



A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems

Mohamed A. Tawhid^{1,2} · Vimal Savsani^{3,4}

Published online: 26 April 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

There are many diverse fields and applications such as data mining, engineering, operations research, economics, and science can be formulated as multi-objective optimization problems. In this paper, we describe and propose a novel and a useful multi-objective artificial algae algorithm (MO-AAA) to solve multi-objective engineering design problems. Our proposed algorithm, (MO-AAA), is based on the search technique of artificial algae algorithm(AAA) algorithm. MO-ADA applies the elitist non-dominated sorting and crowding distance approach to preserve the diversity among the optimal set of solutions and obtains various non-domination levels, respectively. Also, we evaluate the effectiveness of the proposed algorithm by applying it on different multi-objective benchmark problems (20 challenging benchmark problems from CEC 2009 for unconstrained and constrained multi-objective optimization problems) and engineering design benchmark problems with distinctive features. Finally, our results show that MO-AAA efficiently generates the Pareto front and is easy to implement, promising and competitive compared to other state-of-the-art algorithms considered in this work.

Keywords Multi-objective optimization · Artificial algae algorithm · Multi-objective engineering design problems · Pareto optimal solutions

1 Introduction

One of the critical research areas in optimization is multi-objective optimization problems (MOOPs), because MOOPs have a wide diverse of applications such as data mining [8], economic and finance issues [43, 54, 82], utility theory, game theory, linear production theory and economics

[31], preferences in MOPs [13], airline operations [22], scheduling problems [45, 55, 86], portfolio optimization [23], combinatorial problems [10, 24, 25, 37], portfolio optimization [22], engineering problems [5, 14, 32, 49], manpower planning [77], reservoir management [2], automatic cell planning problems [48], radiation therapy [38], jury selection [75], can be formulated as MOOPs. We refer the interested reader to further applications in survey articles [69, 97], and other books [21, 29, 52].

There is more than one objective function in MOOPs. These objective functions need to be optimized simultaneously. It turns out solving MOOPs is a challenging job because the objective functions are conflicting, and the solution set acquired from solving MOOPs has the following characteristic: no solution is subaltern to the other solutions, and each solution is a trade-off of the objectives. This set is known as the Pareto-optimal set. Normally, in most of the applications mentioned above, the number of solutions in the Pareto-optimal set grows exponentially with the problem size. Therefore, MOOPs are NP-hard problems.

Therefore, it might be expensive concerning the computational point of view to use traditional methods to solve MOOPs. Among the drawbacks of using the traditional techniques are the following: they require differentiability

✉ Mohamed A. Tawhid
mtawhid@tru.ca
Vimal Savsani
vimal.savsani@gmail.com

¹ Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC, V2C 0C8, Canada

² Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Moharam Bey 21511, Alexandria, Egypt

³ Department of Mechanical Engineering, Pandit Deendayal Petroleum University, Gandinagar, Gujarat, India

⁴ Postdoctoral Fellow, Department of Mathematics and Statistics, Faculty of Science, Thompson Rivers University, Kamloops, BC, V2C 0C8, Canada

of objective functions and/or constraints; computing the Jacobian and/or Hessian is expensive; they need a good initial point; they are not efficient when the underlying problems are discontinuous and/or discrete in the search space and stuck at a sub-optimal solution; and the objective functions and/or constraints might be non-smooth.

It is appealing to many researchers to develop effective techniques and algorithms to solve MOOPs so that these algorithms overcome the drawbacks of the traditional methods. Among of these efficient algorithms are evolutionary algorithms (EAs). It is shown [98] that EAs can search for the Pareto optimal solutions of MOOPs that are too difficult to be solved by the traditional methods within a reasonable computation time.

In the past decades, many biologically and nature inspired algorithms have been proposed and established to solve MOOPs, such as multi-objective genetic algorithm (MOGA) [30], Niche Pareto genetic algorithm (NPGA) [35], non-dominated sorting genetic algorithm (NSGA) [79], NSGA-II (an elitist based non-dominated sorting genetic algorithm – an improved version of NSGA) [19], strength Pareto evolutionary algorithm (SPEA) [99], SPEA2 (an improved version of SPEA) in [101], Pareto-archived evolutionary strategy (PAES) [42], Pareto envelope-based selection algorithm (PESA and PESA-II) in [15], multi-objective particle swarm optimization (MOPSO) [11] and an improved version of MOPSO in [16], a multi-objective evolutionary algorithm based on decomposition (MOEA/D) in [95] and its improved versions described in detail in [97], indicator-based evolutionary algorithm (IBEA) [100], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [95], archived multi-objective simulated annealing (AMOS) [9], and preference-inspired co-evolutionary algorithm (PICEA) [85]. Also, swarm intelligence based search algorithms for multi-objective optimization [1, 4, 6, 7, 11, 12, 33, 36, 50, 56, 58, 65, 81, 84, 89, 94] have also been developed and applied to a variety of MOOPs. Lately, several more developed multi-objective algorithms [62, 90, 92] based on the search techniques other than the ones used in genetic algorithm or EAs and particle swarm optimization [41] have also shown a great potential in obtaining the Pareto front closest to the true Pareto front.

Nature-based meta-heuristics have caught the attention of various researchers due of its capability to solve problems which are multi-dimensional, multi-modal, combinatorial or large search space problems. The concept of nature based meta-heuristic was suggested by Holland in 1975 [34] with the introduction of genetic algorithm (GA), after that many population based effective algorithms were developed in next three decades like ant colony optimization (ACO) [20], artificial immune algorithm (AIA) [28], differential evolution (DE) [80], bacteria foraging algorithm (BFA) [61], shuffled frog leap (SFL) [27] and particle swarm

optimization (PSO) [41]. These algorithms were applied to various engineering, scientific and management real-life problems. The effectiveness of such algorithms has motivated researchers further to develop new algorithms based on the principle of nature. The development of nature-based algorithms was remarkable since 2005, and many new algorithms were developed after that period. It is difficult to mention all, but few such algorithm can include; The invasive weed optimization (IWO) [51], artificial bee colony (ABC) [39], biogeography-based optimization (BBO) [78], gravitational search algorithm (GSA) [68], gray wolf optimization (GWO) [53], firefly algorithm (FA) [87], cuckoo search (CS) algorithm [91], bat algorithm (BA) [88], grenade explosion method (GEM) [3], charged system search (CSS) [40], teaching-learning based optimization (TLBO) [66, 67]. The animal migration optimization (AMO) [47], water cycle algorithm (WCA) [26], runner root algorithm (RRA) (Merrikh-Bayat, 2015), mine blast algorithm (MBA) [71], heat transfer search (HTS) [63], lightning search algorithm (LSA) [76], lion optimization algorithm (LOA) [93], Heat transfer search [63], passing vehicle search (PVS) [72] and many more. Many of these algorithms have proven their ability for the applications on real-life applications. Many of these algorithms have their effective multi-objective versions like GA [12, 17], PSO [69, 84], ABC [4, 58, 94], ACO [6, 86], AIA [7, 33], GSA [54], BBO [36, 70], IWA [57], FFA [90], CSA [92], BA [89] and TLBO [43, 62, 63].

In addition to the above listed algorithms, artificial algae algorithm (AAA) is also an effective and recently developed algorithm for the single objective optimization [83] which works on the the living behavior of the microalgae. So far, multi-objective version for AAA is not developed and investigated. The basic search mechanism of AAA and its effectiveness for the single objective optimization can be used as a motivation to develop its multi-objective version and to check its efficacy for challenging optimization problems. So, this paper is mainly focused on developing a useful technique by using multi-objective optimization method along with artificial algae algorithm(AAA). This proposed method will be denoted as MO-AAA method. AAA is an effective meta-heuristic developed in 2015 [83]. This technique has proved its effectiveness and robustness regarding accuracy, computational efforts, and convergence. Later in this paper it is shown that the MO-AAA is capable of generating non-dominated solutions and true Pareto front (PF). The first contribution of this paper is to introduce a new concept of multi-objective AAA algorithm. Second contribution is to show the correctness of MO-AAA method for the engineering design problems. The proposed method is investigated on different multi-objective benchmark problems (20 challenging benchmark problems from CEC 2009 for unconstrained and constrained

multi-objective optimization problems) and engineering design problems. These benchmark problems offer challenges to multi-objective optimization algorithms. We have compared our proposed method with about nearly 16 (recently developed) other algorithms for unconstrained problems and 9 (recently developed) algorithms for constrained problems. The proposed approach is compared with the other well-known multi-objective optimization algorithms.

The organization of the paper is as follows: Section 2 describes the definitions of multi-objective optimization problems and Pareto efficiency or optimality. Section 3 presents the artificial algae algorithm(AAA) algorithm in details. Section 4 covers the proposed algorithm, MO-AAA algorithm. Section 5 investigates the performance of MO-AAA on various benchmark problems followed by Section 5 which gives the performance of MO-AAA on different engineering design problems. Finally, Section 6 concludes some final comments and future work.

2 Multi-objective optimization problem

In this paper, we assume that all objective functions are of the minimization type. It is known that minimization type for objective functions can be transformed into a maximization type by multiplying negative one. A minimization multi-objective decision problem (multi-objective optimization problem) with n objectives is defined as follows [10, 44, 52]:

Given a k -dimensional decision variable vector $\vec{x} = \{x_1, x_2, \dots, x_k\}$ in the solution space S , find a vector x^* that minimizes a given set of n objective functions

$$f(\vec{x}^*) = \left\{ f_1(\vec{x}^*), f_2(\vec{x}^*) \dots f_n(\vec{x}^*) \right\}$$
. The solution space S is generally confined by a series of constraints, such as $g_i(\vec{x}) \leq 0, i = 1, 2, \dots, I$, $h_j(\vec{x}) = 0, j = 1, 2, \dots, J$, and bounds on the decision variables $(\vec{x})^l \leq \vec{x} \leq (\vec{x})^u$. The objective space is defined as $F = \{f = (f_1, f_2, \dots, f_n) : f_i = f_i(\vec{x}), \forall \vec{x} \in S, i = 1, 2, \dots, n\}$.

In various real-life problems, objective functions under consideration conflict with each other. Thus, minimizing x for a single objective usually is not suitable for the other objective functions. Therefore, we aim a multi-objective solution that simultaneously optimizes each objective function which is an impossible task. A reasonable solution to a multi-objective problem is to examine a set of solutions so that each solution in the set satisfies the objective functions at an acceptable level without being dominated by any other solutions. If all objective functions are minimized, a feasible solution x is said to dominate another feasible solution y ($x \succ y$), if and only if, $f_i(x) \leq f_i(y)$, for

$i = 1, 2, \dots, n$ and $f_j(x) < f_j(y)$ for least one objective function j . A solution is said to be Pareto optimal if it is not dominated by any other solution in the solution space. A Pareto optimal solution cannot be improved concerning any objective without deteriorating least one other objective. The set of all feasible non-dominated solutions in S is referred to as the Pareto optimal set, and for a given Pareto optimal set, the corresponding objective function values in the objective space are called the Pareto front. For many problems, the number of Pareto optimal solutions is enormous (perhaps infinite).

Our aim in a multi-objective optimization algorithm is to find solutions in the Pareto optimal set. Nonetheless, finding the entire Pareto optimal set, for many multi-objective problems, is practically impossible due to the large size of the problems. Therefore, a practical approach to multi-objective optimization is to examine a set of solutions (the best-known Pareto set) that represent the Pareto optimal set as well as possible.

3 Artificial Algae Algorithm (AAA)

The AAA algorithm, proposed by [83], is based on the living behavior of the microalgae. Artificial algae update each solution in the feasible region by idealizing the characteristics of algae. Based on the natural behavior of algae, the AAA consists of three basic parts, namely, evolutionary process, adaption process and helical movement. In the AAA, each population member is called algal colony. In the evolutionary process, if the algal colony receives enough light in sufficient nutrient conditions, it grows and reproduces itself to generate new algae cells in time. On the other side, the algal colony does not receive enough survival light for short duration and dies eventually. Based on this concept, algal colony provides good solution to grow more and replicate by the smallest algal colony dying in the evolutionary process. So, the evolution process requires the size of algal colony (G), here in algorithm G is the fitness value of the objective function. In adaption process, an insufficient grown algal colony tries to resemble itself to the biggest algal colony in the environment. This process is taken care of the algorithm in the form of starvation. The initial starvation value is zero for each artificial algae, and it increases with time, so the artificial algae having the highest starvation level are adapted based on the value of adaption parameter A_p which is constant on the interval $[0,1]$. Algae generally swim and try to stay close to the water surface to have adequate light for the survival. They swim helically in the liquid to restrict gravity and drag force of the liquid. In AAA, the gravity restricting movement is taken as 0, and viscous drag is considered in the form of shear force, which is proportional to the size of algal cell. It is in the spherical

shape therefore frictional surface becomes the surface of the hemisphere.

$$\tau(x_i) = 2\pi \left(\sqrt[3]{\frac{3G_i}{4\pi}} \right)^2 \tag{1}$$

Where $\tau(x_i)$ is the friction surface.

For one-dimensional problem algal colony moves in a single direction, for two-dimensional problem algal movement is sinusoidal and for three or more-dimensional problem algae follow the helical movement.

The step-wise implementation of basic AAA is explained as below:

Step-1: Set algorithm parameters shear force (Δ), loss of energy (e), energy (E) and adaption parameter (A_p), and define number of design variable (DV) and maximum number of generation.

Define function for the optimization

Minimize $f(X)$, Subject to $X = \{x_1, x_2, \dots, x_{DV}\}$.

Step-2: Initialize a population of algae colonies and evaluate the algae size (G_i) for each algal colony and calculate friction surface τ_i .

Step-3: Set $starvation=true$ and update the solution by using the following condition Till $E(x_i)>0$ and $starvation=true$.

Choose j from the population and three dimensions of j randomly for helical movement $k, l, \text{ and } m$.

Update the solution by using helical movement equation between x_i and x_j :

$$x_{ik}^{t+1} = x_{ik}^t + (x_{jk}^t - x_{ik}^t) (\Delta - \tau_i) \cos\alpha$$

$$x_{il}^{t+1} = x_{il}^t + (x_{jl}^t - x_{il}^t) (\Delta - \tau_i) \sin\beta$$

$$x_{im}^{t+1} = x_{im}^t + (x_{jm}^t - x_{im}^t) (\Delta - \tau_i) p$$

α, β are random angles in the range $[0, 2\pi]$, and p is a random number in the range $[-1, 1]$.

Calculate energy loss as $E(x_i) = E(x_i) - (e/2)$.

Check if new solution is better than the update solution and set $starvation=false$ else again reduce energy $E(x_i)$ if still $E(x_i)>0$ and $starvation=true$ then increase the adaption parameter (A_p) and set $x_i = x_s$.

Step-4: Evaluate the size (G) of population and choose one dimension for reproduction r and replace $x_{worst}_r = x_{best}_r$.

Step-5: Generate any random number ($rand$) and check if $rand < A_p$ than the update starving solution (x_s) by using the following equation:

$$x_s^{t+1} = x_s^t + rand(x_{best}^t - x_s^t)$$

Step-6: Repeat the procedure till the termination criteria are attained.

As seen from the above procedure, AAA is a population-based method, and it follows the concept of the living behavior of the microalgae to update its solution. Genetic Algorithm (GA) is the basic nature-based algorithm, and most of the recent algorithms can be algorithmically linked to it to understand different algorithm better and richer [18, 59]. Such algorithmic linking can provide appropriate direction to modify the existing algorithms and can be useful to even predict the effect of specific modifications to an algorithm. It can be understood that basic GA follows recombination and mutation process to update the solution. AAA also follows the recombination operator that involves two parent solutions, one being the serial parent as per the sequence of the population and the other parent is randomly chosen from the population. Mutation in AAA is performed on the worst solution from the population by replacing any one dimension directly from the best solution in that generation. AAA follows an elite-preservation operation in which new solutions are directly compared with the current ones, and the best one is preserved. Also, AAA does not change all the dimensions in the current solution to get the new one; rather it only updates three dimensions at a time to have perturbation phenomenon for updating a solution.

4 Multi-Objective Artificial Algae Algorithm (MO-AAA)

We introduce elitist non-dominated sorting method and diversity preserving crowding distance approach of NSGA-II in the proposed MO-AAA algorithm for sorting of the population in different non-domination levels with computed crowded distance. We describe an elitist non-dominated sorting for obtaining different non-domination levels, and then we explain the crowding distance approach to preserve the diversity among the optimal set of solutions.

Firstly, for each solution obtained from the basic AAA algorithm or from initially generated random population P_o , all the objectives from the objective vector F are evaluated. Also, a domination count n_p is defined as the number of solutions dominating the solution p . S_p is a set of solutions dominated by solution p and calculated. Secondly, all the solutions p are assigned a domination count zero and are put in first non-dominated level also known as Pareto Front (PF), and their non-domination rank (NDR_p) is set to 1. Thirdly, for each solution p with $n_p = 0$, each member q of the set S_p is visited, and its domination count n_q is reduced by one. While reducing n_q count if it falls to zero the corresponding solution q is put in second non-domination level, and NDR_q is set to 2. The procedure is repeated for each member of the second non-domination level to obtain the third non-domination level, and subsequently, the procedure should be repeated until

the whole population is sorted into different non-domination levels.

In crowding distance approach for maintaining diversity among the obtained solutions firstly the population is sorted according to value of each objective function in ascending order. An infinite crowding distance is then assigned to the boundary solutions, $i = 1$ and $i = l$, of each objective. Here l is the total number of solutions in a particular non-dominated set. The boundary solutions are the minimum ($i = 1$) and maximum ($i = l$) function values. Except the boundary solutions all the other solutions of the sorted population ($i = 2$ to $l-1$) for each objective j ($j = 1, 2, \dots, m$) are assigned the the crowding distance (d_j^i) as:

$$d_j^i = \frac{f_j^{i+1} - f_j^{i-1}}{f_j^{\max} - f_j^{\min}}$$

In above equation, the right-hand side term is the difference in values of objective function j for two neighboring solutions ($i+1$ and $i-1$) of solution i . Once all the solutions in the sorted population of a particular non-dominated set are assigned the crowding distance then each solution i is assigned two entities, non-domination rank NDR_i and crowding distance CD_i . A crowded comparison operator ($<_n$) is used as follows to compare two solutions (i and j) as follows $i <_n j$, if ($NDR_i < NDR_j$) or ($(NDR_i = NDR_j)$ and ($CD_i > CD_j$)). This is, between two solutions, the one with the lower non-domination rank is preferred and if both the solutions have same non-domination rank, then one with the higher crowding distance is preferred.

It can be observed from the previous section that AAA algorithm cannot be directly used for the multi-objective optimization. It requires critical modifications to be done in the basic AAA to make it suitable for multi-objective optimization because multi-objective optimization consists of a set of solutions (Pareto solutions) rather than a single optimum solution. All the Pareto solutions are equally important with respect to the objective functions. Hence, it is difficult to identify or indicate any solutions as the best solution (x_{best}) or the worst solution (x_{worst}). It is also difficult to measure a quantity like $f(X') < f(X)$, where X' and X being the new solution and the existing solution, respectively. The reason for the above challenge is due to the involvement of more than one objective in the multi-objective problems. So, modifications are done in the basic AAA algorithm to make it adaptive for the multi-objective optimization problems. To compare the updated solution with the existing solution, we have used the concept of rank and the crowding distance. The updated solution is considered better if it possesses better rank compared to the existing solution. It may happen that the rank of both existing and the updated solution remain same. In that

case the solution with better distance value is considered superior solution. The solution with the better rank and the distance value is considered as the best solution from the population i.e x_{best} and by using the same concept the solution with inferior rank and distance value will be considered as x_{worst} . So, it can be observed that the modification is done by not losing the concept of basic AAA algorithm. However, the mathematical expressions for updating the solution remain unaltered as that of basic AAA algorithm. It can also be noted that the updated solution may fly away from the boundary and it is required to set back within the boundary conditions. Padhye et al. [60] has presented different concepts to handle boundary violation. MO-AAA uses set on boundary approach to handle boundary violations. In this approach, if the updated solution is less than the lower limit, it is set as the lower limit value and if the updated solution is more than the upper limit, it is set as the upper limit value.

Algorithm 1 MO-AAA

Generate P_o randomly in S and evaluate F for the generated P_o
 Sort the P_o based on the elitist non-dominated sort method and find the NDR and fronts.
 Compute CD for each front.
 Update solutions (P_j) using AAA algorithm.
 Merge P_o and P_j to create $P_i = P_o \cup P_j$.
 For P_i perform Step 2.
 Based on NDR and CD sort P_i .
 Replace P_o with P_i for first N_{pop} members of P_i

The procedure of the proposed MO-AAA algorithm has been shown in Algorithm-1. Firstly, initialize parameters such as population size (N_{pop}), termination criteria, and the maximum number of generation (N_{gen}) to run the algorithm. Secondly, a random parent population P_o in feasible region S is generated and each objective function of the objective vector F for P_o is evaluated. Next, elitist based non-dominated sorting and crowding distance computation as explained in earlier section is applied on P_o . Thirdly, AAA algorithm is employed to create the new population P_j , which is then merged onto P_o to form the merged population P_i . This P_i is sorted based on elitism non-domination, and based on the computed values of NDR and CD . The best N_{pop} solutions are updated to form a new parent population. This process is repeated till the maximum number of generations (iterations) are reached. It should be noted that the same algorithm can also be used with the termination criteria based on the number of function evaluations. Since non-dominated sorting and crowding distance assignment of MO-AAA are adopted from NSGA-II, the computational complexity of MO-AAA

Table 1 Controlling parameter for MO-AAA

Parameter	Value
Shear force (Δ)	2
Loss of energy (e)	0.3
Energy (E)	1
Adaption parameter (A_p),	0.5

is also same as NSGA-II which is $O(m(N_{pop})^2)$, where m is the total number of objective functions and N_{pop} is the population size.

5 Numerical experimentation

This section presents the numerical experimentation and performance evaluation of MO-AAA for classical multi-objective benchmark problems, challenging benchmark problems of CEC 2009 and engineering design benchmark problems. The results of MO-AAA are compared with other efficient as well as recently developed multi-objective optimization algorithms. The controlling parameters for MO-AAA are given in Table 1.

5.1 Benchmark problems-1

We carry out our numerical experiments on various classical benchmark multi-objective problems in this subsection. We consider classical and well-established benchmark multi-objective functions such as SCH, ZDT1, ZDT2, ZDT3, and LZ [46, 74, 98], which have distinct characteristics. For example, SCH is a one-dimensional problem with a convex Pareto front. ZDT1 has convex Pareto front, whereas ZDT2 has non-convex Pareto front. The Pareto front of ZDT3 has

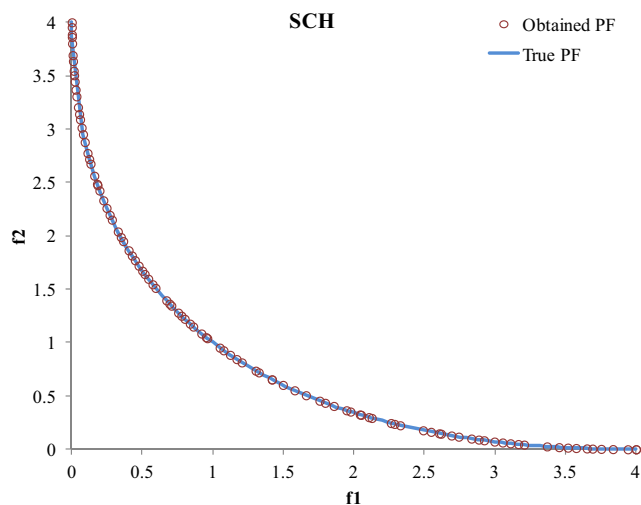


Fig. 1 Pareto front for SCH function

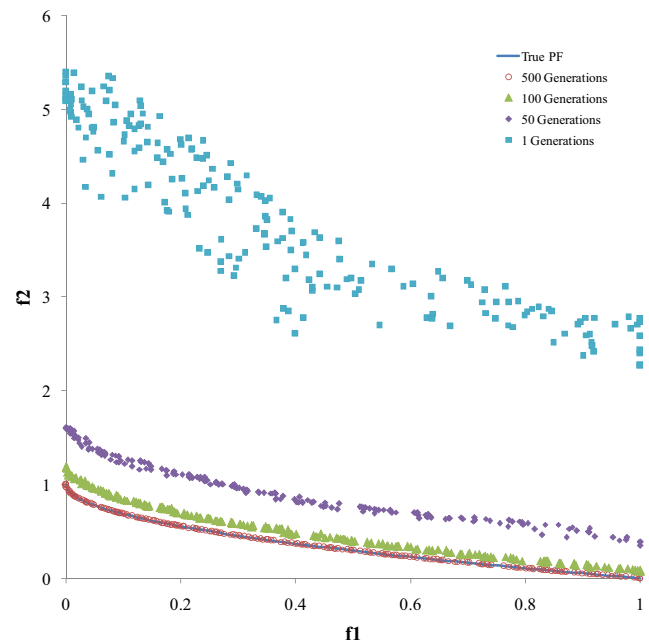


Fig. 2 Convergence ability and Pareto front for ZDT1 function

the discontinuous Pareto front. The objective functions in LZ function are multi-modal, and they add an additional challenge for the algorithm to find the true Pareto front. In this work, the performance of the algorithms is measured by Generational Distance (GD), Spacing (S) or Spread (Δ) [17]. We calculate GD between the true Pareto front (PF^t) and the obtained Pareto front (PF^o). GD indicates the

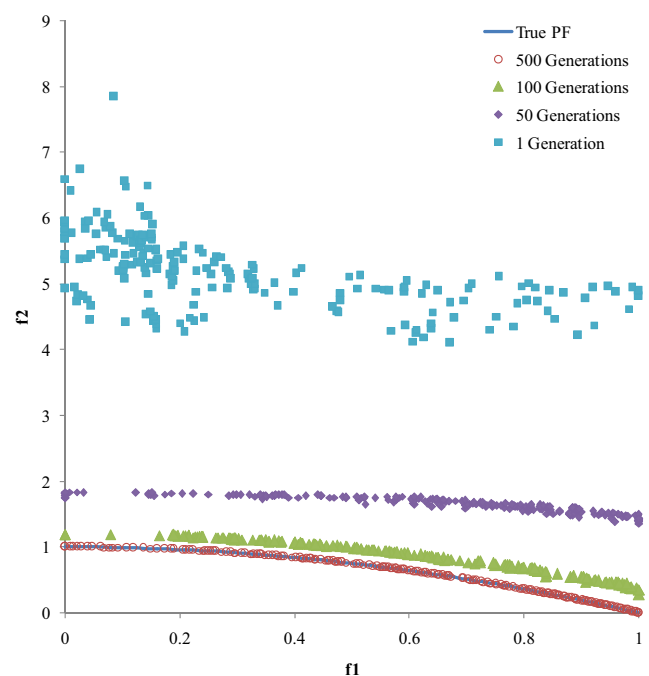


Fig. 3 Convergence ability and Pareto front for ZDT2 function

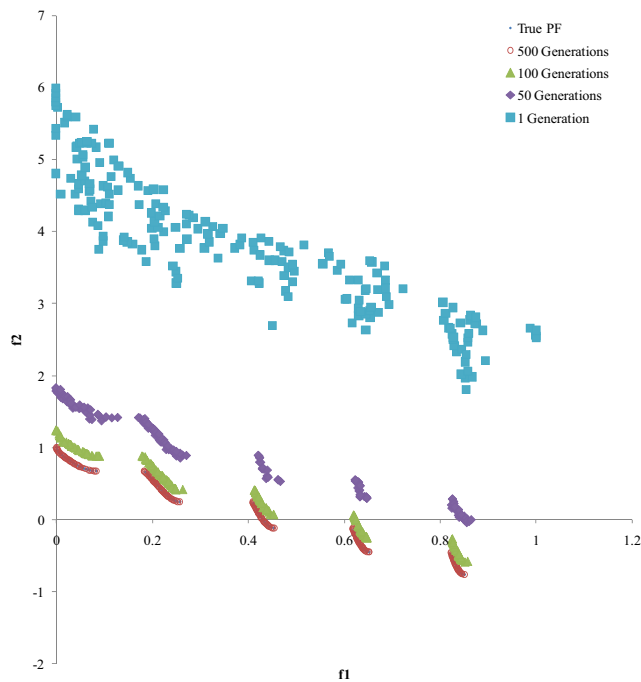


Fig. 4 Convergence ability and Pareto front for ZDT3 function

average distance between the obtained Pareto front and the true Pareto front. GD is defined as:

$$GD = \frac{\left(\sum_{p=1}^k (d_i)\right)}{k}$$

Where, $d_i = \left(\sum_{i=1}^n \left(PF_{i,p}^O - PF_{i,p}^t\right)^2\right)^{1/2}$

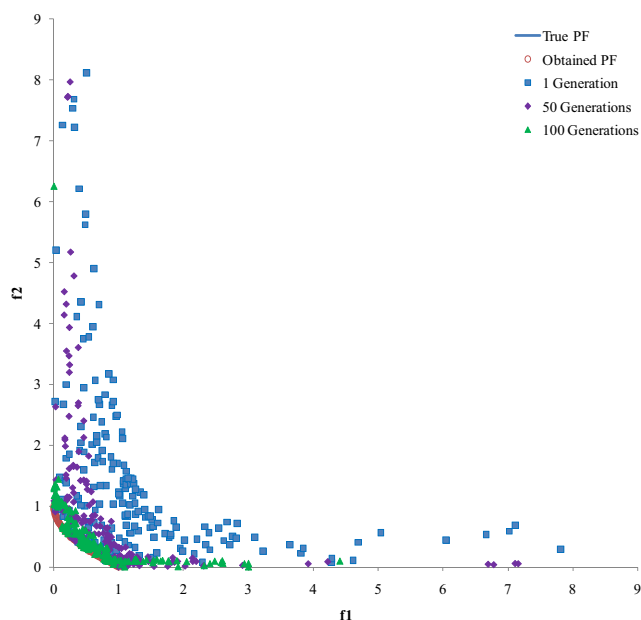


Fig. 5 Convergence ability and Pareto front for LZ function

Table 2 Results (GD) for benchmark multi-objective problems

	SCH	ZDT1	ZDT2	ZDT3	LZ
VEGA	6.98E-02	3.79E-02	2.37E-03	3.29E-01	1.47E-03
NSGA-II	5.73E-03	3.33E-02	7.24E-02	1.14E-01	2.77E-02
MODE	9.32E-04	5.80E-03	5.50E-03	2.15E-02	3.19E-03
DEMO	1.79E-04	1.08E-03	7.55E-04	1.18E-03	1.40E-03
Bees	1.25E-02	2.40E-02	1.69E-02	1.91E-01	1.88E-02
SPEA	5.17E-03	1.78E-03	1.34E-03	4.75E-02	1.92E-03
MO-AAA	5.63E-05	3.29E-05	1.89E-05	1.54E-03	1.98E-03

Where, k is the number of Pareto solutions, n is the number of objective functions, $PF_{i,p}^O$ indicates the p^{th} obtained Pareto solution for the i^{th} objective and $PF_{i,p}^t$ indicates the nearest point on true Pareto front from $PF_{i,p}^O$. Spacing is defined as the following:

$$S = \frac{1}{k-1} \sum_{p=2}^k (D_p - \bar{D})^2$$

Where, D_i is the absolute difference between the two consecutive solutions in the obtained Pareto front (PF^O). It is defined as:

$$D_p = \sum_i^n \left| PF_{i,(p-1)}^O - PF_{i,p}^O \right|$$

\bar{D} is the average of all D_p .

Spacing(S) specifies the spread of the obtained Pareto front. It gives the standard deviation of D_g . The small value of S indicates the uniform spacing of the obtained Pareto solutions. Spread is defined as:

$$\Delta = \frac{\sum_{j=1}^n d_j^{ex} + \sum_{p=2}^k |d_p - \bar{d}|}{\sum_{j=1}^n d_j^{ex} + (k-1)\bar{d}}$$

Where, d_p is the Euclidian distance between two consecutive points of the obtained Pareto front and d_j^{ex} is the Euclidian distance between the obtained Pareto front and the true Pareto front. \bar{d} is the average of d_p . Spread checks the condition for the obtained Pareto front to cover the true Pareto front. The smaller value of Δ indicates better spread for the obtained Pareto front with uniform distribution.

Table 3 Results (Δ) for the multi-objective benchmark problems

	ZDT1	ZDT2	ZDT3
NSGA-II(RC)	0.3903	0.4307	0.7385
NSGA-II(BC)	0.4632	0.4351	0.5756
SPEA	0.7301	0.6781	0.6657
PAES	1.2297	1.1659	0.78992
MO-AAA	0.2547	0.3245	0.4326

Table 4 Results (mean IGD) for unconstrained multi-objective benchmark problems of CEC 2009

A*	UF1	UF2	UF3	UF4	UF5	UF6	UF7	UF8	UF9	UF10
1	3.91E-03	6.72E-03	7.62E-02	3.56E-02	4.13E-02	4.82E-02	6.42E-03	5.98E-02	2.00E-01	5.12E-01
2	8.79E-03	7.84E-03	6.93E-02	7.01E-02	9.79E-02	5.53E-02	6.80E-02	9.93E-02	1.64E-01	1.79E-01
3	4.35E-03	6.79E-03	7.42E-03	6.39E-02	1.81E-01	5.87E-03	4.44E-03	5.84E-02	7.90E-02	4.74E-01
4	6.46E-03	6.15E-03	5.31E-02	2.36E-02	1.49E-02	5.92E-02	4.08E-02	1.13E-01	1.14E-01	1.53E-01
5	1.04E-02	6.79E-03	3.34E-02	4.27E-02	3.15E-01	6.67E-02	1.03E-02	6.84E-02	4.90E-02	3.22E-01
6	6.18E-03	4.84E-03	5.12E-02	5.80E-02	7.78E-02	6.54E-02	5.57E-02	6.73E-02	6.15E-02	1.95E-01
7	7.85E-03	1.23E-02	1.50E-02	4.35E-02	1.62E-01	1.76E-01	7.30E-03	8.24E-02	9.39E-02	4.47E-01
8	6.20E-03	6.40E-03	4.29E-02	4.76E-02	1.79E+00	5.56E-01	7.60E-03	2.45E-01	1.88E-01	5.65E-01
9	5.34E-03	1.20E-02	1.06E-01	2.65E-02	3.93E-02	2.51E-01	2.52E-02	2.49E-01	8.25E-02	4.33E-01
10	7.70E-02	2.83E-02	9.35E-02	3.39E-02	1.67E-01	1.26E-01	2.42E-02	2.16E-01	1.41E-01	3.70E-01
11	2.99E-02	2.28E-02	5.49E-02	5.85E-02	2.47E-01	8.71E-02	2.23E-02	2.38E-01	2.93E-01	4.11E-01
12	3.59E-02	1.62E-02	7.00E-02	4.06E-02	9.41E-02	1.29E-01	5.71E-02	1.71E-01	1.89E-01	3.24E-01
13	8.56E-02	3.06E-02	2.71E-01	4.62E-02	1.69E-01	7.34E-02	3.35E-02	1.92E-01	2.32E-01	6.28E-01
14	1.22E-02	8.10E-03	1.03E-01	5.13E-02	4.30E-01	1.92E-01	5.85E-02	9.45E-02	9.83E-02	7.43E-01
15	5.96E-02	1.89E-02	9.90E-02	4.27E-02	2.25E-01	1.03E-01	1.97E-02	4.23E-01	3.42E-01	3.62E-01
16	1.15E-02	1.24E-02	1.06E-01	5.84E-02	5.66E-01	3.10E-01	2.13E-02	8.63E-02	7.19E-02	8.45E-01
17	5.79E-03	7.84E-03	6.93E-02	3.59E-02	3.79E-02	5.53E-02	2.14E-02	9.93E-02	1.17E-01	1.79E-01

*A-Algorithms: 1-Multi-objective moth flame optimization (NS-MFO), 2-Multi-objective Teaching Learning Based Optimization (MO-TLBO), 3-(Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D), 4-Multiple Trajectory Search (MTS), 5-Dynamical Multi-objective Evolutionary Algorithm with Decomposition Technique (DMOEADD), 6-Multi-objective ABC (MOABC), 7-Multi-objective Evolutionary Algorithm Based on Determined Weight and Sub-regional Search (Liu Li Algorithm), 8-MOEA/D with Guided Mutation and Priority Update (MOEADGM), 9-Generalized Differential Evolution 3 (GDE3), 10-Differential Evolution with Self-Adaptation and Local Search for Constrained Multi-objective Optimization Algorithm (DECMOSA-SQP), 11-Clustering Multi-objective Evolutionary Algorithm (Clustering MOEA), 12-Archive-based Micro Genetic Algorithm (AMGA), 13-Orthogonal Multi-objective Evolutionary Algorithm (OMOE-II), 14-Multi-objective Self-adaptive Differential Evolution Algorithm with Objective-wise Learning Strategies (OW-MOSaDE), 15-Multi-objective Evolutionary Programming (MOEP) using fuzzy rank-sum with diversified selection, 16-Non-dominated Genetic Algorithm with Local Search (NSGAILS), 17-Multi-objective Biogeography Based Optimization with ACO (MO-BBO_ACO)

Table 5 Results (mean IGD) for constrained multi-objective benchmark problems of CEC 2009

A*	CF1	CF2	CF3	CF4	CF5	CF6	CF7	CF8	CF9	CF10
1	6.82E-02	3.92E-03	4.25E-02	2.83E-03	6.23E-02	4.01E-02	2.88E-02	3.11E-02	7.62E-02	9.99E-02
2	2.49E-02	1.20E-02	9.80E-02	8.68E-03	7.64E-02	1.64E-02	3.33E-02	2.98E-01	8.10E-02	3.01E-01
3	1.13E-02	2.10E-03	5.63E-02	6.99E-03	1.58E-02	1.50E-02	1.91E-02	4.75E-02	1.43E-01	1.62E-01
4	8.50E-04	4.20E-03	1.83E-01	1.42E-02	1.10E-01	1.39E-02	1.04E-01	6.07E-02	5.05E-02	1.97E-01
5	1.92E-02	2.68E-02	1.04E-01	1.11E-02	2.08E-02	1.62E-02	2.47E-02	1.09E+00	8.51E-02	1.38E-01
6	1.08E-02	8.00E-03	5.13E-01	7.07E-02	5.45E-01	2.07E-01	5.36E-01	4.06E-01	1.52E-01	3.14E-01
7	1.08E-01	9.46E-02	1.00E+06	1.53E-01	4.13E-01	1.48E-01	2.60E-01	1.76E-01	1.27E-01	5.07E-01
8	2.94E-02	1.60E-02	1.28E-01	7.99E-03	6.80E-02	6.20E-02	4.17E-02	1.39E-01	1.15E-01	4.92E-01
9	6.92E-03	1.18E-02	2.40E-01	1.58E-02	1.84E-01	2.01E-02	2.33E-01	1.11E-01	1.06E-01	3.59E-01

*A-Algorithms: 1-Multi-objective Moth Flame Optimization (NS-MFO), 2-Multi-objective Teaching Learning Based Optimization (MO-TLBO), 3-Dynamical Multi-objective Evolutionary Algorithm with Decomposition Technique (DMOEADD), 4- Multi-objective Evolutionary Algorithm Based on determined weight and sub-regional search (Liu Li Algorithm), 5- Multiple Trajectory Search (MTS), 6- MOEA/D with Guided Mutation and Priority Update (MOEADGM), 7- Differential Evolution with Self-adaptation and Local Search for Constrained Multi-objective Optimization Algorithm (DECMOSA-SQP), 8- Generalized Differential Evolution 3 (GDE3), 9-Non-dominated Genetic Algorithm with Local Search (NSGAILS)

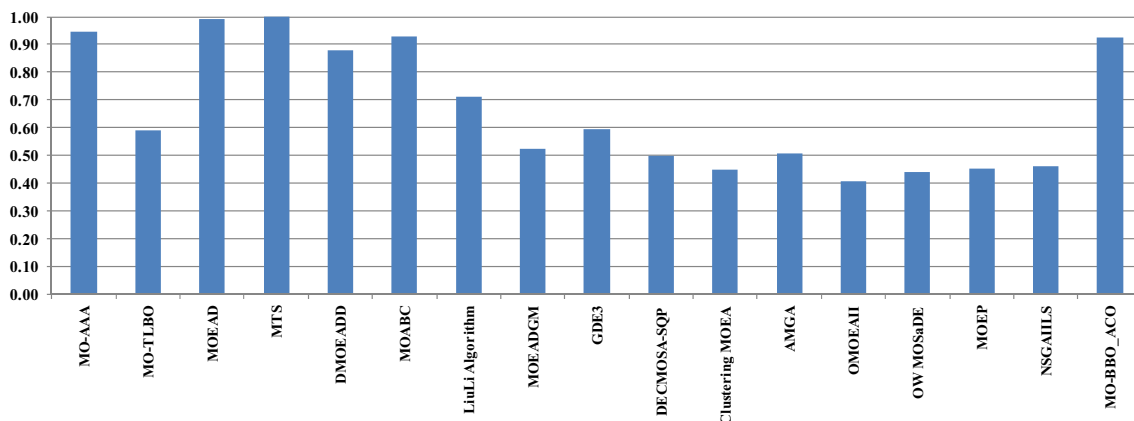
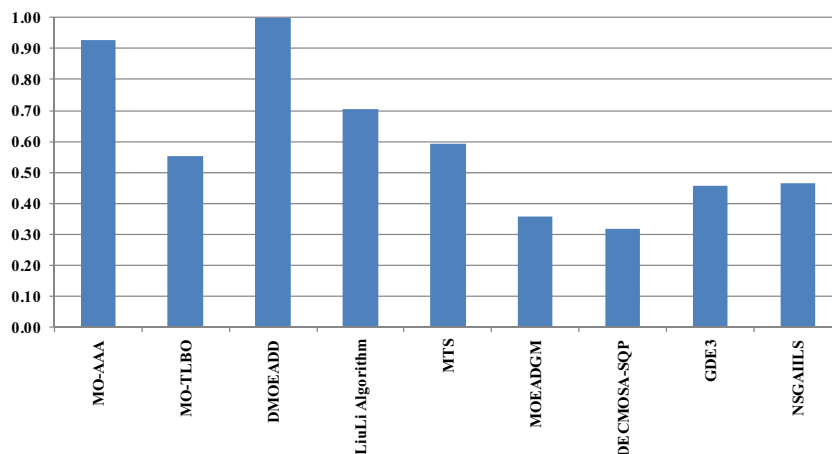


Fig. 6 Comparison of multi-objective optimization algorithms for unconstrained functions based on Friedman rank test

For the numerical experimentation, the population size is set equal to the Pareto solutions required in PF, and maximum numbers of generations are set as 500 for ZDT1, ZDT2, ZDT3 and SCH functions. For LZ function, the number of generations is set as 1000 to maintain the uniformity of experimental setup with the published results. For all the problems, the number of Pareto Solutions (k) is considered as 200. The Pareto front for SCH, ZDT1, ZDT2, ZDT3, and LZ are shown in Figs. 1, 2, 3, 4 and 5, respectively. Figures 2 to 5 also show the convergence ability of MO-AAA for ZDT1, ZDT2, ZDT3, and LZ, relies on various types of multi-objective problems. The result presents the graphical presentation of the Pareto front at 1st, 50th, 100th and final generations. All the figures show the true Pareto front and the obtained Pareto front. It can be noted from the results that for all the functions, MO-AAA is capable of approximating the true Pareto front. It can also be observed from the convergence figures that MO-AAA can converge to an optimum solution for the multi-objective functions..The rate of convergence is more in first 100 generations, and the solutions try to get near Pareto

solutions within that range. However, with the increase in the computation, solutions try to achieve true Pareto front. The effectiveness of the results is checked by the above-mentioned performances measures. The results for the GD for all the functions are summarized in Table 2, where the results are compared with the state-of-the-art multi-objective algorithms such as NSGA-2, SPEA, VEGA, MODE, DEMO, and MO-Bees. The results except MO-AAA are reproduced from Yang, 2013 and Yang and Deb, 2013. The results summarized in Table 2 are the average result obtained in 25 independent runs. It can also be observed from the results that for SCH function, MO-AAA has produced better GD then all the rest of the algorithms. For ZDT1, ZDT2, MO-AAA is better compared to all the algorithms. For ZDT3 it is only inferior to DEMO. For LZ, MO-AAA is inferior to MODE and SPEA. It can be summarized that MO-AAA is capable of finding true Pareto solutions for the problems with convex, non-convex and discrete Pareto front. The results are also compared based on the spread (Δ) with the other multi-objective optimization algorithms such as NSGA-II (real coded), NSGA-II (binary

Fig. 7 Comparison of multi-objective optimization algorithms for constrained functions based on Friedman rank test



coded), SPEA and PAES. The result for the Δ is shown in Table 3. It can be noted from the results that MO-AAA has shown better Δ for ZDT1, ZDT2, and ZDT3.

5.2 Benchmark problems-2

Several challenging benchmark functions are provided by Congress on Evolutionary Computation (CEC) for testing performance parameters of multi-objective optimization algorithms. Benchmark functions for multi-objective optimization are provided in CEC 2009 [96]. The effectiveness of MO-AAA algorithm is evaluated by investigating on 20 well-defined benchmark functions of CEC 2009. Out of all these 20 benchmark problems, ten functions are multi-objective unconstrained problems (UF1-UF10), and the remaining 10 are multi-objective constrained functions (CF1-CF10). The detailed mathematical formulations of the considered test functions are given in [96]. The performance of the algorithms is evaluated by inverted generational distance (IGD) measure. The value of IGD indicates the distance of the approximate Pareto front from the true Pareto front. A low value of IGD indicates that the obtained Pareto front is closely approached to the true Pareto front. To calculate IGD, assume P^* be a set of uniformly distributed points along the true Pareto front in the objective space. Let A be a set of points representing obtained Pareto front. Then the average distance from P^* to A is known as IGD and describe as follows:

$$IGD(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|}$$

Where, $d(v, A)$ is the minimum Euclidian distance between v and the other points in A . If $|P^*|$ is large enough to represent the Pareto front very well, both diversity and convergence of the approximated set A could be measured using $IGD(A, P^*)$. An optimization algorithm will try to minimize the value of $IGD(A, P^*)$ measure. To obtain smaller values of this measure, the approximated set A must be very close to the true Pareto front and cannot miss any part of the whole Pareto front. To maintain the consistence in the comparison of the competitive algorithms, a common platform is provided in [96]. The total number of function evaluations is set as $3.0E05$, and population size is set as 50. The comparison is based on the average value of IGD obtained in 30 independent runs. The result of MO-AAA is compared with 16 other well-known algorithms for unconstrained problems. These algorithms are derived from other optimization methods such as genetic algorithm, differential evolutions, artificial bee colony technique, biogeography-based optimization or teaching learning-based optimization techniques. In Table 4, the results of average IGD are listed based on 30 runs for ten unconstrained multi-objective benchmark problems

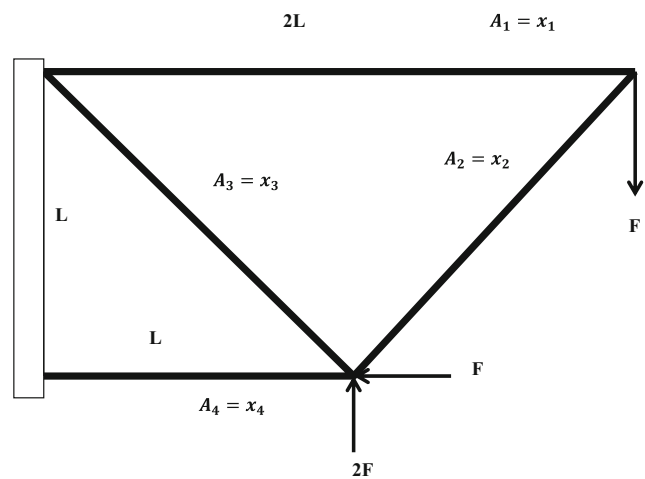


Fig. 8 Four bar truss problem

(UF1-UF10). In Table 5, the results of MO-AAA for constrained benchmark problems are compared with eight other algorithms, and the results of average IGD are listed. All the results except MO-AAA are rewritten from [4, 64, 73] for the comparison purpose. Friedman rank test is performed to get an idea about the performance rank of the algorithms for unconstrained and constrained problems separately based on average IGD. Variations of Normalized Friedman rank value for unconstrained problems are shown in Fig. 6 and for constrained problems are shown in Fig. 7. It can be observed from the results that for unconstrained multi-objective problems MO-AAA has ranked 3rd among 17 algorithms and it can be considered competitive with MOEAD, MTS, MOABC, and MO_BBO-ACO considering the range above 90%. MO-AAA has secured 2nd rank among nine algorithms for constrained problems, and it is comparable with DMOEADD within the top 90% range.

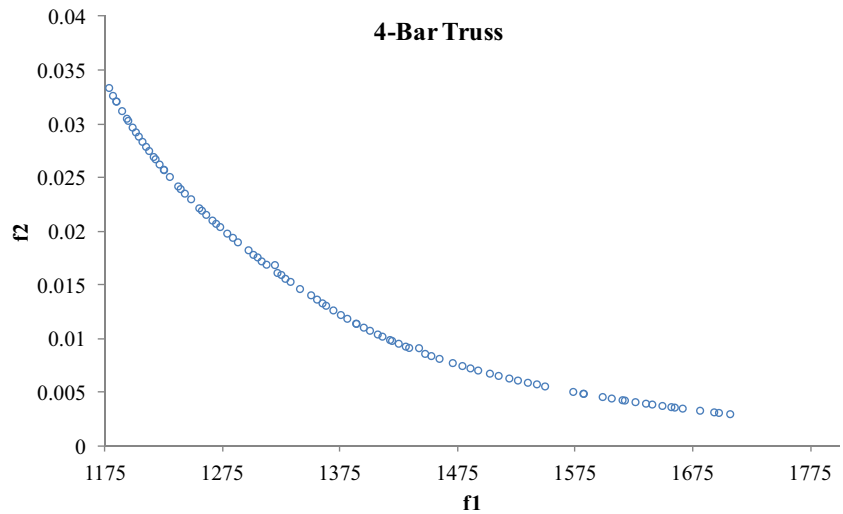
5.3 Engineering design benchmark problems

In this subsection, MO-AAA is experimented on six different multi-objective engineering design problems which are widely used in the research to investigate the multi-objective

Table 6 Result (S) for the four-bar truss problem

Algorithms	S	
	Mean	SD
NSGA-II	2.3635	0.2551
MOPSO	2.5303	0.2275
Micro-GA	8.2742	16.8311
PAES	3.2314	5.9555
MOWCA	2.5816	0.0298
MO-AAA	2.3861	0.2314

Fig. 9 Pareto front for the four-bar truss problem



optimization algorithms. All the considered problems possess different characteristics. The algorithms parameters for the MO-AAA are same as that for the benchmark problems considered in the previous subsection. All the engineering problems experiment with a population size of 100 (solutions required in the Pareto front) and 500 iterations. For all the problems, 100 Pareto solutions are generated, and the results mentioned in this work are the average result obtained in the 30 runs to maintain the same experimental settings with [71].

5.3.1 Four bar truss problem

The aim of this problem is to minimize volume and displacement of joints simultaneously. Area of each link is considered as design variables. This problem is an

unconstrained problem with all the continuous design variables. The system is shown in Fig. 8.

The problem can be stated as below

$$\text{Minimize : } f_1(x) = L(2x_1 + \sqrt{2}x_2 + \sqrt{x_3} + x_4)$$

$$\text{Minimize : } f_2(x) = \left(\frac{F}{E}\right) \left(\frac{2}{x_2} + 2\frac{\sqrt{2}}{x_2} - 2\frac{\sqrt{2}}{x_3} + \frac{2}{x_4}\right)$$

Where,

$$F = 10, E = 2e5, L = 200$$

$$1 \leq x_1, x_4 \leq 3, \sqrt{2} \leq x_2, x_3 \leq 3.$$

This problem has a convex Pareto front in the objective space. The extreme point found for this problem is (1174.200, 0.0341) and (1727.739, 0.00276). As the mathematical expression for the exact Pareto front is not available for this problem in the literature, the performance is checked based on the value of spacing (S) due to its availability in the literature for different algorithms. The value of S for MO-AAA along with different algorithms such as NSGA-II, MOPSO, Micro-GA, PAES, and MOWCA [71] are given in Table 6. It can be observed from the results that MO-AAA has produced better or nearly same spacing compared to all the algorithms. The obtained

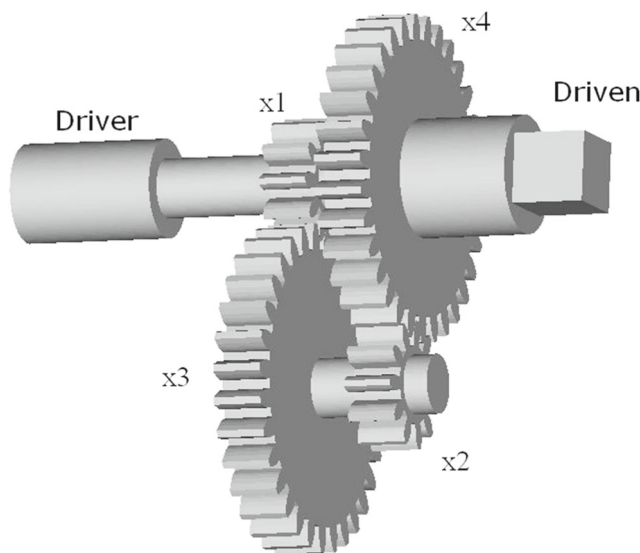
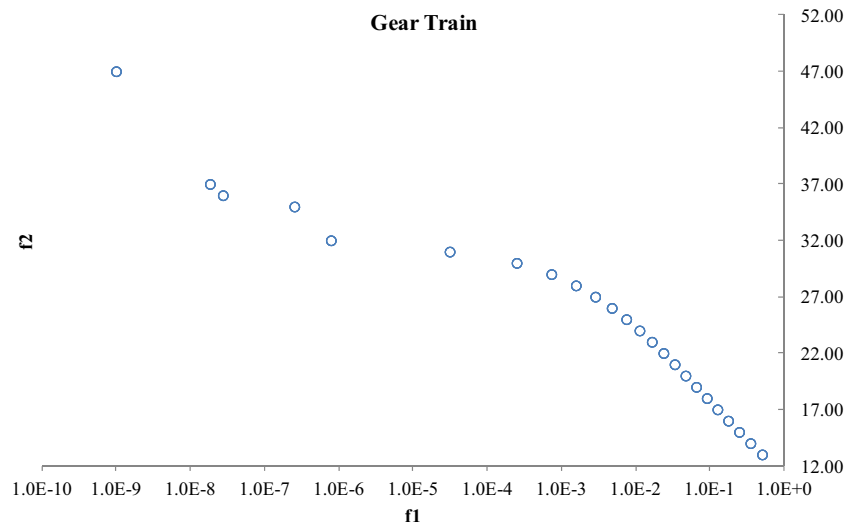


Fig. 10 Gear train

Table 7 Result (extreme Pareto solutions) for the gear train problem

Algorithms		Objective function	
		f_1	f_2
NSGA-II	$f_1 \rightarrow \min$	1.83e-8	37
	$f_2 \rightarrow \min$	5.01e-1	13
MOWCA	$f_1 \rightarrow \min$	4.5e-9	43
	$f_2 \rightarrow \min$	7.32e-1	12
MO-AAA	$f_1 \rightarrow \min$	9.92e-10	47
	$f_2 \rightarrow \min$	7.32e-1	12

Fig. 11 Pareto front for gear train problem



Pareto front is presented in Fig. 9, which justifies the results presented in the Table 5. The obtained Pareto front matches well with the available Pareto front in the literature [71].

5.3.2 Gear train problem

The purpose of this problem is to minimize the maximum size of any one of the gear simultaneously with the minimization of the gear ratio error with reference gear ratio of 1/6.931. All the design variables considered in this example can only take integer values as they represent the number of teeth on each gear. Hence, this problem is a min-max problem with discrete design variables, which offers additional challenges to an optimization algorithm. The system is shown in Fig. 10.

The problem can be stated as:

$$\text{Minimize } f_1(x) = \left(\left(\frac{1}{6.931} \right) - \left(\frac{x_1 x_2}{x_3 x_4} \right) \right)^2$$

$$\text{Minimize } f_2(x) = \max([x_1, x_2, x_3, x_4])$$

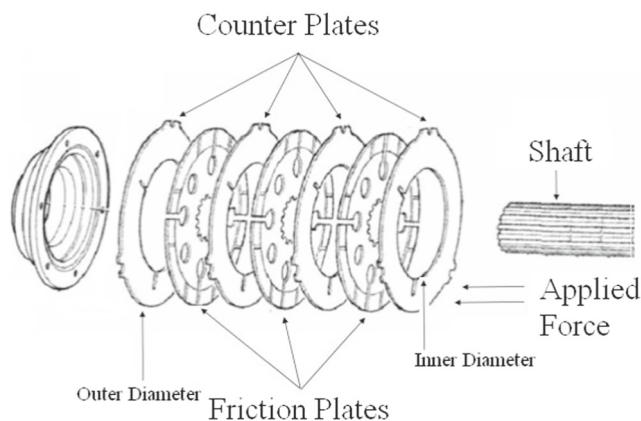


Fig. 12 Exploded view for the multi-plate disk brake

Where, $12 \leq x_1, x_2, x_3, x_4 \leq 60$.

The performance measures such as GD, S, and Δ are not available for the gear train problems. So, the results are compared based on the extreme Pareto solutions available in the literature. The results are compared with NSGA-II and MOWCA. The results are summarized in Table 7.

It can be observed from the results that the extreme Pareto solutions obtained by MO-AAA is (9.92e-10, 47) and (7.32e-1, 12). These extreme solutions are better than NSGA-II and MOWCA for both the conditions. This problem has a discrete Pareto front, and it is shown in Fig. 11, obtained by MO-AAA.

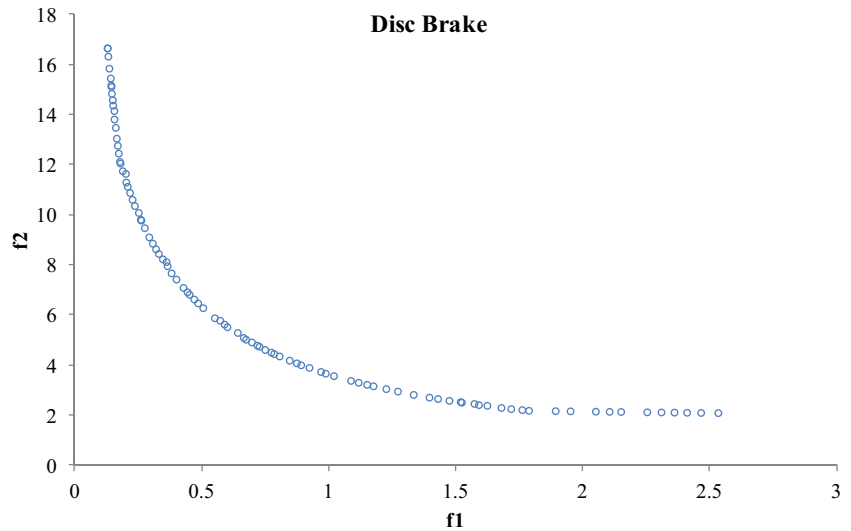
5.3.3 Multi-plate disk brake design

Multi-plate disk brake finds its applications in airplanes, to apply effective braking while landing. The exploded view of the multi-plate disk brake is shown in Fig. 12. The purpose of the problem is to simultaneously minimize the mass of the brake and the stopping time. There are four design variables for the inner radius, outer radius, the engaging force (applied force) and the number of friction surfaces (number of friction plates). Out of these four design variables, the number of friction surfaces can only take a discrete integer values which make it a mixed-integer problem. This is also

Table 8 Result (Δ) for the disk brake problem

Algorithm	Δ	
	Mean	SD
NSGA-II	0.79717	0.06608
<i>paε - ODEMO</i>	0.8401	0.20085
MOWCA	0.46041	0.10961
MO-AAA	0.25824	0.0342

Fig. 13 Pareto front for disk brake problem



a constrained optimization problem for which five different restrictions are introduced for the distance between the radii of the friction plates, length of the brake, pressure sustained by the plates, the maximum limitation for the temperature generated and the braking torque. The problem can be stated as follows:

$$\text{Minimize } f_1(x) = 4.9e - 5 (x_2^2 - x_1^2) (x_4 - 1)$$

$$\text{Minimize } f_2(x) = (9.82e6) \frac{x_2^2 - x_1^2}{x_3 x_4 (x_2^3 - x_1^3)}$$

$$g_1 = 20 + x_1 - x_2$$

$$g_2 = 2.5 (x_4 + 1) - 30$$

$$g_3 = \frac{x_3}{3.14 (x_2^2 - x_1^2)^2} - 0.4$$

$$g_4 = 2.22e - 3x_3 \frac{x_2^3 - x_1^3}{(x_2^2 - x_1^2)^2} - 1$$

$$g_5 = 900 - \left(\frac{2.66e - 2x_3 x_4 (x_3^3 - x_1^3)}{x_2^2 - x_1^2} \right)$$

$$\forall g \leq 0$$

$55 \leq x_1 \leq 80, 75 \leq x_2 \leq 110, 1000 \leq x_3 \leq 3000, 2 \leq x_4 \leq 20, x_4 \in I.$

The extreme Pareto solutions obtained are (0.12740 16.6549) and (2.7915, 2.07204). The results are compared for the value of spacing (Δ). The results are shown in Table 8, where the performance of MO-AAA is compared with NSGA-II, *paε - ODEMO* and MOWCA. It can be noted from the results that MO-AAA has produced better Δ compared to NSGA-II, *paε - ODEMO*, and MOWCA. The Pareto front obtained by using MO-AAA is shown in Fig. 13, which is either same or better than the Pareto front given in [71]. It is useful for the designer to note that the Pareto solutions become nearly stagnant for f_1 at $f_1 \cong 0.2$ and for f_2 at $f_2 \cong 1.75$.

5.3.4 Speed reducer problem

The aim of this problem is to simultaneously optimize the weight of the gear assembly and the transverse deflection of the shaft. The assembly of the speed reducer is shown in Fig. 14. The problem is subjected to different constraints on bending stress of the gear teeth, surfaces stress, transverse

Fig. 14 Speed reducer

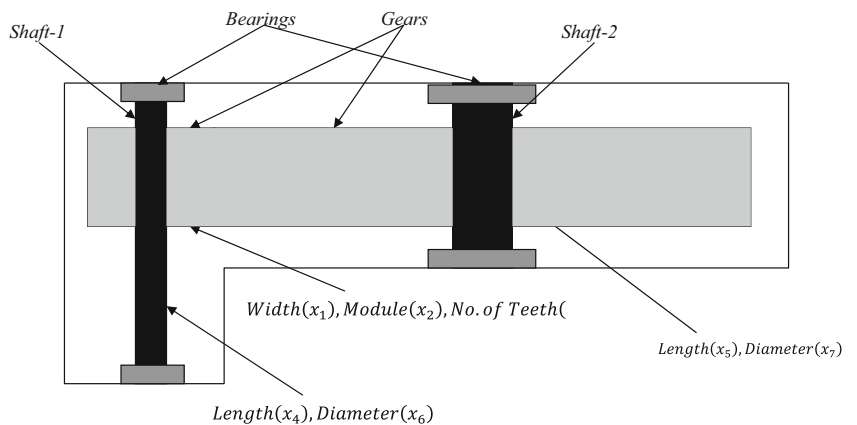


Table 9 Result (S) for speed reducer problem

Algorithms	S	
	Mean	SD
NSGA-II	2.765	3.534
Micro-GA	47.80	32.80
PAES	16.20	4.268
MOWCA	16.68	2.697
MO-AAA	18.2234	3.432

deflections of the shafts and stresses in the shafts. The design variables are the face width, module of teeth, number of teeth in the pinion, length of the first shaft between bearings, length of the second shaft between bearings and the diameter of the first and second shafts, respectively. All the variables are indicated in the Fig. 14. The third variable is an integer, the rest of them are continuous. So, this problem can be classified as a mixed-integer problem.

The problem can be stated as follows:

$$\text{Minimize } f_1 = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

$$\text{Minimize } f_2 = \frac{\sqrt{\left(745 \frac{x_4}{x_2x_3}\right)^2 + 1.69e7}}{0.1 * x_6^3}$$

$$g_1 = -(27/(x_1x_2^2x_3) - 1)$$

$$g_2 = -(397.5/(x_1x_2^2x_3^3) - 1)$$

$$g_3 = -\left(1.93 \frac{x_4^3}{x_2x_3x_6^4} - 1\right)$$

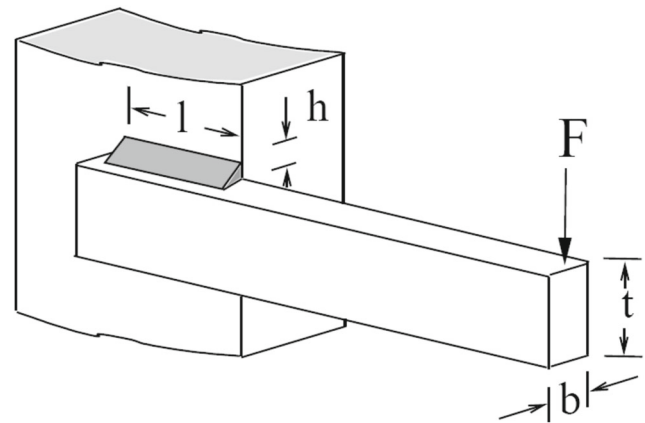


Fig. 16 Welded beam

$$g_4 = -\left(1.93 \frac{x_5^3}{x_2x_3x_7^4} - 1\right)$$

$$g_5 = -\left(\frac{\sqrt{\left(745 \frac{x_4}{x_2x_3}\right)^2 + 16.9e6}}{0.1x_6^3} - 1\right)$$

$$g_6 = -\left(\frac{\sqrt{\left(745 \frac{x_5}{x_2x_3}\right)^2 + 157.5e6}}{0.1x_7^3} - 1\right)$$

$$g_7 = -\left(x_2 * \frac{x_3}{40} - 1\right)$$

$$g_8 = -\left(5 * \frac{x_2}{x_1} - 1\right)$$

$$g_9 = -\left(\frac{x_1}{12 * x_2} - 1\right)$$

Fig. 15 Pareto front for speed reducer problem

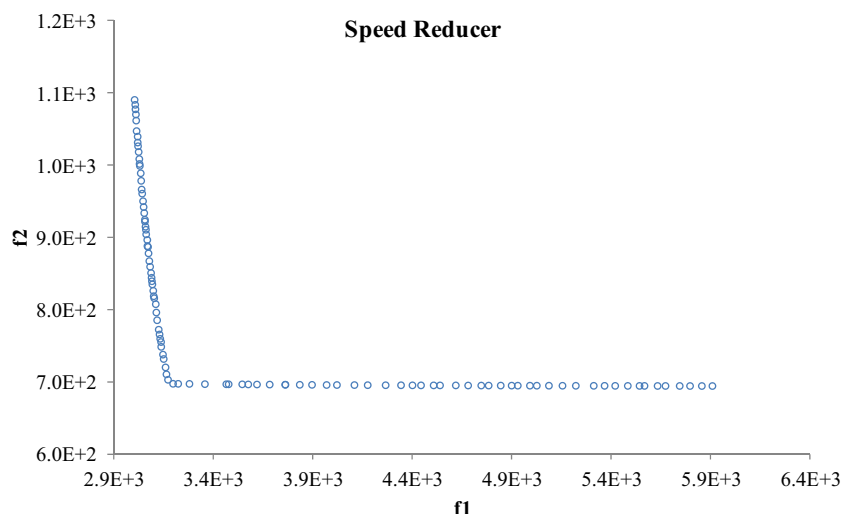


Table 10 Results (Δ) for welded beam problem

Algorithm	Δ	
	Mean	SD
NSGA-II	0.8898	0.1197
<i>paε - ODEMO</i>	0.5860	0.0436
MOWCA	0.2247	0.0928
MO-AAA	0.2349	0.1023

$$g_{10} = - \left(\frac{1.5 * x_6 + 1.9}{x_4} - 1 \right)$$

$$g_{11} = - \left(\frac{1.1 * x_7 + 1.9}{x_5} - 1 \right)$$

$$\forall g_i \geq 0$$

$$2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 x_5 \leq 8.3,$$

$$2.9 \leq x_6 \leq 3.9, 5 \leq x_7 \leq 5.5$$

The extreme Pareto solutions obtained by MO-AAA are (3001.592,1091.24) and (5904.21, 694.728). The results are compared based on the Spacing (S) with the other algorithms such as NSGA-II, Micro-GA, PAES and MOWCA [71]. The results are presented in the Table 9.

It can be observed from the results that, the Spacing of NSGA-II better compared to all the algorithms, but the extreme points of the Pareto front obtained by NSGA-II are inferior compared to other algorithms. Also, the value of S is inferior for MO-AAA to the other algorithms. The Pareto front is given in Fig. 15. It is useful for the designer to note that the Pareto solutions become nearly stagnant for f_2 at $f_2 \cong 697$.

5.3.5 Welded beam design

The purpose of this problem is to minimize the cost simultaneously with the end deflection. The problem is subjected to the constraints on shear stress, bending stress,

weld length, and the buckling load. The four different design variables are the height and the length of the welded joint and thickness and the width of the beam. All the design variables are continuous. The schematic diagram is shown in Fig. 16, where all the variables are shown clearly. The value of objective functions differs much in its values and so such problems are difficult to solve with the weighted sum method. The problem can be stated as:

$$\text{Minimize } f_1 = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2)$$

$$\text{Minimize } f_2 = del$$

$$g_1 = - (tau - taumax)$$

$$g_2 = - (sig - sigmax)$$

$$g_3 = - (x_1 - x_4) \quad g_4 = - (P - pc)$$

Where

$$pc = \left(\frac{4.013 * E \left(\sqrt{\frac{x_3^2 x_4^6}{36}} \right)}{L^2} \right) \left(1 - \left(\left(\frac{x_3}{2L} \right) \sqrt{\frac{E}{4G}} \right) \right)$$

$$del = \frac{4PL^3}{Ex_3^3x_4}$$

$$sig = \frac{6PL}{x_4x_3^2}$$

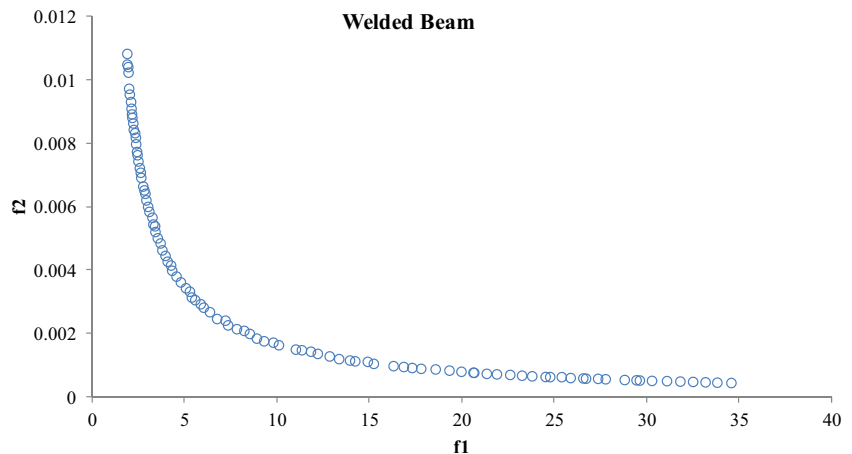
$$J = 2 * \left(\sqrt{2}x_1x_2 \left(\left(\frac{x_2^2}{12} \right) + \left(\frac{x_1 + x_3}{2} \right)^2 \right) \right)$$

$$R = \sqrt{\left(\frac{x_2^2}{4} \right) + \left(\frac{x_1 + x_3}{2} \right)^2}$$

$$M1 = P * \left(L + \frac{x_2}{2} \right)$$

$$tau2 = \frac{M1R}{J}$$

Fig. 17 Pareto front for welded beam problem



$$\tau_{11} = \frac{P}{\sqrt{2}x_1x_2}$$

$$\tau = \sqrt{\tau_{11}^2 + 2 * \tau_{11} * \tau_{12} * \frac{x_2}{2 * R} + \tau_{12}^2}$$

$$\forall g \geq 0$$

$P=6000, L=14, E=30e6, G=12e6, \tau_{max}=13600, \sigma_{max}=30000,$

$del_{max} = 0.25$

$$0.125 \leq x_1x_4 \leq 5, 0.1 \leq x_2x_3 \leq 10.$$

The extreme Pareto solutions obtained by MO-AAA are (1.800, 0.0243) and (35.552, 0.000449). This problem is compared with the other techniques such as NSGA-II, *paε - ODEMO*, and MOWCA [71]. The results are compared based on Δ, and the mean values along with the calculated standard deviation are summarized in Table 10. It can be observed from the results that MO-AAA possesses better spreading compared to the other algorithms and nearly same to that of MOWCA. The Pareto front obtained by MO-AAA is shown in Fig. 17, where it justifies the results.

5.3.6 Spring design problem

The purpose of this problem is to minimize stress and volume simultaneously. The constraints are imposed on the minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the wire diameter (*d*), the mean coil diameter (*D*), and the number of active coils (*N*). The schematic view of the spring is shown in Fig. 18. This problem is special because all the design variables possess different characteristics. The number of turns can only take integer values whereas the diameter of the wire is standardized, and it has to be selected from the set of available diameters. The mean coil diameter can be considered as a continuous variable. So, this problem is the mixed-integer-discrete problem. The problem can be stated as follows:

$$\text{Minimize } f_1(x) = \left(0.253.14^2x_2^2x_3(x_1 + 2)\right)$$

Table 11 Result (extreme Pareto solutions) for spring problem

Algorithms		f_1	f_2
NSGA-II	$f_1 \rightarrow \min$	2.690	187053
	$f_2 \rightarrow \min$	24.189	61949
MOWCA	$f_1 \rightarrow \min$	2.668	188448
	$f_2 \rightarrow \min$	26.93	58752
MO-AAA	$f_1 \rightarrow \min$	2.6972	187552.9
	$f_2 \rightarrow \min$	26.35	57904

$$\text{Minimize } f_2(x) = 8K P_{max} \frac{x_3}{3.14x_2^3}$$

$$g_1 = 1.05x_2(x_1 + 2) + \frac{P_{max}}{k} - l_{max}$$

$$g_2 = d_{min} - x_2$$

$$g_3 = x_2 + x_3 - D_{max}$$

$$g_4 = 3 - C$$

$$g_5 = del_p - del_{pm}$$

$$g_6 = del_w - \frac{P_{max} - P}{k}$$

$$g_7 = 8K P_{max} \frac{x_3}{3.14x_2^3} - S$$

$$g_8 = \left(0.253.14^2x_2^2x_3(x_1 + 2)\right) - V_{max}$$

Where

$$P = 300$$

$$D_{max} = 3$$

$$V_{max} = 30$$

$$P_{max} = 1000$$

$$del_w = 1.25$$

$$l_{max} = 14$$

$$del_{pm} = 6$$

$$d_{min} = 0.2$$

$$S = 189000$$

$$G = 11500000$$

$$C = \frac{x_3}{x_2}$$

Fig. 18 Spring

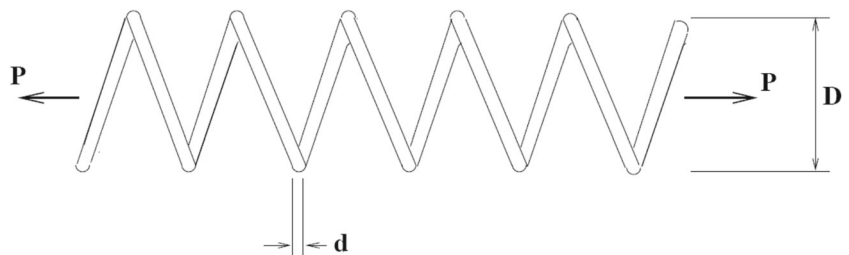
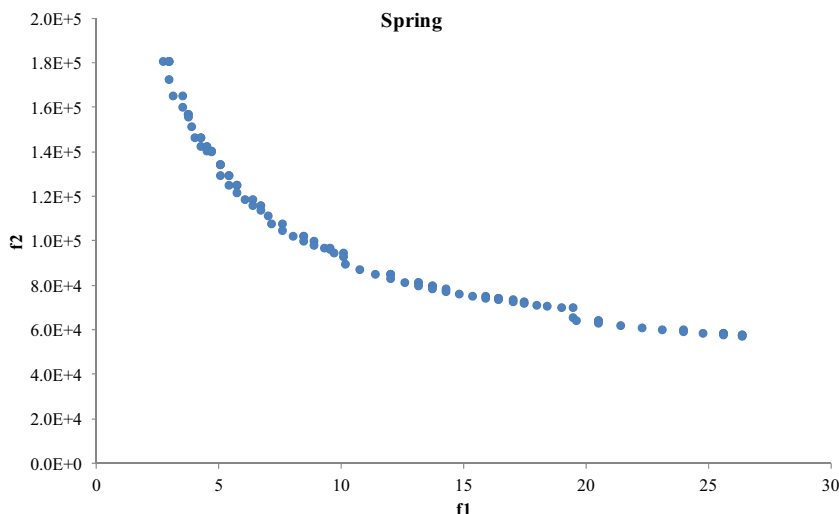


Fig. 19 Pareto front for spring problem



$$k = G \frac{x_2^4}{8x_1x_3^3}$$

$$del_p = \frac{P}{k}$$

$$K = \frac{4C - 1}{4C - 4} + \left(0.615 \frac{x_2}{x_3}\right)$$

$$\forall g \leq 0$$

$$1 \leq x_1 \leq 32, 1 \leq x_3 \leq 30, x_1 \in I$$

$x_2 \in [0.009, 0.0095, 0.0104, 0.0118, 0.0128, 0.0132, 0.014, 0.015, 0.0162, 0.0173, 0.018, 0.020, 0.023, 0.025, 0.028, 0.032, 0.035, 0.041, 0.047, 0.054, 0.063, 0.072, 0.080, 0.092, 0.105, 0.120, 0.135, 0.148, 0.162, 0.177, 0.192, 0.207, 0.225, 0.244, 0.263, 0.283, 0.307, 0.331, 0.362, 0.394, 0.4375, 0.5]$.

This value for GD, S or Δ is not available in the literature for this problem, so the results are compared with the extreme Pareto solutions. The extreme Pareto solutions are compared with NSGA-II and MOWCA. The results are presented in the Table 11. It can be observed from the results that the extreme points obtained by MO-AAA are nearly same compared to NSGA-II and MOWCA [71]. The Pareto front is shown in Fig. 19, which justifies the results. The Pareto front of this problem is discontinuous and of overlapping nature. This occurs because each discrete value for d possesses the certain portion of the Pareto front, which can be observed from the Pareto front for different discrete values of d .

6 Conclusions and future work

In this work, we suggest a new multi-objective version for artificial algae algorithm (AAA) algorithm, called MO-AAA. We test the proposed algorithm on various numerical

benchmark problems with different characteristics of the Pareto front. Also, we apply MO-AAA on different multi-objective benchmark problems (20 challenging benchmark problems from CEC 2009 for unconstrained and constrained multi-objective optimization problems) and engineering design problems and calculate its performances based on generational distance, spacing and spreading.

These benchmark problems offer challenges to multi-objective optimization algorithms. We have compared our proposed method with about nearly 16 (recently developed) other algorithms for unconstrained problems and 9 (newly developed) algorithms for constrained problems.

Moreover, we compare the proposed algorithm with a different multi-objective version of GA, PSO, DE, Bees, and WCA. Our results show that the proposed method is better or as par with the existing algorithms. This work motivates us to do and investigates various directions of research as the followings:

- Apply MO-AAA on solving an economic load dispatch problem with incommensurable objectives, multi-objective flow shop scheduling problems, and analysis of Pareto improvement models in electricity markets.
- Modify MO-AAA so that we can solve multiple-objective optimization problems and large-scale multi-objective optimization problems.
- Study some theory analysis for the MO-AAA algorithm, such as, convergence analysis, and complexity.

Acknowledgments We are thankful to the editor-in-chief and referees for their insightful and constructive comments and suggestions that significantly improved the clarity of the paper. The research of the 1st author is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The postdoctoral fellowship of the 2nd author is supported by NSERC.

References

- Agrawal S, Dashora Y, Tiwari MK, Son YJ (2008) Interactive particle swarm: a Pareto-adaptive metaheuristic to multiobjective optimization. *IEEE Trans Syst Man Cybern Syst Hum* 38(2):258–277
- Agrell PJ, Lence BJ, Stam A (1998) An interactive multicriteria decision model for multipurpose reservoir management: the shellmouth reservoir. *J Multi-Criteria Decis Anal* 7:61–86
- Ahrari A, Atai AA (2010) Grenade explosion method—a novel tool for optimization of multimodal functions. *Appl Soft Comput* 10(4):1132–1140
- Akbari R, Hedayatzadeh R, Ziarati K, Hassanizadeh B (2012) A multi-objective artificial bee colony algorithm. *Swarm Evol Comput* 2:39–52
- Andersson J (2000) A survey of multiobjective optimization in engineering design. Department of Mechanical Engineering, Linköping University, Linköping Sweden, Technical Report No: LiTH-IKP
- Angus D, Woodward C (2009) Multiple objective ant colony optimisation. *Swarm Intell* 3(1):69–85
- Aydin I, Karakose M, Akin E (2011) A multi-objective artificial immune algorithm for parameter optimization in support vector machine. *Appl Soft Comput* 11(1):120–129
- Bandaru S, Ng AH, Deb K (2017) Data mining methods for knowledge discovery in multi-objective optimization: part A-survey. *Expert Syst Appl* 70:139–159
- Bandyopadhyay S, Saha S, Maulik U, Deb K (2008) A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Trans Evol Comput* 12(3):269–283
- Bérubé JF, Gendreau M, Potvin JY (2009) An exact ϵ -constraint method for bi-objective combinatorial optimization problems: application to the traveling salesman problem with Profits. *Eur J Oper Res* 194(1):39–50
- Coello CAC, Lechuga MS (2002) MOPSO: A Proposal for multiple objective particle swarm optimization. In: Proceedings of the congress on evolutionary computation (CEC'2002), Honolulu, HI, vol 1, pp 1051–1056
- Coello CAC, Van Veldhuizen DA, Lamont GB (2002) Evolutionary algorithms for solving multi-objective problems, vol 242. Kluwer Academic, New York
- Coello CC (2000) Handling preferences in evolutionary multi-objective optimization: a survey. In: Proceedings of the 2000 Congress on Evolutionary Computation, vol 1. IEEE, pp 30–37
- Coello CC, Pulido GT, Montes EM (2005) Current and future research trends in evolutionary multiobjective optimization. In: Information processing with evolutionary algorithms. Springer, London, pp 213–231
- Corne D, Jerram NR, Knowles J, Oates MJ (2001) PESA-II: region-based selection in evolutionary multiobjective optimization. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation. Morgan Kaufmann Publishers Inc., pp 283–290
- Daneshyari M, Yen GG (2008) Cultural MOPSO: a cultural framework to adapt parameters of multiobjective particle swarm optimization. In: 2008 Congress on Evolutionary Computation (CEC'2008). IEEE Service Center, Hong Kong, pp 1325–1332
- Deb K (2001) Multi-objective optimization using evolutionary algorithms, vol 16. Wiley, New York
- Deb K, Padhye N (2014) Enhancing performance of particle swarm optimization through an algorithmic link with genetic algorithms. *Comput Optim Appl* 57(3):761–794
- Deb K, Agrawal S, Pratab A, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6:182–197
- Dorigo M (1992) Optimization, learning and natural algorithms. Ph. D Thesis, Politecnico di Milano, Italy
- Ehrgott M (2005) Multicriteria optimization, 2nd edn. Springer, Berlin
- Ehrgott M, Ryan DM (2002) Constructing robust crew schedules with bicriteria optimization. *J Multi-Criteria Decis Anal* 11:139–150
- Ehrgott M, Klamroth K, Schwehm S (2004) An MCDM approach to portfolio optimization. *Eur J Oper Res* 155:752–77
- Ehrgott M, Gandibleux X (2002) Multiobjective combinatorial optimization—theory, methodology, and applications. In: Multiple criteria optimization: State of the art annotated bibliographic surveys. Springer, US, pp 369–444
- Ehrgott M, Gandibleux X (2004) Approximative solution methods for multiobjective combinatorial optimization. *Top* 12(1):1–63
- Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput Struct* 110:151–166
- Eusuff MM, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. *J Water Resour Plan Manag* 129(3):210–225
- Farmer JD, Packard NH, Perelson AS (1986) The immune system, adaptation, and machine learning. *Physica D* 22(1):187–204
- Figueira J, Greco S, Ehrgott M (eds) (2005) Multiple criteria decision analysis: state of the art surveys. Kluwer, New York
- Fonseca CM, Fleming PJ (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization. In: Proceedings of the fifth international conference on genetic algorithms, San Mateo, USA, pp 416–423
- Gal T, Hanne T (1997) On the development and future aspects of vector optimization and MCDM. A tutorial. In: Climaco J (ed) Multicriteria analysis. Proceedings of the XIth International Conference on MCDM. Springer, Berlin, pp 130–145
- Goicoechea A, Hansen DR, Duckstein L (1982) Multiobjective decision analysis with engineering and business applications. Wiley, New York
- Gong M, Jiao L, Du H, Bo L (2008) Multiobjective immune algorithm with nondominated neighbor-based selection. *Evol Comput* 16(2):225–255
- Holland JH (1975) Adaption in natural and artificial systems. The University of Michigan Press, Ann Arbor
- Horn J, Nafpliotis N, Goldberg DE (1994) A niched Pareto genetic algorithm for multiobjective optimization. In: Proceedings of the IEEE conference on evolutionary computation, IEEE world congress on computational intelligence, Piscataway, USA, pp 82–87
- Jamuna K, Swarup KS (2012) Multi-objective biogeography based optimization for optimal PMU placement. *Appl Soft Comput* 12(5):1503–1510
- Jaszkiewicz A, Ishibuchi H, Zhang Q (2012) Multiobjective-memetic algorithms. In: Handbook of Memetic Algorithms. Springer, Berlin, pp 201–217
- Küfer KH, Scherrer A, Monz M, Trinkaus F, Alonso H, Bortfeld T, Thieke C (2003) Intensity-modulated radiotherapy - a large scale multi-criteria programming problem. *OR Spectrum* 25:223–249
- Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39(3):459–471
- Kaveh A (2014) Charged system search algorithm. In: Advances in Metaheuristic Algorithms for Optimal Design of Structures. Springer International Publishing, pp 41–85

41. Kennedy V, Eberhart R (1995) Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp 1942–1948
42. Knowles J, Corne D (1999) The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99, vol 1. IEEE, pp 98–105
43. Krishnanand KR, Panigrahi BK, Rout PK, Mohapatra A (2011) Application of multi-objective teaching-learning-based algorithm to an economic load dispatch problem with incommensurable objectives. In: Swarm, Evolutionary, and Memetic Computing. Springer, Berlin, pp 697–705
44. Laumanns M, Thiele L, Zitzler E (2006) An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *Eur J Oper Res* 169(3):932–942
45. Lei D (2009) Multi-objective production scheduling: a survey. *Int J Adv Manuf Technol* 43(9-10):926–938
46. Li H, Zhang Q (2009) Multiobjective optimization problems with complicated Pareto sets, MOEA/d and NSGA-II. *IEEE Trans Evol Comput* 13(2):284–302
47. Li X, Zhang J, Yin M (2014) Animal migration optimization: an optimization algorithm inspired by animal migration behavior. *Neural Comput Applic* 24(7-8):1867–1877
48. Luna F, Durillo JJ, Nebro AJ, Alba E (2010) Evolutionary algorithms for solving the automatic cell planning problem: a survey. *Eng Optim* 42(7):671–690
49. Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. *Struct Multidiscip Optim* 26(6):369–395
50. Mavrotas G (2009) Effective implementation of the ϵ -constraint method in multi-objective mathematical programming problems. *Appl Math Comput* 213(2):455–465
51. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. *Ecological informatics* 1(4):355–366
52. Miettinen K (2012) Nonlinear multiobjective optimization. Springer, Berlin
53. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
54. Mondal S, Bhattacharya A, nee Dey SH (2013) Multi-objective economic emission load dispatch solution using gravitational search algorithm and considering wind power penetration. *Int J Electr Power Energy Syst* 44(1):282–292
55. Moslehi G, Mahnam M (2011) A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search. *Int J Prod Econ* 129(1):14–22
56. Mostaghim S, Teich J (2004) Covering pareto-optimal fronts by subswarms in multi-objective particle swarm optimization. In: Congress on Evolutionary Computation, 2004. CEC2004, vol 2. IEEE, pp 1404–1411
57. Nikoofard AH, Hajimirsadeghi H, Rahimi-Kian A, Lucas C (2012) Multiobjective invasive weed optimization: Application to analysis of Pareto improvement models in electricity markets. *Appl Soft Comput* 12(1):100–112
58. Omkar SN, Senthilnath J, Khandelwal R, Naik GN, Gopalakrishnan S (2011) Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures. *Appl Soft Comput* 11(1):489–499
59. Padhye N, Bhardawaj P, Deb K (2013) Improving differential evolution through a unified approach. *J Glob Optim* 55(4):771
60. Padhye N, Mittal P, Deb K (2015) Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. *Comput Optim Appl* 62(3):851–890
61. Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst* 22(3):52–67
62. Patel V, Savsani V (2014) A multi-objective improved teaching-learning based optimization algorithm (MO-ITLBO). *Information Sciences*
63. Patel V, Savsani V (2015) Heat Transfer Search (HTS): a novel optimization algorithm. *Inf Sci* 324:217–246
64. Patel V, Savsani V (2016) A multi-objective improved teaching-learning based optimization algorithm (MO-ITLBO). *Inf Sci* 357:182–200
65. Pradhan PM, Panda G (2012) Solving multiobjective problems using cat swarm optimization. *Expert Syst Appl* 39(3):2956–2964
66. Rao RV, Savsani V, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. *Comput Aided Des* 43(3):303–315
67. Rao RV, Savsani V, Vakharia DP (2012) Teaching-learning-based optimization: an optimization method for continuous nonlinear large scale problems. *Inf Sci* 183(1):1–15
68. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: A gravitational search algorithm. *Inf Sci* 179(13):2232–2248
69. Reyes-Sierra M, Coello CC (2006) Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int J Comput Intell Res* 2(3):287–308
70. Roy PK, Ghoshal SP, Thakur SS (2010) Biogeography based optimization for multi-constraint optimal power flow with emission and non-smooth cost function. *Expert Syst Appl* 37(12):8221–8228
71. Sadollah A, Eskandar H, Kim JH (2015) Water cycle algorithm for solving constrained multi-objective optimization problems. *Appl Soft Comput* 27:279–298
72. Savsani P, Savsani V (2016) Passing Vehicle Search (PVS): a novel metaheuristic algorithm. *Appl Math Model* 40:3951–3978
73. Savsani P, Jhala RL, Savsani V (2014) Effect of hybridizing Biogeography-Based Optimization (BBO) technique with Artificial Immune Algorithm (AIA) and Ant Colony Optimization (ACO). *Appl Soft Comput* 21:542–553
74. Schaffer JD (1985) Some experiments in machine learning using vector evaluated genetic algorithms. Vanderbilt University, Nashville
75. Schniederjans MJ, Hollcroft E (2005) A multi-criteria modeling approach to jury selection. *Socio Econ Plan Sci* 39:81–102
76. Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. *Appl Soft Comput* 36:315–333
77. Silverman J, Steuer RE, Whisman AW (1988) A multi-period, multiple criteria optimization system for manpower planning. *Eur J Oper Res* 34:160–170
78. Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12(6):702–713
79. Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol Comput* 2(3):221–248
80. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J Glob Optim* 11(4):341–359
81. Tan KC, Goh CK, Mamun AA, Ei EZ (2008) An evolutionary artificial immune system for multi-objective optimization. *Eur J Oper Res* 187(2):371–392
82. Tapia MGC, Coello CAC (2007) Applications of multi-objective evolutionary algorithms in economics and finance: a survey. In: IEEE Congress on Evolutionary Computation, 2007. CEC 2007. IEEE, pp 532–539
83. Uymaz SA, Tezel G, Yel E (2015) Artificial algae algorithm (AAA) for nonlinear global optimization. *Appl Soft Comput* 31:153–171
84. Wang Y, Yang Y (2009) Particle swarm optimization with preference order ranking for multi-objective optimization. *Inf Sci* 179(12):1944–1959

85. Wang Y, Yang Y (2009) Particle swarm optimization with preference order ranking for multi-objective optimization. *Inf Sci* 179(12):1944–1959
86. Yagmahan B, Yenisey MM (2008) Ant colony optimization for multi-objective flow shop scheduling problem. *Comput Ind Eng* 54(3):411–420
87. Yang XS (2009) Firefly algorithms for multimodal optimization. In: *Stochastic algorithms: foundations and applications*. Springer, Berlin, pp 169–178
88. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*. Springer, Berlin, pp 65–74
89. Yang XS (2011) Bat algorithm for multi-objective optimisation. *Int J Bio-Inspired Comput* 3(5):267–274
90. Yang XS (2013) Multiobjective firefly algorithm for continuous optimization. *Eng Comput* 29(2):175–184
91. Yang XS, Deb S (2010) Engineering optimisation by cuckoo search. *Int J Math Model Numer Optim* 1(4):330–343
92. Yang XS, Deb S (2013) Multiobjective cuckoo search for design optimization. *Comput Oper Res* 40(6):1616–1624
93. Yazdani M, Jolai F (2015) Lion Optimization Algorithm (LOA): a nature-inspired metaheuristic algorithm. *Journal of Computational Design and Engineering*
94. Zhang H, Zhu Y, Zou W, Yan X (2012) A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production. *Appl Math Model* 36(6):2578–2591
95. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
96. Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S (2008) Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report
97. Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm Evol Comput* 1(1):32–49
98. Zitzler E (1999) *Evolutionary algorithms for multiobjective optimization: methods and applications* zurich ETH. Swiss Federal Institute of Technology, Switzerland
99. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
100. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: *International Conference on Parallel Problem Solving from Nature*. Springer, Berlin, pp 832–842
101. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization. In: *Proceedings of the EUROGEN 2001 – evolutionary methods for design optimisation and control with applications to industrial problems*, Barcelona, Spain



Mohamed A. Tawhid got his PhD in Applied Mathematics from the University of Maryland Baltimore County, Maryland, USA. From 2000 to 2002, he was a Postdoctoral Fellow at the Faculty of Management, McGill University, Montreal, Quebec, Canada. Currently, he is a full professor at Thompson Rivers University. His recent research interests are best described as metaheuristic/evolutionary computing/artificial intelligence algorithms and their applications in engineering and data science. He has published in journals such as *Memic Computing*, *Soft Computing*, *Applied intelligence*, *Engineering Applications of Artificial Intelligence*, *Neural Computing and Applications*, *Computational Optimization and Applications*, *J. Optimization and Engineering*, *Journal of Optimization Theory and Applications*, *European Journal of Operational Research*, *Journal of Industrial and Management Optimization*, *Journal Applied Mathematics and Computation*, etc. Mohamed Tawhid published more than 70 referred papers and edited 5 special issues in *J. Optimization and Engineering* (Springer), *J. Abstract and Applied Analysis*, *J. Advanced Modeling and Optimization*, and *International Journal of Distributed Sensor Networks*. Also, he has served on editorial board several journals. Also, he has worked on several industrial projects in BC, Canada.



Dr. Vimal J. Savsani has obtained his PhD. from SV National Institute of Technology, Surat, India in 2011 with the specialization in the applications of advanced optimization techniques in machine design. He has proposed several metaheuristic optimization techniques such as Teaching Learning Based Optimization (TLBO), Heat Transfer Search (HTS) and Passing Vehicle Search (PVS). He has published more than 60 research papers in reputed International conferences and reputed journals of Elsevier, Springer, ASME, IEEE etc. He has obtained “most cited paper” award in the year 2013 from Elsevier, USA for one of his paper. He has to his credit one book called “Mechanical Design Optimization using Advanced Optimization Technique” Published by Springer London Heidelberg. His Specialization includes advanced meta-heuristics techniques, Finite Element Analysis, Vibration Engineering and Machine Design. His area of research includes Investigation of the advanced optimization techniques for the engineering problems of Automobile suspension, Wind farm layout optimization, structure topology optimization and evaluation of offshore wind energy for India. His Specialization includes advanced optimization techniques, Finite Element Analysis, Vibration Engineering and Machine Design. His Research Interest includes multi-objective optimization techniques for engineering design problems, Suspension Design for automobiles and structure topology optimization.