a function of incoming wind conditions, turbine features, and location of turbines. For the sake of simplicity, a single wind direction scenario is considered. Additionally, the land area is determined using the Layout-based land usage, by which the land area for any given layout of turbines can be calculated without pre-defining farm boundaries [26].

A final set of Pareto solutions is generated by applying the Pareto filter to all the non-dominated solutions obtained after 10 runs. Fig. 4 shows the results obtained by MOMDPSO for the constrained bi-objective MINLP problem. In Fig. 4(a) the solutions are showed in the frame as the cited reference [24]. However, the resulted presented in Fig. 4(b) have more solutions as indicated by square box.

Fig. 5 shows the results obtained by MOMDPSO for the multi-objective wind farm layout problem for 25 turbines. Since the single wind direction scenario is considered, the variation of CF is relatively small. However, we obtained 20 non-dominated solutions in this range within a relatively large range of land usage.

CONCLUSION

In this paper, we developed a new MOMDPSO by making fundamental advancements to the previously developed single-objective MDPSO algorithm. This new MOMDPSO algorithm is capable of solving highly constrained, highly nonlinear, high dimensional, and multi-modal problems with different types of design variables. Two important modifications are made to extend the MDPSO algorithm to a multi-objective optimizer:

i by using the non-dominated local and global sets, we successfully turned the MDPSO as a multi-objective optimizer; while the basic features of MDPSO algorithm can be well preserved, including easy-implementation, fast convergence, diversity preservation, and mixed-discrete variable handling.

i the diversity preservation is modified to be applied for multi-objective optimization problems by defining the fractional domain near the extreme solutions in the intermediate Pareto frontier.

The application of crowding distance to the non-dominated local set enables the algorithm to maintain a well diversity from the local search level; while the non-dominated global set provides multiple global leaders for the entire swarm to find the Pareto frontier. The modified diversity preservation technique can avoid the pre-mature stagnation and helps the swarm to explore the full Pareto frontier. A series of unconstrained continuous, constrained continuous multi-objective test functions, one constrained bi-objective MINLP problem, and a multi-objective wind farm layout optimization problem were used to evaluate the performance of MOMDPSO. The comparison of the results with those obtained by NSGA-II will be provided in the final paper. Future work will include the test problems with three objectives. In addition, more complex problems will be tested to explore the potential of MOMDPSO algorithm.

REFERENCES

[1] Zitzler, E., 1999. "Evolutionary algorithms for multiobjective optimization: Methods and applications". PhD thesis, Swiss Federal Institute of Technology Zürich, Zürich, Switzerland, November.

[2] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., 2002. "A fast and elitist multiobjective genetic algorithm: NSGA-II". *IEEE Transactions on Evolutionary Computation,* ***6***(2), pp. 182–197.

[3] Zitzler, E., and Thiele, L., 1998. An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. Tech. Rep. 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zürich, Switzerland.

[4] Zitzler, E., Laumanns, M., and Thiele, L., 2001. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Tech. Rep. 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland.

[5] Laumanns, M., Rudolph, G., and Schwefel, H.-P., 1998. "A spatial predator-prey approach to multi-objective optimization: A preliminary study". *Parallel Problem Solving from Nature – PPSN V,* ***1498***, pp. 241–249.

[6] Kennedy, J., and Eberhart, R., 1995. "Particle swarm optimization". In IEEE International Conference on Neural Networks IV, IEEE, pp. 1942–1948.

[7] Angeline, P. J., 1998. "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences". *Evolutionary Programming VII,* ***1447***, pp. 601–610.

[8] Xu, X., and Eberhart, R., 2002. "Multiobjective optimization using dynamic neighborhood particle swarm optimization". In Evolutionary Computation 2002 (CEC '02), Vol. 2, IEEE, pp. 1677 – 1681.

[9] Xu, X., and Eberhart, R., 2003. "Particle swarm with extended memory for multiobjective optimization". In Swarm Intelligence Symposium 2003 (SIS '03), IEEE, pp. 193–197.

[10] Parsopoulos, K. E., Tasoulis, D. K., Vrahatis, M. N., and Words, K., 2004. "Multiobjective optimization using parallel vector evaluated particle swarm optimization". In In Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004, ACTA Press, pp. 823–828.

[11] Coello, C. A. C., Pulido, G. T., and Lechuga, M. S., 2004. "Handling multiple objectives with particle swarm optimization". *Evolutionary Computation IEEE Transactions,* ***8***(3), pp. 256–279.

[12] Abido, M. A., 2010. "Multiobjective particle swarm optimization with nondominated local and global sets". *Nat Comput,* ***9***, pp. 747–766.

[13] Lalwani, S., Singhal, S., Kumar, R., and Gupta, N., 2013. "A comprehensive survey: Applications of multi-objective particle swarm optimization (mopso) algorithm". *Transactions on Combinatories,* ***2***(1), pp. 39–101.

[14] Chowdhury, S., Tong, W., Messac, A., and Zhang, J. "A mixed-discrete particle swarm optimization algorithm with explicit diversity-preservation".

[15] Banks, A., Vincent, J., and Anyakoha, C., 2008. "A review of particle swarm optimization. part ii: Hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications". pp. 109–124.

[16] Deb, K., and Jain, S., 2002. "Running performance metrics for evolutionary multi-objective optimization". In 4th Asia-Pacific Conference Simulated Evolutionary Learning, N. T. University, ed., pp. 13–20.

[17] Okabe, T., Jin, Y., and Sendhoff, B., 2003. "A critical survey of performance indices for multi-objective optimisation". In Proc. of 2003 Congress on Evolutionary Computation, IEEE Press, pp. 878–885.

[18] Zitzler, E., and Thiele, L., 2000. "Comparison of multiobjective evolutionary algorithms: empirical results". *Evolutionary Computation,* ***8***(2), pp. 173–195.

[19] Fonseca, C. M., and Fleming, P. J., 1995. "An overview of evolutionary algorithms in multiobjective optimization". *Evolutionary Computation,* ***3***, March, pp. 1–16.

[20] Coello, C. A. C., Veldhuizen, D. A. V., and Lamont, G. B., 2002. *Evolutionary Algorithms for Solving Multi-objective Problems*. Kluwer Academic Publishers.

[21] Deb, K., 2002. *Multi-objective Optimization Using Evolutionary Algorithms,*. Wiley & Sons, Chichester, UK.

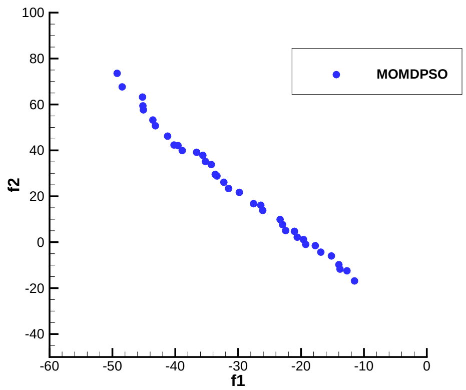
[22] Binh, T. T., and Korn, U., 1997. "Mobes: a multiobjective evolution strategy for constrained optimization problems". In Proc. of The Third International Conference on Genetic Algorithms (Mendel 97), pp. 176–182.

[23] Osyczka, A., and Kundu, S., 1995. "A new method to solve generalized multicriteria optimization problems using the simple genetic algorithm". *Structural Optimization,* ***10***, pp. 94–99.

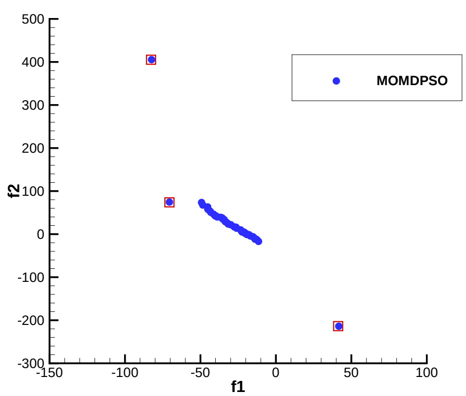
[24] Dimkou, T., and Papalexandri, K., 1998. "A parametric optimization approach for multiobjective engineering problems involving discrete decisions". *Computers and Chemical Engineering,* ***22***(1), March, pp. S951–S954.

[25] Chowdhury, S., Zhang, J., Messac, A., and Castillo, L., 2012. "Unrestricted wind farm layout optimization (uwflo): Investigating key factors influencing the maximum power generation". *Renewable Energy,* ***38***(1), February, pp. 16–30.

[26] Tong, W., Chowdhury, S., and Messac, A., 2014. "A consolidated visualization of wind farm energy production potential and optimal land shapes under different land area and nameplate capacity decisions". In AIAA Science and Technology Forum and Exposition (SciTech2014), AIAA.



(a) MINLP shown in referenced frame



(b) MINLP in full size

FIGURE 4: MOMDPSO results for standard unconstrained continuous test functions

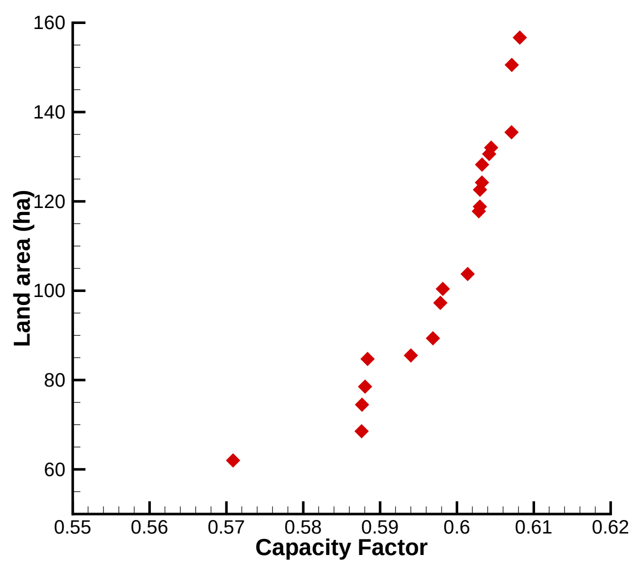


FIGURE 5: The variation of predicted capacity factor with land usage

TABLE 2: Standard unconstrained continuous test functions

Test function	m	Upper & lower bounds	Objective function	Analytical solution
ZDT1	30	$x_i \in [0,1]$	$f_1 = x_1$ $g = 1 + 9\sum_{i=2}^{m} \frac{x_i}{m-1}, h = 1 - \sqrt{\frac{f_1}{g}}$ $f_2 = h \cdot g$	Set $g = 1$
ZDT2	30	$x_i \in [0,1]$	$f_1 = x_1$ $g = 1 + 9\sum_{i=2}^{m} \frac{x_i}{m-1}, h = 1 - \left(\frac{f_1}{g}\right)^2$ $f_2 = h \cdot g$	Set $g = 1$
ZDT3	30	$x_i \in [0,1]$	$f_1 = x_1$ $g = 1 + 9\sum_{i=2}^{m} \frac{x_i}{m-1}$ $h = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f_1)$ $f_2 = h \cdot g$	Set $g = 1$
ZDT4	10	$x_i \in [0,1]$, $x_i \in [-5,5]$, $i = 2,3,\dots,m$	$f_1 = x_1$ $g = 1 + 10(m-1) + \sum_{i=2}^{m} x_i^2 - 10\cos(4\pi x_i)$ $h = h = 1 - \sqrt{\frac{f_1}{g}}$ $f_2 = h \cdot g$	Set $g = 1$
ZDT5	11	$x_i \in [0,1]$, 30 bit resolution $x_i \in [-5,5]$, $i = 2,3,\dots,m$ 5 bit resolution	$f_1 = 1 + u(x_1)$ $u(x_i) =$ number of “1”in the binary form of x_i $g = \sum_{i=2}^{m} v(u(x_i)), h = \frac{1}{f_1}$ $v(x_i) = \begin{cases} 2 + u(x_i) & if u(x_i) < 5 \\ 1 & if u(x_i) = 5 \end{cases}$ $f_2 = h \cdot g$	Set $g = 10$
ZDT6	10	$x_i \in [0,1]$	$f_1 = 1 - \exp[-4x_1 \sin^6(6\pi x_1)]$ $g = 1 + 9\left(\frac{\sum_{i=2}^{m} x_i}{m-1}\right)^{0.25}, h = 1 - \left(\frac{f_1}{g}\right)^2$ $f_2 = h \cdot g$	Set $g = 1$

Standard Unconstrained Continuous Test Functions (continued)

Test function	m	Upper & lower bounds	Objective function	Analytical solution
Fonseca-Fleming	3	$x_i \in [-4,4]$	$f_1 = 1-\exp\left[-\sum_{i=2}^{m}\left(x_i-\frac{1}{\sqrt{m}}\right)^2\right]$ $f_2 = 1-\exp\left[-\sum_{i=2}^{m}\left(x_i+\frac{1}{\sqrt{m}}\right)^2\right]$	Set $f_2 = 1-\exp\left[-\left(2-\sqrt{-\ln(1-f_1)}\right)^2\right]$
Coello	2	$x_i \in [0,1]$	$f_1 = x_1$ $f_2 = (1+10x_2)\left[1-\left(\frac{x_1}{1+10x_2}\right)^2 - \frac{x_1}{1+10x_2}\sin(8\pi x_1)\right]$	Set $f_2 = 1-f_1^2-f_1\sin(8\pi f_1)$

TABLE 4: Constrained Continuous Test Function

Test function	m	Upper & lower bounds	Objective function	Constraints
CONSTR [21]	2	$x_1 \in [0.1,1], x_2 \in [0,5]$	$f_1 = x_1$ $f_2 = \frac{1+x_2}{x_1}$	$9x_1+x_2-6 \geq 0$ $9x_1-x_2-1 \geq 0$
SRN [21]	2	$x_i \in [-20,20]$	$f_1 = (x_1-2)^2+(x_2-1)^2+2$ $f_2 = 9x_1-(x_2-1)^2$	$-x_1^2-x_2^2+225 \geq 0$ $-x_1+3x_2-10 \geq 0$
TNK [21]	2	$x_i \in [0,\pi]$	$f_1 = x_1$ $f_2 = x_2$	$x_1^2+x_2^2-1 - 0.1\cos\left[16\arctan\left(\frac{x_1}{x_2}\right)\right] \geq 0$ $-(x_1-0.5)^2-(x_2-0.5)^2+0.5 \geq 0$
Binh-Korn [22]	2	$x_1 \in [0,5], x_2 \in [0,3]$	$f_1 = 4x_1^2+4x_2^2$ $f_2 = (x_1-5)^2+(x_2-5)^2$	$-(x_1-5)^2-x_2^2+25 \geq 0$ $(x_1-8)^2-(x_2+3)^2-7.7 \geq 0$
Osyczka-Kundu [23]	6	$x_1 \in [0,10], x_2 \in [0,10]$ $x_3 \in [1,5], x_4 \in [0,6]$ $x_5 \in [1,5], x_1 \in [0,10]$	$f_1 = -25(x_1-2)^2-(x_2-2)^2 - (x_3-1)^2-(x_4-4)^2 - (x_5-1)^2$ $f_2 = \sum_{i=1}^{m} x_i^2$	$x_1+x_2-2 \geq 0$ $-x_1-x_2+6 \geq 0$ $x_1-x_2+2 \geq 0$ $-x_1+3x_2+2 \geq 0$ $-(x_3-3)^2-x_4+4 \geq 0$ $(x_5-3)^2+x_6-4 \geq 0$