

# Memetic algorithm with Preferential Local Search using adaptive weights for multi-objective optimization problems

J. Bhuvana · Chandrabose Aravindan

Published online: 15 February 2015  
© Springer-Verlag Berlin Heidelberg 2015

**Abstract** Evolutionary multi-objective optimization algorithms are generally employed to generate Pareto optimal solutions by exploring the search space. To enhance the performance, exploration by global search can be complemented with exploitation by combining it with local search. In this paper, we address the issues in integrating local search with global search such as: how to select individuals for local search; how deep the local search is performed; how to combine multiple objectives into single objective for local search. We introduce a Preferential Local Search mechanism to fine tune the global optimal solutions further and an adaptive weight mechanism for combining multi-objectives together. These ideas have been integrated into NSGA-II to arrive at a new memetic algorithm for solving multi-objective optimization problems. The proposed algorithm has been applied on a set of constrained and unconstrained multi-objective benchmark test suite. The performance was analyzed by computing different metrics such as Generational distance, Spread, Max spread, and HyperVolume Ratio for the test suite functions. Statistical test applied on the results obtained suggests that the proposed algorithm outperforms the state-of-art multi-objective algorithms like NSGA-II and SPEA2. To study the performance of our algorithm on a real-world application, Economic Emission Load Dispatch was also taken up for validation. The performance was studied with the help of measures such as Hypervolume and Set Coverage Metrics. Experimental results substantiate that our algorithm has

the capability to solve real-world problems like Economic Emission Load Dispatch and is able to produce better solutions, when compared with NSGA-II, SPEA2, and traditional memetic algorithms with fixed local search steps.

**Keywords** Multi-objective optimization · Memetic algorithm · Preferential Local Search · Economic emission load dispatch

## 1 Introduction

Many of the real-world problems like scheduling, resource allocation, aerodynamic design, supply chain management, microprocessor chip design, mechanical component design, and medical image reconstruction (Bagchi 1999; Cheng and Li 1997; Joines et al. 2002; Aguilar and Miranda 1999; Deb 2001) have multiple competing objectives to optimize. These are called multi-objective optimization problems (MOOPs), where the objectives need to be simultaneously optimized. Almost all of these real-world multi-objective problems have constraints that are to be satisfied. A constrained multi-objective optimization problem with  $M$  objectives is formulated as,

$$\left. \begin{array}{l} \text{Minimize/maximize } f_m(x), \quad m = 1, 2, \dots, M; \\ \text{subject to } g_j(x) \leq 0, \quad j = 1, 2, \dots, J; \\ h_k(x) = 0, \quad k = 1, 2, \dots, K; \\ x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad i = 1, 2, \dots, I; \end{array} \right\} \quad (1)$$

There are  $J$  inequality and  $K$  equality constraints with  $I$  decision variables in the above equation. Constrained multi-objective problems divide the search space into two regions as feasible and infeasible regions (Deb 2001). Solution  $x^{(i)}$  that satisfies all the constraints is a feasible solution.

Communicated by V. Loia.

J. Bhuvana (✉) · C. Aravindan  
Department of Computer Science and Engineering, SSN College  
of Engineering, Kalavakkam, Chennai 603110, India  
e-mail: bhuvanaj@ssn.edu.in

C. Aravindan  
e-mail: aravindanc@ssn.edu.in

Multi-objective optimization problems require a set of Pareto optimal solutions rather than a single optimal solution. These Pareto optimal solutions exhibit a trade-off among the objectives (Coello 2000; Konak et al. 2006) and are non-dominating in nature. A solution  $x$  is said to dominate another solution  $y$ , when it is not worse than  $y$  in all objectives and strictly better than  $y$  in at least one objective (Deb 2001). If  $x$  does not dominate  $y$  and  $y$  does not dominate  $x$ , then  $x$  and  $y$  are non-dominant solutions. Such solutions are Pareto optimal and are not dominated by any other solution in the population. For a Pareto optimal solution, improvement in one objective of the solution will worsen at least one of the other objectives. Objective values corresponding to Pareto optimal solutions form Pareto optimal front in the objective space.

The goals of any multi-objective optimization algorithm that solves a MOOP are to converge towards optimal front and to generate uniformly spread solutions across the Pareto front in low computation time (Zitzler et al. 2000; Mostaghim 2004). Multi-objective evolutionary algorithms (MOEAs) (Fonseca and Fleming 1995) are widely used for solving MOOPs. Evolutionary algorithms inherently deal with a population of solutions and have the ability to generate a set of optimal solutions while solving multiple objectives. Genetic Algorithm (GA) (Holland 1975; Goldberg 1989) is one kind of evolutionary algorithm that performs a stochastic search and imitates the evolution of nature. Genetic Algorithms are designed to explore the entire search space and generate Pareto optimal solution set when used to optimize multiple contradicting objectives. Genetic Algorithms are broadly categorized under global search algorithms.

Multi-objective evolutionary algorithms are generally classified into elitist and nonelitist algorithms with respect to whether any explicit measures are taken to retain potential solutions or not. Several nonelitist and elitist algorithms have been reported so far. VEGA (Schaffer 1985), MOGA (Fonseca and Fleming 1993), NPGA (Horn et al. 1994), NSGA (Srinivas and Deb 1994), and WBGA (Hajela and Lin 1992) are a few of the nonelitist multi-objective evolutionary algorithms, whereas NSGA-II (Deb et al. 2000, 2002), SPEA (Zitzler and Thiele 1999), SPEA2 (Zitzler et al. 2001), and PAES (Knowles and Corne 2000) are some of the widely used elitist multi-objective evolutionary algorithms. A detailed survey of such algorithms can be found in Konak et al. (2006), Coello and Zacatenco (2006), Coello (2000), and Coello (2002).

Advantages of evolutionary algorithms (EAs) are many when compared with other optimization algorithms. They are simple, are easy to customize, have wide range of applications, are suitable for complex search spaces, and can outperform classical techniques for optimization (Blickle 1996). Despite having advantages, EAs have a few draw-

backs such as lack of fine tuning the solutions in the search space (Krasnogor and Smith 2005) and premature convergence (McGinley et al. 2011). To overcome these drawbacks, a global search algorithm like GA can be combined with a local search procedure resulting in memetic algorithm (MA) (Moscato 1989) or hybrid algorithm. Performance of any search algorithm can be enhanced further by combining with other search algorithms, by accelerating the nature of problem solving (Ong et al. 2010). GAs can be easily combined with any other traditional optimization techniques (Fogel 1995; Abraham 2005). This kind of integration of two different search procedures will improve the search performance of the hybrid algorithm. The research issues to be considered while integrating two search procedures are (Ong et al. 2010; Krasnogor and Smith 2005):

- Whether all the individuals or only a few individuals are chosen for local search;
- How often a local search can be applied;
- How deep a local search is allowed;
- How multi-objectives can be converted into a single objective for performing local search.

Our proposed algorithm addresses the abovelisted issues by preferentially selecting individuals on the basis of their potential, limiting the number of local searches applied, facilitating iterative deepening of search and using a new adaptive weight scheme for combining multiple objectives into a single one. These ideas have been incorporated into the existing NSGA-II algorithm arriving at a new algorithm called memetic algorithm with Preferential Local Search using adaptive weights (MAPLS-AW).

Performance of the proposed algorithm has been evaluated on constrained and unconstrained benchmark test problems using four metrics: Generational Distance, Spread, Max spread and HyperVolume Ratio. The proposed algorithm has also been applied on Economic Emission Load Dispatch (EELD), a real-world application and performance was evaluated using Hypervolume metric and Set Coverage Metrics. The results observed show that MAPLS-AW performs better than NSGA-II, SPEA2, and traditional memetic algorithms with fixed local search steps.

This paper is organized as follows. Section 2 gives rest of background and presents various issues in designing hybrid search algorithms. Proposed algorithm MAPLS-AW is presented in Sect. 3. Section 4 summarizes the details of implementation and experiments conducted along with discussion of results. Application of proposed MAPLS-AW on a real-world problem, EELD, is discussed in Sect. 5. Section 6 gives the conclusion of the proposed algorithm with future directions.

## 2 Background and related works

### 2.1 Multi-objective optimization

Conventionally, multi-objective problems are solved by combining multiple objectives into a single objective. Classical aggregation methods such as weighted sum, value function, and  $\epsilon$ -constraint method generate single optimal solution per run. To obtain a set of Pareto optimal solutions, a classical optimization method has to be repeatedly executed with different parameters. Several algorithms have been developed to simultaneously solve multiple objectives in order to generate a set of Pareto optimal solutions. The generated Pareto optimal solutions should be close to the optimal front and also be uniformly distributed across the Pareto front by exploring the extreme regions of the search space (Konak et al. 2006).

Evolutionary GA is a nature inspired computation strategy that imitates the evolutionary process. Each individual in GA is called a chromosome that carries distinct elements called genes. To solve an optimization problem using GA, the first essential step is to encode the solutions into chromosomes. Several different encoding schemes are available to represent the solutions, such as binary encoding, value encoding, real encoding, and permutation encoding. Selecting a specific encoding technique depends on the problem to be solved. Binary encoding (Goldberg 1989) is the most common representation used to encode chromosomes and has got a variety of applications. When some real-world problems require real values to encode decision parameters, real coding scheme finds its use (Herrera et al. 1998). A collection of chromosomes forms the population. With iterative application of genetic operators such as crossover and mutation on selected solutions evolves the population towards the optimal solutions.

As GAs are designed to work with population of solutions, they are suitable to solve multi-objective problems. Such multi-objective GAs differ in their working with respect to Pareto ranking principle or fitness sharing strategy. Most of the recent GAs use elitism to preserve the elite solutions (Coello and Zlatencic 2006). Two methods have been used to employ elitism. They are maintaining an elite population set and maintaining a separate archive for elite solutions which will be used during the evolution. NSGA-II is an elitist multi-objective GA that preserves elitist solutions in the population itself. This algorithm remains as one of widely used MOEAs, due to its effective performance (Coello and Zlatencic 2006) and hence we have decided to adapt NSGA-II for our proposed work.

NSGA-II is a ranking-based evolutionary algorithm which has measures to maintain elitism and preserve the diversity of solutions. Once the population is initialized with random solutions, the population is sorted into different ranks using non-dominated sorting method. Individuals in the same rank

are placed in the same front. NSGA-II uses the crowded sorting operator for selecting an individual from the same front. This operator sorts the individuals in the same rank with respect to the density of surrounding solutions. A solution from less dense space is given the highest preference for selection (Deb et al. 2002). This feature is one that explicitly maintains the Spread of optimal solutions.

The Pareto optimal solutions generated by the optimization process should be from the feasible region of search space. In such case, the selection procedure in the evolutionary cycle needs to select feasible solutions from the search space. Constrained binary tournament selection (Deb 2001) is one such selection procedure used in constrained optimization problems. Two solutions say  $x$  and  $y$  are chosen at random from the population.

- If  $x$  and  $y$  both are feasible solutions and are in different ranks, a solution with better rank is selected. If both solutions are in same rank then a solution from less crowded space will be selected.
- If one out of  $x$  and  $y$  is feasible and other is an infeasible solution, then a feasible solution is selected.
- If  $x$  and  $y$ , both are infeasible solutions then the one with less constraint violation will be selected.

NSGA-II calls the above procedure as crowded comparison operator.

#### *Crossover and its characteristics*

The primary search operator that explores the search space in the nature inspired evolutionary algorithms is the crossover operator (De Jong and Spears 1992; Črepinšek et al. 2013). Crossover operation determines the extent of exploration to be performed in the search space. Search space can be divided into exploration zone and exploitation zone (Herrera et al. 2003). When similar parents undergo crossover they lead to exploitation of the region than exploration. McGinley et al. (2011) introduces an adaptive crossover by varying the crossover probability. Two different measures such as Standard Population Diversity (SPD) and Healthy Population Diversity (HPD) are used for crossover adaptation thereby to improve the diversity of the population.

Simulated binary crossover (SBX) (Deb and Kumar 1995; Deb and Agrawal 1995) is one of real-coded crossover operators. This has the same search ability as that of single-point binary crossover. If  $P_1$  and  $P_2$  are the two parents selected for SBX crossover, new offspring say  $C_1$  and  $C_2$  will be generated as per the following procedure: A random value  $u$  between 0 and 1 is selected.  $\beta$ , the Spread factor is calculated with respect to  $u$  value. If  $u$  is less than 0.5 then  $\beta = (2 * u)^{1/\eta_c + 1}$  otherwise  $\beta = 1 / (2 * (1 - u)^{1/\eta_c + 1})$ .  $\eta_c$  is the distribution factor that decides whether children should

be generated nearer to the parents or not. Higher the value of  $\eta_c$  closer the children to the parents otherwise they will be generated far away from the parents.

$$\begin{aligned} C_1 &= 0.5 * [(P_1 + P_2) - \beta * |P_2 - P_1|] \\ C_2 &= 0.5 * [(P_1 + P_2) + \beta * |P_2 - P_1|] \end{aligned} \quad (2)$$

Balanced exploration and exploitation (Črepinšek et al. 2013) always generate diverse set of good quality solutions. Values suggested in Deb et al. (2002) for distribution parameter  $\eta_c$  is between 5 and 20. To achieve maximum exploration of search space, we decided to eliminate the need of this parameter in our proposed work. By reducing the significance of  $\eta_c$ , the extreme regions of the search space can be explored. The new offspring will be generated in the unexplored areas of the search space.

To introduce diversity at gene level, mutation is performed on the new offspring and not all the offspring will undergo mutation. High mutation rate will affect the probability of getting the global optimum and hence always limited to a lesser value (Weise 2009). Polynomial mutation is the kind of mutation used by real-coded GAs which tries to simulate the binary mutation. Polynomial mutation, mutates the new offspring as follows. A new mutated individual  $Y_i$  is given by

$$Y_i = x_i + (x_i^U - x_i^L) * \delta_i \quad (3)$$

where  $x_i$  is the  $i$ th parameter selected with mutation probability  $p_m$ ,  $x_i^U$  and  $x_i^L$  are the lower and upper bounds of  $x_i$ , respectively. Mutation probability  $p_m$  is usually set to  $1/n$ , where  $n$  is the number of decision parameters in the multi-objective problem (Deb and Goyal 1996).  $\delta_i$  is given by

$$\delta_i = \begin{cases} (2r_i)^{1/\eta_m+1} - 1 & \text{if } r_i \leq 0.5, \\ 1 - (2(1 - r_i))^{1/\eta_m+1} - 1 & \text{if } r_i \geq 0.5. \end{cases} \quad (4)$$

where  $r_i$  takes a random value between  $[0,1]$  and  $\eta_m$ , the mutation constant is any non-negative real number. As a measure to impose elitism, NSGA-II combines the parent solutions with new offspring and non-dominated sorting is applied on the combined population. We have used the template of NSGA-II to fit in the design of our proposed algorithm.

Another kind of elitist multi-objective Evolutionary Algorithm where an external archive of non-dominating solutions is maintained is Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al. 2001). It is different from its predecessor SPEA, in terms of fitness assignment, density estimation and truncation operations (Guliashki et al. 2009). The size of external archive can be set as same as that of population size or may be different. When the number of non-dominating solutions exceeds the size of archive, they are truncated with respect to a procedure similar to  $k$ th nearest

neighbor. Fitness of any individual in this algorithm does not solely depend on the functional objective, but also on a strength value of dominated individuals and density information. NSGA-II and SPEA2 are the two most widely used elitist MOEAs in the literature (Zhou et al. 2011). The performance of our proposed algorithm has been compared with these two state-of-art algorithms.

## 2.2 Local optimization

Local search method is one of the metaheuristic techniques which is otherwise referred to as trajectory method or iterative improvement method. Local optimization or local search techniques start with an initial solution and search its neighborhood (Gaspero 2003; Blum and Roli 2003), by applying changes to the current solution. The new solution will be accepted if it is better than the previous one and by this manner an initial solution is enhanced through local optimization.

Local search techniques can be classified into derivative and nonderivative techniques (Jang et al. 2004). Nonderivative methods are simplex method (Nelder and Mead 1965), tabu search (Glover and Laguna 1997), etc, and derivative techniques include Gradient descent methods and Newton methods. Nonderivative methods rely on objective function to guide the search process (Deb 2001), whereas the derivative methods make use of gradient information from the objective space to decide the search direction.

Steepest descent or Gradient descent is one of the derivative-based algorithms which, 'begins with an initial solution and repeatedly subtracts a fraction of locally calculated gradient from the current solution' (Salomon 1998). That is, the new solution  $x_{k+1}$  is obtained by the following equation.

$$x_{k+1} = x_k + \lambda d_k \quad (5)$$

where  $x \in \mathbb{R}$ ,  $\lambda$  is the step size. Descent direction is given by  $d_k = -\nabla f(x_k)$ , where  $f$  is the function to be optimized.

### Issues in scalarizing objectives

Local search requires the multiple objectives to be converted into a single objective. Weighted sum approach (Zadeh 1963; Koski 1988; Coello 2000) is one among the most widely used classical aggregating methods. Weighted sum optimization scalarizes the objectives such that the sum of the weights should be equal to 1, i.e.,  $\sum_i^M W_i = 1$ , where  $M$  is the number of objectives. The decision for providing different weights to the objectives depends on the importance of the objectives in the optimization problem. Here are the possible ways one can assign weights to the objectives **Case a:** Knowing the preferences or the significance of each of the objectives in the problem, the decision maker can assign the



weights to the objectives. This is the most simplest or reasonable way of using the weighted sum method. Multi-objective real-world applications are mostly optimized by this method of accepting user given preferences.

**Case b:** But one may not always know the significance of the objectives. In that case, following are the options that can be used.

**Case b.1:** Assign random weights to the objectives. The uncertainty in the weights will affect the direction of optimization and also the time taken to converge towards the optimal solutions.

**Case b.2:** Provide equal weights to all the objectives (Steuer 1986; Bhuvana and Aravindan 2011a). An objective that does not deserve to get equal weight as others will mislead the optimization process. That is, the equal weight will help the optimization process towards wrong direction and hence will consume more than expected time to converge towards the optimal solution. Another aspect is the weights assigned will remain fixed during the entire evolution and may also have its impact over the diversity of solutions obtained.

**Case b.3:** Using adaptive weights for the objectives is the other way of weight assignment. Adaptive weights are weights that change across the iterations with respect to geometric nature of the Pareto front (Kim and De Weck 2005; Jiang et al. 2011) and with the help of nadir and utopia points (Gen and Lin 2005). Hamada et al. (2010) uses one weighted scalarization method, where the initial point and the weight decide the convergence. Weight is randomly generated initially for each point in the search space and no weight adaptation performed for first iteration. The weights are assigned as the midpoint between solutions using subdivision method for the new search points. The solution is optimized for the new weight and repeated until the movement between points is too short. Scalability (Hamada et al. 2010) is limited to six or seven objectives, since the number of search points increases exponentially with respect to the number of objectives. Above drawback was overcome using barycentric subdivision and new weight adaptation scheme in Hamada et al. (2011). In Li and Landa-Silva (2008) adaptive weights are assigned according to their nearness to non-dominated neighboring solution in the population. In general, the weighted sum aggregation approach has its own set of drawbacks. Jubril (2012) discusses the drawbacks of weighted sum such as:

1. Missing the nonconvex surface of Pareto front.
2. Diversity is not controlled.
3. Distribution depends on relative scaling of objectives.

Proposed algorithm has introduced one form of adaptive weights for scalarizing the objectives before performing the

local search. This adaptive weight strategy is scalable and hence not affected by increasing the number of objectives. Our adaptive weight assignment neither depends on the geometric nature of Pareto front, since it is not known in advance, nor it relies on the nadir or utopia points. Our proposed adaptive weight assignment algorithm overcomes the drawbacks listed earlier. The obtained results confirm that our proposed algorithm is not affected by any of the listed drawbacks.

### 2.3 Hybridization of evolutionary algorithms:

In MAs, the memes refer to the ‘unit of cultural evolution’ (Ong et al. 2006) that enhances or fine tunes an individual by a separate procedure. Therefore, the GA will be responsible for the exploration and the local search will take care of exploitation of local neighborhood. The main advantage of hybridization of Genetic Algorithms is to prevent premature convergence of optimization process (El-Mihoub et al. 2006). Memetic algorithms can be classified according to the types of algorithms used, level of hybridizations applied and type of inheritance performed. A survey about such classification can be found in Chen et al. (2011).

Challenges and issues in combining global and local search algorithms have been mentioned in Sect. 1. Local search (LS) is itself a separate optimization process and when it is embedded into another algorithm, the complexity of the procedure significantly increases. To reduce the complexity of the hybrid algorithm few solutions have been identified so far. LS cannot be applied on all individuals always and requires some selection strategy for filtering the individuals which can undergo the local search (Krasnogor and Smith 2005; Mongus et al. 2012). The frequency of performing local optimization during the evolutionary cycle needs to be decided. When the LS is applied on individuals, how deep the local optimization be allowed is one important issue to be addressed. Performing LS till every solution reaches the optimal value will definitely have its impact on total time consumed by the procedure. To overcome this, if the number of LS steps taken is restricted, then the fine tuning of good individuals may not happen. This requires a balanced and optimal way of integrating LS into EA to generate optimal solutions.

Having identified major issues while embedding a LS into EA, our paper proposes a memetic approach addressing how deep a local search can be applied in an optimal manner and what set of population can undergo the local search. The proposed algorithm also addresses how the weights can be assigned in an adaptive manner before applying LS. In our proposed algorithm the global search is performed by NSGA-II and local search is performed by steepest descent method.

### 3 Proposed algorithm

#### 3.1 Design of memetic algorithm

The goal of memetic algorithm is to generate quality solutions by combining global and local search together. The purpose of combining two different search processes is to perform exploration of the search space and exploitation of the neighborhood locality. Issues related to combining global search and local search have been listed and discussed in the previous section. Two major proposals in this paper are adaptive weight assignment scheme for performing local search and Preferential Local Search. These two key ideas can be incorporated into any of the global search algorithms to arrive at a new memetic algorithm. In this paper, we have integrated the above two approaches into NSGA-II arriving at a new algorithm, MAPLS-AW. Balance should be maintained, when a global search is integrated with a local search process. Our proposed algorithm maintains that balance between exploration and exploitation through preferential local search and using adaptive weight assignment scheme. Due to this, the explicit exploration parameter,  $\eta_c$  in Simulated binary crossover used by NSGA-II is not needed. We have eliminated the need of that parameter in our proposed algorithm and kept  $\eta_c$  as a constant. In the following subsections, we have introduced the working of adaptive weight assignment and Preferential Local Search.

#### 3.2 Adaptive weights

Using the weighted sum method, multi-objectives are combined together into single objective before the local optimization process is performed. This requires the weights to be assigned to the functional objectives. An adaptive weight mechanism has been introduced in this paper that dynamically adapts weights for the objectives.

Multi-objective problems with minimizing objectives are considered in this work, since any maximization problem can be converted (Runarsson and Yao 2000) into minimization problem.

Different cases of assigning weights have been discussed in Sect. 2.2. Uniformly distributed optimal solutions are the solutions expected out of the evolutionary process, where providing equal weights will affect the diversity of solutions obtained. Equal weights may never explore the extreme regions of the optimal front. Hence, we have decided to provide weights for the multiple objectives in an adaptive manner (Bhuvana and Aravindan 2011b). Classical aggregation techniques aggregate the objectives before the evolution begins and also does it only once. We are proposing a new aggregation method that aggregates during evolution and it is prudent to use information available in the objective space.

Adaptive weights are assigned by collecting information about a solution from its multidimensional objective space. Weights computed from these functional values will keep on changing during the course of evolutionary process.

The aim is to provide lesser preference for any objective which has larger functional value in a minimization problem. Higher preference can be given to sustain a lesser functional objective and vice versa. While associating weight in this manner to one objective function, we need to consider other objectives at the same time. If we do not consider other objectives then the evolution may take the solutions towards one particular region and make them crowded. This will affect the diversity of optimal solutions. Instead, we need to move an objective functional value towards its optimal minimum with respect to every other functional objectives in the search space. By this a solution is shifted proportionately towards the Pareto optimal. Proportionate movement in the objective space is achieved with the help of Euclidean norm.

If  $f_i^{(x)}$  is the  $i$ th functional objective of a solution  $x$ , then the proportionate movement is given by  $\omega_i$ .

$$\omega_i = \frac{f_i^{(x)}}{\|f^{(x)}\|} \quad (6)$$

where  $\|f^{(x)}\|$ , the Euclidean norm is given by

$$\|f^{(x)}\| = \sqrt{f_1^{2(x)} + f_2^{2(x)} + f_3^{2(x)} + \dots + f_M^{2(x)}} \quad (7)$$

Since sum of weights in the weighted sum aggregation approach should be equal to 1, following scaling of  $\omega_i$  is done:

$$\alpha_i = \frac{\omega_i}{\sum_{i=1}^M \omega_i} \quad (8)$$

Once the individual weights are determined for all the objectives, they are combined together into a single objective  $F$  and is given by,

$$F = \alpha_1 f_1 + \alpha_2 f_2 + \alpha_3 f_3 + \dots + \alpha_M f_M \quad (9)$$

where the sum of the weights,  $\alpha_1 + \alpha_2 + \alpha_3 + \dots + \alpha_M = 1$ .

Local search applied after such dynamic weight adaptation will overcome the drawbacks, such as convergence time, taking optimization in the wrong direction and diversity of obtained optimal solutions.

#### 3.3 Preferential Local Search (PLS)

The objective of integrating a local search in a global search process is to enhance the quality of solutions by fine tuning them. To establish a balance between exploration and

**Algorithm 1** Algorithm for Preferential Local Search(PLS)

---

```

1: procedure Preferential_Local_Search(Input_Solution Iold)
2:   ▷ Procedure to compute one-step Preferential Local Search to improve Iold to Inew
3:   for  $i \leftarrow 1$  to  $M$  do   ▷  $M$ , the number of objectives
4:     Compute proportionate weight  $\omega_i$  with respect to Iold, using Eq. 6
5:     Compute adaptive weight  $\alpha_i$  with respect to Iold using equation 8.
6:   end for
7:   Construct Single objective function F using Eq. 9.
8:   Compute local gradient  $\nabla F$  of Iold
9:   Identify new neighbor individual, Inew using Eq. 5
10:  Evaluate Fitness(Inew)
11:  if (Inew is better than Iold) then
12:    return Inew
13:  else
14:    return Iold
15:  end if
16: end procedure

```

---

exploitation, we are proposing Preferential Local Search (PLS). PLS addresses the issues related to combining global and local searches together. Issues addressed by PLS are choosing individuals for local search, deciding the depth of local search and determining the frequency of local search.

*Choosing individuals for LS*

Time incurred in allowing all the individuals for local search adds to the complexity of the memetic algorithm. Decision should be made to selectively allow a few for the local search. An individual in the population can be passed on to next generation only when it has potential enough to survive and compete with other peer solutions and offspring. In any  $i$ th generation where the population is a mix of varied set of solutions, PLS identifies the elite solutions. Preference can be given to such elite solutions to undergo local search (Krasnogor and Smith 2005; Mongus et al. 2012). These kinds of preferences to good solutions strengthen them to counter new offspring.

The offspring that are generated after the genetic operations may lose their chance in the evolution when compete with the potential parents. PLS selects the new offspring for depth-limited local search. That is, each solution will undergo at least one depth limited local search once they are newly generated. If they survive next generation, and has the potential to optimize further, local search will be deepened further.

*Depth of LS*

PLS is designed in such away to limit the depth of local search, that is, potential solutions will undergo depth-limited LS. Depth of LS is determined by predetermined number of steps. If potential solutions survive the next generation, local search will be deepened further. This way the local search is

continued and iteratively deepened on good solutions across generations. Lesser the potential of one candidate solution, lesser the depth of local searches applied on it. Greater the potential, deeper will be the local search applied on that solution across generations. The potential of a candidate depends on the fitness of that individual, which is associated with both exploration of the search space and exploitation of its neighborhood. We decided to use the steepest descent as local search method, since it suits the decision of limiting the depth of applying local search.

Thus PLS chooses individuals for depth-limited local search and fine tunes them. This fine tuning spreads across generations and iteratively deepens. These steps are collectively referred to as Preferential Local Search. Algorithm for PLS incorporating the adaptive weight computation is given in Algorithm 1.

## 3.4 Memetic algorithm with Preferential Local Search using adaptive weights (MAPLS-AW)

Having defined the adaptive weight (AW) procedure and the concept of Preferential Local Search (PLS), the actual proposed approach, memetic algorithm with Preferential Local Search using Adaptive Weights (MAPLS-AW), is given below. These two ideas are incorporated into NSGA-II and MAPLS-AW has been arrived at. Detailed procedure is given in Algorithm 2.

MAPLS-AW begins with random initialization of population according to problem specific boundary constraints. Fitness of every individual is evaluated and non-dominated sorting is applied. Crowding distance is computed for every individual. This distance helps in selecting individuals from less crowded region and maintains diversity of solutions across the Pareto front. Binary tournament selection chooses the parents and SBX crossover is applied to generate new offspring. In SBX crossover, the distribution parameter is nullified to explore the vast search space. Polynomial mutation

---

**Algorithm 2** Memetic algorithm with Preferential Local Search using Adaptive weights (MAPLS-AW)
 

---

```

1: procedure MAPLS- AW
2: ▷  $P_c$  is the current population of size  $P$ , FE is number of Fitness Evaluations,  $E_i$  is an Elite solution,  $O$  is the
   set of new offsprings
3:   Initialize the population to size  $P$ 
4:   Evaluate Fitness for every individual in population
5:    $FE \leftarrow P$                                      ▷ Initialize the fitness evaluations count
6:   Apply non-dominated sorting.
7:   Compute Crowding Distance
8:   while  $FE < FE_{MAX}$  do
9:      $O \leftarrow \{\}$                                    ▷ Initialize the set of Offspring to empty
10:    for  $k \leftarrow 1$  to  $P/2$  do
11:      Select Parents using Binary Tournament.
12:      Apply SBX with nullified  $\eta_c$  to generate two new offsprings  $O_1$  and  $O_2$ 
13:      Apply Polynomial Mutation on both  $O_1$  and  $O_2$  with probability,  $p_m = 1/n$ 
14:      ▷  $n$  is the number of decision parameters
15:      for  $j \leftarrow 1$  to 2 do
16:         $O_j \leftarrow \text{Preferential\_Local\_Search}(O_j)$ 
17:         $FE \leftarrow FE + 1$                              ▷ Update the number of FEs
18:         $O \leftarrow O \cup \{O_j\}$ 
19:      end for
20:    end for
21:     $P_c \leftarrow P_c \cup O$ 
22:    Apply non-dominated sorting and shrink the population size to  $P$ 
23:    Compute Crowding Distance
24:    Identify elite solutions,  $\langle E_1, E_2, \dots, E_s \rangle$ 
25:    for  $i \leftarrow 1$  to  $s$  do
26:       $E_i \leftarrow \text{Preferential\_Local\_Search}(E_i)$ 
27:       $FE \leftarrow FE + 1$                              ▷ Update the number of FEs
28:    end for
29:  end while
30: end procedure

```

---

with respect to mutation probability modifies the new offspring. Single-step Preferential Local Search is applied on the new offsprings. These locally optimized individuals are now merged with the parent population. Non-dominated sorting is applied on combined population and it is resized to meet the population size. Crowding distance is computed for the resized population. MAPLS-AW now identifies the elite or best individuals in the current population which will undergo Preferential Local Search (PLS). To apply PLS, the objectives of the individual are combined into single using Eq. (9). To combine the objectives using weighted sum method, the adaptive weights are calculated. LS will be continued on the same individual in the next generation if it is competent enough to survive. Fitness for the new individual is computed and replaces the original one in population. Steps from selection are repeated until maximum fitness evaluations reached. MAPLS-AW performs both exploration and exploitation in a balanced manner by nullifying the distribution parameter in the SBX crossover and by Preferential Local Search. MAPLS-AW computes the fitness only when a new individual is added to the population. By design of the algorithm, MAPLS-AW reduces the computation overhead of traditional memetic algorithm by limiting the depth of local search.

## 4 Implementation and experiments

To assess the performance of our proposed memetic algorithm with Preferential Local Search using adaptive weight (MAPLS-AW), we have carried out several experiments. Experimental results have been evaluated in comparison with other MOEAs. This section presents implementation details of MAPLS-AW, few variants, control parameters used, benchmark test functions, and performance criteria used for evaluation. Our proposed algorithm, MAPLS-AW, has adapted NSGA-II as its global search heuristic procedure and steepest descent as the local search procedure. We have followed the guidelines suggested in Črepinšek et al. (2012, 2014) to conduct the experiments in this work.

### 4.1 Parameters used

Initial population for the benchmark test problems are randomly generated. These random values are generated with respect to the boundary constraints of the decision parameters of a chosen benchmark test problem. The population size is set to 50. Termination condition has been fixed as 25000 fitness evaluations for all the variants of search algorithm. For the depth-limited Preferential Local Search, the step size



**Table 1** Memetic algorithm control parameters

Parameter	Value
Category	Elitist MOEA
Population size	50
Termination criteria	25,000 fitness evaluations
Coding	Real-coded decimal representation
Selection	Constrained binary tournament
Crossover probability	1
Crossover	SBX crossover With distribution index 1
Mutation probability	1/Number of variables
Mutation	Polynomial mutation
Step size, $\lambda$	0.001
Number of runs	50

is 0.001 and the depth of LS is set to 1. In order to compare the performance, we have used the same initial population for all the variants of the search algorithm. To observe the consistency in performance of the algorithms, all the implementations are run for 50 times with different initial population and the mean of results is reported here. Other parameters used for implementation are given in Table 1.

#### 4.2 Validation on benchmark problems

20 multi-objective benchmark test problems were used to validate our work. These are the test problems for which the optimal solutions are already known. To demonstrate the superiority of an EA over other algorithms, the new algorithm is applied over benchmark test problems (Huband et al. 2006) and the results are studied for evaluating the performance. As per Nguyen et al. (2012), benchmark test problems should exhibit flexibility, simplicity and efficiency in evaluating an algorithm and also should resemble real-world problems. Huband et al. (2006) have analyzed and reviewed a set of unconstrained test problems available in the literature. The test suites addressed are Deb’s toolkit, Zitzler’s ZDT test suite, Deb’s DTLZ test suite, and Van Veldhuizen’s tool kit. For validation, we have used fifteen unconstrained benchmark test problems in this paper. They are ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, SCH1, FONSECA, SCH2, DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, and DTLZ7. Among the ZDT series, ZDT3 has a disconnected Pareto front whereas the ZDT2 and ZDT6 have concave Pareto front each. Pareto fronts obtained from ZDT1 and ZDT4 are convex in nature. In DTLZ series, we have used seven DTLZ problems, where DTLZ1, DTLZ3 have multimodal Pareto fronts which are linear and concave, respectively. DTLZ2, DTLZ4, DTLZ5, DTLZ6 and DTLZ7 all have unimodal Pareto fronts whereas DTLZ4 has a concave Pareto front. DTLZ7 has got a disconnected Pareto front. In case of SCH1 and SCH2,

both have convex Pareto fronts and FONSECA has a concave Pareto front. The number of decision variables for these test problems differs and is in the range of 1 and 30. The DTLZ series of problems are considered with 3 functional objectives.

Apart from unconstrained problems, we have used five constrained test problems. They are BINH2, SRN, TNK, CTP1, and CTP2 (Deb 2001). The first three test problems have 2 decision variables each, whereas the last two have 4 decision variables each.

#### 4.3 Performance metrics

Quality of solutions determines the performance of any multi-objective optimization algorithm. Two criteria that were used to evaluate the performance of a multi-objective optimization are, how close the obtained Pareto optimal solutions are to the known optimal set and how diverse are the obtained solutions across the Pareto front. Performance metrics such as Generational Distance (GD), Spread, Max spread and HyperVolume Ratio (HVR) (Van Veldhuizen and Lamont 1999; Deb 2001) are used as measures of the above two criteria. All the above four metrics require the knowledge of the known Pareto optimal set P.

Generational Distance (GD) is the distance between the known optimal set of a problem and the obtained optimal set. It is given by

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}}{|Q|} \tag{10}$$

with  $p = 2$ ,  $d_i$  is the Euclidean distance between solutions of obtained set Q and the nearest member in the known optimal set, P. The second metric is Spread  $\Delta$ , that measures how well the solutions are distributed and its extent across the Pareto front.

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N - 1)\bar{d}} \tag{11}$$

where  $d_f$  and  $d_l$  are the distance between the extreme solutions of known and obtained optimal set;  $d_i$  is the Euclidean distance between the consecutive solutions of the obtained optimal set and  $\bar{d}$  is the mean of the above distance. The third performance metric is Max spread (Zitzler 1999). It measures the length of the diagonal of hyperbox formed by the extreme objective values in the obtained non-dominated set (Deb 2001). It is given by

$$\bar{D} = \sqrt{\frac{1}{M} \sum_{m=1}^M \left(\frac{\max_{i=1}^{|Q|} f_m^i - \min_{i=1}^{|Q|} f_m^i}{F_m^{\max} - F_m^{\min}}\right)^2} \tag{12}$$

$F_m^{\max}$  and  $F_m^{\min}$  are the maximum and minimum value of  $m$ th objective in obtained optimal solution set. The final measure

is the HyperVolume Ratio (HVR), which computes the ratio of HV between obtained Pareto set and known Pareto set, where HV calculates the volume covered by the Pareto set (Van Veldhuizen 1999)

$$\text{HVR} = \frac{\text{HV}(Q)}{\text{HV}(P)}$$

$$\text{HV} = \text{volume} \left( \bigcup_{i=1}^{|Q|} v_i \right) \quad (13)$$

where  $v_i$  is a hypercube formed between the chosen reference point and solutions of  $Q$ . Among the four performance metrics used, the first one, GD is the metric that measures the closeness criteria and Spread and Max spread test the diversity measure, whereas the Hypervolume metric measures both the closeness and diversity criteria.

Smaller value of GD metric shows that the obtained non-dominated Pareto points are closer to the known Pareto solutions. That is, the distance between the obtained and known optimal solutions is small, which is the desired one for any multi-objective evolutionary algorithm. For Spread, lesser the value, uniform will be the Spread and for Max spread metrics, higher the value better the distribution that covers the extreme regions of the Pareto front. Higher the HVR metric, larger the volume covered by the optimal solutions and closer towards the optimal front. Performance of proposed MAPLS-AW was evaluated using the above four metrics by applying the algorithm on 15 unconstrained and 5 constrained benchmark multi-objective problems.

#### 4.4 Experiments conducted

Different experiments were conducted to observe the performance of our proposed algorithm MAPLS-AW. The objectives of these experiments are as follows:

- To demonstrate the overall performance of proposed MAPLS-AW algorithm as a Multi-objective evolutionary algorithm.
- To study the effectiveness of Preferential Local Search;
- To study the performance of assigning adaptive weights to multi-objectives while combining them into a single objective.

We have carried out three sets of experiments by implementing different variants of search algorithms. They are:

1. Our proposed MAPLS-AW algorithm, that combines Preferential Local Search and adaptive weight mechanism.
2. TLS-EW, traditional hybrid algorithm with depth-limited local search for 10 steps on every solution in the popula-

tion. To perform local search, multiple objectives are combined into single objective using equal weights.

3. TLS-AW, traditional hybrid algorithm with depth-limited local search for 10 steps on every solution in the population but uses adaptive weights.
4. Two elitist multi-objective evolutionary algorithms: NSGA-II and SPEA2.

All the variations of search algorithms were implemented in C programming language in Intel(R) Core(TM)i5-3470 CPU @3.20GHZ system. The performance of all the abovelisted algorithms was evaluated by applying them on a set of benchmark test suite and by analyzing the metrics obtained. Experiments have been carried out in the following order,

1. The first set of experiments were conducted to study the overall performance of MAPLS-AW as a memetic or hybrid MOEA and compared with NSGA-II and SPEA2.
2. NSGA-II, TLS-EW and TLS-AW were compared to study the performance of memetic algorithm and to analyze the nature of adaptive weights.
3. TLS-AW and MAPLS-AW were compared to study the collective performance of adaptive weights and preferential local search.

The above three experiments can be categorized into two, first that evaluates the performance of MAPLS-AW by comparing it with NSGA-II & SPEA2 and second category includes the experiments 2 and 3. Experiment 1 used same initial population to perform the evaluation on MAPLS-AW, NSGA-II, and SPEA2. For each trial of experiments 2 and 3, we have used same initial population to evaluate the variants TLS-EW, TLS-AW and to compare them with NSGA-II and MAPLS-AW. This is repeated for 50 trials, where each trial used different seed to generate different initial population. The implementations of existing algorithms, NSGA-II and SPEA2, are checked for correctness by statistically comparing their results with those of Deb et al. (2002). There are 7 common problems reported in Deb et al. (2002) and in our paper. Independent two-tailed  $t$  test was applied on Spread metric obtained for those 7 problems to check whether our implementation was error free. From the obtained  $t$  stat value, 0.330977 and the  $p$  value, 0.74823, we infer that solutions obtained by both implementations are similar with no significant difference in means and hence our implementation of NSGA-II is inferred to be error free.

Quality Metrics were computed on the final population for all the experiments carried out. Percentage of improvement was calculated on the results obtained from different variant algorithms. We have tested the data normality and homoscedasticity of the results obtained, using Chi-square test. After verifying that data normality and homoscedasticity hold good on the obtained results, we have applied

ANOVA test, since multiple comparisons have been carried out. From the F values of ANOVA test, we have verified that the populations under comparison are significantly different. We then applied *t* test, whose *p* values are adjusted by Holm–Bonferroni correction to control Type-I error (Veček et al. 2014). The tables reporting the *p* values are corrected with respect to the number of comparisons made in that context.

#### 4.5 Performance of proposed algorithm: MAPLS-AW

To demonstrate the overall performance of our proposed MAPLS-AW as a multi-objective evolutionary algorithm, we have carried out the first set of experiments. This section presents the comparison of performances between our proposed MAPLS-AW and the two other multi-objective evolutionary algorithms, NSGA-II and SPEA2. The mean of the four performance metrics such as Generational Distance (GD), Spread, Max spread, and HyperVolume Ratio (HVR) were observed.

Table 2 shows the performance of NSGA-II, SPEA2, and MAPLS-AW for the benchmark unconstrained and constrained problems through different metrics. Values tabulated are the mean values computed after 50 runs. Table 3 shows the superiority of the proposed MAPLS-AW algorithm through percentage of improvement on the four met-

rics between NSGA-II, SPEA2, and MAPLS-AW for all the test problems.

Performance of MAPLS-AW algorithm under GD metric is better than other two MOEAs, NSGA-II and SPEA2. The highest difference has been observed in DTLZ7 with 68 % of improvement and 46.5 % of improvement in DTLZ4 test problem over NSGA-II. In case of SPEA2, MAPLS-AW has shown a maximum of 76.2 and 54 % of improvement in DTLZ3 and DTLZ7, respectively. All these problem instances in which the MAPLS-AW outperforms are multidimensional problems with 3 objectives each. The non-dominating points generated by our algorithm are more closer to the known Pareto front than the other two algorithms. This is because of the exploration capability exhibited by the hybrid nature of the algorithm. Advantage of using weighted sum in the local search has guided the optimization properly and realized through the better convergence towards known Pareto.

For the Spread metric, the empirical values show a largest improvement of about 77.4 % for DTLZ1 test problem over NSGA-II algorithm. This reveals the exploration ability of our algorithm with constrained problems and ability to generate evenly distributed non-dominating points across the Pareto front. The second best value of Spread was observed with TNK for about 75.5 % when compared with NSGA-II and in SCH2 89.8 % of improvement is shown over SPEA2.

**Table 2** Performance metrics of NSGA-II, SPEA2, and MAPLS-AW

Test problem	NSGA-II				SPEA2				MAPLS-AW			
	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR
ZDT1	0.1511	0.7076	0.8079	0.2001	0.1435	0.5354	0.5966	0.0677	0.1359	0.2163	0.8868	0.2841
ZDT2	0.1607	0.9450	0.8134	0.2127	0.1403	0.8408	0.8188	0.1746	0.1358	0.7453	0.8722	0.6262
ZDT3	0.1558	0.1308	0.7857	0.2311	0.1548	0.1461	0.1380	0.1816	0.1529	0.1172	0.8421	0.3167
ZDT4	0.1595	0.4642	0.5811	0.1164	0.1818	0.4533	0.7459	0.1226	0.1489	0.5100	0.9673	0.2598
ZDT6	0.1715	0.5572	0.7477	0.0650	0.1846	0.5767	0.2693	0.0207	0.1608	0.5200	0.9131	0.0807
DTLZ1	0.1581	0.2513	0.6632	0.0282	0.1368	0.1493	0.6315	0.0332	0.1046	0.0568	0.7661	0.0967
DTLZ2	0.1858	0.1289	0.6026	0.5484	0.2801	0.2420	0.5831	0.5287	0.1505	0.0621	0.7103	0.6376
DTLZ3	0.1480	0.1187	0.5319	0.0020	0.5781	0.0861	0.1721	0.0013	0.1377	0.2732	0.8954	0.0027
DTLZ4	0.4286	0.2148	0.4942	0.6199	0.3544	0.2356	0.7248	0.5673	0.2294	0.1577	0.8756	0.6506
DTLZ5	0.2639	0.8144	0.6438	0.1543	0.3213	0.9804	0.0200	0.6754	0.2269	0.3564	0.9240	0.7696
DTLZ6	0.2867	0.5955	0.8122	0.2115	0.2463	0.7006	0.4235	0.5634	0.2338	0.5462	0.8637	0.7581
DTLZ7	0.7759	0.0519	0.7046	0.1264	0.5394	0.0699	0.8501	0.0602	0.2480	0.0353	0.9419	0.7764
SCH1	0.1859	0.5152	0.9048	0.0385	0.1533	0.4303	0.9009	0.1182	0.1384	0.2352	0.9797	0.7607
SCH2	0.1590	0.0276	0.5766	0.0178	0.1710	0.1858	0.6024	0.0178	0.1531	0.0190	0.6688	0.0896
FONSECA	0.2051	0.1825	0.4945	0.5175	0.2028	0.4053	0.5893	0.5610	0.1928	0.1537	0.7381	0.7024
SRN	0.1459	0.3344	0.3714	0.0015	0.1484	0.1993	0.4327	0.0012	0.1408	0.1813	0.4981	0.0023
TNK	0.1953	0.4524	0.2175	0.2313	0.1684	0.1566	0.2008	0.3358	0.1625	0.1107	0.2525	0.6361
CTP1	0.1410	0.9527	0.9197	0.2632	0.1412	0.6898	0.9333	0.4639	0.1341	0.6532	0.9456	0.7769
CTP2	0.1598	0.1159	0.8599	0.7097	0.1662	0.2188	0.8716	0.6151	0.1469	0.0443	0.9168	0.7903
BINH2	0.1372	0.1392	0.8576	0.0011	0.1424	0.4534	0.8831	0.0039	0.1325	0.0903	0.9244	0.0072

**Table 3** Percentage of improvement in metrics between NSGA-II, SPEA2, and MAPLS-AW

Test problem	MAPLS-AW over NSGA-II				MAPLS-AW over SPEA2			
	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR
ZDT1	10.1	69.4	8.9	29.6	5.3	59.6	32.7	76.2
ZDT2	15.5	21.1	6.7	66.0	3.2	11.4	6.1	72.1
ZDT3	1.9	10.4	6.7	27.0	1.2	19.8	83.6	42.6
ZDT4	6.7	9.0	39.9	55.2	18.1	11.1	22.9	52.8
ZDT6	6.3	6.7	18.1	19.4	12.9	9.8	70.5	74.3
DTLZ1	33.8	77.4	13.4	70.8	23.5	62.0	17.6	65.7
DTLZ2	19.0	51.9	15.2	14.0	46.3	74.4	17.9	17.1
DTLZ3	7.0	56.6	40.6	24.8	76.2	68.5	80.8	52.2
DTLZ4	46.5	26.6	43.6	4.7	35.3	33.0	17.2	12.8
DTLZ5	14.0	56.2	30.3	79.9	29.4	63.6	97.8	12.2
DTLZ6	18.5	8.3	6.0	72.1	5.1	22.0	51.0	25.7
DTLZ7	68.0	32.0	25.2	83.7	54.0	49.5	9.7	92.2
SCH1	25.6	54.3	7.6	94.9	9.7	45.3	8.0	84.5
SCH2	3.7	31.0	13.8	80.1	10.4	89.8	9.9	80.2
FONSECA	6.0	15.8	33.0	26.3	4.9	62.1	20.2	20.1
SRN	3.5	45.8	25.4	36.4	5.1	9.0	13.1	47.7
TNK	16.8	75.5	13.9	63.6	3.5	29.3	20.5	47.2
CTP1	4.9	31.4	2.7	66.1	5.1	5.3	1.3	40.3
CTP2	8.1	61.8	6.2	10.2	11.6	79.7	4.9	22.2
BINH2	3.4	35.1	7.2	84.2	6.9	80.1	4.5	45.6

Showing such high amount of improvement for a test problem with a discontinuous Pareto front clearly states that adaptive weighted sum in our algorithm has overcome the primary drawback of weighted sum approaches. ZDT4 is a complex test problem with 21<sup>9</sup> local Pareto solutions (Deb 2001). Our proposed algorithm MAPLS-AW has shown an improvement in Spread metric of about 9% over NSGA-II and 11% over SPEA2 for this test problem. This substantiates the strength of MAPLS-AW to explore and exploit the search space and ability to generate Pareto optimal solutions with better Spread across the Pareto front.

Max spread metric obtained for proposed MAPLS-AW algorithm has shown a 43.6% of improvement over NSGA-II in DTLZ4. This test problem has its Pareto front which is concave in nature. 97.8% of improvement for this metric over SPEA2 has been observed in DTLZ5 problem, which has a convex Pareto front. These observations clearly show that adaptive weighted sum approach in our MAPLS-AW algorithm has not missed out the convex and nonconvex parts of Pareto front.

For HVR, 94.9% of improvement was observed for SCH1 test problem over NSGA-II, which has shown the amount of area covered under the optimal front by the proposed MAPLS-AW. 92.2% of improvement has been observed for DTLZ7 test problem over SPEA2. The increase in performance was due to the exploitation of the neighborhood of potential solutions by hybrid MAPLS-AW search algorithm.

One-tailed  $t$  test was applied on all the 50 samples of obtained metric values for the three algorithms, NSGA-II, SPEA2, and MAPLS-AW. Computed  $t$  stat values and adjusted  $p$  values are reported in Table 4. The number of comparisons involved here is two and the  $p$  values reported in Table 4 are after Holm–Bonferroni correction. For both GD and  $\Delta$  metrics, computed  $t$  stat values are larger than  $t$  critical value and from the sign of  $t$  value, we infer that the means of GD and  $\Delta$  metric obtained by NSGA-II are greater than means of those metrics obtained by MAPLS-AW.

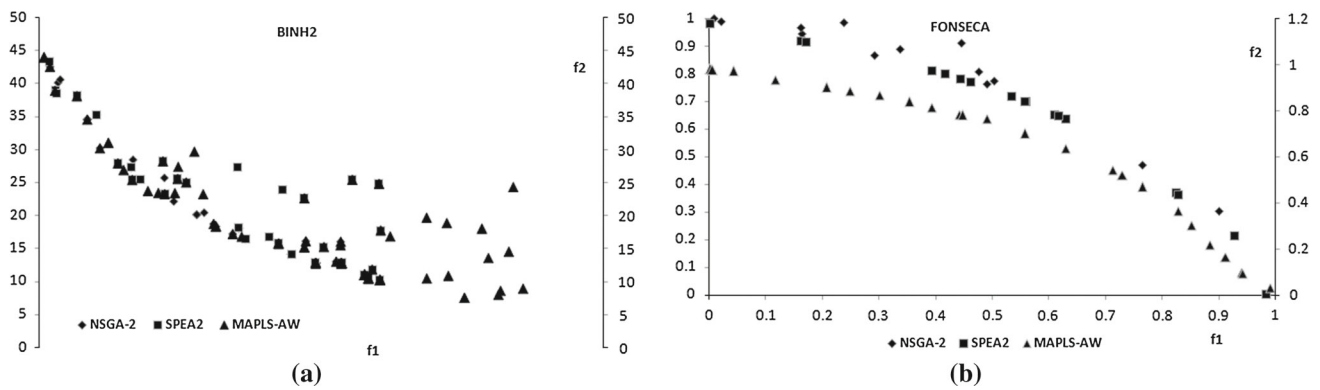
For the Max spread and HVR metrics, the absolute  $t$  stat values are greater than  $t$  critical value and from the sign of  $t$  value, we infer that means of these two metrics of MAPLS-AW are greater than NSGA-II. These are supported by corresponding  $p$  values, which are less than 0.05, implying a high confidence on the inferences made.

Finally we did a  $t$  test on the means of performance metrics obtained from 20 test problems. 2.0785, 3.465,  $-5.53$ , and  $-4.211$  are the  $t$  stat values obtained for GD,  $\Delta$ , MS, and HVR when NSGA-II and MAPLS-AW are compared. Their corresponding adjusted  $p$  values are 4.15E-02, 2.59E-03, 2.47E-05, and 4.72E-04. Computed  $t$  stat values between MAPLS-AW and SPEA2 for all the test problems are, 2.557, 3.5,  $-4.05$ , and  $-4.355$  and their corresponding adjusted  $p$  values are 1.93E-02, 2.39E-03, 6.78E-04, and 3.40E-04, respectively. From this, we infer that MAPLS-AW is in general better than NSGA-II and SPEA2.

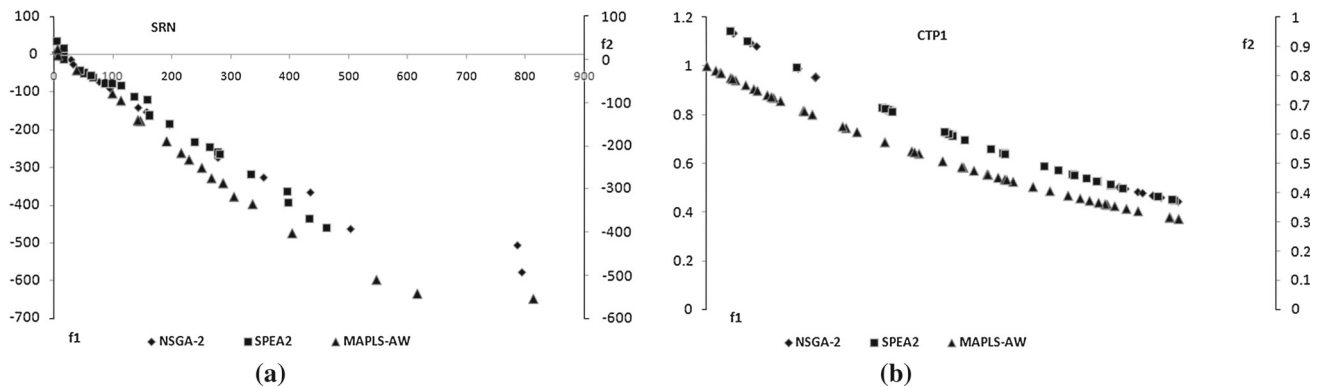


**Table 4**  $t$  stat and adjusted  $p$  values of  $t$  test when applied on performance metrics of NSGA-II, SPEA2, and MAPLS-AW

Test problem	NSGA-II with MAPLS-AW				SPEA2 with MAPLS-AW			
	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR
ZDT1	11.443, 9.5E-16	45.003, 8.0E-42	-2.588, 6.3E-03	-8.796, 6.0E-12	12.588, 5.7E-17	46.114, 5.0E-42	-4.593, 3.1E-05	-72.724, 1.4E-51
ZDT2	12.245, 1.6E-16	24.309, 5.4E-29	-6.281, 8.6E-08	-16.703, 7.2E-22	6.855, 5.6E-09	7.153, 1.9E-09	-2.622, 5.8E-03	-15.878, 3.0E-21
ZDT3	2.633, 5.6E-03	2.157, 1.8E-02	-5.723, 6.2E-07	-4.813, 7.3E-06	2.601, 1.2E-02	2.239, 3.0E-02	-36.157, 2.7E-37	-7.299, 1.1E-09
ZDT4	4.143, 6.8E-05	7.478, 6.0E-10	-10.486, 2.0E-14	-4.543, 3.6E-05	13.374, 5.6E-18	3.633, 6.7E-04	-32.329, 1.0E-34	-3.970, 2.3E-04
ZDT6	2.214, 1.6E-02	2.278, 1.4E-02	-11.297, 1.5E-15	-2.437, 9.3E-03	10.635, 2.5E-14	29.094, 1.4E-32	-12.849, 2.6E-17	-6.310, 7.8E-08
DTLZ1	35.940, 3.6E-37	17.530, 9.5E-23	-2.470, 8.5E-03	-6.627, 2.5E-08	111.780, 1.1E-60	3.213, 1.2E-03	-4.133, 1.4E-04	-6.018, 2.2E-07
DTLZ2	22.493, 8.9E-28	5.037, 3.4E-06	-2.825, 6.8E-03	-1.986, 2.6E-02	129.612, 8.2E-64	5.711, 6.5E-07	-2.546, 7.0E-03	-2.419, 9.7E-03
DTLZ3	58.058, 7.7E-47	4.485, 2.2E-05	-8.827, 5.4E-12	-1.943, 2.9E-02	8.162, 5.4E-11	4.979, 8.3E-06	-31.321, 4.6E-34	-5.237, 3.4E-06
DTLZ4	3.098, 1.6E-03	2.186, 1.7E-02	-8.628, 2.1E-11	-0.749, 0.0E+00	4.234, 1.0E-04	3.915, 2.8E-04	-3.479, 5.3E-04	-2.176, 1.7E-02
DTLZ5	2.176, 1.7E-02	31.673, 1.4E-34	-14.607, 8.8E-20	-16.904, 4.4E-22	4.165, 1.3E-04	101.858, 1.1E-58	-105.216, 2.2E-59	-2.593, 1.3E-02
DTLZ6	2.765, 4.0E-03	5.449, 8.2E-07	-1.857, 3.5E-02	-20.162, 2.3E-25	12.260, 1.5E-16	13.892, 1.3E-18	-9.343, 1.8E-12	-8.472, 3.7E-11
DTLZ7	10.492, 2.0E-14	4.023, 9.9E-05	-7.837, 3.4E-10	-41.798, 2.7E-40	16.458, 1.3E-21	6.379, 6.1E-08	-5.309, 1.3E-06	-52.460, 5.1E-45
SCH1	56.617, 2.6E-46	10.844, 1.3E-14	-3.988, 1.1E-04	-35.434, 1.4E-36	6.077, 8.9E-08	8.071, 7.4E-11	-4.344, 7.0E-05	-23.965, 1.0E-28
SCH2	5.305, 1.3E-06	2.219, 1.6E-02	-7.201, 3.2E-09	-11.971, 1.8E-16	7.925, 2.5E-10	6.481, 4.2E-08	-4.775, 8.3E-06	-11.991, 1.7E-16
FONSECA	4.206, 1.1E-04	2.370, 1.1E-02	-8.552, 2.8E-11	-3.837, 1.8E-04	3.371, 7.3E-04	13.435, 4.7E-18	-5.823, 2.2E-07	-3.617, 3.5E-04
SRN	4.812, 7.3E-06	9.520, 5.0E-13	-5.026, 7.1E-06	-8.990, 6.1E-12	8.260, 7.7E-11	26.542, 9.8E-31	-2.300, 1.3E-02	-7.317, 1.1E-09
TNK	27.304, 2.7E-31	28.545, 3.4E-32	-3.064, 1.8E-03	-11.868, 5.1E-16	3.608, 3.6E-04	3.925, 1.4E-04	-4.526, 3.8E-05	-10.632, 2.5E-14
CTP1	3.315, 8.6E-04	6.418, 5.3E-08	-2.409, 2.0E-02	-22.353, 2.3E-27	3.781, 4.3E-04	6.077, 8.9E-08	-1.917, 3.1E-02	-7.682, 2.9E-10
CTP2	23.149, 2.4E-28	13.382, 2.8E-18	-2.825, 3.4E-03	-3.305, 8.9E-04	13.859, 1.4E-18	11.488, 1.7E-15	-6.255, 9.4E-08	-9.245, 2.5E-12
BINH2	18.401, 1.2E-23	7.745, 2.3E-10	-2.239, 1.5E-02	-13.177, 1.0E-17	11.299, 1.5E-15	8.572, 2.6E-11	-7.223, 3.0E-09	-7.231, 2.9E-09



**Fig. 1** Pareto fronts of Unconstrained benchmark problems. **a** Pareto front of BINH2. **b** Pareto front of FONSECA



**Fig. 2** Pareto fronts of Constrained benchmark problems. **a** Pareto front of SRN. **b** Pareto front of CTP1

The sample Pareto fronts shown in Figs. 1, 2, and 3 pictorially depict, how the obtained non-dominant solutions are wide spread and distributed across the Pareto front. The Pareto points were uniformly distributed and were not crowded in one region.

The results obtained reveal that the MAPLS-AW exploits the neighborhood region of potential parent solutions with better exploration of search space. The adaptive weights provide the functional objectives with the weight they deserve and take the optimization process in the proper direction. Hence the unnecessary random searching has been avoided here.

#### 4.6 Performance of adaptive weights and Preferential Local Search

We have established that memetic algorithm MAPLS-AW works better than global search algorithm like NSGA-II and SPEA2. We have introduced two new features in our work, adaptive weights and Preferential Local Search, the following two subsections will discuss the individual performances of these two contributing features of MAPLS-AW. To enable this study, couple of variants of search algorithms TLS-EW and TLS-AW were implemented and experiments were car-

ried out to compare their performances. Same initial population was used to compare the variants.

##### 4.6.1 Adaptive weights

This section discusses about the performance of two variant memetic algorithms, TLS-EW and TLS-AW over a global search algorithm like NSGA-II. The objective is to study the performance of adaptive weights used to combine multiple objectives into a single objective.

TLS-EW, a hybrid or memetic algorithm was implemented with an integrated local search (LS), where the LS has been depth limited to 10 steps. The multi-objectives of the test problems were combined into a single objective problem on which the LS was applied. Weighted sum approach was used where the multiple objectives were assigned with equal weights. The mean of the four performance metrics such as Generational Distance (GD), Spread, Max spread, and HyperVolume Ratio (HVR) and time consumed by each of the algorithms were computed.

Table 5 shows the performance difference quantified in terms of metrics between NSGA-II and TLS-EW for the benchmark unconstrained and constrained problems. Values tabulated are the mean values computed after 50 runs.

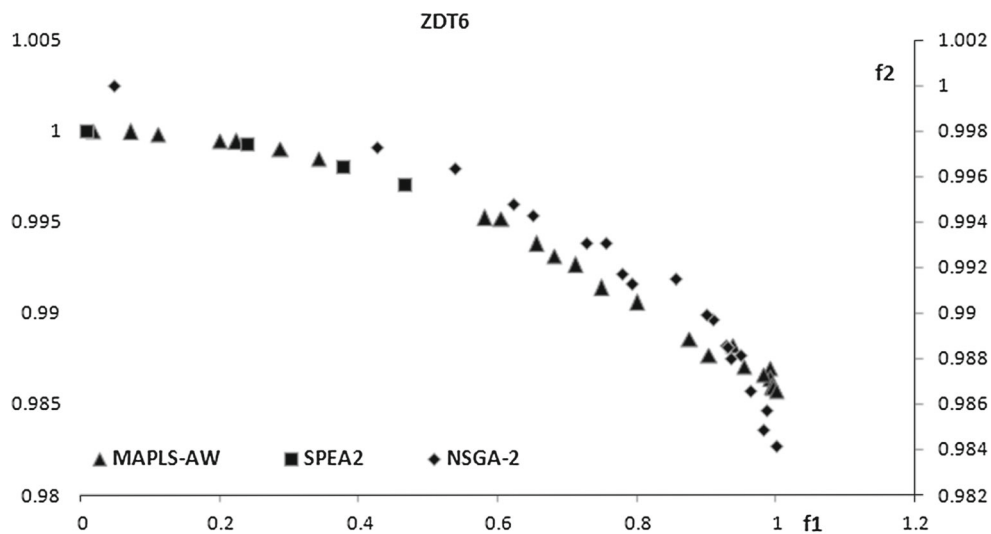


Fig. 3 Pareto front of ZDT6, an unconstrained benchmark problem

Table 5 Performance metrics of NSGA-II, TLS-EW, TLS-AW, and MAPLS-AW

Test problem	NSGA-II				TLS-EW				TLS-AW				MAPLS-AW			
	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR
ZDT1	0.1776	0.9287	0.7608	0.5196	0.1551	0.8872	0.8737	0.6934	0.1489	0.5133	0.9246	0.7487	0.1435	0.4669	0.9457	0.8133
ZDT2	0.1743	0.9362	0.8265	0.1452	0.1453	0.8932	0.8564	0.1722	0.1423	0.8418	0.9085	0.2515	0.1382	0.6918	0.9595	0.5570
ZDT3	0.1792	0.3474	0.8641	0.3075	0.1547	0.1883	0.9161	0.3259	0.1505	0.1261	0.9267	0.6041	0.1469	0.0862	0.9397	0.7078
ZDT4	0.1761	0.4082	0.8895	0.1366	0.1576	0.4413	0.9070	0.1209	0.1473	0.4773	0.9303	0.1659	0.1392	0.6895	0.9582	0.1890
ZDT6	0.2030	0.6101	0.8483	0.1505	0.1532	0.5819	0.8625	0.2003	0.1471	0.5791	0.9196	0.2522	0.1440	0.5152	0.9376	0.4941
DTLZ1	0.6139	0.3351	0.4925	0.0311	0.5371	0.2663	0.7537	0.0913	0.4337	0.1386	0.8696	0.1391	0.3285	0.0501	0.9389	0.2940
DTLZ2	0.2438	0.8534	0.2295	0.0854	0.1489	0.7872	0.2580	0.1430	0.1155	0.7288	0.3573	0.1882	0.0684	0.5909	0.7340	0.6061
DTLZ3	0.6647	0.1539	0.5772	0.0020	0.5580	0.0911	0.6712	0.0092	0.4465	0.0598	0.7536	0.0624	0.3353	0.0336	0.8402	0.2511
DTLZ4	0.3412	0.2966	0.4294	0.1230	0.2843	0.1899	0.5580	0.1818	0.2488	0.1353	0.6564	0.3194	0.1852	0.0966	0.7927	0.3463
DTLZ5	0.5067	0.4052	0.6016	0.0706	0.4174	0.2469	0.7730	0.1063	0.3482	0.1698	0.8125	0.1164	0.2864	0.1138	0.8682	0.2953
DTLZ6	0.5612	0.1881	0.5285	0.0742	0.4643	0.1215	0.6731	0.1007	0.3500	0.0936	0.7709	0.1243	0.3060	0.0487	0.8319	0.1808
DTLZ7	0.2972	0.1635	0.5465	0.0474	0.1991	0.1159	0.7109	0.1201	0.1183	0.0645	0.7993	0.1415	0.0362	0.0436	0.8683	0.2620
SCH1	0.1756	0.6099	0.2644	0.0033	0.1547	0.3432	0.6962	0.0470	0.1439	0.0567	0.7904	0.0861	0.1345	0.0296	0.9035	0.1246
SCH2	0.1848	0.1023	0.2442	0.0221	0.1793	0.0756	0.2644	0.0619	0.1750	0.0489	0.5470	0.1255	0.1547	0.0296	0.6962	0.2256
FONSECA	0.2497	0.5188	0.8182	0.3382	0.2190	0.4017	0.9095	0.5256	0.2083	0.3309	0.9673	0.6063	0.1930	0.2128	0.9844	0.7661
SRN	0.1647	0.1712	0.2549	0.0014	0.1546	0.1063	0.3194	0.0103	0.1501	0.0899	0.3624	0.0159	0.1434	0.0529	0.4046	0.1273
TNK	0.1979	0.0846	0.1848	0.3935	0.1632	0.0694	0.1914	0.4544	0.1515	0.0591	0.1979	0.5033	0.1011	0.3129	0.2325	0.5759
CTP1	0.1671	0.1710	0.9024	0.2712	0.1618	0.1146	0.9360	0.3698	0.1548	0.0634	0.9588	0.4919	0.1431	0.0326	0.9753	0.6298
CTP2	0.1824	0.5650	0.7305	0.4317	0.1734	0.4409	0.8470	0.5174	0.1695	0.2792	0.9005	0.6761	0.1375	0.1568	0.9475	0.8694
BINH2	0.1639	0.6825	0.3553	0.0068	0.1506	0.3430	0.4478	0.0102	0.1460	0.2893	0.6132	0.0129	0.1341	0.1656	0.9077	0.0329

The percentage of improvement in the magnitude of the four metrics between TLS-EW and NSGA-II for all the unconstrained and constrained test problems is tabulated in Table 6. The increase in performance was due to the fine tuning through integrated LS performed on all the solutions by

TLS-EW and substantiates the superiority of TLS-EW over NSGA-II.

To study the performance of adaptive weights, we implemented TLS-AW, where the LS has been depth limited to 10 steps. Adaptive weights are calculated for each solution

**Table 6** Percentage of improvement in metrics between NSGA-II, TLS-AW, TLS-AW and MAPLS-AW

Test problem	TLS-EW over NSGA-II				TLS-AW over TLS-EW				MAPLS-AW over TLS-AW			
	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR	GD	$\Delta$	MS	HVR
ZDT1	12.68	4.47	12.93	25.07	3.99	42.14	5.51	7.39	3.67	9.04	2.23	7.94
ZDT2	16.65	4.59	3.49	15.70	2.06	5.76	5.74	31.52	2.85	17.81	5.31	54.85
ZDT3	13.69	45.78	5.68	5.66	2.68	33.02	1.14	46.05	2.41	31.67	1.39	14.65
ZDT4	10.52	7.50	1.92	11.49	6.55	7.54	2.51	27.16	5.49	30.77	2.92	12.19
ZDT6	24.52	4.63	1.64	24.83	3.99	0.48	6.21	20.59	2.10	11.03	1.92	48.97
DTLZ1	12.52	20.51	34.65	65.94	19.25	47.96	13.33	34.34	24.25	63.86	7.38	52.69
DTLZ2	38.93	7.75	11.07	40.31	22.41	7.42	27.77	24.02	40.80	18.92	51.32	68.95
DTLZ3	16.05	40.80	14.00	77.81	19.98	34.32	10.94	85.33	24.91	43.92	10.31	75.15
DTLZ4	16.66	35.99	23.04	32.32	12.50	28.75	14.99	43.08	25.56	28.56	17.19	7.77
DTLZ5	17.63	39.06	22.18	33.54	16.57	31.22	4.86	8.69	17.75	32.97	6.41	60.58
DTLZ6	17.27	35.43	21.48	26.34	24.62	22.97	12.68	18.99	12.57	48.00	7.33	31.23
DTLZ7	32.99	29.10	23.12	60.52	40.60	44.41	11.05	15.09	69.38	32.41	7.95	46.02
SCH1	11.88	43.72	62.02	93.06	7.03	83.47	11.91	45.36	6.52	47.74	12.52	30.87
SCH2	3.00	26.13	7.64	64.26	2.36	35.31	51.65	50.67	11.61	39.35	21.44	44.37
FONSECA	12.30	22.57	10.04	35.66	4.92	17.63	5.98	13.31	7.31	35.69	1.73	20.86
SRN	6.09	37.92	20.18	86.41	2.94	15.43	11.87	35.41	4.49	41.12	10.44	87.47
TNK	17.52	17.94	3.44	13.41	7.15	14.93	3.29	9.71	33.25	81.13	14.85	12.61
CTP1	3.16	32.96	3.59	26.68	4.33	44.69	2.37	24.81	7.54	48.57	1.69	21.90
CTP2	4.94	21.97	13.76	16.55	2.27	36.68	5.94	23.48	18.85	43.82	4.96	22.23
BINH2	8.08	49.74	20.66	33.18	3.05	15.67	26.97	20.57	8.18	42.75	32.45	60.77

according to their positional values in objective space using Eqs. 6, 8, and 9. Table 6 shows the difference in performance between TLS-EW and TLS-AW for the benchmark unconstrained and constrained problems. Looking at the performance of TLS-AW variant of search algorithm under GD metric, results show that it performs better in all the test problems with a lesser GD value compared to the equal weight version TLS-EW. There was a 40.6 % of percentage of improvement for DTLZ7 problem shown in Table 6 and a minimum of 2.06 % was observed for ZDT2. This shows that the adaptive weight variant of the memetic algorithm really have taken the optimization towards the known optimal solutions. The metrics computed for Spread and Max spread show that TLS-AW outperforms TLS-EW on all the benchmark problems. Spread shows an improvement from 7.42 to 83.47 %, where for the Max spread the percentage of improvement is in the range 1.14–51.65 %. The increase in performance is due to the exploitation of the neighborhood of potential solutions by hybrid TLS-AW search algorithm. Under HVR metric, the DTLZ3 test problem has shown a highest percentage of improvement of 85.33 % than any other problem in the test suite.

One-tailed  $t$  test was applied and the computed  $t$  stat values are given in Table 7 along with the  $p$  values. The  $p$  values are reported after the Holm–Bonferroni adjustment. For the GD

and  $\Delta$  metrics, we can infer that means of metrics obtained by TLS-EW are less than those of NSGA-II and means of metrics obtained by TLS-AW are less than those of TLS-EW. For MS and HVR metrics, the absolute  $t$  stat values are greater than  $t$  critical value and from the sign of  $t$ -stat values, we have inferred that means of these metrics obtained by NSGA-II are lesser than those of TLS-EW and means of metrics obtained by TLS-EW are lesser than those of TLS-AW.

#### 4.6.2 Preferential Local Search

The second feature contributing to the working of MAPLS-AW is the application of Preferential Local Search. To study the effectiveness of Preferential Local Search, we have conducted the experiments on variant search algorithms with PLS and without PLS. This section discusses the results obtained from proposed MAPLS-AW and hybrid algorithm TLS-AW. Former has Preferential Local Search along with adaptive weight mechanism whereas the later has got fixed 10-step local search incorporated using adaptive weights but without PLS feature. This is the third category of experiment carried out where the mean of metrics are given in Table 5.

When the metrics obtained are examined (last two columns of the Table 5), it show that MAPLS-AW outper-



**Table 7** *t* stat and adjusted *p* values of *t* test applied on metrics obtained using NSGA-II, TLS-EW, TLS-AW, and MAPLS-AW

Problem	NSGA-II with TLS-EW					TLS-EW with TLS-AW				
	GD	$\Delta$	MS	HVR		GD	$\Delta$	MS	HVR	
ZDT1	10.868, 1.2E-14	11.163, 4.6E-15	-2.819, 3.5E-03	-3.044, 1.9E-03		2.916, 2.7E-03	32.271, 1.1E-34	-4.608, 2.9E-05	-1.990, 2.3E-15	
ZDT2	15.745, 8.4E-21	4.764, 1.7E-05	-2.738, 4.3E-03	-2.408, 9.9E-03		3.557, 4.2E-04	5.386, 2.0E-06	-5.425, 1.8E-06	-8.350, 5.6E-11	
ZDT3	28.307, 5.0E-32	8.878, 9.0E-12	-1.850, 3.5E-02	-1.395, 1.9E-05		7.081, 5.0E-09	11.914, 4.4E-16	-2.068, 4.4E-02	-10.970, 8.5E-15	
ZDT4	6.045, 1.0E-07	4.394, 3.0E-05	-1.950, 2.8E-02	-2.115, 4.0E-02		6.862, 1.1E-08	4.472, 4.6E-05	-2.922, 5.3E-03	-7.128, 4.2E-09	
ZDT6	8.528, 3.0E-11	1.998, 2.6E-02	-2.763, 8.0E-03	-4.280, 4.3E-05		2.259, 2.8E-02	2.378, 2.1E-02	-5.651, 4.0E-07	-5.088, 5.7E-06	
DTLZ1	1.837, 3.6E-02	5.206, 1.9E-06	-6.335, 7.1E-08	-12.295, 1.4E-16		1.715, 2.3E-02	7.470, 1.2E-09	-3.599, 3.7E-04	-5.174, 2.1E-06	
DTLZ2	3.738, 4.9E-04	2.786, 7.6E-03	-2.025, 2.4E-02	-12.581, 5.8E-17		6.033, 2.1E-07	2.164, 1.8E-02	-2.552, 1.4E-02	-4.415, 2.8E-05	
DTLZ3	1.860, 1.7E-02	2.414, 2.0E-02	-2.392, 2.1E-02	-10.675, 2.2E-14		2.332, 2.4E-02	2.098, 2.0E-02	-1.799, 3.9E-02	-15.634, 1.1E-20	
DTLZ4	2.557, 1.4E-02	4.800, 1.5E-05	-2.578, 1.3E-02	-2.757, 4.1E-03		1.682, 2.5E-02	3.521, 4.7E-04	-2.050, 2.3E-02	-12.907, 2.2E-17	
DTLZ5	2.154, 3.6E-02	3.871, 3.2E-04	-8.523, 3.1E-11	-6.163, 1.3E-07		1.746, 4.4E-02	13.920, 1.2E-18	-1.757, 2.1E-02	-2.508, 7.7E-03	
DTLZ6	1.683, 4.9E-02	7.484, 1.2E-09	-3.074, 3.4E-03	-2.033, 2.4E-02		2.636, 1.1E-02	5.198, 2.0E-06	-2.074, 4.3E-02	-2.584, 1.3E-02	
DTLZ7	45.379, 1.1E-41	12.796, 3.1E-17	-2.664, 1.0E-02	-9.125, 3.8E-12		44.843, 1.9E-41	20.182, 2.2E-25	-1.928, 3.0E-02	-2.900, 2.8E-03	
SCH1	13.555, 3.4E-18	16.398, 1.6E-21	-22.324, 2.5E-27	-2.219, 1.6E-02		43.523, 8.0E-41	46.415, 3.7E-42	-5.143, 2.4E-06	-4.879, 1.2E-05	
SCH2	3.995, 2.2E-04	2.164, 1.8E-02	-2.356, 1.1E-02	-7.808, 1.9E-10		3.118, 1.5E-03	2.380, 2.1E-02	-5.702, 6.7E-07	-9.180, 3.2E-12	
FONSECA	7.689, 5.7E-10	6.397, 5.7E-08	-17.516, 4.9E-23	-3.491, 1.0E-03		24.653, 2.8E-29	3.732, 2.5E-04	-41.707, 6.1E-40	-1.884, 1.6E-02	
SRN	15.454, 1.8E-20	13.011, 1.6E-17	-11.748, 7.4E-16	-9.116, 4.0E-12		5.224, 1.8E-06	3.617, 3.5E-04	-6.147, 7.0E-08	-3.339, 8.1E-04	
TNK	42.373, 2.9E-40	6.647, 2.3E-08	-5.565, 1.1E-06	-2.323, 1.2E-02		17.288, 8.5E-23	3.717, 2.6E-04	-5.155, 2.3E-06	-2.701, 9.5E-03	
CTP1	2.663, 2.1E-02	2.914, 2.7E-03	-2.539, 7.2E-03	-2.883, 5.8E-03		4.602, 3.0E-05	6.713, 1.8E-08	-2.771, 7.9E-03	-3.395, 6.8E-04	
CTP2	5.850, 4.0E-07	3.880, 1.6E-04	-6.578, 3.0E-08	-5.289, 1.4E-06		2.315, 1.2E-02	4.303, 8.0E-05	-2.929, 5.2E-03	-6.071, 1.8E-07	
BINH2	5.574, 1.1E-06	9.201, 3.0E-12	-1.726, 2.3E-02	-5.043, 6.7E-06		3.456, 5.7E-04	2.031, 2.4E-02	-2.407, 2.0E-02	-2.057, 2.3E-02	
Problem	TLS-AW with MAPLS-AW									
	GD	$\Delta$	MS	HVR		GD	$\Delta$	MS	HVR	
ZDT1	3.072, 3.5E-03	2.429, 9.4E-03	-1.917, 1.2E-15	-2.190, 9.4E-03						
ZDT2	4.375, 6.3E-05	3.638, 3.3E-04	-4.861, 1.2E-05	-15.251, 3.1E-20						
ZDT3	4.818, 7.2E-06	8.651, 9.8E-12	-2.524, 1.5E-02	-2.402, 2.0E-02						
ZDT4	6.280, 8.7E-08	11.403, 2.2E-15	-4.079, 1.7E-04	-2.166, 3.5E-02						
ZDT6	1.941, 2.9E-02	5.995, 2.4E-07	-1.963, 2.8E-02	-6.144, 1.4E-07						
DTLZ1	2.313, 2.5E-02	6.533, 3.5E-08	-6.001, 2.3E-07	-10.143, 1.3E-13						
DTLZ2	2.323, 1.2E-02	3.034, 3.9E-03	-9.310, 2.0E-12	-13.287, 7.3E-18						
DTLZ3	2.345, 2.3E-02	1.973, 2.7E-02	-1.779, 2.0E-02	-5.832, 2.1E-07						
DTLZ4	4.601, 3.0E-05	2.839, 6.6E-03	-2.384, 2.1E-02	-1.864, 3.4E-02						
DTLZ5	2.070, 2.2E-02	2.245, 1.5E-02	-2.223, 3.1E-02	-3.978, 2.3E-04						

Table 7 continued

Problem	TLS-AW with MAPLS-AW			
	GD	$\Delta$	MS	HVR
DTLZ6	1.797, 3.9E-02	15.273, 2.9E-20	-1.779, 4.1E-02	-5.113, 5.2E-06
DTLZ7	5.467, 7.7E-07	3.366, 7.5E-04	-1.782, 2.0E-02	-9.791, 4.0E-13
SCH1	11.058, 3.2E-15	3.261, 1.0E-03	-14.845, 9.2E-20	-8.176, 1.0E-10
SCH2	14.297, 4.1E-19	2.496, 1.6E-02	-2.898, 5.6E-03	-12.916, 2.1E-17
FONSECA	4.334, 3.6E-05	6.916, 9.0E-09	-48.046, 7.0E-43	-6.132, 1.5E-07
SRN	11.526, 1.5E-15	5.305, 2.7E-06	-7.142, 4.0E-09	-6.183, 1.2E-07
TNK	79.395, 2.9E-102	12.948, 2.0E-17	-8.153, 1.1E-10	-4.468, 4.7E-05
CTP1	4.646, 2.6E-05	4.778, 1.7E-05	-2.929, 5.2E-03	-2.895, 5.7E-03
CTP2	9.206, 2.9E-12	4.288, 8.4E-05	-2.646, 5.5E-03	-7.312, 2.2E-09
BINH2	14.201, 5.4E-19	6.102, 1.6E-07	-5.936, 2.9E-07	-6.361, 6.5E-08

forms the TLS-AW. This is because of the fact that, the potentially good solutions in every generation are identified and undergo local search. Those solutions are iteratively deepened further, if they survive across generations. This has been reflected in the four metrics. Percentage of improvement is shown in Table 6, where a maximum of 69.38 % in DTLZ7 in GD, 63.86 % of Spread is achieved in DTLZ1. HVR attains a maximum of 75.15 % for DTLZ3 test problem. These three test problems are multidimensional test problems which show that MAPLS-AW works better than traditional hybrid algorithm TLS-AW on multidimensional objective space as well. Quality of solutions obtained from MAPLS-AW is due to identification of potential solutions in each generation. Instead of applying N fixed local search steps on every solution, MAPLS-AW prefers individuals who survive across generations for local search and hence local search is iteratively deepened for good individuals. Statistical one-tailed *t* test was applied between TLS-AW and MAPLS-AW, whose adjusted *p* values are shown in last column of Table 7. We can infer that means of GD,  $\Delta$  of MAPLS-AW are lesser than those of TLS-AW and means of MS, HVR of MAPLS-AW are better than those of TLS-AW. The time consumed by NSGA-II, TLS-EW, TLS-AW, and MAPLS-AW are given in Table 8. To summarize, we have substantiated that MAPLS-AW outperforms traditional memetic algorithm, TLS-AW, with fixed local search steps. This has been observed through performance metrics computed from the non-dominated solutions obtained.

## 5 Economic emission load dispatch

To demonstrate that the proposed algorithm can be used to solve real-world problems, we applied it on Economic Emission Load Dispatch (EELD) with valve point loading. The reason for choosing EELD with valve point model is to establish that our MAPLS-AW works well with the nonsmooth functions and produces better trade-off solutions. EELD is a bi-objective optimization problem that requires the objectives to be optimized for best fuel cost and emission cost. In general, the power plants want to reduce the cost involved in power generation, often neglecting the amount of pollutants they produce. Due to increase in awareness over the environmental pollution (Talaq et al. 1994; Abido 2006), there is an increase in need that demands the pollution to be controlled and to be reduced. That is, the EELD problem is to balance the two conflicting objectives of minimizing fuel cost and emission cost.

### 5.1 Problem formulation

The objective of Economic Emission Load Dispatch is to bring out a schedule of power units that generate power to

**Table 8** Time consumed by each of NSGA-II, TLS-EW, TLS-AW, and MAPLS-AW in milliseconds

Test problem	NSGA-II	TLS-EW	TLS-AW	MAPLS-AW
ZDT1	3.238	3.567	3.140	2.252
ZDT2	3.277	4.016	2.958	2.694
ZDT3	4.232	5.058	4.068	3.619
ZDT4	3.244	3.052	3.491	2.059
ZDT6	3.456	2.877	3.044	1.605
DTLZ1	3.076	3.405	3.394	3.030
DTLZ2	4.645	4.069	3.789	3.616
DTLZ3	4.499	4.779	4.231	4.186
DTLZ4	4.475	5.276	3.246	2.567
DTLZ5	5.556	3.807	3.686	3.471
DTLZ6	3.666	3.590	2.654	2.636
DTLZ7	3.730	3.407	3.406	3.333
SCH1	0.648	0.806	0.784	0.458
SCH2	3.568	3.472	2.542	2.332
FONSECA	1.727	1.626	1.500	0.976
SRN	1.212	1.072	0.937	0.648
TNK	0.348	0.076	0.057	0.051
CTP1	3.644	3.117	3.270	3.129
CTP2	3.834	3.643	3.531	2.573
BINH2	0.347	0.082	0.062	0.054

meet the load demand. The schedule should be in such a way that it minimizes the costs involved in it by satisfying the equality and inequality constraints (Abido 2003; Balakrishnan et al. 2003).

5.1.1 Minimization of fuel cost

One of the objectives of Economic Emission Load Dispatch is to minimize the cost function  $F(P_G)$

$$\text{Minimize } F(P_G) = \sum_{i=1}^N f_i(P_G^i) \tag{14}$$

where  $N$  is the number of generating units and  $f_i(P_G^i)$  is the cost of generation of each unit represented by a quadratic function,

$$f_i(P_G^i) = a_i(P_G^i)^2 + b_i P_G^i + c_i \tag{15}$$

In the above equation,  $a_i$ ,  $b_i$  and  $c_i$  are the cost coefficients. Considering the valve point effect, the function is represented as a sum of quadratic and sinusoidal functions,

$$f_i(P_G^i) = a_i + b_i P_G^i + c_i(P_G^i)^2 + |d_i \sin(e_i(P_G^i \text{min} - P_G^i))| \tag{16}$$

$d_i$ ,  $e_i$ , and  $P_G^i \text{min}$  are the cost coefficients and lower power generation limit of  $i$ th unit.

5.1.2 Minimization of emission cost

The other objective of EELD is to minimize the total amount of emission from the system,

$$\text{Minimize } E(P_G) = \sum_{i=1}^N E_i(P_G^i) \tag{17}$$

where  $E_i(P_G^i)$  is the emission dispatch from each generating unit given by sum of quadratic and exponential function.

$$E_i(P_G^i) = \alpha_i + \beta_i P_G^i + \gamma_i (P_G^i)^2 + \eta_i \exp(\delta_i P_G^i) \tag{18}$$

where  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$ , and  $\eta_i$ ,  $\delta_i$  are the emission coefficients of  $i$ th unit.

5.1.3 Constraints

(i) Power balance constraints

The total power generated by all the units should be equal to the sum of total demand  $P_D$  and transmission loss  $P_L$ ,

$$\sum_{i=1}^N P_G^i - P_D - P_L = 0 \tag{19}$$

$P_L$  is calculated by B coefficients method (Wood and Woltenberg 2011) which can be represented by

$$P_L = \sum_{i=1}^N \sum_{j=1}^N P_G^i B_{ij} P_G^j \tag{20}$$

(ii) Capacity constraints

The power generated by each unit is restricted by upper and lower power limits

$$P_G^i \text{min} \leq P_G^i \leq P_G^i \text{max} \quad i \in N \tag{21}$$

5.2 Implementation of MAPLS-AW to solve EELD problem

Population is initialized with randomly generated individuals within the capacity constraints. Each chromosome will be a vector of  $N$  parameters, where each parameter refers to the power generated by each generator.

$$\langle P_1 \ P_2 \ P_3 \ P_4 \ \dots \ P_N \rangle \tag{22}$$

Fitness of each individual is evaluated and the steps of MAPLS-AW namely, identifying the parents, applying Preferential Local Search on them, computing ranks and fronts using non-domination relation, computing crowding distance, constructing the mating pool using constrained

binary tournament selection are carried out. This is followed by the recombination operations over selected parents. SBX crossover with nullified distribution parameter is applied over every gene of the chromosome. Polynomial mutation is performed on the new offspring with respect to mutation probability. The new set of offspring undergoes the one-step preferential local search and then combined with parents for the next generation.

Adaptive weight is computed for each individual and used for the calculation of the weight to be assigned for the individual objectives. Stopping criteria is set to 1,500 fitness evaluations. We have used the inputs for cost and emission coefficients from [Basu \(2011\)](#).

### 5.2.1 Performance metrics

To evaluate the performance of the MAPLS-AW on a real-world problem, we have used two metrics to examine the quality of obtained solutions. They are Hypervolume (HV) and Set Coverage metric (SCM). Since the optimum solutions are not known for EELD problem, these two metrics will help in evaluating the proposed MAPLS-AW algorithm.

The SCM compares two sets of solutions and yields an outcome in the range 0 and 1, with respect to the amount of coverage between the sets. Let A and B be the two solution sets, Set Coverage Metric ([Zitzler and Thiele 1999](#)) between A and B is given by

$$SCM(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \prec b\}|}{|B|} \quad (23)$$

$SCM(A, B) = 1$ , when solutions of A are dominating all the solutions of B and  $SCM(A, B) = 0$ , when none of A is dominating the solutions of B. SCM is not symmetric that is,  $SCM(A, B) \neq SCM(B, A)$ . The resultant set generated

by MAPLS-AW has been compared with the set obtained from NSGA-II and SPEA2. That is,  $SCM(MAPLS-AW, NSGA-II)$ , and  $SCM(NSGA-II, MAPLS-AW)$  have been computed to analyze the performance of MAPLS-AW. Similarly, SCM between MAPLS-AW and SPEA2 has been computed.

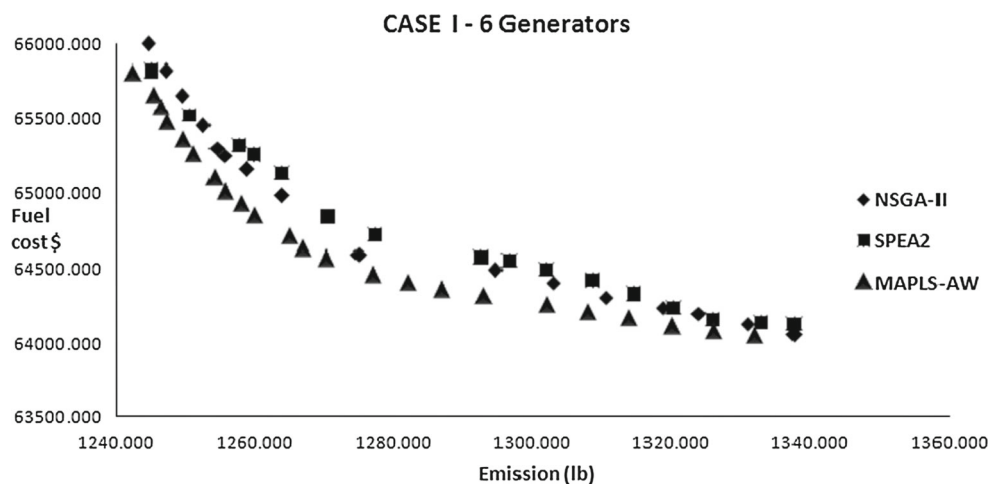
Three different test cases are used to evaluate the performance of the MAPLS-AW algorithm. They are 6, 10, and 40 generator systems to generate 1,200 MW, 2,000 MW, and 10,500 MW power, respectively.

## 5.3 Performance of MAPLS-AW on EELD problem

### 5.3.1 Case I

This is a 6 generator system with simple quadratic fuel and emission cost functions and with the transmission loss to generate 1,200 MW power. The test data are taken from [Basu \(2011\)](#). The results of best fuel cost and emission cost obtained from NSGA-II, SPEA2, and memetic algorithm with Preferential Local Search using adaptive weights (MAPLS-AW) are shown in Fig 4. The Hypervolume metric obtained for this test case is listed in Table 9. The proposed MAPLS-AW resulted in higher value, i.e., the volume covered under the optimal front by the MAPLS-AW algorithm was larger when compared with the rest of the algorithms. MAPLS-AW shown an improvement of about 6.6 and 3.5% over NSGA-II and SPEA2, respectively.

The adjusted  $p$  values of  $t$  test over Hypervolume metric shown in Table 10 also confirmed that there is a significant difference between MAPLS-AW, NSGA-II, and SPEA2 algorithms. Along with Hypervolume metric, we have computed SCM. Mean of SCM values after 50 runs has been computed and tabulated in Table 9. The SCM between the proposed MAPLS-AW and NSGA-II for case



**Fig. 4** Pareto front of 6 generator case

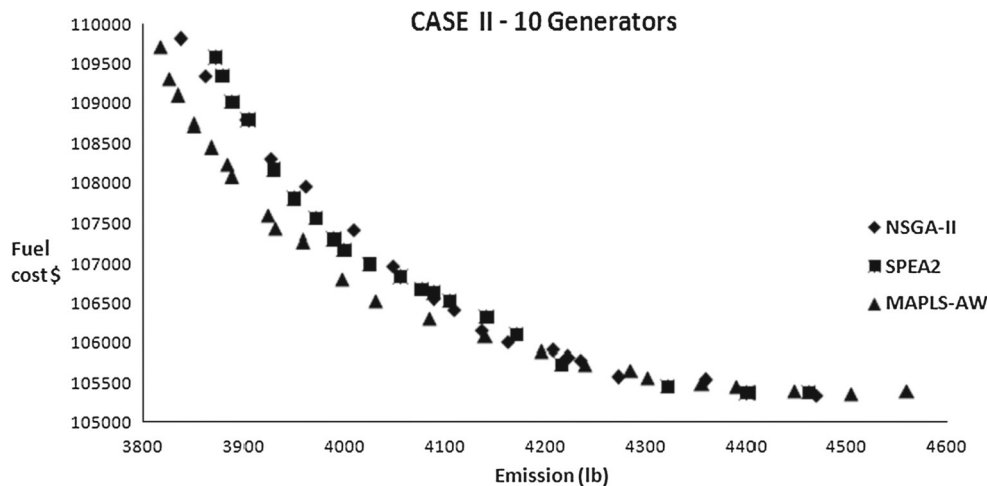


**Table 9** Performance metrics obtained from NSGA-II, SPEA2, and MAPLS-AW

Test case	HV			SCM			
	NSGA-II	SPEA2	MAPLS-AW	NSGA-II < MAPLS-AW	MAPLS-AW < NSGA-II	SPEA2 < MAPLS-AW	MAPLS-AW < SPEA2
I	0.5357	0.5514	0.5711	0.0400	0.0879	0.0127	0.0772
II	0.1392	0.1619	0.3838	0.0927	0.1696	0.0199	0.0821
III	0.7873	0.7780	0.8221	0.4993	0.6182	0.3377	0.5886

**Table 10** *t* stat and adjusted *p* values of *t* test applied on metrics obtained from NSGA-II, SPEA2, MAPLS-AW

Test case	HV		SCM	
	NSGA-II with MAPLS-AW	SPEA2 with MAPLS-AW	NSGA-II with MAPLS-AW	SPEA2 with MAPLS-AW
I	-3.631, 1.43E-06	-12.450, 2.87E-03	-7.488, 1.49E-02	-4.417, 3.12E-04
II	-5.059, 1.36E-19	-5.852, 1.03E-17	-2.030, 5.52E-05	-3.186, 3.12E-05
III	-7.656, 1.46E-02	-11.424, 4.42E-03	-2.061, 1.11E-03	-4.611, 1.68E-09



**Fig. 5** Pareto front of 10 generator case

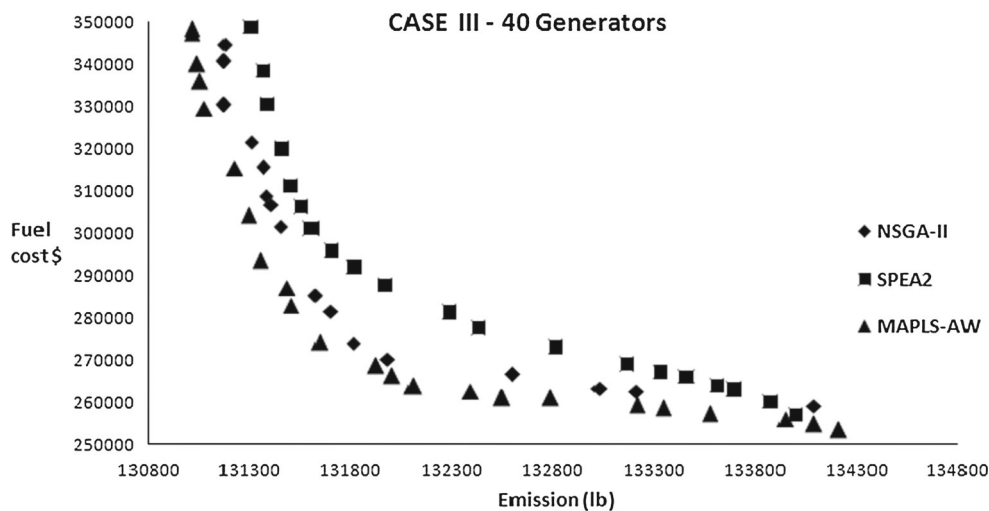
I, 6 generator system shows the effective coverage of non-dominating solutions of MAPLS-AW than NSGA-II. That is, number of non-dominating solutions generated by MAPLS-AW is 54.49% more than non-dominating solutions obtained from NSGA-II. There was a massive 83.54% of improvement in the SCM metric between MAPLS-AW and SPEA2.

### 5.3.2 Case II

The second case is a system with 10 generators considering the nonsmooth functions of valve point effect and with transmission loss to generate 2,000 MW power. Optimal fuel and emission cost obtained generated by three algorithms are shown in Fig. 5, which depicts that proposed MAPLS-AW

obtained best emission and fuel cost outperforming the other two algorithms. The Hypervolume metric tabulated in Table 9 shows that proposed MAPLS-AW has covered larger area under the optimal front and this has been supported by the *t* test results in Table 10. This 10 generator case has shown an improvement of about 63.73% over NSGA-II and 57.81% over SPEA2 in Hypervolume metric. It is evident that this kind of improvement is due to the presence of two contributing features, Preferential Local Search and adaptive weights that enable a wide exploration of extreme regions of search space.

The SCM computed between NSGA-II and MAPLS-AW reveals that less number of non-dominating solutions have been obtained from NSGA-II than MAPLS-AW. Similarly, the amount of coverage by MAPLS-AW over SPEA2 was



**Fig. 6** Pareto front of 40 generator case

way better with 75.7% of improvement than set coverage of SPEA2 over MAPLS-AW. The adjusted  $p$  values of  $t$  test over SCM for these algorithms are given in Table 10, where the adjusted probability values for the one-tailed were well below the level of significance, which is 5%. The  $t$  test confirmed that proposed MAPLS-AW algorithm is significantly better than NSGA-II and SPEA2.

### 5.3.3 Case III

As the third case, we took the 40 generator system with valve point loading with demand of 10,500 MW of power using nonsmooth fuel cost and emission functions. Cost coefficients and generator capacity constraints are taken from Basu (2011). 40 generator system has been taken as a no loss case. We applied MAPLS-AW algorithm, NSGA-II, and SPEA2 to solve this multi-objective problem. Non-dominated solutions obtained from three algorithms are shown in Fig. 6. The amount of area covered under Pareto front measured in terms of Hypervolume obtained by MAPLS-AW is better than other two algorithms. Table 9 presents the performance of algorithms in comparison where there is a 4.6% of improvement over NSGA-II and 5.6% of improvement over SPEA2.

The mean of SCM obtained by comparing non-dominated solutions obtained from MAPLS-AW, NSGA-II, and SPEA2 for this case is given in Table 9. There is an improvement of 74.2% observed by MAPLS-AW over SPEA2 and 19.2% over NSGA-II. The  $t$  test was applied over the metrics HV & SCM. The computed  $t$  stat values and the adjusted  $p$  values are presented in Table 10. From these values, we can infer that the metrics of MAPLS-AW are better than those of NSGA-II & SPEA2. This shows that MAPLS-AW is able to achieve quality solutions on complex real-world applications as well.

## 6 Conclusion

In this paper, we have introduced a hybrid algorithm referred to as MAPLS-AW. This memetic algorithm addresses the challenges that arise while combining a global search with local search procedure. MAPLS-AW deals with which individuals are chosen for local search, how often a local search can be applied, and how deep a local search is allowed. MAPLS-AW identifies potential solutions for Preferential Local Search and facilitates iterative deepening of local search over them. We have proposed an adaptive weight mechanism which is used while converting multi-objectives into single objective. This adaptive weight assigned to every objective is dynamic and adapts across generations. Adaptive weights are computed by taking the current positional information of a solution from the objective space.

To examine the overall performance of proposed MAPLS-AW algorithm as a multi-objective evolutionary algorithm, we compared it with NSGA-II and SPEA2. We tested the performance of the proposed algorithm by applying it on 20 constrained and unconstrained benchmark test problems and a real-world application EELD.

The performance metrics reveal that the proposed MAPLS-AW stands out as a better algorithm. This is due to the preferential identification of good solutions and the fine tuning applied on them. Statistical  $t$  test confirmed that proposed MAPLS-AW algorithm is performing better than NSGA-II and SPEA2.

When applied on a real-world application like EELD, MAPLS-AW is able to produce the better fuel and emission cost in all the three different case studies taken. The performance of proposed algorithm on EELD is measured using Hypervolume metric and Set Coverage Metrics and tested using statistical  $t$  test. The results reveal that our proposed

algorithm has the ability to provide a trade-off among the nonsmooth functional objectives.

In the current work, we have used a hard fixed termination condition of 25000 fitness evaluations. Instead, proper decision making can be incorporated to decide upon when to stop the evolution of the memetic algorithm. In this paper, the idea of Preferential Local Search with adaptive weights has been incorporated into NSGA-II to develop the hybrid algorithm. But these ideas can be easily integrated with any of the other evolutionary optimization algorithms.

**Acknowledgments** We would like to thank the management of SSN College of Engineering for funding the High Performance Computing Lab (HPC Lab) where this research was carried out. We also express our sincere thanks to K.Kannan, Dean, SASTRA University, Thanjavur, India, for his valuable suggestions in statistical analysis. We thank the anonymous reviewers for their valuable comments in improving the quality of the paper.

## References

- Abido MA (2003) A niched pareto genetic algorithm for multiobjective environmental/economic dispatch. *Int J Elec Power Energy Syst* 25(2):97–105
- Abido MA (2006) Multiobjective evolutionary algorithms for electric power dispatch problem. *Evol Comput IEEE Trans* 10(3):315–329
- Abraham A (2005) Evolutionary computation., Handbook for measurement systems design Wiley, London
- Aguilari J, Miranda P (1999) Approaches based on genetic algorithms for multiobjective optimization problems. In: GECCO, pp 3–10
- Bagchi TP (1999) Multiobjective scheduling by Genetic Algorithms. Springer
- Balakrishnan S, Kannan PS, Aravindan C, Subathra P (2003) On-line emission and economic load dispatch using adaptive hopfield neural network. *Appl Soft Comput* 2(4):297–305
- Basu M (2011) Economic environmental dispatch using multi-objective differential evolution. *Appl Soft Comput* 11(2):2845–2853
- Bhuvana J, Aravindan C (2011a) Design of hybrid genetic algorithm with preferential local search for multiobjective optimization problems. In: Information technology and mobile communication, communications in computer and information science, vol 147. Springer, Berlin, pp 312–316
- Bhuvana J, Aravindan C (2011b) Preferential local search with adaptive weights in evolutionary algorithms for multiobjective optimization problems. In: Soft computing and pattern recognition (SoCPaR), 2011 international conference of IEEE, pp 358–363
- Blickle T (1996) Theory of evolutionary algorithms and application to system-synthesis. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland
- Blum C, Rolli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv CSUR* 35(3):268–308
- Chen X, Ong YS, Lim MH, Tan KC (2011) A multi-facet survey on memetic computation. *Evol Comput IEEE Trans* 15(5):591–607
- Cheng FY, Li D (1997) Multiobjective optimization design with pareto genetic algorithm. *J Struct Eng* 123(9):1252–1261
- Coello CA (2000) An updated survey of GA-based multiobjective optimization techniques. *ACM Comput Surv CSUR* 32(2):109–143
- Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Methods Appl Mech Eng* 191(11–12):1245–1287
- Coello CA, Zacetenco CSP (2006) 20 years of evolutionary multi-objective optimization: what has been done and what remains to be done. *Comput Intell Princ Pract Chap* 4:73–88
- Črepinšek M, Liu SH, Mernik L (2012) A note on teaching-learning-based optimization algorithm. *Inf Sci* 212:79–93
- Črepinšek M, Liu SH, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput Surv CSUR* 45(3):1–33
- Črepinšek M, Liu SH, Mernik M (2014) Replication and comparison of computational experiments in applied evolutionary computing: common pitfalls and guidelines to avoid them. *Appl Soft Comput* 19:161–170
- De Jong KA, Spears WM (1992) A formal analysis of the role of multi-point crossover in genetic algorithms. *Ann Math Artif Intell* 5(1):1–26
- Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley (Asia) Pte Ltd
- Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. *Complex Syst* 9:115–148
- Deb K, Kumar A (1995) Real-coded genetic algorithms with simulated binary crossover: studies on multi-modal and multi-objective problems. *Complex Syst* 9(6):431–454
- Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. *Comput Sci Inf* 26:30–45
- Deb K, Agrawal S, Pratap A, Meyarivan T (2000) A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lect Notes Comput Sci* 1917:849–858
- Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evol Comput IEEE Trans* 6(2):182–197
- El-Mihoub TA, Hopgood AA, Nolle L, Battersby A (2006) Hybrid genetic algorithms: a review. *Eng Lett* 13(2):124–137
- Fogel DB (1995) Evolutionary computation: toward a new philosophy of machine intelligence. IEEE Press, Piscataway
- Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. *ICGA* 93:416–423
- Fonseca CM, Fleming PJ (1995) An overview of evolutionary algorithms in multiobjective optimization. *Evol Comput* 3(1):1–16
- Gaspero LD (2003) Local search techniques for scheduling problems: algorithms and software tools. PhD thesis, Dipartimento di Matematica e Informatica-Università degli Studi di Udine
- Gen M, Lin L (2005) Multi-objective hybrid genetic algorithm for bicriteria network design problem. *Complex Int* 11(11):73–83
- Glover F, Laguna M (1997) Tabu Search. Kluwer
- Goldberg DE (1989) genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading
- Guliashki V, Toshev H, Korsemov C (2009) Survey of evolutionary algorithms used in multiobjective optimization. *Prob Eng Cybern Robot* 60:42–54
- Hajela P, Lin CY (1992) Genetic search strategies in multicriterion optimal design. *Struct Optim* 4(2):99–107
- Hamada N, Nagata Y, Kobayashi S, Ono I (2010) Adaptive weighted aggregation: a multiobjective function optimization framework taking account of spread and evenness of approximate solutions. In: Evolutionary computation (CEC), 2010 IEEE Congress on, pp 1–8
- Hamada N, Nagata Y, Kobayashi S, Ono I (2011) Adaptive weighted aggregation 2: more scalable AWA for multiobjective function optimization. In: Evolutionary computation (CEC), 2011 IEEE congress on, pp 2375–2382
- Herrera F, Lozano M, Verdegay JL (1998) Tackling real-coded genetic algorithms: operators and tools for behavioural analysis. *Artif Intell Rev* 12(4):265–319

- Herrera F, Lozano M, Sánchez AM (2003) A taxonomy for the crossover operator for real-coded genetic algorithms: an experimental study. *Int J Intell Syst* 18(3):309–338
- Holland JH (1975) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press
- Horn J, Nafpliotis N, Goldberg DE (1994) A niched pareto genetic algorithm for multiobjective optimization. In: *Evolutionary computation, 1994. IEEE world congress on computational intelligence., Proceedings of the first IEEE conference on*, pp 82–87
- Huband S, Hingston P, Barone L, While L (2006) A review of multi-objective test problems and a scalable test problem toolkit. *Evol Comput IEEE Trans* 10(5):477–506
- Jang JSR, Sun CT, Mizutani E (2004) *Neuro-fuzzy and soft computing*. PHI, Pearson Education, New Delhi
- Jiang S, Cai Z, Zhang J, Ong YS (2011) Multiobjective optimization by decomposition with pareto-adaptive weight vectors. In: *Natural computation (ICNC), 2011 seventh international conference on*, IEEE, vol 3. pp 1260–1264
- Joines JA, Gupta D, Gokce MA, King RE, Kay MG (2002) Manufacturing supply chain applications 1: supply chain multi-objective simulation optimization. In: *Proceedings of the 34th conference on winter simulation: exploring new frontiers, winter simulation conference, WSC '02*, pp 1306–1314
- Jubril AM (2012) A nonlinear weights selection in weighted sum for convex multiobjective optimization. *Facta Univ Ser Math Inf* 27(3):357–372
- Kim IY, De Weck O (2005) Adaptive weighted-sum method for bi-objective optimization: pareto front generation. *Struct Multidisc Optim* 29(2):149–158
- Knowles JD, Corne DW (2000) Approximating the nondominated front using the pareto archived evolution strategy. *Evol Comput* 8(2):149–172
- Konak A, Coit DW, Smith AE (2006) Multi-objective optimization using genetic algorithms: a tutorial. *Reliab Eng Syst Saf* 91(9):992–1007
- Koski J (1988) *Multicriteria truss optimization. Multicriteria optimization in engineering and in the sciences*. Springer, US, pp 263–307
- Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *Evol Comput IEEE Trans* 9(5):474–488
- Li H, Landa-Silva D (2008) Evolutionary multi-objective simulated annealing with adaptive and competitive search direction. In: *Evolutionary computation, 2008. CEC 2008. (IEEE World congress on computational intelligence). IEEE congress on*, pp 3311–3318
- McGinley B, Maher J, O’Riordan C, Morgan F (2011) Maintaining healthy population diversity using adaptive crossover, mutation, and selection. *Evol Comput IEEE Trans* 15(5):692–714
- Mongus D, Repnik B, Mernik M, Žalik B (2012) A hybrid evolutionary algorithm for tuning a cloth-simulation model. *Appl Soft Comput* 12(1):266–273
- Moscato P (1989) *On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms*. Caltech concurrent computation program, C3P. Report 826:1989
- Mostaghim S (2004) *Multi-objective evolutionary algorithms. Data structures, convergence, and diversity*. PhD thesis, University of Paderborn, Germany
- Nelder JA, Mead R (1965) A simplex method for function minimization. *Comput J* 7(4):308–313
- Nguyen TT, Yang S, Branke J (2012) Evolutionary dynamic optimization: a survey of the state of the art. *Swarm Evol Comput* 6:1–24
- Ong YS, Lim MH, Zhu N, Wong KW (2006) Classification of adaptive memetic algorithms: a comparative study. *Syst Man Cybern Part B Cybern IEEE Trans* 36(1):141–152
- Ong YS, Lim M, Chen X (2010) Memetic computation: past, present and future [Research Frontier]. *Comput Intell Mag IEEE* 5(2):24–31
- Runarsson T, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *Evol Comput IEEE Trans* 4(3):284–294
- Salomon R (1998) Evolutionary algorithms and gradient search: similarities and differences. *Evol Comput IEEE Trans* 2(2):45–55
- Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: *Proceedings of the 1st international conference on genetic algorithms*, L. Erlbaum Associates Inc., pp 93–100
- Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput* 2(3):221–248
- Steuer RE (1986) *Multiple criteria optimization: theory, computation and application* Wiley, New York
- Talaq J, El-Hawary F, El-Hawary M (1994) A summary of environmental/economic dispatch algorithms. *Power Syst IEEE Trans* 9(3):1508–1516
- Van Veldhuizen DA (1999) *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. PhD thesis, Air Force Institute of Technology, Wright Patterson AFB, OH, USA
- Van Veldhuizen DA, Lamont GB (1999) Multiobjective evolutionary algorithm test suites. In: *Proceedings of the 1999 ACM symposium on Applied computing*, New York, USA, SAC '99, pp 351–357
- Veček N, Mernik M, Črepinšek M (2014) A chess rating system for evolutionary algorithms: a new method for the comparison and ranking of evolutionary algorithms. *Inf Sci* 277:656–679
- Weise T (2009) *Global optimization algorithms: theory and application*, 2nd edn. Self-Published
- Wood AJ, Wollenberg BF (2011) *Power generation, operation, and control*. Wiley, New Delhi
- Zadeh L (1963) Optimality and non-scalar-valued performance criteria. *Autom Control IEEE Trans* 8(1):59–60
- Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multi-objective evolutionary algorithms: a survey of the state of the art. *Swarm Evol Comput* 1(1):32–49
- Zitzler E (1999) *Evolutionary algorithms for multiobjective optimization: methods and applications*. PhD thesis, ETH Zurich, Switzerland
- Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *Evol Comput IEEE Trans* 3(4):257–271
- Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. *Evol Comput* 8(2):173–195
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization. In: *Evolutionary methods for design optimization and control with applications to industrial problems*, pp 95–100