Advances in Computational Design, *Vol. 1, No. 1 (2016) 87-103*
DOI: http://dx.doi.org/10.12989/acd.2016.1.1.087

An efficient multi-objective cuckoo search algorithm for design optimization

A. Kaveh* and T. Bakhshpoori

Centre of Excellence for Fundamental Studies in Structural Engineering, Iran University of Science and Technology, Narmak, Tehran-16, Iran

(Received August 12, 2015, Revised November 5, 2015, Accepted November 9, 2015)

Abstract. This paper adopts and investigates the non-dominated sorting approach for extending the single-objective Cuckoo Search (CS) into a multi-objective framework. The proposed approach uses an archive composed of primary and secondary population to select and keep the non-dominated solutions at each generation instead of pairwise analogy used in the original Multi-objective Cuckoo Search (MOCS). Our simulations show that such a low computational complexity approach can enrich CS to incorporate multi-objective needs instead of considering multiple eggs for cuckoos used in the original MOCS. The proposed MOCS is tested on a set of multi-objective optimization problems and two well-studied engineering design optimization problems. Compared to MOCS and some other available multi-objective algorithms such as NSGA-II, our approach is found to be competitive while benefiting simplicity. Moreover, the proposed approach is simpler and is capable of finding a wide spread of solutions with good coverage and convergence to true Pareto optimal fronts.

Keywords: multi-objective optimization; engineering design; cuckoo search; metaheuristic

1. Introduction

Many real-world decision problems involve simultaneous optimization of multiple objectives. Such problems are known as Multi-Objective Problems (MOPs). In a MOP, different objectives stay in conflict with each other and results in a set of optimal solutions (largely known as Pareto-optimal solutions), instead of a single optimal solution. MOPs can be solved using different methods (Branke *et al*. 2001, Deb *et al*. 2002). Pareto based methods are the most common and interesting ones because of offering a suitable way namely the dominance concept to deal with the conflicting relationship between objectives (Zeltni and Meshoul 2014). Solving a MOP using Pareto dominance concept consists of identifying the set of non-dominated solutions that represent the possible tradeoffs between objectives. Considering this, population based metaheuristics offer attractive tools to solve MOPs as they deal simultaneously with a set of possible solutions (the so-called population) that allows us to find several members of the Pareto-optimal set in one single run (Zeltni and Meshoul 2014). Multi-objective Evolutionary Algorithms (MOEAs) have been

*Corresponding author, Professor, E-mail: alikaveh@iust.ac.ir

http://www.techno-press.com/journals/acd&subpage=7 ISSN: 2383-8477 (Print), 2466-0523 (Online)

proposed and successfully applied to solve MOPs in science and engineering, such as NSGA-II (Deb *et al*. 2002), and MOPSO (Coello *et al*. 2004), among some others (Yahya and Saka 2014). In the last two decades, population based metaheuristic optimization methods have been proposed and successfully contributed to the discipline of single objective optimization (Kaveh 2014). The Cuckoo Search (CS) algorithm is one of the efficient algorithms among others (Yang and Deb 2010). In order to cope with multi-objective optimization problems, important contributions should be made to extend such methods to the multi-objective frameworks with efficiency in convergence and diversity.

CS is a relatively recent population based metaheuristic inspired by cuckoo parasitism developed by Yang and Deb (2010). This algorithm is based on two fundamental concepts namely the reproductive behavior of cuckoos and Lévy flight behavior of some birds and fruit flies. CS has been found to be successful in a wide variety of single objective optimization tasks (Kaveh and Bakhshpoori 2013, Shayanfar *et al*. 2013, Kaveh *et al*. 2014) but until recently it had not been seriously extended to deal with MOPs. The original Multi-objective Cuckoo Search (MOCS) proposed by Yang and Deb (2013) was one of the first such studies conducted. The main problems of the original MOCS approach can be as follows: 1) High computational complexity because of considering multiple eggs for each cuckoo equal to the number of objective functions of the MOP at hand. This strategy is benefited to incorporate multi-objective needs. 2) The lack of leader selection strategy (Zeltni1 and Meshoul 2014). According to the dynamics of the basic CS, best solutions are used to globally update the current generation. In the original MOCS (Yang and Deb 2013) and some limited number of applications of MOCS (Kanagaraj *et al*. 2013, Hanoun *et al*. 2014, Srivastav and Agrawal 2015) it is not mentioned how this task has been performed. The second issue is addressed by Zeltni and Meshoul (2014). They investigated five leader selection strategies resulting in significant affect on the convergence and diversity of the algorithm.

These issues are addressed in this paper in the form of an efficient and simplified variant of the MOCS. 1) Adopting an archive composed of primary and secondary populations to select and keep the non-dominated solutions at each generation instead of pairwise analogy used in the CS and the original MOCS. Our simulations show that such a low computational complexity can enable CS to incorporate multi-objective requirements instead of considering multiple eggs for cuckoos. 2) In a multi-objective optimization process along with convergence to the Pareto-optimal set, it is also desirable the optimizer to maintain a good spread of solutions in the set of solutions. Benefitting two main and unique features of the NSGA-II (Deb *et al*. 2002), one of the best algorithms available at present, the second issue will be addressed. These features correspond to fast non-dominated sorting approach and diversity preservation. The leader selection strategy will be investigated by considering the cuckoo belonging to the first optimal front with far more neighboring ones. The proposed MOCS is tested on a set of multi-objective optimization problems and two well-studied engineering design optimization problems (bi-objective beam design and design of a disc brake). Comparison with MOCS and some other available multi-objective algorithms such as NSGA-II, the results indicate that our approach is competitive. Moreover, the proposed approach is simpler and is capable of finding a wide spread of solutions with good coverage and convergence to the true Pareto optimal fronts.

In the reminder of this paper, the single objective CS is outlined in Section 2. Thereafter in Section 3 we extend the CS to a multi-objective framework and formulate an efficient and simplified multi-objective cuckoo search. Section 4 tests the proposed MOCS on a set of multi-objective optimization problems and two well-studied engineering design problems. Finally the findings of this paper are concluded at Section 5.

2. Single objective cuckoo search

CS is one of the recent population based metaheuristic inspired by cuckoo parasitism which is developed by Yang and Deb (2010). CS is based on two fundamental concepts namely the reproductive behavior of cuckoos, and Lévy flight behavior of some birds and fruit flies. Lévy flight is used successfully by Yahya and Saka (2014) in the context of artificial bee colony algorithm for multi-objective construction site layout planning task. In order to extend the single objective CS to a multiobjective framework, the pseudo-code of the single objective CS is briefly reviewed as it follows (Kaveh and Bakhshpoori 2013):

2.1 Initialize the cuckoo search algorithm parameters

In the first step, the CS parameters are set consisting of the number of nests (n), the step size parameter (α) and discovering probability (pa).

2.2 Generate initial nests or eggs of the host birds

The initial locations of the nests are determined by a set of values randomly assigned to each decision variable as

$$nest_{i,j}^{(0)} = x_{j,\min} + rand.(x_{j,\max} - x_{j,\min}) \tag{1}$$

where $nest_{i,j}^{(0)}$ determines the initial value of the jth variable for the ith nest; $x_{j,min}$ and $x_{j,max}$ are the minimum and the maximum allowable values for the jth variable respectively; $rand$ is a random number in the interval [0, 1].

2.3 Generate new cuckoos by Lévy flights

In this step, all the nests except the best one are replaced by the new cuckoo eggs produced with Lévy flights from their positions based on their quality as

$$nest_i^{(t+1)} = nest_i^{(t)} + \alpha.S.(nest_i^{(t)} - nest_{best}^{(t)}).r \tag{2}$$

where $nest_i^t$ is the ith nest current position, α is the step size parameter; r is a random number from a standard normal distribution and $nest_{best}$ is the position of the best nest so far; and S is a random walk based on the Lévy flights. The Lévy flight essentially provides a random walk while the random step length is drawn from a Lévy distribution. In fact, Lévy flights have been observed among foraging patterns of albatrosses, fruit flies and spider monkeys. One of the most efficient and yet straightforward ways of applying Lévy flights is to use the so-called Mantegna algorithm. In Mantegnas algorithm, the step length S is calculated by

$$S = \frac{u}{|v|^{1/\beta}} \tag{3}$$

where β is a parameter between the interval [1, 2] and here it is considered as 1.5; u and v are obtained from normal distribution as

$$u \sim N(0,\ \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \tag{4}$$

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\ \sin(\pi\beta/2)}{\Gamma\left[(1+\beta)/2\right]\ \beta\ 2^{(\beta-1)/2}} \right\}^{1/\beta}, \qquad \sigma_v = 1 \tag{5}$$

2.4 Alien eggs discovery

The alien eggs discovery is performed for each component of each solution in terms of the probability matrix as

$$P_{ij} = \begin{cases} 1 & if\ \ rand < pa \\ 0 & if\ \ rand \geq pa \end{cases} \tag{6}$$

where *rand* is a random number in the interval [0, 1], and *pa* is the discovering probability. Existing eggs are replaced considering their quality by the newly generated ones from their current positions through random walks with step size of

$$\begin{aligned} & S = rand\,.\left(nests\left(randperm1(n),:\right) - nests\left(randperm2(n),:\right)\right) \\ & nest^{t+1} = nest^{t} + S.*P \end{aligned} \tag{7}$$

where *randperm1* and *randperm2* are random permutation functions used for different rows permutation applied on nests matrix and *P* is the probability matrix.

2.5 Termination criterion

The generating new cuckoos and discovering alien eggs steps are alternately performed until a termination criterion is satisfied.

3. Simplified multi-objective cuckoo search algorithm

In the original multi-objective cuckoo search algorithm (MOCS) proposed by Yang and Deb (2013), the basic rules are as follows: 1) Each cuckoo lays *k* eggs at a time, and dumps them in a randomly chosen nest. Egg *k* corresponds to the solution of the *k*th objective. 2) The best nests with eggs of high quality (solutions) will carry over to the next generations. 3) Each nest will be abandoned with a probability *pa* and a new nest with *k* eggs will be built, according to the similarities / differences of the eggs. Some random mixing can be used to provide diversity.

The main problems with the MOCS are as follows: 1) According to the basic rules of the single objective CS it can be observed that the first and last rules are modified to fulfill the multi-objective optimization problem with *k* different solutions. Considering *k* eggs for each cuckoo (egg *k* corresponds to the solution of the *k*th objective) resulting in computationally complex algorithm. 2) The lack of leader selection strategy. According to the dynamics of the basic CS as proposed in (Yang and Deb 2010), best solution is used to update globally the current generation. This issue is further addressed by Zeltni and Meshoul (2014). Considering this matter they investigated five leader selection strategies which significantly affected the convergence and diversity of the algorithm with no superiority for any of these strategies.

In this study, these two issues are addressed in the form of an efficient simplified variant of the MOCS. 1) Adopting an archive composed of primary and secondary population to select and keep the non-dominated solutions at each generation instead of pairwise analogy used in the CS and MOCS. Simulation results show that such a low computational complexity enables the CS to incorporate multi-objective needs instead of considering multiple eggs for cuckoos. 2) In a multi-objective optimization process along with convergence to the Pareto-optimal set, it is also desired that the optimizer maintains a good spread of solutions in the obtained set of solutions. Benefitting two main and unique features of the NSGA-II, one of the best algorithms available at present: fast non-dominated sorting approach and diversity preservation, this issue can be addressed. Considering the cuckoo belonging to the primary optimal front with far more neighboring cuckoos can be considered as an efficient leader selection strategy. In the following subsections these features are introduced and adopted for the MOCS. Then the basic steps of the proposed multi-objective cuckoo search is summarized in Subsection 3.4.

3.1 Pareto front

In a minimization multi-objective problem, solution vector *u* is said to dominate another vector *v* if and only if no component of *u* is larger than *v*, and at least one component is smaller. Therefore an individual is called non-dominated solution if no other individual can be found that dominates it. The Pareto Front (*PF*) of a multi-objective problem can be defined as a set of non-dominated solutions which is called as the true Pareto optimal front. A population based metaheuristic multi-objective optimizer tries to converge its population toward the true Pareto optimal front. Let us denote the estimated true Pareto front by the optimizer in its *t*th iteration by PF_1^t which represents the solutions that cannot be dominated by any other individuals. In this way, PF_n^t shows the *n*th front with solutions dominated with *n*-1 other ones. Therefore in the *t*th iteration of the optimizer *n* can be an integer number between 1 and the number of algorithm population minus 1. In the early iterations of the optimizer in which the randomness plays the main rule, *n* will be large and will be decreased to unity along with algorithm procession. Generating these fronts has been addressed by Deb *et al*. (2002) using a fast non-dominated sorting approach. It should be noted that along with convergence to the true Pareto optimal front, it is also desirable that an optimizer maintains a good spread of solutions in the obtained set of solutions. Considering this feature for the first Pareto front (PF_1^t), the leader selection strategy can be addressed as it follows in the next subsection.

3.2 Leader selection strategy

For achieving a well-diversified Pareto front a sharing function approach for NSGA, and the crowded comparison operator using density estimation approach is used in NSGAII, by Deb *et al*. (2002). Density estimation approach eliminates the two difficulties of sharing function approach: high overall complexity and depending largely on the chosen sharing parameter. Implementing the density estimation approach in the proposed MOCS framework results in crowding the cuckoos in the vicinity of boundary points. Fig. 1 shows the observed Pareto front of the proposed MOCS using crowded comparison operator in a single run for the SCH function. It should be noted that the population size parameter (*n*) is considered equal to 200. This can be due to considering large density estimation for boundary points to reach a well stretched Pareto front. Based on the study by Yang and Deb (2013), Lévy flights ensure a good diversity of the solutions. Our simulations using the presented MOCS shows that the sole use of Lévy flights cannot ensure finding the well located boundary points and uniformly distributed solutions on the Pareto optimal front. This can be because of the use of a combined archive analogy instead of the pair-wise analogy employed in the original MOCS. Following the above discussion, a simple and efficient approach is proposed in the following for selection of leader:

After generating Pareto fronts, the first Pareto front (PF_1^t) will be used for selecting the leader cuckoo ($nest_{best}$). The cuckoo crowded with less number of neighboring cuckoos (Crowding Number) will be selected as the $nest_{best}$. The pseudo code of determining Crowding Number (CN) for the individuals of the first Pareto front and at the sequence selecting the $nest_{best}$ is depicted in Fig. 2.

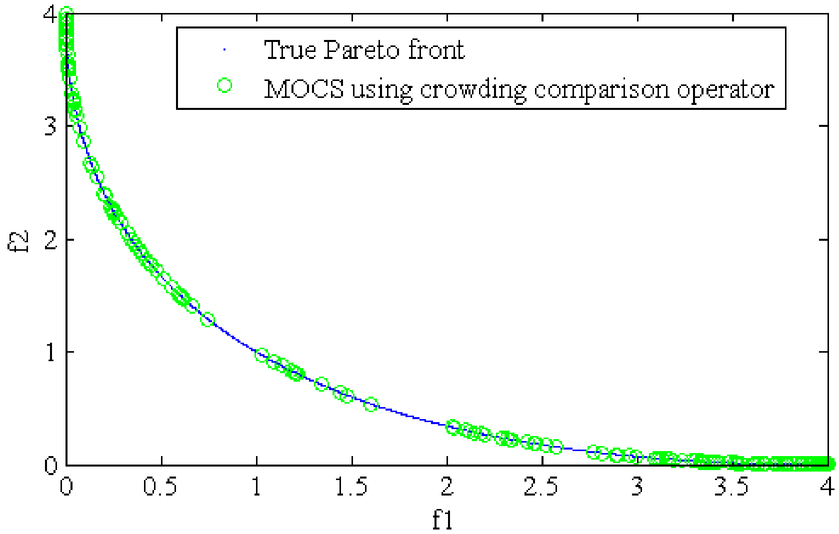

Fig. 1 Pareto front obtained by the MOCS using crowding comparison operator on the test function SCH

Generate PF_1^t and its corresponding nest and fitness matrixes: $Fitness_1^t$ and $Nest_1^t$, respectively

```
CN=zeros [1, size (PF1^t, 2)];                CN: Crowding Number
For i=1: size (Fitness1^t, 2);
    [a, b]=sort [Fitness1^t (:, i)];
    Step=[a (size (a, 1))-a (1)] / [size(PF1^t, 2)];
    for j=1:size (a, 1)
        lb=a(j)-step; ub=a(j)+step;
        for k=1:size(a, 1)
            if a(k)>=lb && a(k)<=ub
                CN (b(j))= CN (b(j))+1;
            end (if)
        end (for k)
    end (for j)
end (for i)
nest_best= Nest1^t(min(CN),:);
```

Fig. 2 Pseudo code of the proposed leader selection strategy

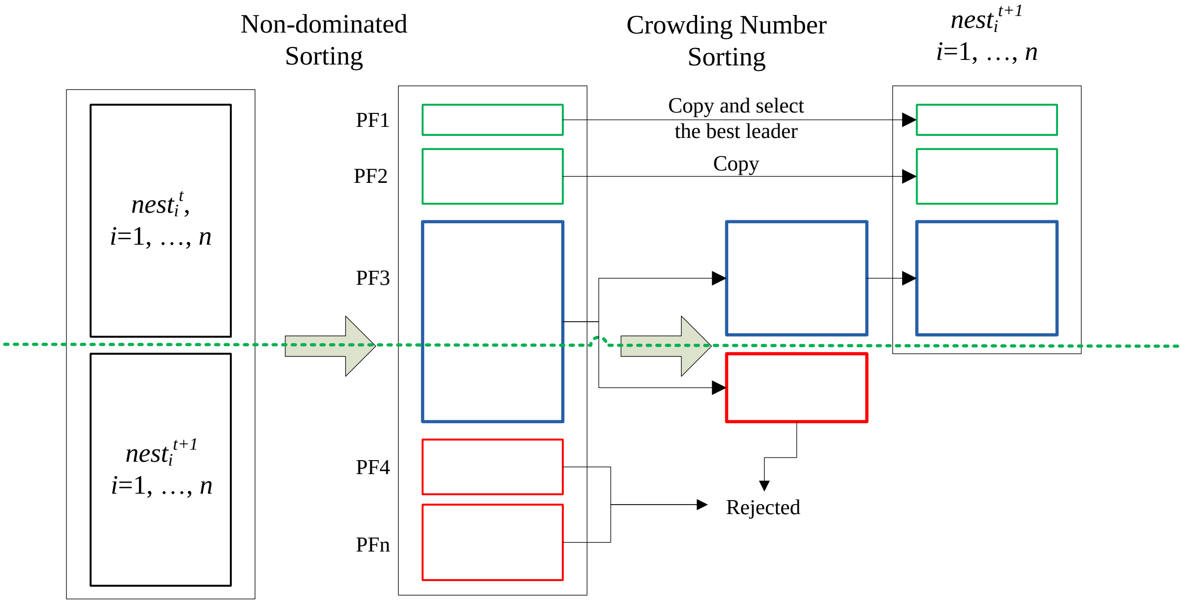

Fig. 3 Schematic representation of the transition from the current iteration to the next iteration

3.3 Transition from the current iteration to the next iteration

Based on the single objective CS, the current nests ($nest_i^t$, i=1,…, n) should be compared and replaced with the newly generated ones ($nest_i^{t+1}$). Our simulations show that such a pairwise analogy cannot fulfill the CS to incorporate the multi-objective needs. Yang and Deb (2013) addresses this issue considering k eggs for each cuckoo (egg k corresponds to the solution to the kth objective) which results in computational complexity of the algorithm. An archive composed of primary ($nest_i^t$) and secondary populations ($nest_i^{t+1}$) is adopted here to select and keep the non-dominated solutions at each generation instead of investigating a pairwise analogy. Such strategy is used successfully in the evolutionary multi-objective optimization algorithms such as NSGAII. Fig. 3 illustrates the transition between two consecutive generations. The population is updated by: (i) Generating Pareto fronts for the combined population of the old and newly generated nests using fast non-dominated sorting approach. This combined population has a size of 2n. (ii) Coping the first Pareto front, if the size of PF1 is smaller than n, and sorting based on the crowding number (introduced in the previous subsection) and determining the head leader ($nest_{best}$). (iii) Coping the subsequent non-dominated fronts until the front which includes the nth nest. For such a front, firstly the crowding number sorting is applied and then the extra last members are rejected from being transferred to the subsequent generation.

3.4 Proposed multi-objective cuckoo search algorithm

Basic steps of the proposed multi-objective cuckoo search can be summarized as follows:

Step 1: Initialization

Initialize the objective functions $f_1(x), \ldots, f_k(x)$. Initialize the algorithm parameters. These parameters consist of the number of nests (n), the step size parameter (α), discovering probability

(pa) and the maximum number of iterations as the stopping criterion. Generate initial nests or eggs of the host birds based on Eq. (1). Evaluate the first population using the objective functions. Determine the head leader $nest_{best}$ using the pseudo code depicted in Fig. 2.

Step 2: Main loop

Step 2.1: Generating new Cuckoos by Lévy flights

In this step, all the nests except the best one are used for generating new ones using Lévy flights based on the Eq. (2). Update the population using the transition process illustrated in Fig. 3.

Step 2.2: Alien eggs discovery

The alien eggs discovery is performed for each component of each solution in terms of the probability matrix (Eq. (6)). All the nests used for generating new ones from their current positions through random walks based on the probability matrix and Eq. (7). Update the population using the transition process defined in subsection 3.3.

Step 3: Termination criterion

The generating new cuckoos and discovering alien eggs steps are alternately performed until a termination criterion is satisfied. The maximum number of algorithm iterations is considered as the stopping criteria in this study.

4. Numerical results

4.1 Parametric studies

The proposed MOCS have been tested using a various range of parameters such as population size (n), the step size parameter (α) and discovering probability (pa) by varying n=5, 10, 15, 20, 50, 100, 150, 200, 300, 400, 500; pa=0.25 to 0.5; and α=0.01 to 0.5. Based on our simulations, the proposed MOCS needs at least 50 number of population to predict a wide diversified Pareto fronts and considering n equal to 200 results in the most efficient performance of the algorithm. Considering pa and α equal to 0.3 and 0.1, respectively, results in good performance for almost all the problems. The maximum number of algorithm iterations equal to 150 is considered as stopping criteria in all the test cases. The consistency of the algorithm was verified by running all the problems for 30 independent runs with different random initial solutions, and the results for the representative sample run are reported.

4.2 Multi-objective test functions

There are many different test functions for multi-objective optimization. But a subset of a few widely used functions provides a wide range of diverse properties in terms of Pareto front and Pareto optimal set. Yang and Deb (2013) have selected a subset of functions with convex, non-convex and discontinuous Pareto fronts to validate the original MOSC. Here in a similar manner the proposed MOCS is validated on this subset of problems including the SCH, ZDT1, ZDT2, and ZDT3. These problems are described in Table 1. The table also shows the number of variables, their bounds, the true Pareto-optimal solutions, and the nature of the Pareto-optimal front for each problem.

In order to make a fair comparison with the original MOCS, we use the performance metric proposed by Yang and Deb (2013) to assess the performance of the proposed MOCS: The distance or error between the estimated Pareto front PF^e to its corresponding true front PF^t is as

Table 1 Test function problems used on this study

Problem	n	Variable bounds	Objective functions	Optimal solutions	Comments
SCH	1	$\left[-10^3, 10^3\right]^n$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	$x \in [0, 2]$	convex
ZDT1	30	$[0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1-\sqrt{x_1/g(x)}\right]$ $g(x) = 1+9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2,\ldots,n$	convex
ZDT2	30	$[0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1-\left(x_1/g(x)\right)^2\right]$ $g(x) = 1+9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2,\ldots,n$	nonconvex
ZDT3	30	$[0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)\left[1-\sqrt{x_1/g(x)} - \frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x) = 1+9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$	$x_1 \in [0, 1]$ $x_i = 0,$ $i = 2,\ldots,n$	convex, disconnected

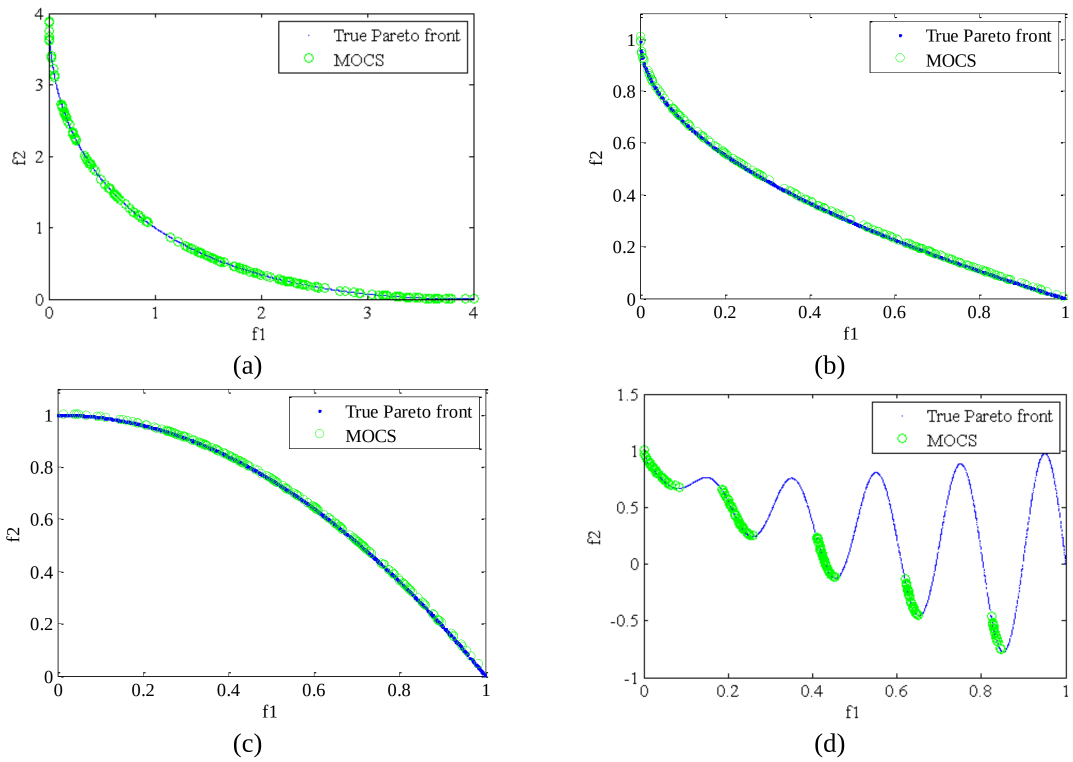

Fig. 4 Non-dominated solutions with the proposed MOCS on: (a) SCH; (b) ZDT1; (c) ZDT2; (d) ZDT3

$$E_f = \left\| PF^e - PF^t \right\| = \sum_{j=1}^{N} (PF_j^e - PF_j^t)^2 \tag{8}$$

where N is the number of points. The convergence property can also be viewed by calculating the error function following the iterations. As this measure is an absolute measure, which depends on the number of points. Sometimes, it is easier to use relative measure using a generalized distance (Deb and Yang 2013) as

$$D_g = \frac{1}{N}\sqrt{\sum_{j=1}^{N} (PF_j^e - PF_j^t)^2} \tag{9}$$

The results for all functions are depicted in Fig. 4. The obtained non-dominated solutions (estimated Pareto fronts) along with the true Pareto fronts are shown for the studied functions. Obviously the proposed MOCS is able to reach a good spread of solutions on the true Pareto fronts. For more investigation of the algorithm performance, the estimated Pareto fronts by proposed MOCS on the first two functions are shown in Figs. 5 and 6, respectively. The most important point that should be noted is that the algorithm not only tries to converge to the true Pareto optimal front along with exceeding the number of iterations but also attempts to reach a good spread of solutions.

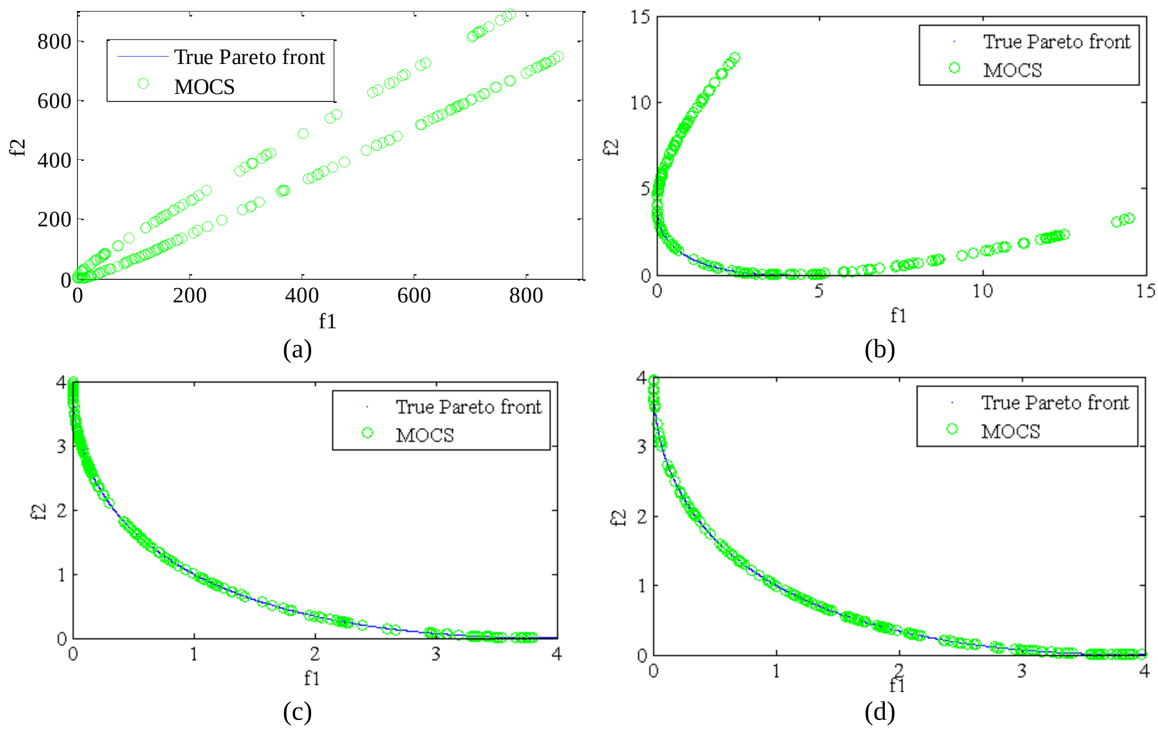

Fig. 5 Pareto fronts with the proposed MOCS on SCH with different number of algorithm iterations: (a) 2; (b) 5; (c) 10; (d) 15

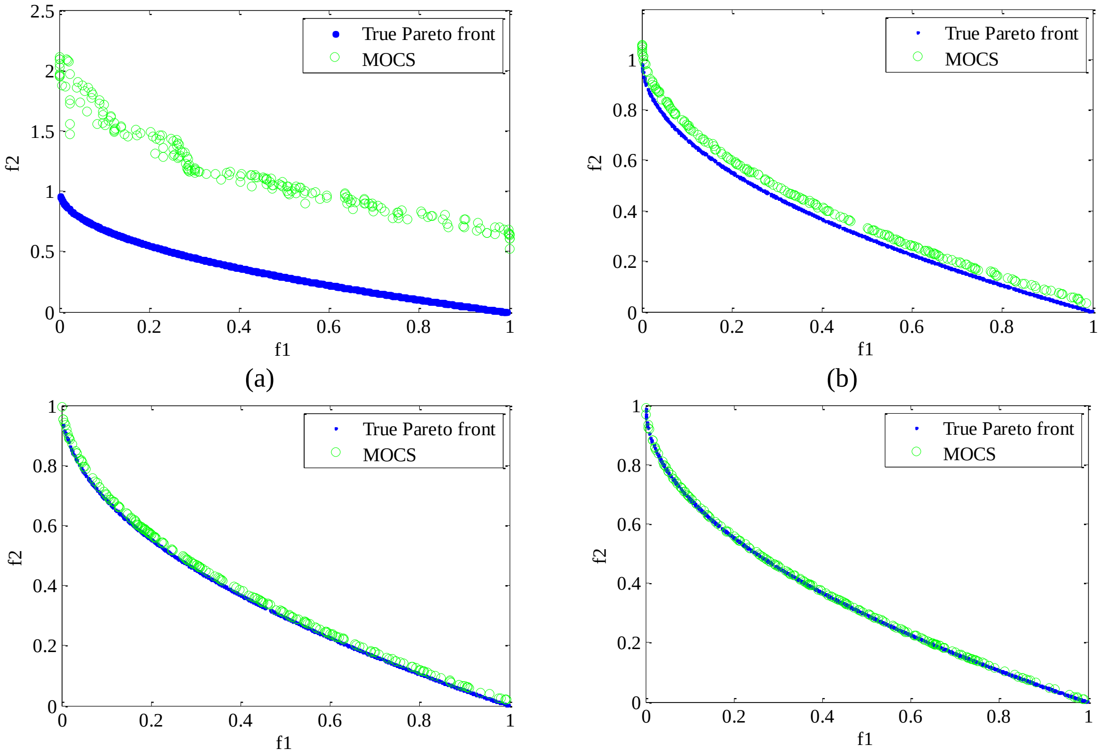

Fig. 6 Pareto fronts with the proposed MOCS on ZDT1 with different number of algorithm iterations: (a) 25; (b) 75; (c) 100; (d) 150

Table 2 Comparison of D_g for the proposed MOCS and the original MOCS and some other well-known methods Yang and Deb (2013)

Methods	SCH	ZDT1	ZDT2	ZDT3
VEGA	6.98E-02	3.79E-02	2.37E-03	3.29E-01
NSGA-II	5.73E-03	3.33E-02	7.24E-02	1.14E-01
MODE	9.32E-04	5.80E-03	5.50E-03	2.15E-02
DEMO	1.79E-04	1.08E-03	7.55E-04	1.18E-03
Bees	1.25E-02	2.40E-02	1.69E-02	1.91E-01
SPEA	5.17E-03	1.78E-03	1.34E-03	4.75E-02
MOCS	1.25E-07	1.17E-07	2.23E-05	2.88E-05
Proposed MOCS	**1.27E-04**	**3.18E-04**	**2.54E-03**	**4.23E-03**

The original MOCS is investigated carefully by Yang and Deb (2013). In order to compare the performance of the original MOCS with other established multi-objective algorithms, they have carefully selected a few algorithms with available results from the literature. In the cases for which the results were not available, they have tried to implement the algorithms using well-documented studies and then generated new results using these algorithms. In particular, they have used other methods for comparison, including vector evaluated genetic algorithm (VEGA), NSGA-II, multi-objective differential evolution (MODE), differential evolution for multi-objective optimization (DEMO), multi-objective bees algorithms (Bees), and strength Pareto evolutionary algorithm (SPEA). The reported results in terms of generalized distance D_g for the original MOCS and other well studied algorithms are used here for evaluating the performance of the proposed MOCS. Table 2 presents generalized distance D_g for the original and the proposed MOCS along with the above mentioned methods. As it is clear, the original MOCS performs far better than all the methods. Proposed MOCS has better and acceptable comparable performance in comparison with other methods. Although the Original MOCS has better accuracy than the proposed MOCS, however, the proposed MOCS has simpler structure and low computational complexity. This reduction in the complexity of the algorithm in no way detracts from the efficiency of the algorithm in comparison to other major well known methods.

4.3 Design optimization problems

From the previous subsection, benchmark unconstrained MOPs; it is observed that the proposed MOCS is able to handle mathematically complex problems efficiently. This subsection deals with solving the engineering design problems. There are many different benchmarks with detailed studies in the literature Ray and Liew (2002) and Gong *et al.* (2009). Among the well-known benchmarks are the welded beam design, and disc brake design which are considered as the constrained engineering MOPs by Yang and Deb (2013) to evaluate the original MOCS. These well-studied engineering design problems are selected here to test the efficiency and applicability of the proposed MOCS for multi-objective design optimization. Algorithm parameter setting is presented in the Subsection 4.1. Note that for calculating the D_g in these two problems the true Pareto-optimal fronts are generated by a rather high computational enumeration effort (Gong *et al*. 2009) using parallel processing techniques in MATLAB software.

Usually engineering design MOPs have multiple constraints. Thus it is essential to handle the constraints efficiently to solve the engineering design problems. A considerable amount of researches on constraint handing techniques has been carried out. However, many of the previous constraint handling methods need to tune some parameters to balance between the objective(s) and constraint(s). In this research, we employ a simpler and efficient constraint handling method proposed by Deb *et al*. (2002) outlined in the following:

In the presence of constraints, each solution can be either feasible or infeasible. Thus, there may be at most three situations when two solutions are compared: 1) both solutions are feasible; 2) one is feasible and other is not; and 3) both are infeasible. In the context of multi-objective optimization, the definition of domination between two solutions i and j should be modified. A solution i is said to constrained-dominate a solution j, if any of the following conditions hold: (i) Solution i is feasible and solution j is not. (ii) Solutions i and j are both infeasible, but solution i has a smaller overall constraint violation. (iii) Solutions i and j are feasible and solution i dominates solution j. Since in none of these cases constraints and objective function values are compared with each other, there is no need for having a penalty parameter.

4.3.1 Welded beam design

The first problem is selected from Yang and Deb (2013), which is to be designed for minimum cost and minimum end deflection (δ) subject to constraints on shear stress ($\tau(x)$), bending stress ($\sigma(x)$) and buckling load ($P(x)$). The problem has four design variables: the width w and length L

of the welded area, the depth d and thickness h of the main beam. The mathematical formulation of the problem is as it follows:

Minimize

$$\begin{cases} f_1(x) = 1.10471 w^2 L + 0.04811 dh(14.0 + L) \\ f_2(x) = \delta \end{cases} \tag{10}$$

Subject to

$$\begin{aligned} &g_1(x) = w - h \le 0, \\ &g_2(x) = \delta(x) - 0.25 \le 0, \\ &g_3(x) = \tau(x) - 13,600 \le 0, \\ &g_4(x) = \sigma(x) - 30,000 \le 0, \\ &g_5(x) = 0.10471 w^2 + 0.04811 hd(14 + L) - 5.0 \le 0, \\ &g_6(x) = 0.125 - w \le 0, \\ &g_7(x) = 6,000 - P(x) \le 0, \end{aligned} \tag{11}$$

where

$$\begin{aligned} &\sigma(x) = 504,000/hd^2, \; Q = 6000(14 + L/2), \\ &D = \frac{1}{2}\sqrt{L^2 + (w + d)^2}, \; J = \sqrt{2} wL \left[\frac{L^2}{6} + \frac{(w + d)^2}{2} \right], \\ &\delta = \frac{65.856}{30,000 hd^3}, \quad \beta = \frac{QD}{J}, \\ &\alpha = \frac{6000}{\sqrt{2} wL}, \; \tau(x) = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}, \\ &P(x) = 0.61423 \times 10^6 \frac{dh^3}{6} (1 - \frac{d\sqrt{30/48}}{28}), \\ &0.1 \le L, d \le 10 \text{ and } 0.125 \le w, h \le 2.0. \end{aligned} \tag{12}$$

The estimated Pareto front generated by 200 non-dominated solutions after 150 iterations is shown in Fig. 7. This is consistent with the results obtained by the original MOCS and other well-known methods (Yang and Deb 2013, Gong *et al*. 2009). It should be noted that the proposed MOCS results in better spread of solutions. For careful evaluation of the original MOCS, Yang and Deb (2013) solved this problem with other well-known methods (introduced in the mathematical optimization section). The results are presented as convergence comparison rates in logarithmic scale. The proposed MOCS convergence history is monitored for the best sample run and depicted along with the original MOCS and other methods in Fig. 8. Obviously the proposed MOCS performs better than other methods and results are comparable with those of the original MOCS.

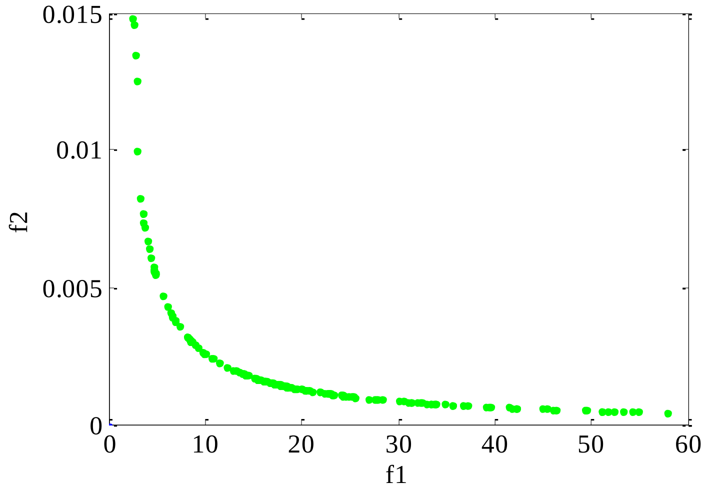

Fig. 7 The non-dominated solutions obtained by the proposed MOCS on welded beam design problem

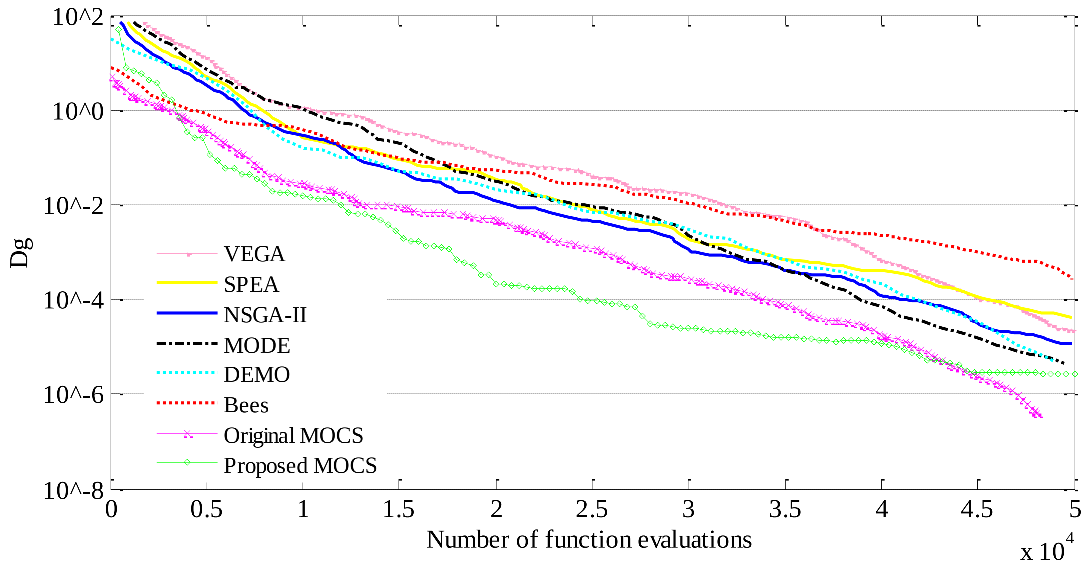

Fig. 8 Convergence comparison for the welded beam design problem, Yang and Deb (2013)

4.3.2 Disc brake design

The second design example consists of optimizing the disc brake design problems Yang and Deb (2013). The objectives of the design are to minimize the mass of the brake and to minimize the stopping time. The variables are the inner radius of the discs, outer radius of the discs, the engaging force and the number of friction surfaces and are represented as *r*, *R*, *F* and *S*, respectively. The constraints for the design include minimum distance between the radii, maximum length of the brake, pressure, temperature and torque limitations. The problem is a mixed, constrained, multi-objective problem. The mathematical description of this problem is as follows (Gong *et al*. 2009)

Minimize

$$\begin{cases} f_1(x) = 4.9\times10^{-5}\left(R^2 - r^2\right)\left(S-1\right) \\ f_2(x) = \dfrac{9.82\times10^{6}\left(R^2 - r^2\right)}{F\,S\left(R^3 - r^3\right)} \end{cases} \tag{13}$$

Subject to

$$\begin{aligned} g_1(x) &= 20 - \left(R - r\right) \le 0, \\ g_2(x) &= 2.5\left(S+1\right) - 30 \le 0, \\ g_3(x) &= \frac{F}{3.14\left(R^2 - r^2\right)} - 0.4 \le 0, \\ g_4(x) &= \frac{2.22\times10^{-3}F\left(R^3 - r^3\right)}{\left(R^2 - r^2\right)^2} - 1 \le 0, \\ g_5(x) &= 900 - \frac{0.0266\,FS\left(R^3 - r^3\right)}{\left(R^2 - r^2\right)} \le 0, \end{aligned} \tag{14}$$

where

$$55 \le r \le 80,\ \ 75 \le R \le 110,\ \ 1000 \le F \le 3000,\ \ 2 \le S \le 20 \tag{15}$$

The estimated Pareto front generated by 200 non-dominated solutions after 150 iterations is shown in Fig. 9. This is consistent with the results obtained by the original MOCS and other well-known methods (Yang and Deb 2013, Gong *et al*. 2009). It should be noted that the proposed MOCS results in better spread of solutions. For careful evaluation of the original MOCS, Yang and Deb (2013) solved this problem with other well-known methods (introduced in the mathematical optimization section). The results are presented as convergence comparison rates in

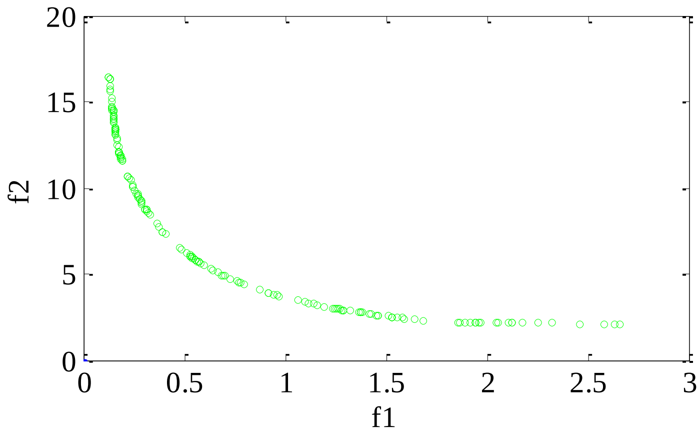

Fig. 9 The non-dominated solutions obtained by the proposed MOCS on disc brake design problem

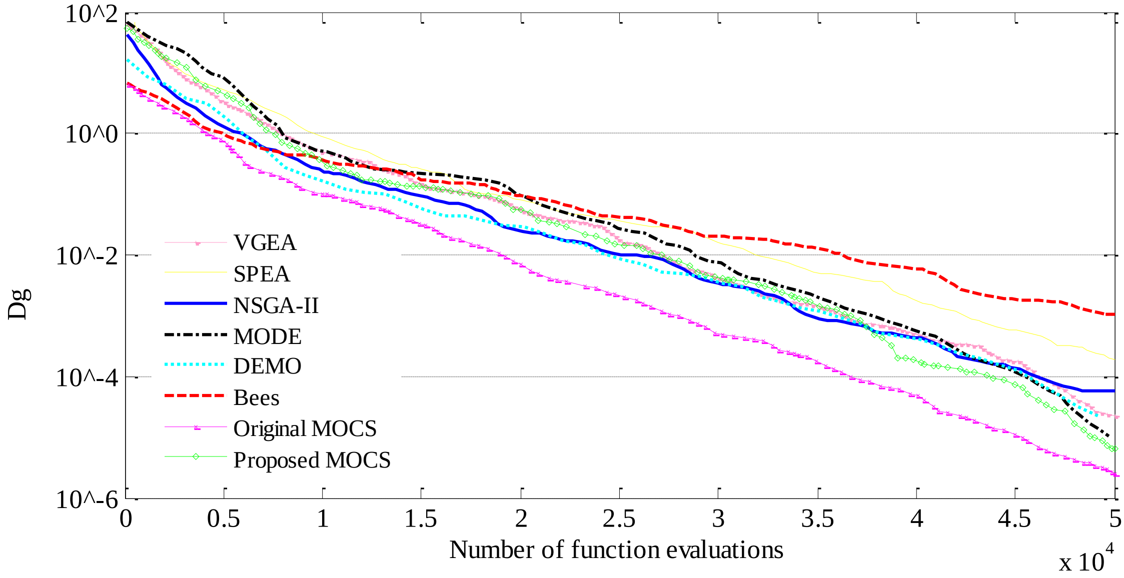

Fig. 10 Convergence comparison for the welded beam design problem, Yang and Deb (2013)

logarithmic scale. The proposed MOCS convergence history is monitored for the best sample run and depicted along with the original MOCS and other methods in Fig. 10. Obviously the proposed MOCS shows comparable performance compared to the original MOCS and other methods.

5. Conclusions

In this paper, an efficient simplified multi-objective Cuckoo Search algorithm is presented to solve MOPs. This algorithm is characterized by a) Adopting an archive composed of primary and secondary populations to select and keep the non-dominated solutions at each generation instead of pairwise analogy used in the CS and original MOCS. Such a strategy results in decreasing the computational complexity. b) Maintaining a good spread of solutions in the obtained set of solutions, along with convergence to the Pareto-optimal set is considered as an efficient strategy to overcome the lack of leader selection strategy in the original MOCS. c) Employing a new and efficient strategy to achieve a well-diversified Pareto front.

The tests are carried out for a quaternary subset of functions with convex, non-convex and discontinuous Pareto fronts and two well-studied engineering design MOPs. The proposed MOCS behaves better and in some cases comparable to some well-known multi-objective algorithms. Decreasing the computational complexity does not significantly reduce the MOCS precision, which makes it of interest for use in real time design MOPs.

References

Branke, J., Kaußler, T. and Schmeck, H. (2001), “Guidance in evolutionary multi-objective optimization”, *Adv. Eng. Softw.*, **32**(6), 499-507.

Coello, C.A.C., Pulido, G.T. and Lechuga M.S. (2014), "Handling multiple objectives with particle swarm optimization", *IEEE Trans. Evol. Comput*., **8**(3), 256-279.
Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE Trans. Evol. Comput.*, **6**(2), 182-197.
Gong, W., Cai, Z. and Zhu, L. (2009), "An efficient multiobjective differential evolution algorithm for engineering design", *Struct. Multidiscip. Optim*., **38**(2), 137-157.
Hanoun, S., Creighton, D. and Nahavandi, S. (2014), "A hybrid cuckoo search and variable neighborhood descent for single and multiobjective scheduling problems", *Int. J. Adv. Manuf. Tech*., **75**(9-12), 1501-1516.
Kaveh, A. (2014), *Advances in mataheuristic algorithms for optimal design of structures*, Springer, Switzerland.
Kaveh, A. and Bakhshpoori, T. (2013), "Optimum design of steel frames using cuckoo search algorithm with Lévy flights", *Struct. Des. Tall Spec. Build*., **22**(13), 1023-1036.
Kaveh, A., Bakhshpoori, T. and Barkhori, M. (2014), "Optimum design of multi-span composite box girder bridges using Cuckoo Search algorithm", *Steel Compos. Struct*., **17**(5), 705-719.
Kanagaraj, G., Ponnambalam, S.G. and Jawahar, N. (2013), "A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems", *Comput. Indust. Eng*., **66**(4), 1115-1124.
Ray, T. and Liew, K.M. (2002), "A swarm metaphor for multiobjective design optimization", *Eng. Optim*., **34**(2), 141-153.
Shayanfar, M.A., Ashoory, M., Bakhshpoori, T. and Farhadi, B. (2013), "Optimization of modal load pattern for pushover analysis of building structures", *Struct. Eng. Mech*., **47**(1), 119-129.
Srivastav, A. and Agrawal, S. (2015), "Multi objective cuckoo search optimization for fast moving inventory items", *Adv. Intell. Syst. Comput*., **320**, 503-510.
Srinivas, N. and Deb, K. (1994), "Multiobjective function optimization using nondominated sorting genetic algorithms", *Evol. Comput*., **2**(3), 221-248.
Yahya, M. and Saka, M.P. (2014), "Construction site layout planning using multi-objective artificial bee colony algorithm with Levy flights", *Autom Construct*., **38**, 14-29.
Yang, X.S. and Deb, S. (2013), "Multiobjective cuckoo search for design optimization", *Comput. Oper. Res*., **40**(6), 1616-1624.
Yang, X.S. and Deb, S. (2010), "Engineering optimisation by cuckoo search", *Int. J. Math. Model Numer. Optim*., **1**(4), 330-343.
Zeltni, K. and Meshoul, S. (2014), "Multi-objective cuckoo Search with leader selection strategies", *Combinatorial optimization, Lecture Notes in Computer Science*, **8596**, 421-432.

CC