

## Research Article

# A Seed-Based Plant Propagation Algorithm: The Feeding Station Model

Muhammad Sulaiman<sup>1,2</sup> and Abdellah Salhi<sup>1</sup>

<sup>1</sup>Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, UK

<sup>2</sup>Department of Mathematics, Abdul Wali Khan University, Mardan, Khyber Pakhtunkhwa, Pakistan

Correspondence should be addressed to Muhammad Sulaiman; [sulaiman513@yahoo.co.uk](mailto:sulaiman513@yahoo.co.uk)

Received 25 December 2014; Revised 10 February 2015; Accepted 10 February 2015

Academic Editor: Xinyu Li

Copyright © 2015 M. Sulaiman and A. Salhi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The seasonal production of fruit and seeds is akin to opening a feeding station, such as a restaurant. Agents coming to feed on the fruit are like customers attending the restaurant; they arrive at a certain rate and get served at a certain rate following some appropriate processes. The same applies to birds and animals visiting and feeding on ripe fruit produced by plants such as the strawberry plant. This phenomenon underpins the seed dispersion of the plants. Modelling it as a queuing process results in a seed-based search/optimisation algorithm. This variant of the Plant Propagation Algorithm is described, analysed, tested on nontrivial problems, and compared with well established algorithms. The results are included.

## 1. Introduction

Plants have evolved a variety of ways to propagate. Propagation with seeds is perhaps the most common of them all and one which takes advantage of all sorts of agents ranging from wind to water, birds, and animals. In [1] a Plant Propagation Algorithm based on the way the strawberry plant propagates using runners has been introduced. Here, we consider the case where the strawberry plant uses seeds to propagate.

Plants rely heavily on the dispersion of their seeds to colonise new territories and to improve their survival [2, 3]. There are a lot of studies and models of seed dispersion particularly for trees [2–6]. Dispersion by wind and ballistic means is probably the most studied of all approaches [7–9]. However, in the case of the strawberry plant, given the way the seeds stick to the surface of the fruit (Figure 1(a)) [10], dispersion by wind or mechanical means is very limited. Animals, however, and birds in particular are the ideal agents for dispersion [2, 3, 11, 12] in this case.

There are many biologically inspired optimization algorithms in the literature [13, 14]. The Flower Pollination Algorithm (FPA) is inspired by the pollination of flowers through different agents [8]; the swarm data clustering algorithm is inspired by pollination by bees [15]; Particle Swarm Optimization (PSO) is inspired by the foraging behavior of

groups of animals and insects [16, 17]; the Artificial Bee Colony (ABC) simulates the foraging behavior of honey bees [18, 19]; the Firefly algorithm is inspired by the flashing fireflies when trying to attract a mate [20, 21]; the Social Spider Optimization (SSO) algorithm is inspired by the cooperative behavior of social spiders [22]. The list could easily be extended.

The Plant Propagation Algorithm (PPA) also known as the strawberry algorithm was inspired by the way plants and specifically the strawberry plants propagate using runners [1, 23]. The attraction of PPA is that it can be implemented easily for all sorts of optimization problems. Moreover, it has few algorithm specific arbitrary parameters. It follows the principle that plants in good spots with plenty of nutrients will send many short runners. They send few long runners when in nutrient poor spots. With long runners PPA tries to explore the search space while short runners enable it to exploit the solution space well. In this paper, we investigate an alternative PPA which is entirely based on the propagation by seeds of the strawberry plant. Because of the periodic nature of fruit and seed production, it amounts to setting up a feeding station for the attention of potential seed-dispersing agents [24]. Hence the feeding station model used here and the resulting Seed-Based Plant Propagation Algorithm or SbPPA.



(a) Strawberry fruit with seeds



(b) Strawberry flower



(c) A strawberry eaten by bird(s)



(d) A bird eating strawberries



(e) Strawberry plants showing runners and seeded fruit

FIGURE 1: Strawberry plant propagation: through seed dispersion [25–28].

SbPPA is tested on both unconstrained and constrained benchmark problems also used in [22, 29, 30]. Experimental results are presented in Tables 4–7 in terms of best, mean, worst, and standard deviation for all algorithms. The paper is organised as follows. In Section 2 we briefly introduce the feeding station model representing strawberry plants in fruit and the main characteristics of the paths followed by different agents that disperse the seeds. Section 3 presents the SbPPA in pseudocode form. The experimental settings, results, and convergence graphs for different problems are given in Section 4.

## 2. Aspects of the Feeding Station Model

Some animals and plants depend on each other to conserve their species [31]. Thus, many plants require, for effective seed

dispersal, the visits of frugivorous birds or animals according to a certain distribution [2, 3, 32, 33].

Seed dispersal by different agents is also called “seed shadow” [32]; this shows the abundance of seeds spread globally or locally around parent plants. Here a queuing model is used which, in the context of a strawberry feeding station model, involves two parts:

- (1) the quantity of fruit or seeds available to agents which implies the rate at which the agents will visit the plants,
- (2) a probability density function that tells us about the service rate with which the agents are served by the plants.

The model estimates the quantity of seeds that is spread locally compared to that dispersed globally [34–38]. There are two aspects that need to be balanced: exploitation, which

is represented by the dispersal of seeds around the plants, and exploration which ensures that the search space is well covered.

Agents arrive at plants in a random process. Assume that at most one agent arrives to the plants in any unit of time (orderliness condition). It is further supposed that the probability of arrivals of agents to the plants remains the same for a particular period of time. This period corresponds to when the plants are in fruit and during which time the number of visitors is stable (stationarity condition). Furthermore, it is assumed that the arrival of one agent does not affect the rest of arrivals (independence).

With these assumptions in mind, the arrival of agents to plants follows a Poisson process [39, 40], which can be formally described as follows. Let  $X'$  be the random variable representing the number of arrivals per unit of time  $t$ . Then, the probability of  $k$  arrivals over  $t$  is

$$P(X' = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}, \quad (1)$$

where  $\lambda$  denotes the mean arrival rate of agents per time unit  $t$ . On the other hand, the time taken by agents in successfully eating fruit and leaving to disperse its seeds, in other words the service time for agents, is expressed by a random variable which follows the exponential probability distribution [41]. This can be expressed as follows:

$$S(t) = \mu e^{-\mu t}, \quad (2)$$

where  $\mu$  is the average number of agents that can feed at time  $t$ . Let us assume that the arrival rate of agents is less than the fruits available on all plants per unit of time; therefore  $\lambda < \mu$ .

We assume that the system is in steady state. Let  $A$  denote the average number of agents in the strawberry field (some already eating and the rest waiting to feed) and  $A_q$  the average number of agents waiting to get the chance to feed. If we denote the average number of agents eating fruits by  $\lambda/\mu$ , then by Little's formula [42], we have

$$A = A_q + \frac{\lambda}{\mu}. \quad (3)$$

Since the plant needs to maximise dispersion, this is equivalent to having a large  $A_q$  in (3). Therefore, from this equation, we need to solve the following problem:

$$\begin{aligned} \text{Maximize} \quad & A_q = A - \frac{\lambda}{\mu}, \\ \text{subject to} \quad & g_1(\lambda, \mu) = \lambda < \mu + 1, \\ & \lambda > 0, \quad \mu > 0, \end{aligned} \quad (4)$$

where  $A = 10$ , which represents the population size in the implementation. The simple limits on the variables are  $0 < \lambda$ ,  $\mu \leq 100$ . The optimum solution to this particular problem is  $\lambda = 1.1$ ,  $\mu = 0.1$ , and  $A_q = 1$ .

Frugivores may travel far away from the plants and hence will disperse the seeds far and wide. This feeding behaviour typically follows a Lévy distribution [43–45]. In the following we present some basic facts about it.

**2.1. Lévy Distribution.** The Lévy distribution is a probability density distribution for random variables. Here the random variables represent the directions of flights of arbitrary birds. This function ranges over real numbers in the domain represented by the problem search space.

The flight lengths of the agents served by the plants follow a heavy tailed power law distribution [14], represented by

$$L(s) \sim |s|^{-1-\beta}, \quad (5)$$

where  $L(s)$  denotes the Lévy distribution with index  $\beta \in (0, 2)$ . Lévy flights are unique arbitrary excursions whose step lengths are drawn from (5). An alternative form of Lévy distribution is [14]

$$L(s, \gamma, \mu) = \begin{cases} \sqrt{\frac{\gamma}{2\pi}} \left( \frac{1}{(s-\mu)} \right)^{3/2} \cdot \exp \left[ -\frac{\gamma}{2(s-\mu)} \right], & 0 < \mu < s < \infty, \\ 0, & \text{Otherwise.} \end{cases} \quad (6)$$

This implies that

$$\lim_{s \rightarrow \infty} L(s, \gamma, \mu) \approx \sqrt{\frac{\gamma}{2\pi}} \left( \frac{1}{s} \right)^{3/2}. \quad (7)$$

In terms of the Fourier transform [14], the limiting value of  $L(s)$  can be written as

$$\lim_{s \rightarrow \infty} L(s) = \frac{\alpha \beta \Gamma(\beta) \sin(\pi \beta / 2)}{\pi |s|^{1+\beta}}, \quad (8)$$

where  $\Gamma(\beta)$  is the Gamma function [46], defined by

$$\Gamma(\beta) = \int_0^\infty x^{\beta-1} e^{-x} dx. \quad (9)$$

The steps  $L(s)$  are generated by Mantegna's algorithm [14]. This algorithm ensures that the behaviour of Lévy flights is symmetric and stable as shown in Figure 3(b).

### 3. Strawberry Plant Propagation Algorithm: The Feeding Station Model

We assume that the arrival of different agents (birds and animals) to the plants to feed is according to the Poisson distribution [40]. As per the solution of problem (4), the mean arrival rate is  $\lambda = 1.1$ , and  $NP = 10$  is the size of the agents population. Let  $k = 1, 2, \dots, A$  be the possible numbers of agents visiting the plants per unit time. With these assumptions the graphic representation of (1) results in Figure 2.

As already stated, it is essential in this algorithm to balance exploration and exploitation. To this end, we choose a threshold value of the Poisson probability that dictates how much exploration and exploitation are done during the search. The probability  $\text{Poiss}(\lambda) < 0.05$  means that

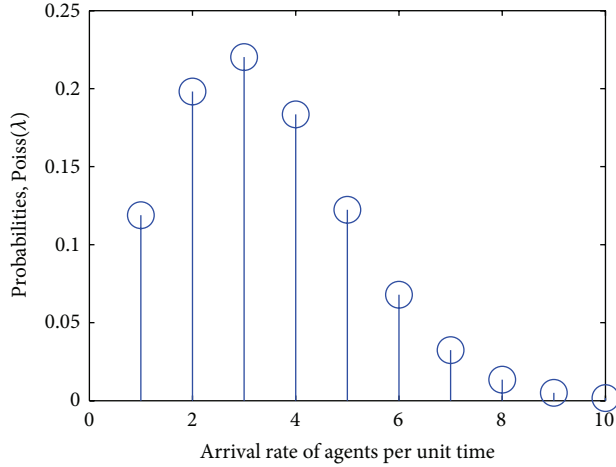


FIGURE 2: Distribution of agents arriving at strawberry plants to eat fruit and disperse seeds.

exploitation is covered. In this case, (10) below is used, which helps the algorithm to search locally:

$$x_{i,j}^* = \begin{cases} x_{i,j} + \xi_j (x_{i,j} - x_{l,j}) & \text{if } PR \leq 0.8; j = 1, 2, \dots, n; \\ & i, l = 1, 2, \dots, NP; i \neq l, \\ x_{i,j}, & \text{Otherwise,} \end{cases} \quad (10)$$

where PR denotes the rate of dispersion of the seeds locally, around SP;  $x_{i,j}^*$  and  $x_{i,j} \in [a_j, b_j]$  are the  $j$ th coordinates of the seeds  $X_i^*$  and  $X_i$ , respectively;  $a_j$  and  $b_j$  are the  $j$ th lower and upper bounds defining the search space of the problem and  $\xi_j \in [-1, 1]$ . The indices  $l$  and  $i$  are mutually exclusive.

On the other hand, if  $Poiss(\lambda) \geq 0.05$  then global dispersion of seeds becomes more prominent. This is implemented by using the following equation:

$$x_{i,j}^* = \begin{cases} x_{i,j} + L_i (x_{i,j} - \theta_j) & \text{if } PR \leq 0.8, \theta_j \in [a_j, b_j], \\ & i = 1, 2, \dots, NP; \\ & j = 1, 2, \dots, n, \\ x_{i,j}, & \text{Otherwise,} \end{cases} \quad (11)$$

where  $L_i$  is a step drawn from the Lévy distribution [14] and  $\theta_j$  is a random coordinate within the search space. Equations (10) and (11) perturb the current solution, the results of which can be seen in Figures 3(a) and 3(b), respectively.

As mentioned in Algorithm 1, we first collect the best solutions from the first NP trial runs to form a population of potentially good solutions denoted by  $pop_{best}$ . The convergence rate of SbPPA is shown in Figures 4 and 5 for different test problems used in our experiments (see Appendices). The statistics values best, worst, mean, and standard deviation are calculated based on  $pop_{best}$ .

The seed-based propagation process of SP can be represented in the following steps.

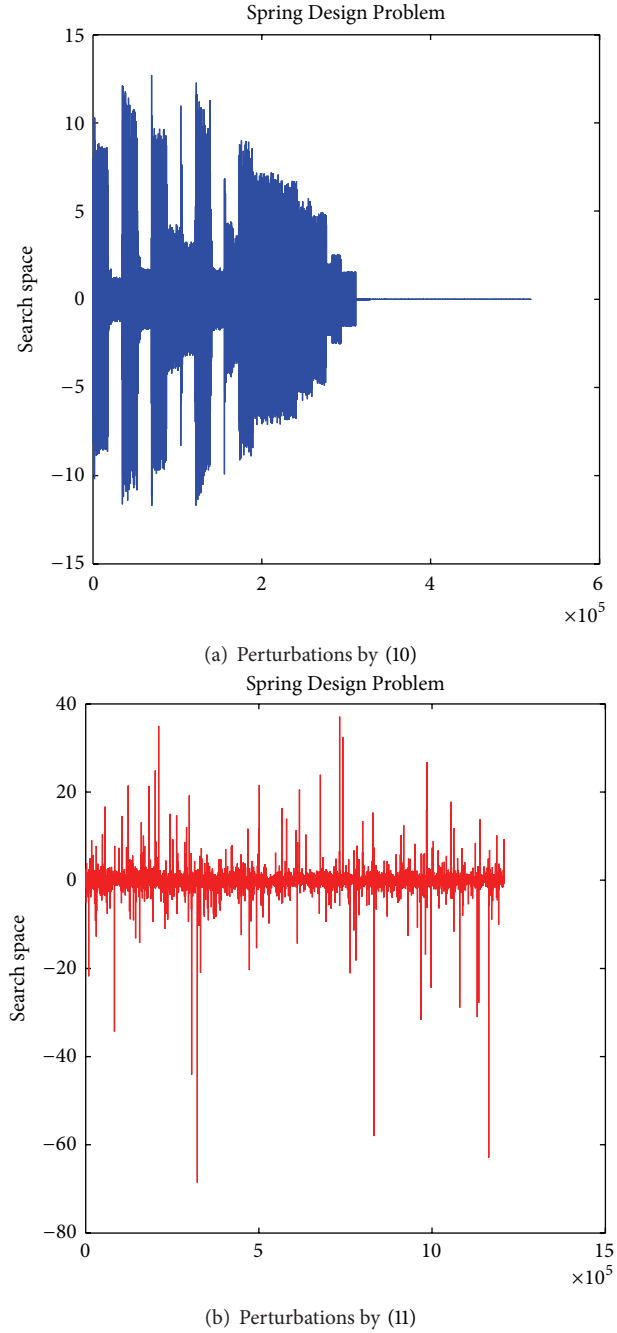


FIGURE 3: Overall performance of SbPPA on Spring Design Problem.

- (1) The dispersal of seeds in the neighbourhood of the SP, as shown in Figure 1(e), is carried out either by fruits fallen from strawberry plants after they become ripe or by agents. The step lengths for this phase are calculated using (10).
- (2) Seeds are spread globally through agents, as shown in Figures 1(c) and 1(d). The step lengths for these travelling agents are drawn from the Lévy distribution [14].
- (3) The probabilities,  $Poiss(\lambda)$ , that a certain number  $k$  of agents will arrive to SP to eat fruits and disperse it,

```

(1) NP ← Population size, r ← Counter of trial runs, MaxExp ← Maximum experiments
(2) for r = 1: MaxExp do
(3)   if r ≤ NP then
(4)     Create a random population of seeds pop = {Xi | i = 1, 2, ..., NP},
           using (12) and collect the best solutions from each trial run, in popbest.
(5)     Evaluate the population pop.
(6)   end if
(7)   while r > NP do
(8)     Use updated population popbest.
(9)   end while
(10)  while (the stopping criteria is not satisfied) do
(11)    for i = 1 to NP do
(12)      if Poiss(λ)i ≥ 0.05, then,                                ▷ (Global or local seed dispersion)
(13)        for j = 1 to n do                                       ▷ (n is number of dimensions)
(14)          if rand ≤ PR then,                                     ▷ (PR = Perturbation Rate)
(15)            Update the current entry according to (11)
(16)          end if
(17)        end for
(18)      else
(19)        for j = 1 to n do
(20)          if rand ≤ PR then,
(21)            Update the current entry according to (10)
(22)          end if
(23)        end for
(24)      end if
(25)    end for
(26)    Update current best
(27)  end while
(28)  Return: Updated population and global best solution.
(29) end for

```

ALGORITHM 1: Seed-based Plant Propagation Algorithm (SbPPA) [47].

is used as a balancing factor between exploration and exploitation.

For implementation purposes, we assume that each SP produces one fruit, and each fruit is assumed to have one seed; by a solution  $X_i$  we mean the current position of the  $i$ th seed to be dispersed. The number of seeds in the population is denoted by NP. Initially we generate a random population of NP seeds using

$$x_{i,j} = a_j + (b_j - a_j)\eta_j, \quad j = 1, \dots, n, \quad (12)$$

where  $x_{i,j} \in [a_j, b_j]$  is the  $j$ th coordinate of solution  $X_i$ ,  $a_j$  and  $b_j$  are the  $j$ th coordinates of the bounds describing the search space of the problem, and  $\eta_j \in (0, 1)$ . This means that  $X_i = [x_{i,j}]$ , for  $j = 1, \dots, n$ , represents the position of the  $j$ th seed in population pop.

#### 4. Experimental Settings and Discussion

In our experiments we tested SbPPA against some recently developed algorithms and some well established and standard ones. Our set of test problems includes benchmark constrained and unconstrained optimization problems [22, 30, 48, 49]. The results are compared in terms of statistics (best, worst, mean and standard deviation) for solutions obtained

by SbPPA; ABC [18, 50]; PSO [51]; FF [21]; HPA [29]; SSO-C [22]; Classical Evolutionary Programming (CEP) [30]; and Fast Evolutionary Programming (FEP) [30]. The detailed descriptions of these problems are given in Appendices.

In Tables 4 and 7, the significance of results is shown in terms of win/tie/loss (see Table 2 in [52]) according to the following notations:

- (i) (+) when SbPPA is better;
- (ii) ( $\approx$ ) when the results are approximately the same as those obtained with SbPPA;
- (iii) (−) when SbPPA is worse.

Moreover, in Tables 5 and 6 the significance of results obtained with SbPPA is highlighted.

**4.1. Parameter Settings.** The parameter settings are given in Tables 1–3.

#### 5. Conclusion

In this paper, a new metaheuristic referred to as the Seed-Based Plant Propagation Algorithm (SbPPA) [47] has been proposed. Plants have evolved a variety of ways to propagate. Propagation through seeds is perhaps the most common of them all and one which takes advantage of all sorts of

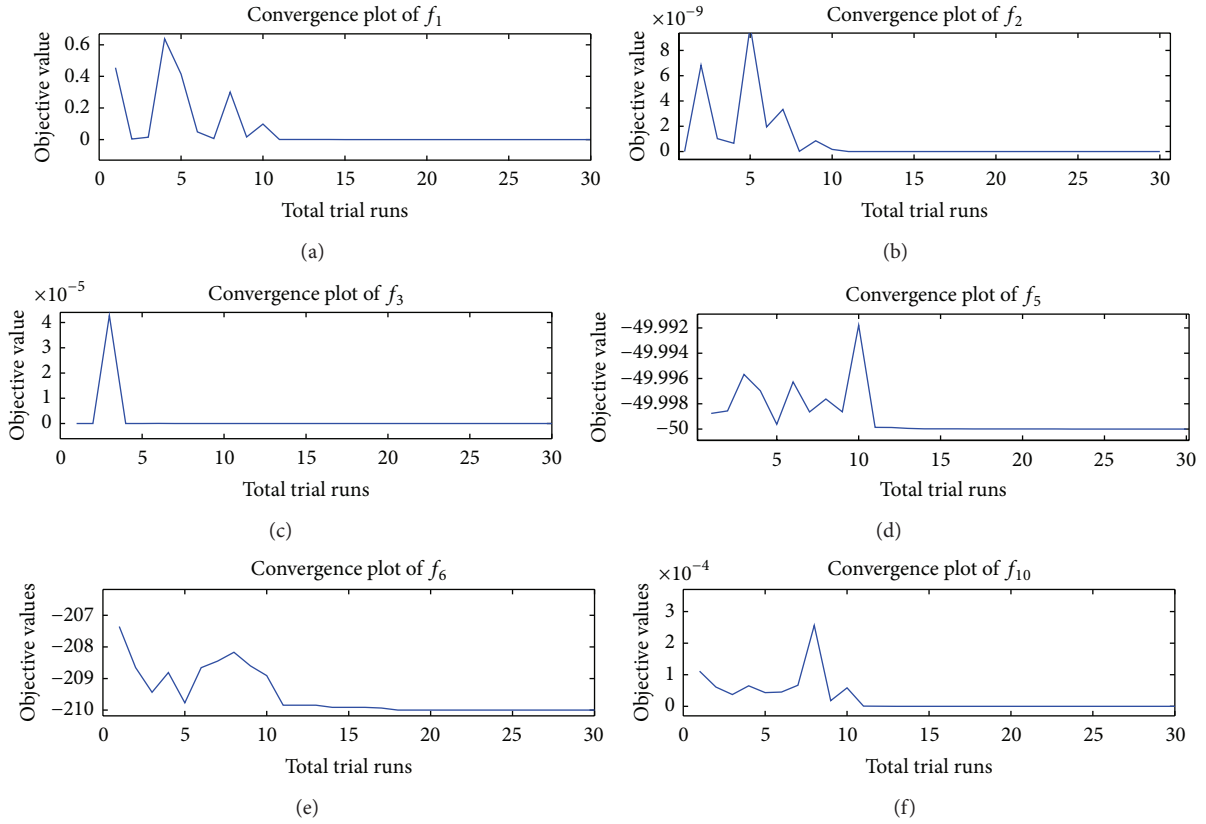


FIGURE 4: Performance of SbPPA on unconstrained global optimization problems.

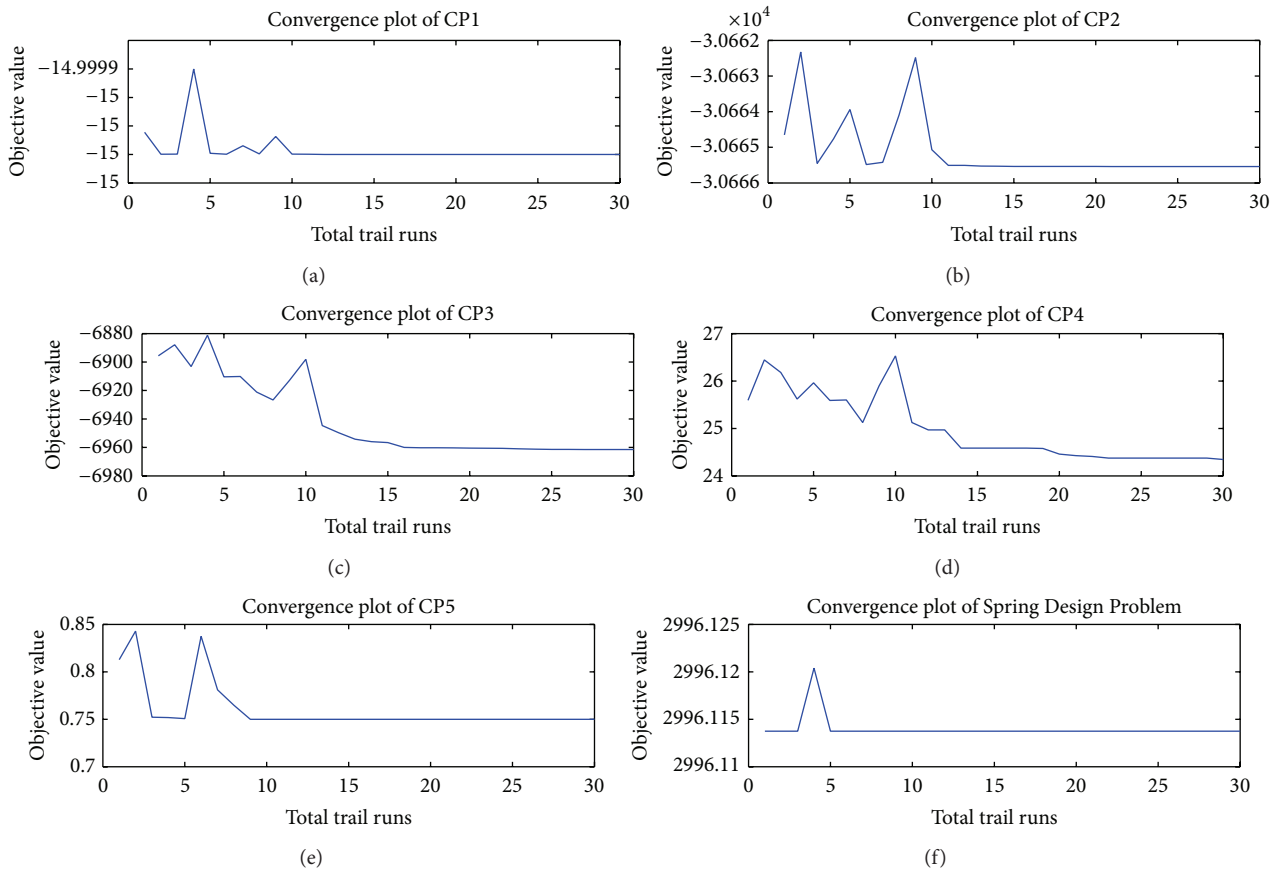


FIGURE 5: Performance of SbPPA on constrained global optimization problems (see Appendices).

TABLE 1: Parameters used for each algorithm for solving unconstrained global optimization problems  $f_1-f_{10}$ . All experiments are repeated 30 times.

PSO [16, 29]	ABC [18, 29]	HPA [29]	SbPPA [47]
$M = 100$	$SN = 100$	$Agents = 100$	$NP = 10$
$G_{max} = \frac{(Dimension \times 20,000)}{M}$	$MCN = \frac{(Dimension \times 20,000)}{SN}$	$Iteration\ number = \frac{(Dimension \times 20,000)}{Agents}$	$Iteration\ number = \frac{(Dimension \times 20,000)}{NP}$
$c_1 = 2$	$MR = 0.8$	$c_1 = 2$	$PR = 0.8$
$c_2 = 2$	$Limit = \frac{(SN \times dimension)}{2}$	$c_2 = 2$	$Poiss(\lambda) = 0.05$
$W = \frac{(G_{max} - iteration_{index})}{G_{max}}$	—	$Limit = \frac{(Agents \times dimension)}{2}$	—
—	—	$W = \frac{(Iteration\ number - iteration_{index})}{Iteration\ number}$	—

TABLE 2: Experimental setup used for each algorithm for solving unconstrained global optimization problems  $f_{11}$ - $f_{26}$ . All experiments are repeated 50 times.

CEP [30, 53, 54]	FEP [30]	SbPPA
Population size $\mu = 100$	Population size $\mu = 100$	NP = 10
Tournament size $q = 10$	Tournament size $q = 10$	PR = 0.8
$\eta = 3.0$	$\eta = 3.0$	Poiss( $\lambda$ ) = 0.05

agents ranging from wind to water, birds, and animals. The strawberry plant uses both runners and seeds to propagate. Here we consider the propagation through seeds that the strawberry plant has evolved, to design an efficient optimization algorithm.

To capture the dispersal process, we adopt a queuing approach which, given the extent of fruit produced, indicates the extent of seeds dispersed and hence the effectiveness of the search/optimization algorithm based on this process. Looking at the random process of agents using the plants (feeding station) it is reasonable to assume that it is of the Poisson type. On the other hand, the time taken by agents in successfully eating fruit and leaving to disperse its seeds, in other words the service time for agents, is expressed by a random variable which follows the exponential probability distribution. To this end, we choose a threshold value of the Poisson probability that dictates how much exploration and exploitation are done during the search. An alternative strategy has been adopted here. This strategy consists in making sure that the initial population is as good as the user can afford it to be by using best solutions found so far. The effects of this strategy on convergence are shown through convergence plots of Figures 4 and 5, for some of the solved problems. SbPPA is easy to implement as it requires less arbitrary parameter settings than other algorithms. The success rate of SbPPA increases as it gets its population of best solutions. It has been implemented for both unconstrained and constrained optimization problems. Its performance, compared to that of other algorithms, points to SbPPA as being superior.

## Appendices

### A. Unconstrained Global Optimization Problems

See Tables 8 and 9.

### B. Set of Constrained Global Optimization Problems Used in Our Experiments

B.1. CP1. Consider the following:

$$\text{Min} \quad f(x) = 5 \sum_{d=1}^4 x_d - 5 \sum_{d=1}^4 x_d^2 - \sum_{d=5}^{13} x_d,$$

$$\begin{aligned} \text{subject to} \quad & g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0, \\ & g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0, \\ & g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0, \\ & g_4(x) = -8x_1 + x_{10} \leq 0, \\ & g_5(x) = -8x_2 + x_{11} \leq 0, \\ & g_6(x) = -8x_3 + x_{12} \leq 0, \\ & g_7(x) = -2x_4 - x_5 + x_{10} \leq 0, \\ & g_8(x) = -2x_6 - x_7 + x_{11} \leq 0, \\ & g_9(x) = -2x_8 - x_9 + x_{12} \leq 0, \end{aligned} \quad (\text{B.1})$$

where bounds are  $0 \leq x_i \leq 1$  ( $i = 1, \dots, 9, 13$ ),  $0 \leq x_i \leq 100$  ( $i = 10, 11, 12$ ). The global optimum is at  $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$ ,  $f(x^*) = -15$ .

B.2. CP2. Consider the following:

$$\begin{aligned} \text{Min} \quad & f(x) = 5.3578547x_2 + 0.8356891x_1x_5 \\ & + 37.293239x_1 - 40792.141, \\ \text{subject to} \quad & g_1(x) = 85.334407 + 0.0056858x_2x_5 \\ & + 0.0006262x_1x_4 - 0.0022053x_3x_5 \\ & - 92 \leq 0, \\ & g_2(x) = -85.334407 - 0.0056858x_2x_5 \\ & - 0.0006262x_1x_4 + 0.0022053x_3x_5 \\ & \leq 0, \\ & g_3(x) = 80.51249 + 0.0071317x_2x_5 \\ & + 0.0029955x_1x_2 - 0.0021813x_2 \\ & - 110 \leq 0, \\ & g_4(x) = -80.51249 - 0.0071317x_2x_5 \\ & + 0.0029955x_1x_2 - 0.0021813x_2 \\ & + 90 \leq 0, \\ & g_5(x) = 9.300961 - 0.0047026x_3x_5 \\ & - 0.0012547x_1x_3 - 0.0019085x_3x_4 \\ & - 25 \leq 0, \\ & g_6(x) = -9.300961 - 0.0047026x_3x_5 \\ & - 0.0012547x_1x_3 - 0.0019085x_3x_4 \\ & + 20 \leq 0, \end{aligned} \quad (\text{B.2})$$

TABLE 3: Parameters used for each algorithm for solving constrained optimization problems. All experiments are repeated 30 times.

PSO [51]	ABC [50]	FF [21]	SSO-C [22]	SbPPA [47]
$M = 250$	SN = 40	Fireflies = 25	$N = 50$	NP = 10
$G_{\max} = 300$	MCN = 6000	Iteration number = 2000	Iteration number = 500	Iteration number = 800
$c_1 = 2$	MR = 0.8	$q = 1.5$	PF = 0.7	PR = 0.8
$c_2 = 2$	—	$\alpha = 0.001$	—	Poiss( $\lambda$ ) = 0.05
Weight factors = 0.9 to 0.4	—	—	—	—

TABLE 4: Results obtained by SbPPA, HPA, PSO, and ABC. All problems in this table are unconstrained.

Fun.	Dim	Algorithm	Best	Worst	Mean	SD
$f_1$	4	ABC	(+) 0.0129	(+) 0.6106	(+) 0.1157	(+) 0.111
		PSO	(-) 6.8991E - 08	(+) 0.0045	(+) 0.001	(+) 0.0013
		HPA	(+) 2.0323E - 06	(+) 0.0456	(+) 0.009	(+) 0.0122
		SbPPA	1.08E - 07	7.05E - 06	3.05E - 06	3.14E - 06
$f_2$	2	ABC	(+) 1.2452E - 08	(+) 8.4415E - 06	(+) 1.8978E - 06	(+) 1.8537E - 06
		PSO	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		HPA	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		SbPPA	0	0	0	0
$f_3$	2	ABC	( $\approx$ ) 0	(+) 4.8555E - 06	(+) 4.1307E - 07	(+) 1.2260E - 06
		PSO	( $\approx$ ) 0	(+) 3.5733E - 07	(+) 1.1911E - 08	(+) 6.4142E - 08
		HPA	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		SbPPA	0	0	0	0
$f_4$	2	ABC	( $\approx$ ) -1.03163	( $\approx$ ) -1.03163	( $\approx$ ) -1.03163	( $\approx$ ) 0
		PSO	( $\approx$ ) -1.03163	( $\approx$ ) -1.03163	( $\approx$ ) -1.03163	( $\approx$ ) 0
		HPA	( $\approx$ ) -1.03163	( $\approx$ ) -1.03163	( $\approx$ ) -1.03163	( $\approx$ ) 0
		SbPPA	-1.031628	-1.031628	-1.031628	0
$f_5$	6	ABC	( $\approx$ ) -50.0000	( $\approx$ ) -50.0000	( $\approx$ ) -50.0000	(-) 0
		PSO	( $\approx$ ) -50.0000	( $\approx$ ) -50.0000	( $\approx$ ) -50.0000	(-) 0
		HPA	( $\approx$ ) -50.0000	( $\approx$ ) -50.0000	( $\approx$ ) -50.0000	(-) 0
		SbPPA	-50.0000	-50.0000	-50.0000	5.88E - 09
$f_6$	10	ABC	(+) -209.9929	(+) -209.8437	(+) -209.9471	(+) 0.044
		PSO	( $\approx$ ) -210.0000	( $\approx$ ) -210.0000	( $\approx$ ) -210.0000	(-) 0
		HPA	( $\approx$ ) -210.0000	( $\approx$ ) -210.0000	( $\approx$ ) -210.0000	(+) 1
		SbPPA	-210.0000	-210.0000	-210.0000	4.86E - 06
$f_7$	30	ABC	(+) 2.6055E - 16	(+) 5.5392E - 16	(+) 4.7403E - 16	(+) 9.2969E - 17
		PSO	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		HPA	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		SbPPA	0	0	0	0
$f_8$	30	ABC	(+) 2.9407E - 16	(+) 5.5463E - 16	(+) 4.8909E - 16	(+) 9.0442E - 17
		PSO	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		HPA	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		SbPPA	0	0	0	0
$f_9$	30	ABC	( $\approx$ ) 0	(+) 1.1102E - 16	(+) 9.2519E - 17	(+) 4.1376E - 17
		PSO	( $\approx$ ) 0	(+) 1.1765E - 01	(+) 2.0633E - 02	(+) 2.3206E - 02
		HPA	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0	( $\approx$ ) 0
		SbPPA	0	0	0	0
$f_{10}$	30	ABC	(+) 2.9310E - 14	(+) 3.9968E - 14	(+) 3.2744E - 14	(+) 2.5094E - 15
		PSO	( $\approx$ ) 7.9936E - 15	(+) 1.5099E - 14	(+) 8.5857E - 15	(+) 1.8536E - 15
		HPA	( $\approx$ ) 7.9936E - 15	(+) 1.5099E - 14	(+) 1.1309E - 14	(+) 3.54E - 15
		SbPPA	7.994E - 15	7.99361E - 15	7.994E - 15	7.99361E - 15

TABLE 5: Results obtained by SbPPA, CEP, and FEP. All problems in this table are unconstrained [30].

Function number	Algorithm	Maximum generations	Mean	SD
$f_{11}$	CEP	2000	$2.60E - 03$	$1.70E - 04$
	FEP		$8.10E - 03$	$7.70E - 04$
	SbPPA		<b><math>9.45E - 13</math></b>	<b><math>4.08E - 12</math></b>
$f_{12}$	CEP	5000	2	1.2
	FEP		0.3	0.5
	SbPPA		<b><math>3.93E - 02</math></b>	<b><math>3.76E - 02</math></b>
$f_{13}$	CEP	20000	6.17	13.61
	FEP		5.06	5.87
	SbPPA		$1.86E + 01$	$2.25E + 00$
$f_{14}$	CEP	1500	577.76	1125.76
	FEP		0	0
	SbPPA		<b>0</b>	<b>0</b>
$f_{15}$	CEP	3000	$1.80E - 02$	$6.40E - 03$
	FEP		$7.60E - 03$	$2.60E - 03$
	SbPPA		<b><math>3.61E - 03</math></b>	<b><math>1.31E - 03</math></b>
$f_{16}$	CEP	9000	$-7.92E + 03$	$6.35E + 02$
	FEP		$-1.26E + 04$	$5.26E + 01$
	SbPPA		$-1.16E + 04$	$6.04E + 01$
$f_{17}$	CEP	5000	89	23.1
	FEP		$4.60E - 02$	$1.20E - 02$
	SbPPA		<b><math>8.73E + 00</math></b>	<b><math>9.88E - 01</math></b>
$f_{18}$	CEP	100	0.398	$1.50E - 07$
	FEP		0.398	$1.50E - 07$
	SbPPA		$3.98E - 01$	<b>0</b>

TABLE 6: Results obtained by SbPPA, CEP, and FEP. All problems in this table are unconstrained [30].

Function number	Algorithm	Maximum generations	Mean	SD
$f_{19}$	CEP	100	3	0
	FEP		3.02	0.11
	SbPPA		3	<b><math>3.05E - 15</math></b>
$f_{20}$	CEP	100	$-3.86E + 00$	$1.40E - 02$
	FEP		$-3.86E + 00$	$1.40E - 05$
	SbPPA		$-3.86E + 00$	<b><math>2.75E - 15</math></b>
$f_{21}$	CEP	200	$-3.28E + 00$	$5.80E - 02$
	FEP		$-3.27E + 00$	$5.90E - 02$
	SbPPA		<b><math>-3.32E + 00</math></b>	<b><math>2.91E - 14</math></b>
$f_{22}$	CEP	100	$-6.86E + 00$	$2.67E + 00$
	FEP		$-5.52E + 00$	$1.59E + 00$
	SbPPA		<b><math>-1.02E + 01</math></b>	<b><math>4.30E - 09</math></b>
$f_{23}$	CEP	100	-8.27	2.95
	FEP		-5.52	2.12
	SbPPA		<b><math>-1.04E + 01</math></b>	<b><math>7.73E - 09</math></b>
$f_{24}$	CEP	100	-9.1	2.92
	FEP		-6.57	3.14
	SbPPA		<b><math>-1.05E + 01</math></b>	<b><math>1.03E - 07</math></b>
$f_{25}$	CEP	100	1.66	1.19
	FEP		1.22	0.56
	SbPPA		<b><math>9.98E - 01</math></b>	<b><math>1.13E - 16</math></b>
$f_{26}$	CEP	4000	$4.70E - 04$	$3.00E - 04$
	FEP		$5.00E - 04$	$3.20E - 04$
	SbPPA		<b><math>3.07E - 04</math></b>	<b><math>6.80E - 15</math></b>

TABLE 7: Results obtained by SbPPA, PSO, ABC, FF, and SSO-C. All problems in this table are standard constrained optimization problems.

Fun. name	Optimal	Algorithm	Best	Mean	Worst	SD
CP1	-15	PSO	( $\approx$ ) -15	( $\approx$ ) -15	( $\approx$ ) -15	(-) 0
		ABC	( $\approx$ ) -15	( $\approx$ ) -15	( $\approx$ ) -15	(-) 0
		FF	(+) 14.999	(+) 14.988	(+) 14.798	(+) 6.40E - 07
		SSO-C	( $\approx$ ) -15	( $\approx$ ) -15	( $\approx$ ) -15	(-) 0
		SbPPA	-15	-15	-15	1.95E - 15
CP2	-30665.539	PSO	( $\approx$ ) -30665.5	(+) -30662.8	(+) -30650.4	(+) 5.20E - 02
		ABC	( $\approx$ ) -30665.5	(+) -30664.9	(+) -30659.1	(+) 8.20E - 02
		FF	( $\approx$ ) -3.07E + 04	(+) -30662	(+) -30649	(+) 5.20E - 02
		SSO-C	( $\approx$ ) -3.07E + 04	( $\approx$ ) -30665.5	(+) -30665.1	(+) 1.10E - 04
		SbPPA	-30665.5	-30665.5	-30665.5	2.21E - 06
CP3	-6961.814	PSO	(+) -6.96E + 03	(+) -6958.37	(+) -6942.09	(+) 6.70E - 02
		ABC	(-) -6961.81	(+) -6958.02	(+) -6955.34	(-) 2.10E - 02
		FF	(+) -6959.99	(+) -6.95E + 03	(+) -6947.63	(-) 3.80E - 02
		SSO-C	(-) -6961.81	(+) -6961.01	(+) -6960.92	(-) 1.10E - 03
		SbPPA	-6961.5	-6961.38	-6961.45	0.043637
CP4	24.306	PSO	(-) 24.327	(+) 2.45E + 01	(+) 24.843	(+) 1.32E - 01
		ABC	(+) 24.48	(+) 2.66E + 01	(+) 28.4	(+) 1.14
		FF	(-) 23.97	(+) 28.54	(+) 30.14	(+) 2.25
		SSO-C	(-) 24.306	(-) 24.306	(-) 24.306	(-) 4.95E - 05
		SbPPA	24.34442	24.37536	24.37021	0.012632
CP5	-0.7499	PSO	( $\approx$ ) -0.7499	(+) -0.749	(+) -0.7486	(+) 1.20E - 03
		ABC	( $\approx$ ) -0.7499	(+) -0.7495	(+) -0.749	(+) 1.67E - 03
		FF	(+) -0.7497	(+) -0.7491	(+) -0.7479	(+) 1.50E - 03
		SSO-C	( $\approx$ ) -0.7499	( $\approx$ ) -0.7499	( $\approx$ ) -0.7499	(-) 4.10E - 09
		SbPPA	0.7499	0.749901	0.7499	1.66E - 07
Spring Design Problem	Not known	PSO	(+) 0.012858	(+) 0.014863	(+) 0.019145	(+) 0.001262
		ABC	( $\approx$ ) 0.012665	(+) 0.012851	(+) 0.01321	(+) 0.000118
		FF	( $\approx$ ) 0.012665	(+) 0.012931	(+) 0.01342	(+) 0.001454
		SSO-C	( $\approx$ ) 0.012665	(+) 0.012765	(+) 0.012868	(+) 9.29E - 05
		SbPPA	0.012665	0.012666	0.012666	3.39E - 10
Welded beam design problem	Not known	PSO	(+) 1.846408	(+) 2.011146	(+) 2.237389	(+) 0.108513
		ABC	(+) 1.798173	(+) 2.167358	(+) 2.887044	(+) 0.254266
		FF	(+) 1.724854	(+) 2.197401	(+) 2.931001	(+) 0.195264
		SSO-C	( $\approx$ ) 1.724852	(+) 1.746462	(+) 1.799332	(+) 0.02573
		SbPPA	1.724852	1.724852	1.724852	4.06E - 08
Speed reducer design optimization	Not known	PSO	(+) 3044.453	(+) 3079.262	(+) 3177.515	(+) 26.21731
		ABC	(+) 2996.116	(+) 2998.063	(+) 3002.756	(+) 6.354562
		FF	(+) 2996.947	(+) 3000.005	(+) 3005.836	(+) 8.356535
		SSO-C	( $\approx$ ) 2996.113	( $\approx$ ) 2996.113	( $\approx$ ) 2996.113	(+) 1.34E - 12
		SbPPA	2996.114	2996.114	2996.114	0

TABLE 8: Unconstrained global optimization problems (Set-1) used in our experiments.

Fun.	Fun. name	$D$	$C$	Range	Min	Formulation
$f_1$	Colville	4	UN	$[-10\ 10]^D$	0	$f(x) = 100(x_1^2 - x_2) + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$
$f_2$	Matyas	2	UN	$[-10\ 10]^D$	0	$f(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$
$f_3$	Schaffer	2	MN	$[-100\ 100]^D$	0	$f(x) = 0.5 + \frac{\sin^2\left(\sqrt{\sum_{i=1}^n x_i^2}\right) - 0.5}{(1 + 0.001(\sum_{i=1}^n x_i^2))^2}$
$f_4$	Six Hump Camel Back	2	MN	$[-5\ 5]^D$	-1.03163	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$
$f_5$	Trid6	6	UN	$[-36\ 36]^D$	-50	$f(x) = \sum_{i=1}^6 (x_i - 1)^2 - \sum_{i=2}^6 x_i x_{i-1}$
$f_6$	Trid10	10	UN	$[-100\ 100]^D$	-210	$f(x) = \sum_{i=1}^{10} (x_i - 1)^2 - \sum_{i=2}^{10} x_i x_{i-1}$
$f_7$	Sphere	30	US	$[-100\ 100]^D$	0	$f(x) = \sum_{i=1}^n x_i^2$
$f_8$	SumSquares	30	US	$[-10\ 10]^D$	0	$f(x) = \sum_{i=1}^n ix_i^2$
$f_9$	Griewank	30	MN	$[-600\ 600]^D$	0	$f(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$
$f_{10}$	Ackley	30	MN	$[-32\ 32]^D$	0	$f(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$

where  $78 \leq x_1 \leq 102$ ,  $33 \leq x_2 \leq 45$ , and  $27 \leq x_i \leq 45$  ( $i = 3, 4, 5$ ). The optimum solution is  $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ , where  $f(x^*) = -30665.539$ . Constraints  $g_1$  and  $g_6$  are active.

B.3. CP3. Consider the following:

$$\begin{aligned}
 \text{Min} \quad & f(x) = (x_1 - 10)^3 + (x_2 - 20)^3, \\
 \text{subject to} \quad & g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0, \\
 & g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leq 0, \\
 & \quad \quad \quad (B.3)
 \end{aligned}$$

where  $13 \leq x_1 \leq 100$  and  $0 \leq x_2 \leq 100$ . The optimum solution is  $x^* = (14.095, 0.84296)$  where  $f(x^*) = -6961.81388$ . Both constraints are active.

B.4. CP4. Consider the following:

$$\begin{aligned}
 \text{Min} \quad & f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 \\
 & + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 \\
 & + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 \\
 & + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45, \\
 \text{subject to} \quad & g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leq 0, \\
 & g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0,
 \end{aligned}$$

$$g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0,$$

$$\begin{aligned}
 g_4(x) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 \\
 &- 7x_4 - 120 \leq 0,
 \end{aligned}$$

$$\begin{aligned}
 g_5(x) &= 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x^4 \\
 &- 40 \leq 0,
 \end{aligned}$$

$$\begin{aligned}
 g_6(x) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 \\
 &- 6x_6 \leq 0,
 \end{aligned}$$

$$\begin{aligned}
 g_7(x) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x_5^2 \\
 &- x_6 - 30 \leq 0,
 \end{aligned}$$

$$g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0, \quad (B.4)$$

where  $-10 \leq x_i \leq 10$  ( $i = 1, \dots, 10$ ). The global optimum is  $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$ , where  $f(x^*) = 24.3062091$ . Constraints  $g_1, g_2, g_3, g_4, g_5$ , and  $g_6$  are active.

B.5. CP5. Consider the following:

$$\begin{aligned}
 \text{Min} \quad & f(x) = x_1^2 + (x_2 - 1)^2, \\
 \text{subject to} \quad & g_1(x) = x_2 - x_1^2 = 0,
 \end{aligned} \quad (B.5)$$

where  $1 \leq x_1 \leq 1$ ,  $1 \leq x_2 \leq 1$ . The optimum solution is  $x^* = (\pm 1/\sqrt{2}, 1/2)$ , where  $f(x^*) = 0.7499$ .

TABLE 9: Unconstrained global optimization problems (Set-2) used in our experiments [30].

Fun. number	Range	$D$	Function	Formulation	$f_{\min}$
$f_{11}$	$[-10, 10]^D$	30	Schwefel's Problem 2.22	$f(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	0
$f_{12}$	$[-100, 100]^D$	30	Schwefel's Problem 2.21	$f(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	0
$f_{13}$	$[-10, 10]^D$	30	Rosenbrock	$f(x) = \sum_{i=1}^{n-1} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	0
$f_{14}$	$[-100, 100]^D$	30	Step	$f(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2$	0
$f_{15}$	$[-1.28, 1.28]^D$	30	Quartic (noise)	$f(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1)$	0
$f_{16}$	$[-500, 500]^D$	30	Schwefel	$f(x) = -\sum_{i=1}^n x_i \sin\left(\sqrt{ x_i }\right)$	-12569.5
$f_{17}$	$[-5.12, 5.12]^D$	30	Rastrigin	$f(x) = [x_i^2 - 10\cos(2\pi x_i) + 10]$	0
$f_{18}$	$[-5, 10] \times [0, 15]$	2	Branin	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	0.398
$f_{19}$	$[-2, 2]^D$	2	Goldstein-Price	$f(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3
$f_{20}$	$[0, 1]^D$	4	Hartman's Family ( $n = 3$ )	$f(x) = -\sum_{i=1}^4 c_i \exp\left[\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right]$	-3.86
$f_{21}$	$[0, 1]^D$	6	Hartman's Family ( $n = 6$ )	$f(x) = -\sum_{i=1}^4 c_i \exp\left[\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2\right]$	-3.32
$f_{22}$	$[0, 10]^D$	4	Shekel's Family ( $m = 5$ )	$f(x) = -\sum_{i=1}^5 \left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	-10
$f_{23}$	$[0, 10]^D$	4	Shekel's Family ( $m = 7$ )	$f(x) = -\sum_{i=1}^7 \left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	-10
$f_{24}$	$[0, 10]^D$	4	Shekel's Family ( $m = 10$ )	$f(x) = -\sum_{i=1}^{10} \left[(x - a_i)(x - a_i)^T + c_i\right]^{-1}$	-10
$f_{25}$	$[-65.536, 65.536]^D$	2	Shekel's Foxholes	$f(x) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6}\right]^{-1}$	1
$f_{26}$	$[-5, 5]^D$	4	Kowalik	$f(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	0.0003075

**B.6. Welded Beam Design Optimisation.** The welded beam design is a standard test problem for constrained design optimisation [55, 56]. There are four design variables: the width  $w$  and length  $L$  of the welded area and the depth  $d$  and thickness  $h$  of the main beam. The objective is to minimise the overall fabrication cost, under the appropriate constraints of shear stress  $\tau$ , bending stress  $\sigma$ , buckling load  $P$ , and maximum end deflection  $\delta$ . The optimization model is summarized as follows, where  $x^T = (w, L, d, h)$ :

$$\text{Minimise } f(x) = 1.10471w^2L + 0.04811dh(14.0 + L),$$

$$\text{subject to } g_1(x) = w - h \leq 0,$$

$$g_2(x) = \delta(x) - 0.25 \leq 0,$$

$$g_3(x) = \tau(x) - 13,600 \leq 0,$$

$$g_4(x) = \sigma(x) - 30,000 \leq 0,$$

$$g_5(x) = 1.10471w^2 + 0.04811dh(14.0 + L) - 5.0 \leq 0,$$

$$g_6(x) = 0.125 - w \leq 0,$$

$$g_7(x) = 6000 - P(x) \leq 0,$$

(B.6)

where

$$\begin{aligned}
 \sigma(x) &= \frac{504,000}{hd^2}, & D &= \frac{1}{2} \sqrt{L^2 + (w+d)^2}, \\
 Q &= 6000 \left( 14 + \frac{L}{2} \right), & \delta &= \frac{65,856}{30,000hd^3}, \\
 J &= \sqrt{2}wL \left( \frac{L^2}{6} + \frac{(w+d)^2}{2} \right), & \alpha &= \frac{6000}{\sqrt{2}wL}, \\
 \beta &= \frac{QD}{J}, & P &= 0.61423 \times 10^6 \frac{dh^3}{6} \left( 1 - \frac{\sqrt[4]{30/48}}{28} \right), \\
 \tau(x) &= \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}.
 \end{aligned}
 \tag{B.7}$$

**B.7. Speed Reducer Design Optimization.** The problem of designing a speed reducer [57] is a standard test problem. It consists of the design variables as face width  $x_1$ , module of teeth  $x_2$ , number of teeth on pinion  $x_3$ , length of the first shaft between bearings  $x_4$ , length of the second shaft between bearings  $x_5$ , diameter of the first shaft  $x_6$ , and diameter of the first shaft  $x_7$  (all variables are continuous except  $x_3$  that is integer). The weight of the speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the shaft [55]. The mathematical formulation of the problem, where  $x^T = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ , is as follows:

$$\begin{aligned}
 \text{Minimise } f(x) &= 0.7854x_1x_2^2 \\
 &\quad \cdot (3.3333x_3^2 + 14.9334x_3 + 43.0934) \\
 &\quad - 1.508x_1(x_6^2 + x_7^3) \\
 &\quad + 7.4777(x_6^3 + x_7^3) \\
 &\quad + 0.7854(x_4x_6^2 + x_5x_7^2), \\
 \text{subject to } g_1(x) &= \frac{27}{x_1x_2^2x_3} - 1 \leq 0, \\
 g_2(x) &= \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0, \\
 g_3(x) &= \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0, \\
 g_4(x) &= \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0, \\
 g_5(x) &= \frac{1.0}{110x_6^3} \sqrt{\left( \frac{745.0x_4}{x_2x_3} \right)^2 + 16.9 \times 10^6} \\
 &\quad - 1 \leq 0,
 \end{aligned}$$

$$\begin{aligned}
 g_6(x) &= \frac{1.0}{85x_7^3} \sqrt{\left( \frac{745.0x_5}{x_2x_3} \right)^2 + 157.5 \times 10^6} \\
 &\quad - 1 \leq 0, \\
 g_7(x) &= \frac{x_2x_3}{40} - 1 \leq 0, \\
 g_8(x) &= \frac{5x_2}{x_1} - 1 \leq 0, \\
 g_9(x) &= \frac{x_1}{12x_2} - 1 \leq 0, \\
 g_{10}(x) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \\
 g_{11}(x) &= \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0.
 \end{aligned}
 \tag{B.8}$$

The simple limits on the design variables are  $2.6 \leq x_1 \leq 3.6$ ,  $0.7 \leq x_2 \leq 0.8$ ,  $17 \leq x_3 \leq 28$ ,  $7.3 \leq x_4 \leq 8.3$ ,  $7.8 \leq x_5 \leq 8.3$ ,  $2.9 \leq x_6 \leq 3.9$ , and  $5.0 \leq x_7 \leq 5.5$ .

**B.8. Spring Design Optimisation.** The main objective of this problem [58, 59] is to minimize the weight of a tension/compression spring, subject to constraints of minimum deflection, shear stress, surge frequency, and limits on outside diameter and on design variables. There are three design variables: the wire diameter  $x_1$ , the mean coil diameter  $x_2$ , and the number of active coils  $x_3$  [55]. The mathematical formulation of this problem, where  $x^T = (x_1, x_2, x_3)$ , is as follows:

$$\begin{aligned}
 \text{Minimize } f(x) &= (x_3 + 2)x_2x_1^2, \\
 \text{subject to } g_1(x) &= 1 - \frac{x_2^3x_3}{7,178x_1^4} \leq 0, \\
 g_2(x) &= \frac{4x_2^2 - x_1x_2}{12,566(x_2x_1^3) - x_1^4} + \frac{1}{5,108x_1^2} \\
 &\quad - 1 \leq 0, \\
 g_3(x) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\
 g_4(x) &= \frac{x_2 + x_1}{1.5} - 1 \leq 0.
 \end{aligned}
 \tag{B.9}$$

The simple limits on the design variables are  $0.05 \leq x_1 \leq 2.0$ ,  $0.25 \leq x_2 \leq 1.3$ , and  $2.0 \leq x_3 \leq 15.0$ .

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

The authors are grateful to anonymous reviewers for their valuable reviews and constructive criticism on earlier version of this paper. This work is supported by Abdul Wali Khan University, Mardan, Pakistan, Grant no. F.16-5/P&D/AWKUM/238.

## References

- [1] A. Salhi and E. S. Fraga, "Nature-inspired optimisation approaches and the new plant propagation algorithm," in *Proceedings of the International Conference on Numerical Analysis and Optimization (ICeMATH '11)*, pp. K2-1–K2-8, Yogyakarta, Indonesia, 2011.
- [2] C. M. Herrera and O. Pellmyr, *Plant Animal Interactions: An Evolutionary Approach*, John Wiley & Sons, 2009.
- [3] C. M. Herrera, "Seed dispersal by vertebrates," in *Plant-Animal Interactions: An Evolutionary Approach*, pp. 185–208, 2002.
- [4] W. G. Abrahamson and T. N. Taylor, *Plant-Animal Interactions*, McGraw Hill, 1989.
- [5] A. N. Andersen and R. W. Braithwaite, "Plant-animal interactions," in *Landscape and Vegetation Ecology of the Kakadu Region, Northern Australia*, pp. 137–154, Springer, 1996.
- [6] J. P. Bryant, "Plant-animal interactions," *Environmental Entomology*, vol. 19, no. 4, pp. 1169–1170, 1990.
- [7] B. J. Glover, *Understanding Flowers and Flowering: an Integrated Approach*, Oxford University Press, Oxford, UK, 2007.
- [8] X.-S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*, pp. 240–249, Springer, 2012.
- [9] X.-S. Yang, M. Karamanoglu, and X. He, "Multi-objective flower algorithm for optimization," in *Proceedings of the 13th Annual International Conference on Computational Science (ICCS '13)*, vol. 18, pp. 861–868, June 2013.
- [10] H. J. du Plessis, R. J. Brand, C. Glyn-Woods, and M. A. Goedhart, "Efficient genetic transformation of strawberry (*Fragaria x Ananassa* Duch.) cultivar selekta," in *III International Symposium on In Vitro Culture and Horticultural Breeding*, ISHS Acta Horticulturae 447, pp. 289–294, ISHS, 1996.
- [11] L. W. Krefting and E. I. Roe, "The role of some birds and mammals in seed germination," *Ecological Monographs*, vol. 19, no. 3, pp. 269–286, 1949.
- [12] D. G. Wenny and D. J. Levey, "Directed seed dispersal by bellbirds in a tropical cloud forest," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, no. 11, pp. 6204–6207, 1998.
- [13] J. Brownlee, *Clever Algorithms: Nature-Inspired Programming Recipes*, 2011.
- [14] X.-S. Yang, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, Beckington, UK, 2011.
- [15] M. Kazemian, Y. Ramezani, C. Lucas, and B. Moshiri, "Swarm clustering based on flowers pollination by artificial bees," *Studies in Computational Intelligence*, vol. 34, pp. 191–202, 2006.
- [16] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS '95)*, pp. 39–43, IEEE, Nagoya, Japan, October 1995.
- [17] M. Clerc, *Particle Swarm Optimization*, vol. 93, John Wiley & Sons, 2010.
- [18] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Erciyes University Press, Kayseri, Turkey, 2005.
- [19] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied Soft Computing Journal*, vol. 8, no. 1, pp. 687–697, 2008.
- [20] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of Bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.
- [21] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using Firefly Algorithm," *Computers and Structures*, vol. 89, no. 23–24, pp. 2325–2336, 2011.
- [22] E. Cuevas and M. Cienfuegos, "A new algorithm inspired in the behavior of the social-spider for constrained optimization," *Expert Systems with Applications*, vol. 41, no. 2, pp. 412–425, 2014.
- [23] M. Sulaiman, A. Salhi, B. I. Selamoglu, and O. B. Kirikchi, "A plant propagation algorithm for constrained engineering optimisation problems," *Mathematical Problems in Engineering*, vol. 2014, Article ID 627416, 10 pages, 2014.
- [24] J. L. Tellería, Á. Ramírez, and J. Pérez-Tris, "Conservation of seed-dispersing migrant birds in Mediterranean habitats: shedding light on patterns to preserve processes," *Biological Conservation*, vol. 124, no. 4, pp. 493–502, 2005.
- [25] Wikipedia Contributors, "Strawberry," 2015, <http://bit.ly/17REoNP>.
- [26] Ruth, Strawberries eaten by animals, 2012, <http://bit.ly/1zZg0jS>.
- [27] Anisognathus, "Blue Winged Mountain-Tanager eating tree strawberries at San isidro Lodge," 2012, <http://bit.ly/1we3thV>.
- [28] lifeisfull@ineedexchange.com, "Looking for nice strawberries to pick among the tiny ones, rotten ones and thistles," 2014, <https://lifeisfull.wordpress.com/page/5/>.
- [29] M. S. Kiran and M. Gündüz, "A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems," *Applied Soft Computing Journal*, vol. 13, no. 4, pp. 2188–2203, 2013.
- [30] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.
- [31] N. E. Stork and C. H. C. Lyal, "Extinction or 'co-extinction' rates?" *Nature*, vol. 366, no. 6453, p. 307, 1993.
- [32] P. Jordano, "Fruits and frugivory," in *Seeds: The Ecology of Regeneration in Plant Communities*, vol. 2, pp. 125–166, CABI, Wallingford, UK, 2000.
- [33] M. Debussche and P. Isenmann, "Bird-dispersed seed rain and seedling establishment in patchy Mediterranean vegetation," *Oikos*, vol. 69, no. 3, pp. 414–426, 1994.
- [34] D. H. Janzen, "Herbivores and the number of tree species in tropical forests," *The American Naturalist*, vol. 104, no. 940, pp. 501–528, 1970.
- [35] S. A. Levin, "Population dynamic models in heterogeneous environments," *Annual Review of Ecology and Systematics*, vol. 7, no. 1, pp. 287–310, 1976.
- [36] S. A. H. Geritz, T. J. de Jong, and P. G. L. Klinkhamer, "The efficacy of dispersal in relation to safe site area and seed production," *Oecologia*, vol. 62, no. 2, pp. 219–221, 1984.
- [37] S. A. Levin, D. Cohen, and A. Hastings, "Dispersal strategies in patchy environments," *Theoretical Population Biology*, vol. 26, no. 2, pp. 165–191, 1984.

- [38] C. K. Augspurger and S. E. Franson, "Wind dispersal of artificial fruits varying in mass, area, and morphology," *Ecology*, vol. 68, no. 1, pp. 27–42, 1987.
- [39] R. B. Cooper, *Introduction to Queueing Theory*, 1972.
- [40] J. A. Lawrence and B. A. Pasternack, *Applied Management Science*, Wiley, New York, NY, USA, 2002.
- [41] A. H. S. Ang and W. H. Tang, "Probability concepts in engineering," *Planning*, vol. 1, no. 4, pp. 1–3, 2004.
- [42] J. D. C. Little, "A proof for the queuing formula:  $L = \lambda W$ ," *Operations Research*, vol. 9, no. 3, pp. 383–387, 1961.
- [43] D. W. Thompson, *On Growth and Form*, Courier, 1942.
- [44] K. S. van Houtan, S. L. Pimm, J. M. Halley, R. O. Bierregaard Jr., and T. E. Lovejoy, "Dispersal of Amazonian birds in continuous and fragmented forest," *Ecology Letters*, vol. 10, no. 3, pp. 219–229, 2007.
- [45] A. M. Reynolds and M. A. Frye, "Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search," *PLoS ONE*, vol. 2, no. 4, article e354, 2007.
- [46] Wikipedia Contributors, Gamma function, 2015, <http://bit.ly/lw6scza>.
- [47] M. Sulaiman and A. Salhi, "A seed-based plant propagation algorithm: the feeding station model," in *Proceedings of the 5th International Conference on Metaheuristics and Nature Inspired Computing*, Marrakesh, Morocco, October 2014, <http://meta-2014.sciencesconf.org/40158>.
- [48] P. N. Suganthan, N. Hansen, J. J. Liang et al., "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," Tech. Rep., Nanyang Technological University, Singapore, 2005.
- [49] J. J. Liang, T. Runarsson, E. Mezura-Montes et al., "Problem definitions and evaluation criteria for the CEC 2006, special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, p. 8, 2006.
- [50] D. Karaboga and B. Akay, "A modified artificial bee colony (abc) algorithm for constrained optimization problems," *Applied Soft Computing Journal*, vol. 11, no. 3, pp. 3021–3031, 2011.
- [51] Q. He and L. Wang, "A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization," *Applied Mathematics and Computation*, vol. 186, no. 2, pp. 1407–1422, 2007.
- [52] H.-Y. Wang, X.-J. Ding, Q.-C. Cheng, and F.-H. Chen, "An improved isomap for visualization and classification of multiple manifolds," in *Neural Information Processing*, M. Lee, A. Hirose, Z.-G. Hou, and R. M. Kil, Eds., vol. 8227 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, Berlin, Germany, 2013.
- [53] D. B. Fogel, *System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling*, Ginn Press, 1991.
- [54] T. Bäck and H.-P. Schwefel, "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, vol. 1, no. 1, pp. 1–23, 1993.
- [55] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "Solving engineering optimization problems with the simple constrained particle swarm optimizer," *Informatica*, vol. 32, no. 3, pp. 319–326, 2008.
- [56] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, no. 4, pp. 330–343, 2010.
- [57] J. Golinski, "An adaptive optimization system applied to machine synthesis," *Mechanism and Machine Theory*, vol. 8, no. 4, pp. 419–436, 1973.
- [58] J. S. Arora, *Introduction to Optimum Design*, Academic Press, 2004.
- [59] A. D. Belegundu and J. S. Arora, "A study of mathematical programming methods for structural optimization. I. Theory," *International Journal for Numerical Methods in Engineering*, vol. 21, no. 9, pp. 1583–1599, 1985.

