



# A novel hybrid hypervolume indicator and reference vector adaptation strategies based evolutionary algorithm for many-objective optimization

Gaurav Dhiman<sup>1</sup> · Mukesh Soni<sup>2</sup> · Hari Mohan Pandey<sup>3</sup> · Adam Slowik<sup>4</sup> · Harsimran Kaur<sup>1</sup>

Received: 10 March 2019 / Accepted: 8 February 2020 / Published online: 25 February 2020  
© Springer-Verlag London Ltd., part of Springer Nature 2020

## Abstract

A novel hybrid many-objective evolutionary algorithm called Reference Vector Guided Evolutionary Algorithm based on hypervolume indicator (H-RVEA) is proposed in this paper. The reference vectors are used in a number of sub-problems to decompose the optimization problem. An adaptation strategy is used in the proposed algorithm to adjust the reference vector distribution. The proposed algorithm is compared over well-known benchmark test functions with five state-of-the-art evolutionary algorithms. The results show H-RVEA's superior performance in terms of the inverted generational distance and hypervolume performance measures than the competitor algorithms. The suggested algorithm's computational complexity is also analysed. The statistical tests are carried out to demonstrate the statistical significance of the proposed algorithm. In order to demonstrate its efficiency, H-RVEA is also applied to solve two real-life constrained many-objective optimization problems. The experimental results indicate that the proposed algorithm can solve the many-objective real-life problems. Note that the source codes of the proposed technique are available at <http://dhimangaurav.com/>.

**Keywords** Many-objective optimization · Hypervolume estimation algorithm · Reference vector guided evolutionary algorithm · Constrained optimization · Pareto optimality

## 1 Introduction

Researchers have developed and designed the number of new evolutionary multi-objective algorithms over the past few years to solve problems with more than two objectives [1–3]. multi-objective evolutionary algorithms (MOEAs) are a type of population based heuristics [4–9] that, during simulation runs, can get a group of solutions. Nonetheless, some real-life issues, such as vehicle engine tuning problems [10], water distribution systems [11], and land use management problems [12] and so on [13–22] also require a large

number of objectives. MOEAs also face some challenges in tackling these multiple objectives. The selection methods are the reason behind the failure of most MOEAs. To converge the population to the Pareto front, such methods are useful. When the number of objectives increases, it is difficult to maintain the diversity, as the solutions are very sparsely allocated within the objective search space.

Non-dominated Sorting Genetic Algorithm (NSGA-II) [23] and Strength Pareto Evolutionary Algorithm 2 (SPEA2) [24] are the most popular multi-objective evolutionary algorithms. Both of these algorithms use dominance based selection strategy that fails to solve problems with more than three objectives.

To address this issue, researchers have developed new optimization algorithms to deal with a large number of objectives [25–30], also referred to as the Many-Objective Problems (MaOPs). The main challenges in MaOPs are:

- Pareto's front visualization in a search space requires special procedures [31].
- The fact of the dominance resistance (DR), in which the gap between the solutions is caused by an increase in the amount of non-dominated solutions [32–34].

✉ Gaurav Dhiman  
gdhiman0001@gmail.com

<sup>1</sup> Department of Computer Science, Government Bikram College of Commerce, Patiala, Punjab 147001, India

<sup>2</sup> Department of Computer Engineering, Smt. S. R. Patel Engineering College, Unjha, Gujarat 384170, India

<sup>3</sup> Department of Computer Science, Edge Hill University, Ormskirk, UK

<sup>4</sup> Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland

- The number of solutions has increased exponentially to define the multi-dimensional Pareto front problems [35, 36].

Most of the MaOPs concentrate on maintaining diversity and enhancing convergence. Such algorithms are not about using preferences for many-objective problems. This is because the researchers may be interested in optimal Pareto solutions and less computational effort [37–44] to achieve a demonstrative subset of the Pareto front [45].

There are several evolutionary algorithms that have been developed to improve the effectiveness of algorithms to solve MaOPs [46–48]. Reference Vector Guided Evolutionary Algorithm (RVEA) [49] is a recently developed decomposition based approach that can produce optimal solutions using reference points for a Pareto subset of solutions. Such points of reference are useful for greater convergence. Despite this, the most common algorithm based on hypervolume indicator is the Hypervolume Estimation Algorithm (HypE) [50] which uses the Monte Carlo simulation to estimate the exact hypervolume values and maintain diversity.

The main contribution of this work is the development of a novel hybrid evolutionary algorithm using the principles of RVEA and HypE algorithms. The algorithm is called Reference Vector Guided Evolutionary Algorithm based on the Hypervolume indicator (H-RVEA). H-RVEA's four main strategies are mating selection, variation, environmental selection, and adaptation strategy for reference vectors. The mating selection strategy is used to exchange information between individuals, and to select from the current population the most promising optimal solution. The variation method utilize operators of recombination and mutation to generate new offspring. It is the task of the environmental selection strategy to break and merge the strategies into non-dominated fronts. The Adaptation technique for the reference vector is used to enhance the search process and to decompose the problem into a sub-problem.

H-RVEA's performance is tested on well-known benchmark test problems. The two performance measures, namely Inverted Generational Distance (IGD) [51] and Hypervolume (HV) [52], have been used to assess the algorithms. The results have been compared with five approaches such as Reference Vector Guided Evolutionary Algorithm (RVEA) [49], Hypervolume Estimation Algorithm (HypE) [50], Non-dominated Sorting Genetic Algorithm (NSGA-III) [53], Many-Objective Evolutionary Algorithm based on Dominance and Decomposition (MOEA/DD) [54], and Approximation-guided Evolutionary Algorithm (AGE-II) [55]. Therefore, to illustrate its usefulness, H-RVEA has been applied to two real-life many-objective problems.

The rest of the paper has the following structure: Sect. 2 represents the basics of many-objective problems and two

many-objective evolutionary algorithms. Section 3 presents the motivation and brief description of proposed H-RVEA algorithm. Results and discussions of experiments are given in Sect. 4. Section 5 presents the performance of H-RVEA on two real-life problems followed by conclusions and future works in Sect. 6.

## 2 Background

In this section, the basic concepts of Multi-Objective Problems (MOPs) are presented. Then, the brief descriptions of RVEA and HypE algorithms are presented.

### 2.1 Basic concepts

Multi-Objective Optimization Problems (MOPs) involve more than one objective which is to be optimized. The MOPs can be formulated as follows:

$$\begin{aligned} &\text{Minimize } f(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}), \dots, f_n(\vec{x})) \\ &\text{Subject to: } \vec{x} \in R \end{aligned} \quad (1)$$

where  $n$  defines the number of objective functions and  $f(\vec{x})$  defines the objective function. However, the feasible space  $R$  is a subset of decision space ( $D$ ), which consists the decision vectors  $\vec{x} = (\vec{x}_1, \vec{x}_2, \dots, \vec{x}_m)^T$ .

### 2.2 Reference Vector Guided Evolutionary Algorithm (RVEA)

Reference Vector Guided Evolutionary Algorithm (RVEA) is an evolutionary algorithm for many-objective optimization problems [49]. The main components of the RVEA algorithm are offspring production, guided selection of the reference vector, and strategies for adapting the reference vector. The genetic operators, as used in NSGA-III and HypE, are employed to build the offspring population. The guided selection method for the reference vector consists of four steps, namely objective value translation, population partitions, Angle Penalized Distance (APD) calculation and elitism selection [49]. RVEA uses their strategies of elitism and offspring as similar to the NSGA-III algorithm. Its selection strategy adopts a set of reference vectors. Angle Penalized Distance (APD) is used to align the convergence and diversity properly. An adaptive technique is used to modify the reference vectors to ensure a consistent distribution of solutions within the objective space. The preferential articulation method based on the reference vector is used to get optimal solutions in a chosen area uniformly distributed by Pareto. The RVEA is described in the Algorithm 1 [49].

**Algorithm 1** RVEA

---

**Input:** Set of reference vectors  $V_0 \leftarrow \{v_1, v_2, v_3, \dots, v_N\}$ , Maximum number of iterations  $Max_{iteration}$   
**Output:** Non-dominated solutions  $P$

- 1: **procedure** RVEA
- 2:   Initialize the population  $P$  and set counter  $c \leftarrow 0$
- 3:   **while**  $c < Max_{iteration}$  **do**
- 4:     Create offspring  $O_c$  from parent population  $P_c$
- 5:     Combine the parent and offspring populations, ( $M_c \leftarrow P_c \cup O_c$ )
- 6:      $P_{c+1} \leftarrow$  ReferenceVectorGuidedSelection ( $c, P_c, V_c$ )
- 7:      $V_{c+1} \leftarrow$  ReferenceVectorAdaptation ( $c, P_{c+1}, V_c, V_0$ )
- 8:      $c \leftarrow c + 1$
- 9:   **end while**
- 10: **return**  $P$
- 11: **end procedure**

---

**2.3 Hypervolume Estimation Algorithm (HypE)**

Bader and Zitzler [50] have proposed a MOEA based rapid hypervolume indicator, namely Hypervolume Estimation Algorithm (HypE). The main concept behind HypE is the rating by the hypervolume of solutions achieved. HypE uses a generalized technique of granting fitness. The hypervolume values are calculated using Monte-Carlo simulation. Mating selection, variation, and environmental selection are key components of HypE. Selection of binary tournament is used to select offspring. The variation employed operators of recombination and mutation to produce offspring. Throughout environmental selection, the populations of parents and children are integrated and decomposed into non-dominated fronts, and the new population is added with the best fronts. The HypE is described in Algorithm 2 [50].

**3.1 Motivation**

The researchers have pointed to difficulties of convergence and diversity with more than four objectives for real-life problems. Therefore, an algorithm must be established which maintains the convergence and diversity. In this paper the hypervolume indicator is used to preserve the diversity. The reason for preferring this measure over others is:

1. Hypervolume indicator removes the lack of selection pressure.
2. The value of this indicator is optimized directly, without the need for niching, which helps to maintain the diversity.

**Algorithm 2** HypE

---

**Input:** Number of sampling points  $M$ , Reference set  $R$ , Population size  $N$ , Maximum number of iterations  $Max_{iteration}$   
**Output:** Non-dominated solutions  $P$

- 1: **procedure** HypE
- 2:   Initialize the randomly generated population  $P$  and set counter  $c \leftarrow 0$
- 3:   **while**  $c < Max_{iteration}$  **do**
- 4:      $P' \leftarrow$  matingSelection ( $P, R, M, N$ )
- 5:      $P'' \leftarrow$  variation ( $P', N$ )
- 6:      $P \leftarrow$  environmentalSelection ( $P \cup P'', R, N, M$ )
- 7:      $c \leftarrow c + 1$
- 8:   **end while**
- 9:   **return**  $P$
- 10: **end procedure**

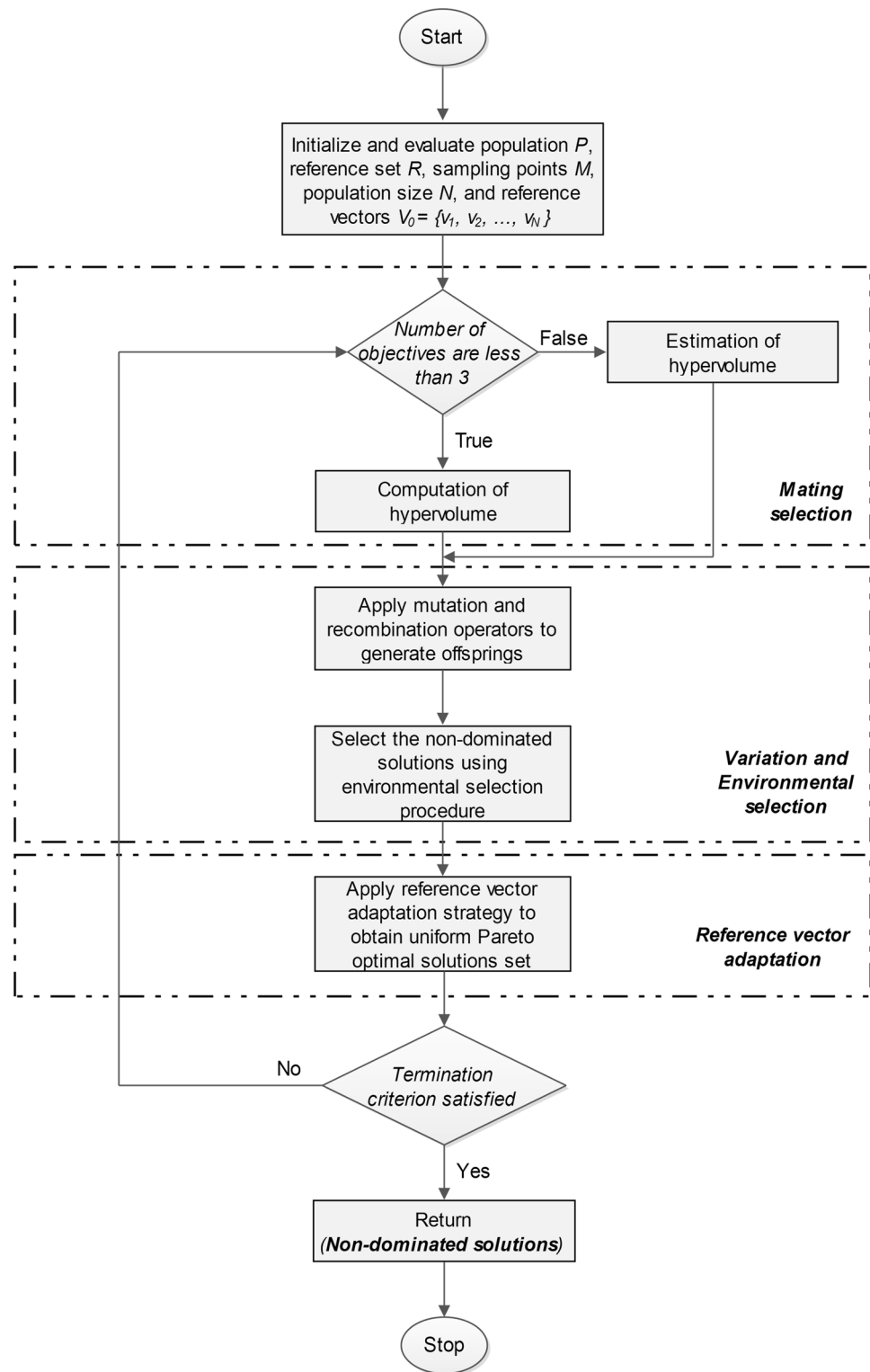
---

**3 Proposed H-RVEA Algorithm**

This section describes the motivation and overall procedure of proposed H-RVEA algorithm.

3. This indicator ensures that any approximation set with a high quality value for a specific problem includes all optimal objective vectors from Pareto [56].

**Fig. 1** Flowchart of the proposed H-RVEA algorithm



Hypervolume calculation, however, requires a high computational effort. We have used fast HypE algorithm to solve this problem. HypE suffers from the overheads of having the information needed. We have used reference vector adaptation strategy to preserve the knowledge. The reference vector

adaptation strategy is taken from RVEA algorithm. Therefore, a novel hybrid algorithm is proposed which employs both HypE and RVEA features.

### 3.2 Proposed algorithm

The intend to propose the H-RVEA is to solve problems that have a large number of objectives. The basic procedure of proposed H-RVEA is mentioned in Algorithm 3. The H-RVEA is inspired by RVEA and HypE algorithms (see Fig. 1). H-RVEA consists of four main components. These are mating selection (Algorithm 4), variation, environmental selection (Algorithm 8), and reference vector adaptation strategy (Algorithm 9). Initially, it starts with the randomly generated population  $P$  and  $N$  number of individuals. The mating selection procedure is used to select the promising solutions in a search space. The variation uses the mutation and recombination operators to generate the  $N$  number of offspring. The environmental selection procedure is used for the selection of the best optimal solution. Finally, the reference vector adaptation strategy is performed to obtain a uniformly distributed Pareto optimal solutions set.

multi-set containing a pair  $(a, w_a)$ , where  $w_a$  is fitness value and  $a \in P$ . The Slicing procedure mentioned in Algorithm 6, recursively cut the dominated space into hyper-rectangles and returns a partial fitness assignment  $\alpha'$ . The scanning process is performed at each recursive level  $l$  with  $u'$ , where  $u'$  represents the current scan position. The scan positions are included in vector  $(s_1, s_2, \dots, s_n)$  which contains all the dimensions. The partial volume PV is updated and reference points are filtered out during whole the recursive process (Line 2 and 3). Example 1 is used to describe the working of Computehypervolume procedure.

The Estimatehypervolume procedure returns a fitness assignment  $\alpha$  corresponding to the estimated hypervolume as mentioned in Algorithm 7. The first step is to determine the sampling box  $S$  which is a superset of approximate hypervolumes. Thereafter, the volume of sampling space  $S$  is computed (Line 6). For sampling process the  $M$  objective vectors

---

#### Algorithm 3 H-RVEA

---

**Input:** Population  $P$ , Reference set  $R$ , Number of sampling points  $M$ , Population size  $N$ , Set of reference vectors  $V_0 \leftarrow \{v_1, v_2, v_3, \dots, v_N\}$ , Control parameter  $F_c$ , Maximum number of iterations  $Max_{iteration}$

**Output:** Non-dominated solutions  $P$

```

1: procedure H-RVEA
2:   Initialize the randomly generated population  $P$ 
3:   Set counter  $c \leftarrow 0$ 
4:   while  $c < Max_{iteration}$  do
5:      $P' \leftarrow$  Matingselection ( $P, R, M, N$ )
6:      $P'' \leftarrow$  Variation ( $P', N$ )
7:      $P \leftarrow$  Environmentalselection ( $P \cup P'', R, N, M$ )
8:      $V_{c+1} \leftarrow$  ReferenceVectorAdaptation ( $c, P, V_c, V_0, F_c$ )
9:      $c \leftarrow c + 1$ 
10:  end while
11:  return  $P$ 
12: end procedure

```

---

#### 3.2.1 Mating selection

The mating selection is a procedure for exchanging the information between individuals and to select the promising optimal solutions from the current population. The mating selection strategy of proposed H-RVEA is described in Algorithm 4. When the number of objectives is less than 3 ( $\leq 3$ ), then the computation of hypervolume is done through Computehypervolume procedure which is mentioned in Algorithm 5. Otherwise, the computation of hypervolume is done through the Estimatehypervolume procedure which is mentioned in Algorithm 7.

The Computehypervolume procedure uses the recursive function Slicing and returns a fitness assignment  $\alpha$  and a

$(y_1, y_2, \dots, y_M)$  are selected from  $S$  randomly. Based on the sampling point an estimation of hypervolume is done. The hypervolumes values are then updated during procedure call.

**Example 1** (Hypervolume computation) Consider the population contains four solutions i.e.,  $w, x, y, z$  with objective vectors  $f(w) = (-9, -2, -1), f(x) = (-7, 0, -7), f(y) = (-5, -7, -9), f(z) = (-3, -4, -10)$ , fitness parameter  $f_p = 2$ , reference points  $r = (-1, 0, 0)$ , and  $y = (-1, -2, -3)$ .

Firstly, call the Slicing procedure in which the value of  $l$  is set to 3 and  $U$  contains all objective vectors and reference points. According to the third vector components, the  $U$  with its elements are sorted in ascending order.

$$U = \begin{cases} f(z) = (-3, -4, -10) \\ f(y) = (-5, -7, -9) \\ f(x) = (-7, 0, -7) \\ y = (-1, -2, -3) \\ f(w) = (-9, -2, -1) \\ r = (-1, 0, 0) \end{cases}$$

The variable  $u'$  assigned the third vector value of these elements to  $f_3(z) = -10, f_3(y) = -9, f_3(x) = -7, y_3 = -3, f_3(w) = -1, r_3 = 0$ . During iteration process,  $U$  is reduced and assign  $u' = f_3(x) = -7$  and update the value of  $PV' = 1 \times (-3 - (-7)) = 4$  with scan positions  $(s_1, s_2, s_3) = (\infty, \infty, -7)$ . Now, these values are passed to next recursion level  $l = 2$  and process the Line 13 in Slicing procedure, where  $U$  is initialized.

In the next level, the sorting is performed according to second dimension of these elements.

$$U = \begin{cases} f(y) = (-5, -7, -9) \\ f(z) = (-3, -4, -10) \\ y = (-1, -2, -3) \\ f(x) = (-7, 0, -7) \\ r = (-1, 0, 0) \end{cases}$$

Therefore, variable  $u'$  assigned the second vector value to  $f_2(y) = -7, f_2(z) = -4, y_2 = -2, f_2(x) = 0, r_2 = 0$ .

$U$  is further reduced and assign  $u' = f_2(x) = 0$ . The value of  $PV' = 1 \times 4 \times (0 - 0) = 0$  and scan positions  $(s_1, s_2, s_3) = (\infty, 0, -7)$ . For the next recursion level  $l = 1$  in which the sorting is performed according to first vector, then the  $U$  is computed as follows.

$$U = \begin{cases} f(x) = (-7, 0, -7) \\ f(y) = (-5, -7, -9) \\ f(z) = (-3, -4, -10) \\ r = (-1, 0, 0) \end{cases}$$

In the second iteration, variable  $u'$  assigned the first vector value to  $f_1(x) = -7, f_1(y) = -5, f_1(z) = -3, r_1 = -1$  and holds  $u' = f_1(y) = -5$ . The value of  $PV' = 1 \times 4 \times 0 \times (-3 - (-5)) = 0$  and scan positions  $(s_1, s_2, s_3) = (-5, 0, -7)$ . Once the recursion level is reached at level 0 ( $l = 0$ ), the  $\beta$  is computed with  $N$  number of population size, fitness parameter  $f_p = 2$ , and hyperrectangle  $RS = 0$ . Therefore, the whole procedure is applied to all slices to identifies all hyperrectangles of the search space.

---

#### Algorithm 4 Mating selection

---

**Input:** Population  $P$ , Reference set  $R$ , Number of sampling points  $M$ , Population size  $N$

**Output:**  $T$

```

1: procedure MATINGSELECTION ( $P, R, M, N$ )
2:   if  $n \leq 3$  then
3:      $\alpha \leftarrow$  Computehypervolume( $P, R, N$ )
4:   else
5:      $\alpha \leftarrow$  Estimatehypervolume( $P, R, N, M$ )
6:   end if
7:    $T \leftarrow \phi$ 
8:   while  $|T| < N$  do
9:     Choose  $(a, w_a), (b, w_b) \in \alpha$  randomly
10:    if  $w_a > w_b$  then
11:       $T \leftarrow T \cup \{a\}$ 
12:    else
13:       $T \leftarrow T \cup \{b\}$ 
14:    end if
15:  end while
16:  return  $T$ 
17: end procedure

```

---

**Algorithm 5** Compute hypervolume**Input:** Population  $P$ , Reference set  $R$ , Fitness parameter  $f_p \in N$ , Number of objectives  $n$ **Output:**  $\alpha$  and  $(a, w_a)$ 

- 1: **procedure** COMPUTEHYPERVOLUME ( $P, R, f_p$ )
- 2:      $\alpha \leftarrow \bigcup_{a \in P} \{(a, 0)\}$
- 3:     **return**  $Slicing(\alpha, R, f_p, n, 1, (\infty, \infty, \dots, \infty))$
- 4: **end procedure**

**Algorithm 6** Slicing**Input:** Population  $P$ , Reference set  $R$ , Fitness parameter  $f_p \in N$ , Fitness assignment  $\alpha$ , Recursion level  $l$ , Partial volume $PV \in R$ , Scan positions  $(s_1, s_2, \dots, s_n) \in R^n$ **Output:** Partial fitness assignment  $\alpha'$ 

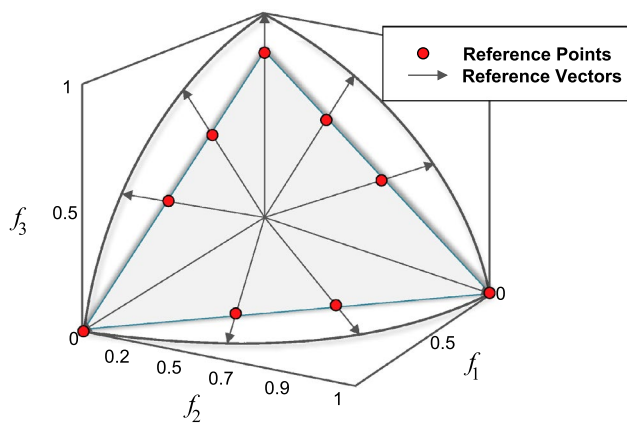
- 1: **procedure** SLICING ( $\alpha, R, f_p, l, PV, (s_1, s_2, \dots, s_n)$ )
- 2:      $RS \leftarrow \bigcup_{(a,w) \in \alpha, \forall i < j \leq n : f_j(a) \leq z_j\{f(a)\}}$
- 3:      $RP \leftarrow \bigcup_{(r_1, r_2, \dots, r_n) \in R, \forall i < j \leq n : r_j \geq z_j\{(r_1, \dots, r_n)\}}$
- 4:     **if**  $l = 0$  and  $RP \neq \phi$  **then**
- 5:          $\beta \leftarrow \prod_{t=1}^{|RS|-1} (f_p - t) / (|\alpha| - t)$
- 6:          $\alpha' \leftarrow \phi$
- 7:         **for all**  $(a, w) \in \alpha$  **do**
- 8:              $\alpha' \leftarrow \alpha' \cup \left( a, w + \frac{\beta}{|RS|} \times PV \right)$
- 9:         **end for**
- 10:     **else if**  $l > 0$  **then**
- 11:          $\alpha' \leftarrow \alpha$
- 12:          $U \leftarrow RS \cup RP$
- 13:         **while**  $U \neq \phi$  **do**
- 14:              $u' \leftarrow \min_{(u_1, u_2, \dots, u_n) \in U} u_l$
- 15:              $U' \leftarrow \{(u_1, u_2, \dots, u_n) \in U \mid u_l > u'\}$
- 16:             **if**  $U' \neq \phi$  **then**
- 17:                  $PV' \leftarrow PV \times ((\min_{(u'_1, u'_2, \dots, u'_n) \in U'} u'_l) - u')$
- 18:                  $\alpha' \leftarrow Slicing(\alpha', R, f_p, l - 1, PV', (s_1, s_2, \dots, s_{l-1}, u', s_{l+1}, s_{l+2}, \dots, s_n))$
- 19:             **end if**
- 20:              $U \leftarrow U'$
- 21:         **end while**
- 22:     **end if**
- 23:     **return**  $\alpha'$
- 24: **end procedure**

**Algorithm 7** Estimate hypervolume**Input:** Population  $P$ , Reference set  $R$ , Fitness parameter  $f_p$ , Sampling points  $M$ , Population size  $N$ **Output:** Fitness assignment  $\alpha$ 

```

1: procedure ESTIMATEHYPERVOLUME ( $P, R, N, M$ )
2:   for  $i = 1$  to  $n$  do
3:      $LB_i \leftarrow \min_{a \in P} f_i(a)$ 
4:      $UB_i \leftarrow \max_{(r_1, r_2, \dots, r_n) \in R} f_i$ 
5:   end for
6:    $S \leftarrow [LB_1, UB_1] \times [LB_2, UB_2] \times \dots \times [LB_n, UB_n]$ 
7:    $W \leftarrow \prod_{i=1}^n \max\{0, (UB_i - LB_i)\}$ 
8:    $\alpha \leftarrow \cup_{a \in P} \{(a, 0)\}$ 
9:   for  $k = 1$  to  $M$  do
10:    Choose  $y \in S$  randomly
11:     $RS \leftarrow \cup_{a \in P, f(a) \leq y} \{f(a)\}$ 
12:    if  $|RS| \leq f_p$  then
13:       $\beta \leftarrow \prod_{LB=1}^{|RS|-1} \frac{f_p - LB}{|P| - LB}$ 
14:       $\alpha' \leftarrow \phi$ 
15:      for all  $(a, w) \in \alpha$  do
16:         $\alpha' \leftarrow \alpha' \cup \left( a, w + \frac{\beta}{|RS|} \times \frac{W}{M} \right)$ 
17:      end for
18:       $\alpha \leftarrow \alpha'$ 
19:    end if
20:  end for
21:  return  $\alpha$ 
22: end procedure

```

**Fig. 2** Uniformly distributed reference vectors in 3D search space**3.2.2 Environmental selection**

Environmental selection is a procedure to obtain a well-distributed population and to keep it in a memory. It chooses the best optimal solutions from the previous and newly created population. The proposed H-RVEA algorithm implements the environmental selection strategy with hypervolume subset selection problem [50]. Algorithm 8 describes the environmental selection strategy for creating a new population. Firstly, the nondominated sorting approach [23, 57] is used for partitioning the union of parent and offspring into disjoint partitions. The partition that fits into the new population is further processed [50]. The fitness value of each partition is computed and remove the individual with the worst fitness assignment. The parents and offsprings are



merged into population  $P$  which truncates the non-fitted solutions. Therefore, the dominated solutions do not affect the overall value of hypervolume. Algorithm 8 is repeated until the desired size of a partition is not found and returns the generated new population  $T$ .

### 3.2.3 Reference vector adaptation strategy

The proposed H-RVEA algorithm uses reference vector adaptation [49] strategy to obtain a set of Pareto optimal solutions as shown in Fig. 2.

---

#### Algorithm 8 Environmental selection

---

**Input:** Population  $P$ , Reference set  $R$ , Number of offspring  $N$ , Number of sampling points  $M$   
**Output:** New population  $T$

```

1: procedure ENVIRONMENTALSELECTION ( $P, R, N, M$ )
2:    $P' \leftarrow P$ 
3:    $T \leftarrow \phi$ 
4:    $T' \leftarrow \phi$ 
5:   do
6:      $T \leftarrow T \cup T'$ 
7:      $T' \leftarrow \phi$ 
8:      $P'' \leftarrow \phi$ 
9:     for all  $a \in P'$  do
10:      if  $b \in P' : b \leq a \Rightarrow a \leq b$  then
11:         $T' \leftarrow T' \cup \{a\}$ 
12:      else
13:         $T'' \leftarrow T'' \cup \{a\}$ 
14:      end if
15:    end for
16:     $P' \leftarrow P''$ 
17:    while  $|T| + |T'| \geq N \vee P' = \phi$ 
18:       $TC \leftarrow |T| + |T'| - N$ 
19:      while  $TC > 0$  do
20:        if  $n \leq 3$  then
21:           $\alpha \leftarrow \text{Computehypervolume}(T', R, TC)$ 
22:        else
23:           $\alpha \leftarrow \text{Estimatehypervolume}(T', R, TC, M)$ 
24:        end if
25:         $T' \leftarrow \phi$ 
26:         $RV \leftarrow 0$ 
27:        for all  $(a, w) \in \alpha$  do
28:          if  $RV = 1$  or  $w \neq \min_{(a,w) \in \alpha} \{w\}$  then
29:             $T' \leftarrow T' \cup \{a\}$ 
30:          else
31:             $RV \leftarrow 1$ 
32:          end if
33:        end for
34:         $TC \leftarrow TC - 1$ 
35:      end while
36:       $T \leftarrow T \cup T'$ 
37:      return  $T$ 
38: end procedure

```

---

**Table 1** The IGD values obtained by proposed H-RVEA and other competitor algorithms on DTLZ test suite

F	M	Analysis	H-RVEA	RVEA	HypE	NSGA-III	MOEA/DD	AGE-II	
DTLZ1	3	Mean	<b>3.91E-05</b>	4.69E-04	1.53E-04	4.90E-04	3.20E-04	3.55E-02	
		Median	<b>4.07E-05</b>	4.99E-04	2.08E-01	1.40E-03	5.90E-04	3.95E-02	
	5	Mean	<b>2.54E-05</b>	3.11E-05	1.85E-01	5.25E-04	2.65E-04	2.34E-02	
		Median	<b>1.21E-05</b>	3.57E-05	2.20E-01	9.91E-04	2.96E-04	2.83E-02	
	10	Mean	<b>5.12E-05</b>	1.50E-04	1.50E-01	2.25E-03	1.90E-03	3.63E-02	
		Median	<b>1.11E-04</b>	6.69E-04	1.92E-01	3.53E-03	2.26E-03	3.89E-02	
	15	Mean	<b>3.07E-05</b>	1.77E-04	1.82E-01	2.95E-03	2.90E-03	1.03E-02	
		Median	<b>2.00E-04</b>	2.04E-04	2.60E-01	5.15E-03	4.25E-03	1.58E-02	
	DTLZ2	3	Mean	<b>1.00E-04</b>	1.71E-04	6.80E-02	1.35E-03	6.81E-04	2.73E-02
			Median	<b>1.03E-04</b>	1.90E-04	6.98E-02	1.42E-03	8.10E-04	2.97E-02
		5	Mean	1.03E-03	<b>1.23E-04</b>	2.80E-01	4.35E-03	1.18E-03	1.35E-02
			Median	1.91E-03	<b>3.11E-04</b>	2.92E-01	5.15E-03	1.40E-03	1.63E-02
10		Mean	1.39E-03	<b>1.35E-03</b>	6.90E-01	1.40E-02	3.30E-03	2.25E-02	
		Median	2.11E-02	<b>2.03E-03</b>	7.01E-01	1.61E-02	3.81E-03	2.69E-02	
15		Mean	<b>2.22E-04</b>	5.03E-04	6.35E-01	1.40E-02	4.60E-03	2.74E-02	
		Median	<b>4.90E-04</b>	5.95E-04	7.62E-01	1.83E-02	5.90E-03	2.95E-02	
DTLZ3		3	Mean	<b>2.71E-05</b>	1.11E-04	1.70E-02	9.80E-04	5.90E-04	2.35E-02
			Median	<b>5.10E-05</b>	2.20E-04	1.82E-02	4.11E-03	1.95E-03	2.79E-02
		5	Mean	<b>3.18E-04</b>	1.33E-02	1.19E-02	3.15E-03	6.25E-04	2.01E-02
			Median	<b>1.10E-03</b>	1.87E-02	1.93E-02	6.01E-03	1.36E-03	2.94E-02
	10	Mean	<b>1.22E-03</b>	2.01E-01	1.86E-02	8.90E-03	1.95E-03	2.38E-02	
		Median	<b>2.08E-03</b>	2.71E-01	2.93E-02	1.25E-02	2.35E-03	2.93E-02	
	15	Mean	<b>2.56E-04</b>	1.98E-01	4.35E-02	1.43E-02	5.90E-03	1.11E-02	
		Median	<b>4.45E-03</b>	2.90E-01	6.93E-02	2.20E-02	7.50E-03	2.04E-02	
	DTLZ4	3	Mean	<b>1.00E-04</b>	5.03E-04	8.49E-02	2.94E-04	1.10E-04	2.23E-02
			Median	<b>1.50E-04</b>	6.84E-04	9.40E-02	6.11E-04	1.71E-04	2.48E-02
		5	Mean	<b>5.61E-05</b>	1.00E-01	2.80E-01	9.90E-04	1.11E-04	1.04E-02
			Median	<b>6.45E-05</b>	1.69E-01	2.93E-01	1.37E-03	1.37E-04	1.54E-02
10		Mean	<b>1.00E-03</b>	1.94E-03	6.81E-01	5.80E-03	1.32E-03	2.30E-02	
		Median	<b>1.52E-03</b>	2.31E-03	6.93E-01	6.43E-03	1.70E-03	2.83E-02	
15		Mean	<b>1.22E-04</b>	1.96E-04	6.08E-01	7.87E-03	1.50E-03	3.02E-02	
		Median	<b>2.00E-04</b>	2.31E-04	6.41E-01	3.11E-02	1.90E-03	3.73E-02	

The obtained best results are in bold

Algorithm 9 mentions this strategy. These solutions are the points of intersection between each reference vector and the Pareto front. However, when the objective function values are normalized in the same range, these reference vectors won't produce uniformly distributed solutions. The range of the reference vectors will be modified as follows to eliminate this conflict:

$$v_{k+1,j} = \frac{v_{0,j} \circ (O_{k+1}^{\max} - O_{k+1}^{\min})}{\| v_{0,j} \circ (O_{k+1}^{\max} - O_{k+1}^{\min}) \|} \tag{2}$$

where  $j = 1, 2, \dots, N$ ,  $v_{k+1,j}$  defines the  $j$ th reference vector adapted for generation  $k + 1$ ,  $v_{0,j}$  represents the  $j$ th uniformly distributed reference vector.  $O_{k+1}^{\min}$  and  $O_{k+1}^{\max}$  are the minimum and maximum values of objective function, respectively. The  $\circ$  operator indicates the Hadamard product that element-wise multiply the two vectors of same size [49]. The adaptation strategy for the reference vector is capable of obtaining uniformly distributed solutions even if the objective functions are not normalized in the same range.

**Table 2** The HV values obtained by proposed H-RVEA and other competitor algorithms on DTLZ test suite

F	M	Analysis	H-RVEA	RVEA	HypE	NSGA-III	MOEA/DD	AGE-II	
DTLZ1	3	Mean	<b>8.51E-02</b>	4.69E-05	4.98E-03	3.15E-04	4.81E-02	3.95E-02	
		Median	<b>3.80E-01</b>	4.13E-06	2.57E-05	3.00E-04	4.11E-03	3.55E-02	
	5	Mean	<b>4.61E-02</b>	4.42E-04	3.49E-05	3.73E-04	3.09E-02	2.00E-02	
		Median	<b>4.90E-01</b>	2.03E-05	2.99E-05	1.63E-04	2.00E-03	1.05E-02	
	10	Mean	<b>5.30E-02</b>	1.39E-03	3.08E-03	2.15E-04	1.49E-02	4.11E-03	
		Median	<b>2.17E-01</b>	1.69E-04	4.18E-04	1.15E-05	2.80E-03	1.79E-04	
	15	Mean	<b>5.07E-04</b>	6.69E-05	4.18E-05	2.15E-05	1.81E-06	3.97E-07	
		Median	<b>7.46E-05</b>	3.00E-06	3.19E-07	3.97E-07	1.03E-06	2.97E-07	
	DTLZ2	3	Mean	<b>4.22E-02</b>	2.69E-05	3.99E-03	4.05E-04	3.85E-02	2.90E-02
			Median	<b>3.69E-01</b>	3.03E-06	2.67E-05	2.97E-04	3.19E-03	3.57E-02
		5	Mean	<b>3.75E-01</b>	3.40E-04	2.09E-05	2.77E-04	2.91E-02	3.87E-02
			Median	<b>2.46E-02</b>	2.03E-05	1.09E-05	1.67E-04	2.01E-03	1.07E-02
10		Mean	<b>3.91E-02</b>	1.19E-03	3.09E-03	3.17E-04	1.57E-02	3.10E-03	
		Median	<b>3.97E-03</b>	1.59E-04	3.18E-04	1.05E-05	2.73E-03	2.70E-04	
15		Mean	<b>4.70E-04</b>	5.60E-05	3.85E-05	3.37E-05	3.01E-06	2.97E-07	
		Median	<b>5.13E-05</b>	3.97E-06	2.19E-07	3.06E-07	1.19E-06	2.00E-07	
DTLZ3		3	Mean	<b>7.48E-02</b>	5.57E-03	3.98E-03	2.05E-04	5.83E-02	2.90E-02
			Median	<b>4.36E-03</b>	4.13E-04	3.07E-04	3.78E-05	3.11E-04	3.97E-04
		5	Mean	<b>3.96E-01</b>	3.47E-02	2.39E-03	1.77E-03	2.49E-02	1.37E-02
			Median	<b>2.36E-02</b>	2.57E-03	2.00E-04	1.95E-04	2.33E-03	1.69E-03
	10	Mean	<b>6.82E-01</b>	2.30E-02	3.18E-03	4.09E-03	2.33E-02	3.03E-02	
		Median	<b>3.48E-02</b>	1.23E-03	4.26E-04	3.37E-04	2.49E-03	2.70E-04	
	15	Mean	<b>4.58E-02</b>	4.60E-03	3.08E-03	2.67E-03	2.98E-02	1.23E-03	
		Median	<b>2.60E-03</b>	3.01E-04	2.10E-04	3.92E-04	2.50E-05	2.67E-04	
	DTLZ4	3	Mean	<b>3.81E-02</b>	3.48E-03	3.00E-03	2.02E-03	3.69E-04	4.57E-05
			Median	<b>2.19E-03</b>	4.49E-04	4.07E-04	2.84E-05	1.66E-04	2.39E-05
		5	Mean	<b>4.70E-02</b>	4.24E-03	2.49E-04	4.58E-03	2.43E-03	3.47E-04
			Median	<b>4.48E-03</b>	3.01E-04	1.90E-05	4.61E-04	5.17E-04	6.69E-05
10		Mean	<b>9.59E-02</b>	2.30E-04	3.57E-03	3.79E-03	2.08E-04	3.96E-03	
		Median	<b>2.62E-03</b>	1.63E-04	4.10E-04	1.15E-05	2.81E-04	1.39E-04	
15		Mean	<b>4.58E-03</b>	6.17E-05	3.58E-04	2.99E-05	3.02E-04	3.43E-05	
		Median	<b>5.95E-04</b>	1.00E-05	3.10E-06	4.69E-07	1.13E-05	2.90E-06	

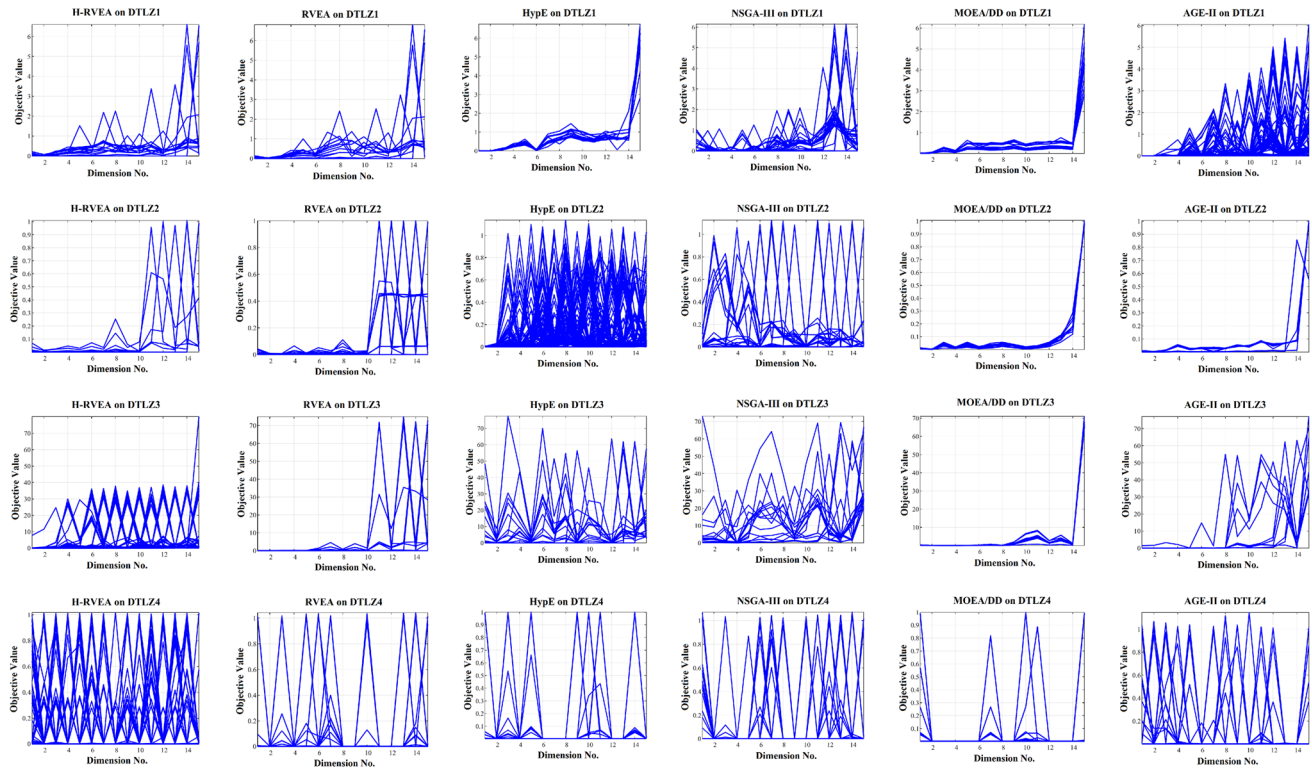
The obtained best results are in bold

**Algorithm 9** Reference vector adaptation strategy

**Input:** Generation index  $k$ , Population  $P$ , Current unit reference vector set  $V_k \leftarrow \{v_{k,1}, v_{k,2}, \dots, v_{k,N}\}$ , Initial unit reference vector set  $V_0 \leftarrow \{v_{0,1}, v_{0,2}, \dots, v_{0,N}\}$ , Control parameter  $F_c$

**Output:** Next generation reference vector set  $V_{k+1}$

- 1: **procedure** REFERENCEVECTORADAPTATION ( $k, P, V_k, V_0, F_c$ )
- 2:   **if** ( $\frac{k}{Max_{iteration}} \bmod F_c == 0$ ) **then**
- 3:     Compute the minimal  $O_{k+1}^{min}$  and maximal  $O_{k+1}^{max}$  objective values, respectively
- 4:     **for**  $j = 1$  to  $N$  **do**
- 5:       
$$v_{k+1,j} \leftarrow \frac{v_{0,j} \circ (O_{k+1}^{max} - O_{k+1}^{min})}{\|v_{0,j} \circ (O_{k+1}^{max} - O_{k+1}^{min})\|}$$
- 6:     **end for**
- 7:   **else**
- 8:      $v_{k+1,j} \leftarrow v_{k,j}$
- 9:   **end if**
- 10: **end procedure**



**Fig. 3** The non-dominated fronts obtained from six algorithms for 15-objectives on DTLZ1, DTLZ2, DTLZ3, and DTLZ4 test functions

There is a  $F_c$  controlling parameter which controls the frequency of use of the adaptation strategy. This parameter means that the reference vector adaptation strategy has to be carried out in a selected generation, since this strategy is not needed very often [58]. Example 2 illustrates the working of control parameter  $F_c$  in adaptation strategy.

**Example 2** (Control parameter) For instance, the value of  $F_c$  is set to 0.4. The reference vector will be adapted only at  $k = 0, k = 0.4 \times \text{Max}_{\text{iteration}}, k = 0.8 \times \text{Max}_{\text{iteration}}, k = 0.12 \times \text{Max}_{\text{iteration}}$  and so forth.

### 3.3 Computational complexity

The computational complexity of the proposed algorithm is debated in this subsection. The proposed algorithm's complexities in terms of time and space are given below.

#### 3.3.1 Time complexity

1. H-RVEA population initialization needs  $\mathcal{O}(M \times N)$  time, where  $M$  and  $N$  represent the number of objectives and population size, respectively.
2. The selection procedure for mating requires time of  $\mathcal{O}(M \times N^2)$ .

3. Variation procedure uses  $\mathcal{O}(N)$  time. Environmental selection method uses  $\mathcal{O}(N \log N)$  and the adaptation technique for the reference vector includes  $\mathcal{O}(N/F_c)$  time, where  $F_c$  represents the control parameter.

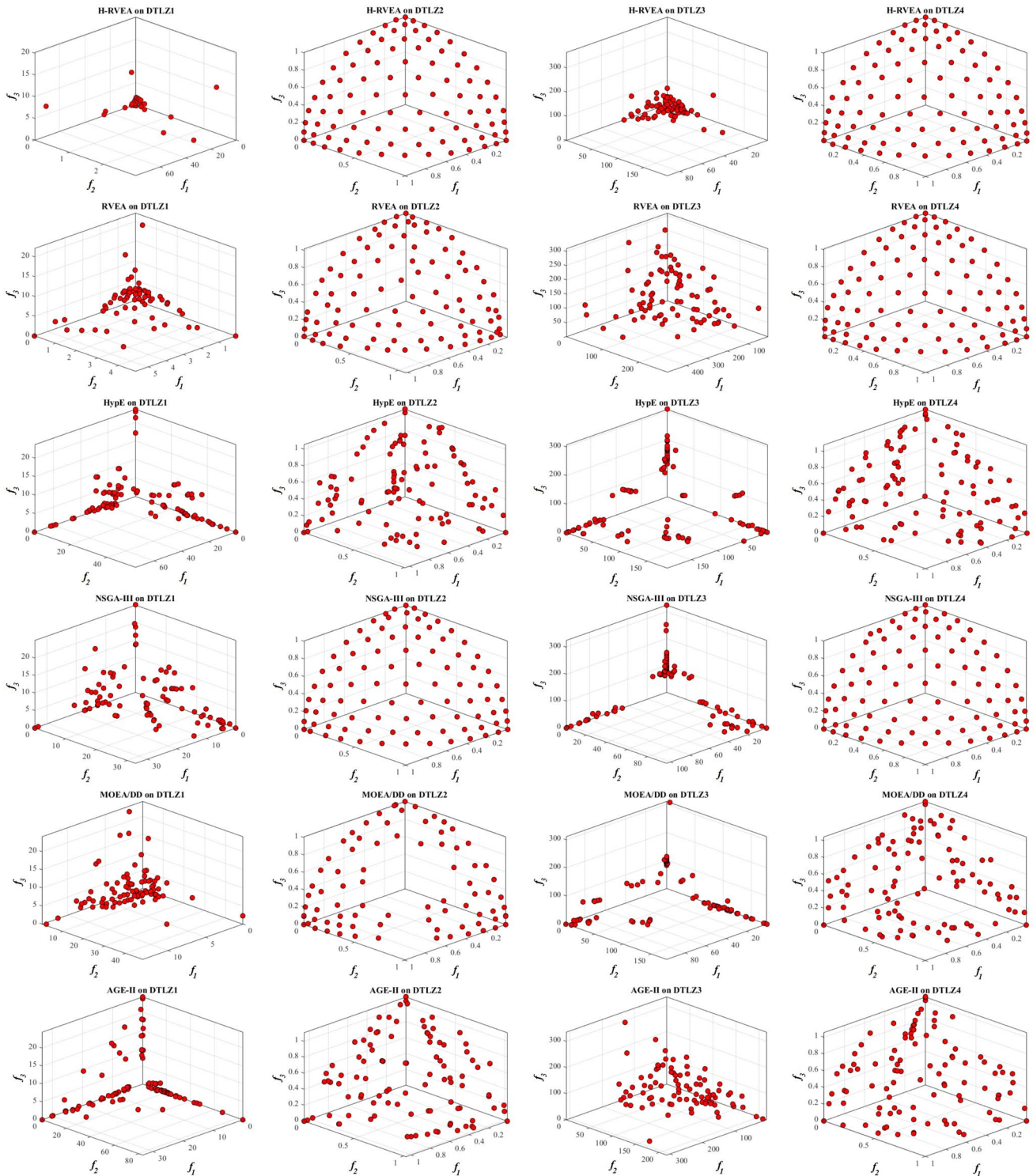
The summary of the complexities of all the above steps and the total time complexity of H-RVEA for the maximum number of generations is therefore  $\mathcal{O}(M \times N^2) + (N/F_c) \times \text{Max}_{\text{iteration}}$ .

#### 3.3.2 Space complexity

H-RVEA algorithm's space complexity is the cumulative amount of space that is perceived during its initialization process at any given point. Hence, the total space complexity of H-RVEA algorithm is  $\mathcal{O}(M \times N)$ .

## 4 Experimental results and discussions

The proposed algorithm H-RVEA is compared with five algorithms, namely Reference Vector Guided Evolutionary Algorithm (RVEA) [49], Hypervolume Estimation Algorithm (HypE) [50], Non-dominated Sorting Genetic Algorithm (NSGA-III) [53], Many-Objective Evolutionary Algorithm based on Dominance and Decomposition



**Fig. 4** The non-dominated solutions obtained from six algorithms for 3-objectives on DTLZ1, DTLZ2, DTLZ3, and DTLZ4 test functions

(MOEA/DD) [54], and Approximation-guided Evolutionary Algorithm (AGE-II) [55]. The well-known benchmark test problems from DTLZ test suite [59] are taken for

experimentation. The findings are measured with two well-known performance measures such as Inverted Generational Distance (IGD) [51] and Hypervolume (HV) [52].

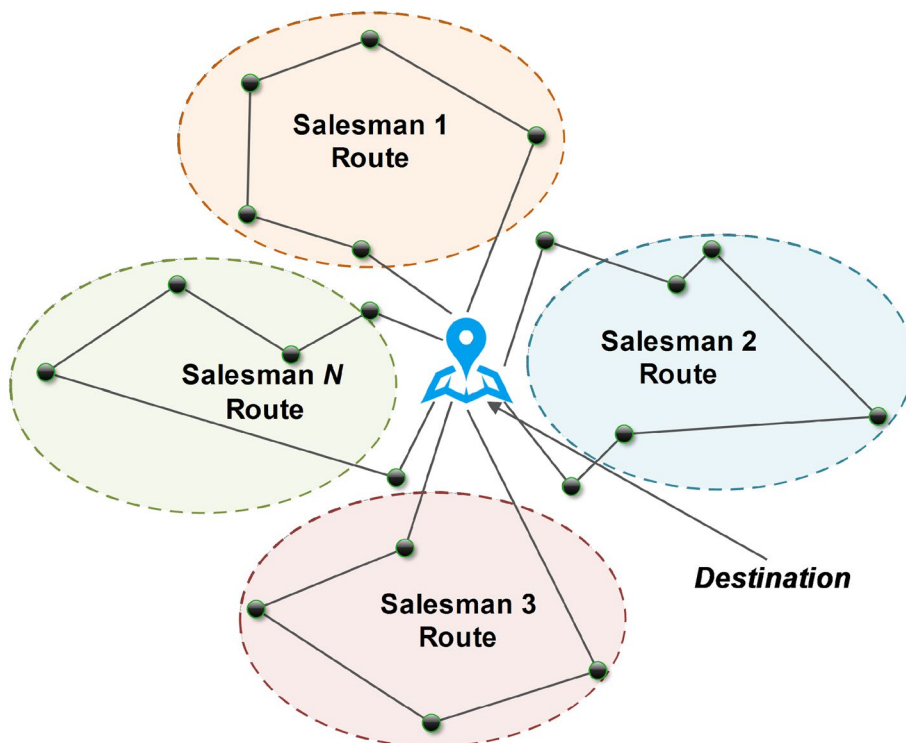
**Table 3** Wilcoxon signed-rank test results between proposed H-RVEA and other algorithms based on average IGD and HV performance measures

Algorithms	IGD	HV
RVEA	=	+
HypE	+	+
NSGA-III	+	−
MOEA/DD	+	+
AGE-II	+	+

### 4.1 Benchmark test problems

The four (DTLZ1-DTLZ4) test functions from DTLZ [59] test suite are employed. The number of objectives is varied from 3 to 15, i.e., {3, 5, 10, 15}. The number of decision variables is set to  $n = m + r - 1$  for DTLZ test problems [59], where  $r = 5$  for DTLZ1 and  $r = 10$  for DTLZ2-DTLZ4.

**Fig. 5** Schematic view of multi-objective travelling salesman problem



### 4.2 Experimental setup

The mean and median solutions are reported in tables which are obtained in final iteration. The simulations are done in Matlab R2019a environment on Microsoft Windows 10 (64-bit) using Core i7 and 3.15 GHz processor with 16 GB main memory.

### 4.3 Performance evaluation on DTLZ test suite

Tables 1 and 2 display the values of IGD and HV obtained through the suggested and competitor algorithms for the DTLZ test suite of 3-, 5-, 10- and 15-objectives. The results reveal that for DTLZ1 test problem H-RVEA offers better values of IGD and HV. Except for H-RVEA, RVEA and MOEA/DD achieve better values of IGD and HV than the

**Table 4** The IGD values obtained by proposed H-RVEA and other competitor algorithms on MOTSP

M	Analysis	H-RVEA	RVEA	HypE	NSGA-III	MOEA/DD	AGE-II
3	Mean	<b>6.11E−05</b>	7.61E−05	2.34E−04	3.24E−04	2.11E−04	1.50E−02
	Median	<b>6.39E−05</b>	3.10E−04	1.77E−03	2.51E−03	4.70E−03	2.99E−01
5	Mean	<b>3.72E−05</b>	2.00E−04	2.83E−02	4.21E−03	4.14E−05	1.31E−03
	Median	<b>4.10E−04</b>	2.54E−03	4.22E−02	7.90E−02	6.66E−04	1.70E−02
10	Mean	<b>9.09E−04</b>	3.39E−03	2.80E−02	7.26E−02	2.10E−02	8.60E−01
	Median	<b>2.70E−03</b>	6.66E−02	1.92E−01	4.01E−01	2.23E−01	9.84E−01
15	Mean	<b>1.00E−05</b>	4.35E−04	7.38E−03	3.45E−02	6.63E−02	2.07E−03
	Median	<b>4.14E−04</b>	7.75E−03	4.34E−02	8.13E−02	9.45E−02	5.67E−02

The obtained best results are in bold

others. RVEA and MOEA/DD are therefore the second and third best algorithms, respectively. Whereas, NSGA-III's output for all objectives is almost identical to the HypE algorithm on DTLZ1 test problem.

For test problem DTLZ2, H-RVEA gets a better value of IGD for 3- and 15-objectives. RVEA performs better results for these objectives than the other algorithms after H-RVEA. H-RVEA performs second best algorithm in terms of the performance measurement IGD for 5- and 10-objectives. RVEA provides better results for those objectives than all other approaches. H-RVEA generates optimal results for all objectives of the DTLZ2 test function for performance measurement HV. Whereas, for the efficiency indicator of HV, RVEA and HypE algorithms are very close to one another.

In terms of IGD and HV performance measures for all objectives, H-RVEA is significantly better than other algorithms on DTLZ3. Nonetheless, MOEA/DD and NSGA-III are the second and third strongest methods, respectively, for IGD efficiency assessment. The NSGA-III obtained the second best solution for HV efficiency estimation after the original H-RVEA algorithm.

For DTLZ4 test issue, H-RVEA's success measurements for 3-, 5-, 10- and 15-objectives are higher than others in terms of IGD and HV. MOEA/DD and NSGA-III comprise the second and third best methods for IGD and HV.

From Tables 1 and 2 it has been found that H-RVEA's consistency and efficiency is highest on most of the test functions. For IGD, H-RVEA performs well on 3 out of 4 assessment features. For values of HV, H-RVEA delivers the best results on 4 out of 4 test functions.

For visualization of the solutions, Figs. 3 and 4 display the non-dominated fronts obtained from the proposed H-RVEA and other 15- and 3-objectives competitor algorithms on the DTLZ test functions. RVEA and MOEA/DD have consistent convergence for 15-objectives DTLZ test functions but fail to reach certain regions of the Pareto front. HypE, NSGA-III, and AGE-II ensure good integration across most of Pareto's fronts. Fig. 3 demonstrates that H-RVEA's non-dominated front provides greater consistency and variety than other strategies. It has been observed for 3-objectives DTLZ test functions that the Pareto front obtained from H-RVEA, RVEA, NSGA-III, and MOEA/DD has uniform convergence and better diversity as shown in Fig. 4. Nonetheless, over the Pareto front HypE and AGE-II struggle to get a good coverage.

#### 4.4 Wilcoxon signed-rank test

It has been observed in the literature [60] that the performance measures IGD and HV do not provide any guarantee for better convergence and diversity, because sometimes the solutions obtained are not close to the optimal Pareto front. The Wilcoxon signed-rank test [61] is performed on

the average value of performance measures of IGD and HV to solve this problem. For increasing issue the gap between each pair of average outcomes is calculated. Such variations are ordered in ascending order, and a rank is given from the smallest to the highest. If more than one discrepancy is equivalent, then each of them is given an average rank [61]. The ranks are subsequently converted to signed ranks. It is used in pair-wise analysis of H-RVEA to other algorithms. The positive rank is granted to the proposed algorithm if it is stronger for a specific output factor (i.e., IGD and HV) than the competitor algorithms. Otherwise, it is assigned negative rank. A meaning level is set to 0.10 for comparison and sums up all the positive and negative rank [61]. Tabulate the effects of the Wilcoxon test in Table 3, where +, -, and = mean that H-RVEA efficiency is superior, inferior, and equivalent to competitor algorithms, respectively. From Table 3, it is observed that H-RVEA outperforms with IGD and HV efficiency measures over all competitor algorithms except for NSGA-III which finds superior on the HV measure.

## 5 H-RVEA for real-life applications

In this section, the performance of proposed H-RVEA has been tested on two real-life problems namely, Multi-Objective Travelling Salesman Problem (MOTSP) and Car Side Impact Problem (CSI). These problems are used to show the efficiency and effectiveness of proposed H-RVEA in real-life problems.

### 5.1 Multi-objective travelling salesman problem (MOTSP)

The traveling salesman problem in combinatorial optimization is a difficult and most studied NP-hard issue. The number of cities and the cost of travel are provided in this problem. The challenge with the traveling salesman is to find

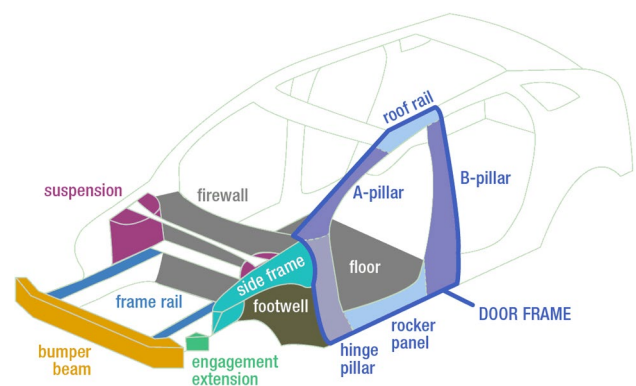


Fig. 6 Schematic view of car side impact problem [64]

the cheapest tour to reach each area precisely once and return to the point of departure (see Fig. 5).

The mathematical formulation of multi-objective travelling salesman problem is as follows [62]: Given a set  $n$  cities and  $p$  cost  $c_{ij}^k$ ,  $k = 1, 2, \dots, p$  to travel from city  $i$  to  $j$ . The main aim of *MOTSP* is to find a tour, i.e., a cyclic permutation  $R$  of  $n$  cities.

$$\text{Minimize } f_k(R) = \sum_{i=1}^{n-1} c_{R(i),R(i+1)}^k + c_{R(n),R(1)}^k, \quad k = 1, 2, \dots, p \quad (3)$$

In this work,  $p$  is set to 3, 5, 10, and 15.

Table 4 shows the IGD value obtained from H-RVEA and other competitor algorithms of Multi-Objective Travelling Salesman Problem (MOTSP). H-RVEA can be seen performing better for all test objectives than other algorithms (i.e., 3, 5, 10, 15). For 3- and 5-objectives, H-RVEA has a best IGD value and the difference between H-RVEA and its competitor algorithms is statistically significant. RVEA is the second best algorithm for 3-objectives. The MOEA/DD is the second best performing algorithm for 5-objectives. H-RVEA outperforms the other five algorithms for the 10- and 15-objectives. RVEA is the second best algorithm for

these test objectives. It is worth mentioning that H-RVEA consistently performing better than other algorithms for all four *MOTSP* test objectives.

## 5.2 Car side impact problem (CSI)

Car side impact problem is a constrained optimization problem to optimize the vehicle side crashworthiness [63]. This problem involves seven design variables namely thickness of B-Pillar inner, B-Pillar inner reinforcement, floor side inner, cross-members, door beam, door beltline reinforcement, and roof rail (see Fig. 6). There are three objective functions of this problem which are described as:

- Weight of the car.
- Pubic force experienced by a passenger.
- Average velocity of the V-Pillar responsible for withstanding the impact load.

The mathematical formulation of this problem is described as follows:

$$\begin{aligned} f_1(z) &= 1.98 + 4.9z_1 + 6.67z_2 + 6.98z_3 + 4.01z_4 + 1.78z_5 + 0.00001z_6 + 2.73z_7, \\ f_2(z) &= 4.72 - 0.5z_4 - 0.19z_2z_3, \\ f_3(z) &= 0.5 \times (10.58 - 0.674z_1z_2 - 0.67275z_2 + 16.45 - 0.489z_3z_7 - 0.843z_5z_6), \\ \text{Subject to:} \\ g_1(z) &= 1.16 - 0.3717z_2z_4 - 0.0092928z_3 \leq 1.0, \\ g_2(z) &= 0.261 - 0.0159z_1z_2 - 0.06486z_1 - 0.019z_2z_7 \\ &\quad + 0.0144z_3z_5 + 0.0154464z_6 \leq 0.32, \\ g_3(z) &= 0.214 + 0.00817z_5 - 0.045195z_1 - 0.0135168z_1 \\ &\quad + 0.03099z_2z_6 - 0.018z_2z_7 + 0.007176z_3 \\ &\quad + 0.023232z_3 - 0.00364z_5z_6 - 0.018z_2z_2 \leq 0.32, \\ g_4(z) &= 0.74 - 0.61z_2 - 0.031296z_3 - 0.031872z_7 + 0.227z_2z_2 \leq 0.32, \\ g_5(z) &= 28.98 + 3.818z_3 - 4.2z_1z_2 + 1.27296z_6 - 2.68065z_7 \leq 0.32, \\ g_6(z) &= 33.86 + 2.95z_3 - 5.057z_1z_2 - 3.795z_2 - 3.4431z_7 \\ &\quad + 1.45728 \leq 0.32, \\ g_7(z) &= 46.36 - 9.9z_2 - 4.4505z_1 \leq 0.32, \\ g_8(z) &= 4.72 - 0.5z_4 - 0.19z_2z_3 \leq 4.0, \\ g_9(z) &= 10.58 - 0.674z_1z_2 - 0.67275z_2 \leq 9.9, \\ g_{10}(z) &= 16.45 - 0.489z_3z_7 - 0.843z_5z_6 \leq 15.7, \\ \text{where,} \\ 0.5 \leq z_1, z_3, z_4 \leq 1.5, & \quad 0.4 \leq z_6, z_7 \leq 1.2, \quad 0.45 \leq z_2 \\ &\quad \leq 1.35, \quad 0.875 \leq z_5 \leq 2.625. \end{aligned} \quad (4)$$



**Table 5** The IGD values obtained by proposed H-RVEA and other competitor algorithms on CSI

Analysis	H-RVEA	RVEA	HypE	NSGA-III	MOEA/DD	AGE-II
Mean	<b>1.36E-03</b>	5.62E-02	4.00E-03	2.55E-03	9.31E-02	2.42E-02
Median	<b>2.51E-01</b>	7.47E-01	6.36E-00	3.80E-01	7.51E-01	4.12E-01

The obtained IGD values by proposed and competitor algorithms are shown in Table 5. H-RVEA generates the smaller IGD value than the others. NSGA-III and HypE are the second and third best optimization algorithms, respectively. It can be seen that H-RVEA provides optimal results and very competitive than the other approaches for CSI problem.

## 6 Conclusions and future works

A novel hybrid many-objective evolutionary algorithm, named H-RVEA, is introduced in this article. H-RVEA uses the Hypervolume Estimation Algorithm (HypE) and Reference Vector Guided Evolutionary Algorithm (RVEA) methods. The proposed H-RVEA was applied and evaluated on well-known test functions. Comparing H-RVEA results to other algorithms, it was found that H-RVEA outperformed RVEA, HypE, NSGA-III, MOEA/DD, and AGE-II. The comparative work was carried out to demonstrate the statistical significance of proposed H-RVEA over benchmark test functions. The proposed H-RVEA was tested on two real-life constrained problems. The findings show that among all the successful algorithms H-RVEA has delivered better performance.

There are several research directions which can be recommended for future works. The variation in operators and selection procedure of the proposed H-RVEA algorithm can be the motivation of future work. Also, to extend this algorithm to solve more real-life constrained many-objective optimization problems can also be seen as a future contribution.

**Acknowledgements** The first author Dr. Gaurav Dhiman would like to thanks to “Shri Maa Kali Devi” for her divine blessing on him.

## References

- Kalyanmoy D (2001) Multi objective optimization using evolutionary algorithms. Wiley, New York
- Coello CAC, Lamont GB, Van Veldhuizen DA et al (2007) Evolutionary algorithms for solving multi-objective problems, vol 5. Springer, Berlin
- Li K, Zheng J, Li M, Zhou C, Lv H (2009) A novel algorithm for non-dominated hypervolume-based multiobjective optimization. In: Systems, man and cybernetics, SMC 2009. IEEE international conference on. IEEE 2009:5220–5226
- Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. *Adv Eng Softw* 114:48–70
- Singh P, Dhiman G (2018) A hybrid fuzzy time series forecasting model based on granular computing and bio-inspired optimization approaches. *J Comput Sci* 27:370–385
- Dhiman G, Kumar V (2018) Multi-objective spotted hyena optimizer: a multi-objective optimization algorithm for engineering problems. *Knowl-Based Syst* 150:175–197
- Singh P, Dhiman G (2018) Uncertainty representation using fuzzy-entropy approach: Special application in remotely sensed high-resolution satellite images (rsrshrs). *Appl Soft Comput* 72:121–139
- Dhiman G, Kumar V (2018) Emperor penguin optimizer: a bio-inspired algorithm for engineering problems. *Knowl-Based Syst* 159:20–50
- Dhiman G, Kaur A (2018) Optimizing the design of airfoil and optical buffer problems using spotted hyena optimizer. *Designs* 2(3):28
- Lygoe RJ, Cary M, Fleming PJ (2013) A real-world application of a many-objective optimisation complexity reduction process. In: International conference on evolutionary multi-criterion optimization. Springer, pp. 641–655
- Fu G, Kapelan Z, Kasprzyk JR, Reed P (2012) Optimal design of water distribution systems using many-objective visual analytics. *J Water Resour Plan Manag* 139(6):624–633
- Chikumbo O, Goodman E, Deb K (2012) Approximating a multi-dimensional pareto front for a land use management problem: a modified moea with an epigenetic silencing metaphor. In: Evolutionary computation (CEC), IEEE congress on. IEEE 2012:1–9
- Singh P, Rabadiya K, Dhiman G (2018) A four-way decision-making system for the indian summer monsoon rainfall. *Mod Phys Lett B* 32(25)
- Dhiman G, Kumar V (2018) Astrophysics inspired multi-objective approach for automatic clustering and feature selection in real-life environment. *Mod Phys Lett B* 32(31)
- Kaur A, Kaur S, Dhiman G (2018) A quantum method for dynamic nonlinear programming technique using schrödinger equation and monte carlo approach. *Mod Phys Lett B* 32(30)
- Singh P, Dhiman G, Kaur A (2018) A quantum approach for time series data based on graph and schrödinger equations methods. *Mod Phys Lett A* 33(35)
- Dhiman G, Guo S, Kaur S (2018) Ed-sho: A framework for solving nonlinear economic load power dispatch problem using spotted hyena optimizer. *Mod Phys Lett A* 33(40)
- Dhiman G, Kumar V (2019) Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems. *Knowl-Based Syst* 165:169–196
- Dhiman G, Kaur A (2019) Stoa: a bio-inspired based optimization algorithm for industrial engineering problems. *Eng Appl Artif Intell* 82:148–174
- Dhiman G, Kumar V (2019) Knrvea: a hybrid evolutionary algorithm based on knee points and reference vector adaptation strategies for many-objective optimization. *Appl Intell* 49(7):2434–2460
- Dhiman G, Singh P, Kaur H, Maini R (2019) Dhiman: a novel algorithm for economic dispatch problem based on optimization

- method using monte carlo simulation and astrophysics concepts. *Mod Phys Lett A* 34(04)
22. Dhiman G (2019) Moshapo: a hybrid multi-objective approach to solve economic load dispatch and micro grid problems. *Appl Intell*:1–19
  23. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Trans Evol Comput* 6(2):182–197
  24. Zitzler E, Laumanns M, Thiele L (2001) *Spea2: Improving the strength pareto evolutionary algorithm*. TIK-report, vol. 103
  25. Helbig M, Engelbrecht AP (2014) Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm Evol Comput* 14:31–47
  26. Tian Y, Cheng R, Zhang X, Cheng F, Jin Y (2017) An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. *IEEE Trans Evol Comput* 22(4):609–622
  27. Singh P, Dhiman G, Guo S, Maini R, Kaur H, Kaur A, Kaur H, Singh J, Singh N (2019) A hybrid fuzzy quantum time series and linear programming model: Special application on taixex index dataset. *Mod Phys Lett A* 34(25)
  28. Dhiman G (2019) Esa: a hybrid bio-inspired metaheuristic optimization approach for engineering problems. *Eng Compute*:1–31
  29. Dhiman G (2019) Multi-objective metaheuristic approaches for data clustering in engineering application (s). Ph.D. dissertation
  30. Dehghani M, Montazeri Z, Malik OP, Dhiman G, Kumar V (2019) Bosa: binary orientation search algorithm. *Int J Innov Technol Explor Eng* 9:5306–5310
  31. Walker DJ, Everson R, Fieldsend JE (2013) Visualizing mutually nondominating solution sets in many-objective optimization. *IEEE Trans Evol Comput* 17(2):165–184
  32. Fonseca CM, Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation. *IEEE Trans Syst Man Cybern Part A Syst Hum* 28(1):26–37
  33. Purshouse RC, Fleming PJ (2007) On the evolutionary optimization of many conflicting objectives. *IEEE Trans Evol Comput* 11(6):770–784
  34. Knowles J, Corne D (2007) Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: *International conference on evolutionary multi-criterion optimization*. Springer, pp 757–771
  35. Ishibuchi H, Akedo N, Nojima Y (2015) Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Trans Evol Comput* 19(2):264–283
  36. Jin Y, Sendhoff B (2003) Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In: *Evolutionary computation, CEC'03. The 2003 Congress on, vol. 3*. IEEE 2003:1910–1917
  37. Chandrawat RK, Kumar R, Garg B, Dhiman G, Kumar S (2017) An analysis of modeling and optimization production cost through fuzzy linear programming problem with symmetric and right angle triangular fuzzy number. In: *Proceedings of sixth international conference on soft computing for problem solving*. Springer, pp 197–211
  38. Singh P, Dhiman G (2017) A fuzzy-lp approach in time series forecasting. In: *International conference on pattern recognition and machine intelligence*. Springer, pp 243–253
  39. Dhiman G, Kaur A (2017) Spotted hyena optimizer for solving engineering design problems. In: *international conference on machine learning and data science (MLDS)*. IEEE 2017:114–119
  40. Verma S, Kaur S, Dhiman G, Kaur A (2018) Design of a novel energy efficient routing framework for wireless nanosensor networks. In: *2018 first international conference on secure cyber computing and communication (ICSCCC)*. IEEE, pp 532–536
  41. Kaur A, Dhiman G (2019) A review on search-based tools and techniques to identify bad code smells in object-oriented systems. In: *Harmony search and nature inspired optimization algorithms*. Springer, pp 909–921
  42. Garg M, Dhiman G (2020) Deep convolution neural network approach for defect inspection of textured surfaces. *J Inst Electron Comput* 2:28–38
  43. Dhiman G, Kaur A (2019) A hybrid algorithm based on particle swarm and spotted hyena optimizer for global optimization. In: *Soft computing for problem solving*. Springer, pp 599–615
  44. Dhiman G, Kumar V (2019) Spotted hyena optimizer for solving complex and non-linear constrained engineering problems. In: *Harmony search and nature inspired optimization algorithms*. Springer, pp 857–867
  45. Li B, Li J, Tang K, Yao X (2015) Many-objective evolutionary algorithms: a survey. *ACM Comput Surv (CSUR)* 48(1):13
  46. Emmerich M, Beume N, Naujoks B (2005) An emo algorithm using the hypervolume measure as selection criterion. In: *International conference on evolutionary multi-criterion optimization*. Springer, pp 62–76
  47. Igel C, Hansen N, Roth S (2007) Covariance matrix adaptation for multi-objective optimization. *Evol Comput* 15(1):1–28
  48. Brockhoff D, Zitzler E (2007) Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In: *Evolutionary computation, CEC 2007*. IEEE Congress on. IEEE 2007:2086–2093
  49. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(5):773–791
  50. Bader J, Zitzler E (2011) Hype: an algorithm for fast hypervolume-based many-objective optimization. *Evol Comput* 19(1):45–76
  51. Bosman PA, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 7(2):174–188
  52. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
  53. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601
  54. Li K, Deb K, Zhang Q, Kwong S (2015) An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans Evol Comput* 19(5):694–716
  55. Wagner M, Neumann F (2013) A fast approximation-guided evolutionary multi-objective algorithm. In: *Proceedings of the 15th annual conference on Genetic and evolutionary computation*. ACM, pp 687–694
  56. Fleischer M (2003) The measure of pareto optima applications to multi-objective metaheuristics. In: *International conference on evolutionary multi-criterion optimization*. Springer, pp 519–533
  57. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. *Mach Learn* 3(2):95–99
  58. Giagkiozis I, Purshouse RC, Fleming PJ (2013) Towards understanding the cost of adaptation in decomposition-based optimization algorithms. In: *Systems, man, and cybernetics (SMC), 2013 IEEE international conference on*. IEEE, pp 615–620
  59. Deb K, Thiele L, Laumanns M, Zitzler E (2005) Scalable test problems for evolutionary multiobjective optimization. In: *Evolutionary multiobjective optimization*. Springer, pp 105–145
  60. Roy PC, Islam MM, Murase K, Yao X (2015) Evolutionary path control strategy for solving many-objective optimization problem. *IEEE Trans Cybern* 45(4):702–715

61. Richardson A (2010) Nonparametric statistics for non-statisticians: a step-by-step approach by gregory w. corder, dale i. foreman. *Int Stat Rev* 78(3):451–452
62. Corne DW, Knowles JD (2007) Techniques for highly multiobjective optimisation: some nondominated points are better than others. In: *Proceedings of the 9th annual conference on Genetic and evolutionary computation*. ACM, pp 773–780
63. Gu L, Yang R, Tho C-H, Makowskit M, Faruquet O, Li YLY (2001) Optimisation and robustness for crashworthiness of side impact. *Int J Veh Des* 26(4):348–360
64. “[status report],” <https://www.iihs.org/iihs/sr/statusreport/article/49/11/1>

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.