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

Muhammad Sulaiman¹,² and Abdellah Salhi¹

¹Department of Mathematical Sciences, University of Essex, Colchester CO4 3SQ, UK
²Department of Mathematics, Abdul Wali Khan University, Mardan, Khyber Pakhtunkhwa, Pakistan

Correspondence should be addressed to Muhammad Sulaiman; sulaiman513@yahoo.co.uk

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

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.
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 $X^t$ be the random variable representing the number of arrivals per unit of time $t$. Then, the probability of $k$ arrivals over $t$ is

$$P(X^t = k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!},$$  \hspace{1cm} (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:

$$S(t) = \mu e^{-\mu t},$$  \hspace{1cm} (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

$$A = A_q + \frac{\lambda}{\mu}. \hspace{1cm} (3)$$

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:

Maximize $A_q = A - \frac{\lambda}{\mu}$, \hspace{1cm} (4)

subject to $g_1(\lambda, \mu) = \lambda < \mu + 1$, 

$\lambda > 0$, \hspace{0.5cm} $\mu > 0$,

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

$$L(s) \sim |s|^{-1-\beta},$$  \hspace{1cm} (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]

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

This implies that

$$\lim_{s \to \infty} L(s, y, \mu) = \sqrt{\frac{\gamma^2}{2\pi}} \left( \frac{1}{s} \right)^{3/2}. \hspace{1cm} (7)$$

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

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

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

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

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 $k = 1, 2, \ldots, A$ 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 Poiss($\lambda$) $< 0.05$ means that
Probabilities, Poiss(λ)

Arrival rate of agents per unit time

0.25
0.2
0.15
0.1
0.05
0
0 2 4 6 8 10

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:

\[
x^*_{i,j} = \begin{cases} 
    x_{i,j} + \xi_j (x_{i,j} - x_{i,l}) & \text{if } PR \leq 0.8; \ j = 1,2,\ldots,n; \\
    x_{i,l} & \text{Otherwise,} \\
  \end{cases} 
\]

(10)

where PR denotes the rate of dispersion of the seeds locally, around SP; \(x^*_{i,j}\) and \(x_{i,j}\) 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 \(\text{Pois}(\lambda) \geq 0.05\) then global dispersion of seeds becomes more prominent. This is implemented by using the following equation:

\[
x^*_{i,j} = \begin{cases} 
    x_{i,j} + L_j (x_{i,j} - \theta_j) & \text{if } PR \leq 0.8, \ \theta_j \in [a_j, b_j], \\
    x_{i,j} & \text{Otherwise,} \\
  \end{cases} 
\]

(11)

where \(L_j\) 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 \(\text{pop}_{\text{best}}\). 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 \(\text{pop}_{\text{best}}\).

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

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, \(\text{Pois}(\lambda)\), 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 = \{X_i | i = 1, 2, \ldots, NP\}, using (12) and collect the best solutions from each trial run, in pop_best.
(5) Evaluate the population pop.
(6) end if
(7) while r > NP do
(8) Use updated population pop_best.
(9) end while
(10) while (the stopping criteria is not satisfied) do
(11) for i = 1 to NP do
(12) if Poiss(\lambda_i) ≥ 0.05, then, ⊳ (Global or local seed dispersion)
(13) for j = 1 to n do
(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, ⊳ (PR = Perturbation Rate)
(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

\[ x_{i,j} = a_j + (b_j - a_j) \eta_j, \quad j = 1, \ldots, n, \tag{12} \]

where \( x_{i,j} \in [a_j, b_j] \) is the j-th coordinate of solution \( X_i \), 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 \( j = 1, \ldots, n \), 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) (=) 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.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO [16, 29]</td>
<td>$M = 100$, $G_{max} = \frac{(\text{Dimension} \times 20,000)}{M}$, $c_1 = 2$, $c_2 = 2$, $W = \frac{(G_{max} - \text{iteration index})}{G_{max}}$</td>
</tr>
<tr>
<td>ABC [18, 29]</td>
<td>$SN = 100$, $MCN = \frac{(\text{Dimension} \times 20,000)}{SN}$, $c_1 = 2$, $c_2 = 2$, $W = \frac{\text{Iteration number} - \text{iteration index}}{\text{Iteration number}}$</td>
</tr>
<tr>
<td>HPA [29]</td>
<td>Agents = 100, $G_{max} = \frac{(\text{Dimension} \times 20,000)}{\text{Agents}}$, $c_1 = 2$, $c_2 = 2$, $W = \frac{\text{Iteration number} - \text{iteration index}}{\text{Iteration number}}$</td>
</tr>
<tr>
<td>SbPPA [47]</td>
<td>NP = 10, $G_{max} = \frac{(\text{Dimension} \times 20,000)}{\text{NP}}$, $c_1 = 2$, $c_2 = 2$, $W = \frac{\text{Iteration number} - \text{iteration index}}{\text{Iteration number}}$</td>
</tr>
</tbody>
</table>
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. The extent of seeds dispersed and hence the effectiveness of the global optimum is at $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$, $f(x^*) = -15$.

**B.2. CP2.** Consider the following:

$$
\text{Min } f(x) = 5.3578547x_2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141,
$$

subject to

$$
g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0,
$$

$$
g_2(x) = 85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0,
$$

$$
g_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_2 - 110 \leq 0,
$$

$$
g_4(x) = 80.51249 - 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_2 + 90 \leq 0,
$$

$$
g_5(x) = 9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 - 25 \leq 0,
$$

$$
g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0,
$$

B.1. CPI. Consider the following:

$$
\text{Min } f(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0,
$$

$$
g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0,
$$

$$
g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0,
$$

$$
g_4(x) = -8x_1 + x_{10} \leq 0,
$$

$$
g_5(x) = -8x_2 + x_{11} \leq 0,
$$

$$
g_6(x) = -8x_3 + x_{12} \leq 0,
$$

$$
g_7(x) = -2x_4 - x_5 + x_{10} \leq 0,
$$

$$
g_8(x) = -2x_5 - x_7 + x_{11} \leq 0,
$$

$$
g_9(x) = -2x_8 - x_9 + x_{12} \leq 0,
$$

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

**Appendices**

**A. Unconstrained Global Optimization Problems**

See Tables 8 and 9.

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

B.1. CPI. Consider the following:

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

subject to

$$
g_1(x) = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0,
$$

$$
g_2(x) = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0,
$$

$$
g_3(x) = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0,
$$

$$
g_4(x) = -8x_1 + x_{10} \leq 0,
$$

$$
g_5(x) = -8x_2 + x_{11} \leq 0,
$$

$$
g_6(x) = -8x_3 + x_{12} \leq 0,
$$

$$
g_7(x) = -2x_4 - x_5 + x_{10} \leq 0,
$$

$$
g_8(x) = -2x_5 - x_7 + x_{11} \leq 0,
$$

$$
g_9(x) = -2x_8 - x_9 + x_{12} \leq 0,
$$

(B.2)
Table 3: Parameters used for each algorithm for solving constrained optimization problems. All experiments are repeated 30 times.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>M</th>
<th>SN</th>
<th>Fireflies</th>
<th>N</th>
<th>NP</th>
<th>G_{max}</th>
<th>MCN</th>
<th>Iteration number</th>
<th>q</th>
<th>PF</th>
<th>PR</th>
<th>( \alpha )</th>
<th>Pois(\lambda)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO [51]</td>
<td>250</td>
<td>40</td>
<td>25</td>
<td>50</td>
<td>10</td>
<td>300</td>
<td>6000</td>
<td>800</td>
<td>1.5</td>
<td>0.7</td>
<td>0.8</td>
<td>0.001</td>
<td>0.05</td>
</tr>
<tr>
<td>SSO-C [22]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Function</th>
<th>Dim</th>
<th>Algorithm</th>
<th>Best</th>
<th>Worst</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1 )</td>
<td>4</td>
<td>ABC</td>
<td>(+) 0.0129</td>
<td>(+) 0.6106</td>
<td>(+) 0.1157</td>
<td>(+) 0.111</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>(-) 6.8991E-08</td>
<td>(+) 0.0045</td>
<td>(+) 0.001</td>
<td>(+) 0.0013</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPA</td>
<td>(+) 2.0323E-06</td>
<td>(+) 0.0456</td>
<td>(+) 0.009</td>
<td>(+) 0.0122</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA</td>
<td>1.08E-07</td>
<td>7.05E-06</td>
<td>3.05E-06</td>
<td>3.14E-06</td>
</tr>
<tr>
<td>( f_2 )</td>
<td>2</td>
<td>ABC</td>
<td>(+) 1.2452E-08</td>
<td>(+) 8.4415E-06</td>
<td>(+) 1.8978E-06</td>
<td>(+) 1.8537E-06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPA</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( f_3 )</td>
<td>2</td>
<td>ABC</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPA</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
<td>(-) 1.03163</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA</td>
<td>-1.031628</td>
<td>-1.031628</td>
<td>-1.031628</td>
<td>0</td>
</tr>
<tr>
<td>( f_4 )</td>
<td>6</td>
<td>ABC</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPA</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
<td>(-) 50.0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA</td>
<td>-50.0000</td>
<td>-50.0000</td>
<td>-50.0000</td>
<td>5.88E-09</td>
</tr>
<tr>
<td>( f_5 )</td>
<td>10</td>
<td>ABC</td>
<td>(+) 2.6055E-16</td>
<td>(+) 5.5392E-16</td>
<td>(+) 4.7403E-16</td>
<td>(+) 9.2969E-17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>(-) 210.0000</td>
<td>(-) 210.0000</td>
<td>(-) 210.0000</td>
<td>(-) 210.0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPA</td>
<td>(-) 210.0000</td>
<td>(-) 210.0000</td>
<td>(-) 210.0000</td>
<td>(+) 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA</td>
<td>-210.0000</td>
<td>-210.0000</td>
<td>-210.0000</td>
<td>4.86E-06</td>
</tr>
<tr>
<td>( f_6 )</td>
<td>30</td>
<td>ABC</td>
<td>(+) 2.9407E-16</td>
<td>(+) 5.5463E-16</td>
<td>(+) 4.8909E-16</td>
<td>(+) 9.0442E-17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPA</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
<td>(-) 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( f_7 )</td>
<td>30</td>
<td>ABC</td>
<td>(+) 2.9310E-14</td>
<td>(+) 3.9968E-14</td>
<td>(+) 3.2744E-14</td>
<td>(+) 2.5094E-15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PSO</td>
<td>(-) 7.9936E-15</td>
<td>(-) 1.5099E-14</td>
<td>(-) 8.5857E-15</td>
<td>(+) 1.8536E-15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>HPA</td>
<td>(-) 7.9936E-15</td>
<td>(-) 1.5099E-14</td>
<td>(+) 1.1309E-14</td>
<td>(+) 3.54E-15</td>
</tr>
</tbody>
</table>
### Table 5: Results obtained by SbPPA, CEP, and FEP. All problems in this table are unconstrained \[30\].

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

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

<table>
<thead>
<tr>
<th>Function number</th>
<th>Algorithm</th>
<th>Maximum generations</th>
<th>Mean</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>(f_{19})</td>
<td>CEP</td>
<td></td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>100</td>
<td>3.02</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>3</td>
<td>3.05E-15</td>
</tr>
<tr>
<td>(f_{20})</td>
<td>CEP</td>
<td>100</td>
<td>-3.86E+00</td>
<td>1.40E-02</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>100</td>
<td>-3.86E+00</td>
<td>1.40E-05</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>-3.86E+00</td>
<td>2.75E-15</td>
</tr>
<tr>
<td>(f_{21})</td>
<td>CEP</td>
<td>200</td>
<td>-3.28E+00</td>
<td>5.80E-02</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>200</td>
<td>-3.27E+00</td>
<td>5.90E-02</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>-3.32E+00</td>
<td>2.91E-14</td>
</tr>
<tr>
<td>(f_{22})</td>
<td>CEP</td>
<td>100</td>
<td>-6.86E+00</td>
<td>2.67E+00</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>100</td>
<td>-5.52E+00</td>
<td>1.59E+00</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>-1.02E+01</td>
<td>4.30E-09</td>
</tr>
<tr>
<td>(f_{23})</td>
<td>CEP</td>
<td>100</td>
<td>-8.27</td>
<td>2.95</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>100</td>
<td>-5.52</td>
<td>2.12</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>-1.04E+01</td>
<td>7.73E-09</td>
</tr>
<tr>
<td>(f_{24})</td>
<td>CEP</td>
<td>100</td>
<td>-9.1</td>
<td>2.92</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>100</td>
<td>-6.57</td>
<td>3.14</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>-1.05E+01</td>
<td>1.03E-07</td>
</tr>
<tr>
<td>(f_{25})</td>
<td>CEP</td>
<td>100</td>
<td>1.66</td>
<td>1.19</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>100</td>
<td>1.22</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>9.98E+01</td>
<td>1.13E+16</td>
</tr>
<tr>
<td>(f_{26})</td>
<td>CEP</td>
<td>4000</td>
<td>4.70E-04</td>
<td>3.00E-04</td>
</tr>
<tr>
<td></td>
<td>FEP</td>
<td>4000</td>
<td>5.00E-04</td>
<td>3.20E-04</td>
</tr>
<tr>
<td></td>
<td>SbPPA</td>
<td></td>
<td>3.07E-04</td>
<td>6.80E-15</td>
</tr>
</tbody>
</table>
Table 7: Results obtained by SbPPA, PSO, ABC, FF, and SSO-C. All problems in this table are standard constrained optimization problems.

<table>
<thead>
<tr>
<th>Fun. name</th>
<th>Optimal</th>
<th>Algorithm</th>
<th>Best</th>
<th>Mean</th>
<th>Worst</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP1</td>
<td>−15</td>
<td>PSO (≈) −15</td>
<td>(≈) −15</td>
<td>(≈) −15</td>
<td>(−) −15</td>
<td>(−) 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABC (≈) −15</td>
<td>(≈) −15</td>
<td>(≈) −15</td>
<td>(−) −15</td>
<td>(−) 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF (+) 14.999</td>
<td>(+) 14.988</td>
<td>(+) 14.798</td>
<td>(+) 6.40E−07</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSO-C (≈) −15</td>
<td>(≈) −15</td>
<td>(≈) −15</td>
<td>(−) −15</td>
<td>(−) 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA −15</td>
<td>−15</td>
<td>−15</td>
<td>−15</td>
<td>1.95E−15</td>
</tr>
<tr>
<td>CP2</td>
<td>−30665.539</td>
<td>PSO (+) −6.96E+03</td>
<td>(+) −6958.37</td>
<td>(+) −6942.09</td>
<td>(+) 6.70E−02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABC (−) −6961.81</td>
<td>(+) −6958.02</td>
<td>(+) −6955.34</td>
<td>(−) 2.10E−02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF (+) −6959.99</td>
<td>(+) −6.95E+03</td>
<td>(+) −6947.63</td>
<td>(−) 3.80E−02</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSO-C (−) −6961.81</td>
<td>(+) −6961.01</td>
<td>(+) −6960.92</td>
<td>(−) 1.10E−03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA −6961.5</td>
<td>−6961.38</td>
<td>−6961.45</td>
<td>0.043637</td>
<td></td>
</tr>
<tr>
<td>CP3</td>
<td>−6961.814</td>
<td>PSO (−) 24.327</td>
<td>(+) 2.45E+01</td>
<td>(+) 24.843</td>
<td>(+) 1.32E−01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABC (+) 24.48</td>
<td>(+) 2.66E+01</td>
<td>(+) 28.4</td>
<td>(−) 1.14</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF (−) 23.97</td>
<td>(+) 28.54</td>
<td>(+) 30.14</td>
<td>(−) 2.25</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSO-C (−) 24.306</td>
<td>(−) 24.306</td>
<td>(−) 24.306</td>
<td>(−) 4.95E−05</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA 24.3442</td>
<td>24.37536</td>
<td>24.37021</td>
<td>0.012632</td>
<td></td>
</tr>
<tr>
<td>CP4</td>
<td>24.306</td>
<td>PSO (−) 0.7499</td>
<td>(+) −0.749</td>
<td>(+) −0.7486</td>
<td>(−) 1.20E−03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABC (−) 0.7499</td>
<td>(+) −0.7495</td>
<td>(+) −0.749</td>
<td>(−) 1.67E−03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF (+) 0.7497</td>
<td>(+) −0.7491</td>
<td>(+) −0.7479</td>
<td>(−) 1.50E−03</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSO-C (−) 0.7499</td>
<td>(≈) −0.7499</td>
<td>(≈) −0.7499</td>
<td>(−) 4.10E−09</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA 0.7499</td>
<td>0.749901</td>
<td>0.7499</td>
<td>1.66E−07</td>
<td></td>
</tr>
<tr>
<td>Spring Design Problem</td>
<td>Not known</td>
<td>PSO (+) 1.846408</td>
<td>(+) 2.011146</td>
<td>(+) 2.237839</td>
<td>(+) 0.108513</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABC (+) 1.798173</td>
<td>(+) 2.167358</td>
<td>(+) 2.887044</td>
<td>(+) 0.254266</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF (+) 1.724854</td>
<td>(+) 2.197401</td>
<td>(+) 2.931001</td>
<td>(+) 0.195264</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSO-C (+) 1.724852</td>
<td>(+) 1.746462</td>
<td>(+) 1.799332</td>
<td>(+) 0.02573</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA 1.724852</td>
<td>1.724852</td>
<td>1.724852</td>
<td>4.06E−08</td>
<td></td>
</tr>
<tr>
<td>Welded beam design problem</td>
<td>Not known</td>
<td>PSO (+) 3044.453</td>
<td>(+) 3079.262</td>
<td>(+) 3177.515</td>
<td>(+) 26.21731</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABC (+) 2996.116</td>
<td>(+) 2998.063</td>
<td>(+) 3002.756</td>
<td>(+) 6.354562</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF (+) 2996.947</td>
<td>(+) 3000.005</td>
<td>(+) 3005.836</td>
<td>(+) 8.356535</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSO-C (+) 2996.113</td>
<td>(+) 2996.113</td>
<td>(+) 2996.113</td>
<td>(+) 1.34E−12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA 2996.114</td>
<td>2996.114</td>
<td>2996.114</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Speed reducer design optimization</td>
<td>Not known</td>
<td>PSO (+) 3044.453</td>
<td>(+) 3079.262</td>
<td>(+) 3177.515</td>
<td>(+) 26.21731</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ABC (+) 2996.116</td>
<td>(+) 2998.063</td>
<td>(+) 3002.756</td>
<td>(+) 6.354562</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>FF (+) 2996.947</td>
<td>(+) 3000.005</td>
<td>(+) 3005.836</td>
<td>(+) 8.356535</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SSO-C (+) 2996.113</td>
<td>(+) 2996.113</td>
<td>(+) 2996.113</td>
<td>(+) 1.34E−12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SbPPA 2996.114</td>
<td>2996.114</td>
<td>2996.114</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Table 8: Unconstrained global optimization problems (Set-1) used in our experiments.

<table>
<thead>
<tr>
<th>Fun.</th>
<th>Fun. name</th>
<th>D</th>
<th>C</th>
<th>Range</th>
<th>Min</th>
<th>Formulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>f₁</td>
<td>Colville</td>
<td>4</td>
<td>UN</td>
<td>[-10 10]D</td>
<td>0</td>
<td>( f(x) = 100 (x_1^2 - x_2) + (x_1 - 1)^2 + (x_2 - 1)^2 + 90 (x_1 - x_2)^2 + 10.1 ((x_2 - 1)^2 + (x_2 - 1)^2) + 19.8 (x_3 (x_4 - 1)) (x_4 - 1) )</td>
</tr>
<tr>
<td>f₂</td>
<td>Matyas</td>
<td>2</td>
<td>UN</td>
<td>[-10 10]D</td>
<td>0</td>
<td>( f(x) = 0.26 x_1^2 + x_2^2 - 0.48 x_1 x_2 )</td>
</tr>
<tr>
<td>f₃</td>
<td>Schaffer</td>
<td>2</td>
<td>MN</td>
<td>[-100 100]D</td>
<td>0</td>
<td>( f(x) = 0.5 + \sin^2 \left( \sqrt{\sum_{i=1}^{n} x_i^2} \right) - 0.5 )</td>
</tr>
<tr>
<td>f₄</td>
<td>Six Hump Camel Back</td>
<td>2</td>
<td>MN</td>
<td>[-5 5]D</td>
<td>1.03163</td>
<td>( f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_2^6 + x_2 - 4x_2^2 + 4x_4^2 )</td>
</tr>
<tr>
<td>f₅</td>
<td>Trid6</td>
<td>6</td>
<td>UN</td>
<td>[-36 36]D</td>
<td>50</td>
<td>( f(x) = \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i x_{i+1} )</td>
</tr>
<tr>
<td>f₆</td>
<td>Trid10</td>
<td>10</td>
<td>UN</td>
<td>[-100 100]D</td>
<td>210</td>
<td>( f(x) = \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i x_{i+1} )</td>
</tr>
<tr>
<td>f₇</td>
<td>Sphere</td>
<td>30</td>
<td>US</td>
<td>[-100 100]D</td>
<td>0</td>
<td>( f(x) = \sum_{i=1}^{n} x_i^2 )</td>
</tr>
<tr>
<td>f₈</td>
<td>SumSquares</td>
<td>30</td>
<td>US</td>
<td>[-10 10]D</td>
<td>0</td>
<td>( f(x) = \sum_{i=1}^{n} x_i^2 )</td>
</tr>
<tr>
<td>f₉</td>
<td>Griewank</td>
<td>30</td>
<td>MN</td>
<td>[-600 600]D</td>
<td>0</td>
<td>( f(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 )</td>
</tr>
<tr>
<td>f₁₀</td>
<td>Ackley</td>
<td>30</td>
<td>MN</td>
<td>[-32 32]D</td>
<td>0</td>
<td>( f(x) = -20 \exp \left( -\frac{1}{4} \sum_{i=1}^{n} x_i^2 \right) - \exp \left( \frac{-1}{4} \sum_{i=1}^{n} \cos (2 \pi x_i) \right) + 20 + e )</td>
</tr>
</tbody>
</table>

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 \( x^* = (78, 33, 29.99525625682, 45, 36.775812905788) \), where \( f(x^*) = 30665.539 \). Constraints \( g_1 \) and \( g_6 \) are active.

B.3. CP3. Consider the following:

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

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:

\[
\text{Min} \quad f(x) = x_1^2 + x_2^2 + x_1 x_2 - 14 x_1 - 16 x_2 \\
\quad \quad \quad + (x_1 - 10)^2 + 4 (x_1 - 5)^2 + (x_5 - 3)^2 \\
\quad \quad \quad + 2 (x_6 - 1)^2 + 5 x_7^2 + (x_8 - 11)^2 \\
\quad \quad \quad + 2 (x_9 - 10)^2 + (x_{10} - 7)^2 + 45, \\
\text{subject to} \quad g_1(x) = -105 + 4 x_1 + 5 x_2 - 3 x_7 + 9 x_9 \leq 0, \\
g_2(x) = 10 x_1 - 8 x_2 - 17 x_7 + 2 x_8 \leq 0, \\
(B.4)
\]

where \( -10 \leq x_i \leq 10 \) (\( i = 1, \ldots, 10 \)). 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:

\[
\text{Min} \quad f(x) = x_1^2 + (x_2 - 1)^2, \\
\text{subject to} \quad g_1(x) = x_2 - x_1^2 = 0, \\
(B.5)
\]

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].

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

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 $x^T = (w, L, d, h)$:

Minimise $f(x) = 1.10471w^2L + 0.04811dh (14.0 + L)$, 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
\[
\sigma(x) = \frac{504,000}{hd^2}, \quad D = \frac{1}{2}\sqrt{L^2 + (w + d)^2},
\]
\[
Q = 6000\left(14 + \frac{L}{2}\right), \quad \delta = \frac{65,856}{30,000hd^3},
\]
\[
J = \sqrt{2\omega L}\left(\frac{L^2}{6} + \frac{(w + d)^2}{2}\right), \quad \alpha = \frac{6000}{\sqrt{2\omega L}},
\]
\[
\beta = \frac{QD}{J}, \quad P = 0.61423 \times 10^6 \frac{dh^5}{6} \left(1 - \frac{\sqrt{30/48}}{28}\right),
\]
\[
\tau(x) = \sqrt{\alpha^2 + \frac{\alpha\beta L}{D} + \beta^2}.
\]

(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:

Minimise
\[
f(x) = 0.7854x_1x_2^2 \cdot \left(3.3333x_2^3 + 14.9334x_3x_4\right)
- 1.508x_1(x_6^2 + x_7^2)
+ 7.4777(x_3^2 + x_4^2)
+ 0.7854(x_4x_6^2 + x_5x_7^2),
\]

subject to
\[
g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0,
\]
\[
g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0,
\]
\[
g_3(x) = \frac{1.93x_4^3}{x_2x_3x_4^2} - 1 \leq 0,
\]
\[
g_4(x) = \frac{1.93x_5^3}{x_2x_3x_5^2} - 1 \leq 0,
\]
\[
g_5(x) = \frac{1.0}{110x_4^2} \sqrt{\left(\frac{745.0}{x_2x_3}\right)^4 + 16.9 \times 10^6}
- 1 \leq 0,
\]

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 string, 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:

Minimise
\[
f(x) = (x_3 + 2)x_2x_1^2,
\]

subject to
\[
g_1(x) = 1 - \frac{x_1^2x_3}{7,178x_1^2} \leq 0,
\]
\[
g_2(x) = \frac{4x_2^2 - x_1x_3}{12,566(x_2x_1^2)} - x_1^4 + \frac{1}{5,108x_1^2} \leq 0,
\]
\[
g_3(x) = 1 - \frac{140.45x_1}{x_2x_3} \leq 0,
\]
\[
g_4(x) = \frac{x_3 + x_1}{1.5} - 1 \leq 0.
\]

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


Submit your manuscripts at
http://www.hindawi.com