ISSN 1 746-7233, England, UK
World Journal of Modelling and Simulation
Vol. 12 (2016) No. 3, pp. 189-203

Use of membrane algorithms for solving constrained engineering design problems*

Garima Singh[1] , Kusum Deep[2†]

Department of Mathematics Indian Institute of Technology Roorkee Roorkee - 247667, India

(*Received August 30 2015, Accepted December 30, 2015*)

Abstract. The concept of membrane algorithms is a recently introduced paradigm which uses the framework of P systems. In this paper membrane algorithms using the rules of Particle Swarm Optimization defined within the parallel framework of cell-like P systems are described so as to be used to solve three well-known constrained engineering design problems. Two approaches are considered for handling constraints namely Deb's constraint handling mechanism and Brute Force Method. The comparative analysis of results is conducted to demonstrate the performance of PSO based membrane algorithms in solving the constrained engineering design problems with varying constraint handling techniques. Also, the efficiency of PSO based membrane algorithms is demonstrated by coupling it with brute force method of constraint handling technique and comparing the results with the same algorithms when coupled with well-known Deb's constraint handling technique.

Keywords: membrane algorithm, p system, engineering design problem, constrained optimization problems, particle swarm optimization

1 Introduction

In Engineering design problems, the optimal parameters are evaluated with an aim to attain maximum or minimum objective function values. However, the obtained parameters must satisfy specified requirements or the set of conditions known as constraints. Due to the imposed constraints such optimization problems are called constrained optimization problems. High non-linearity and complex constraints, involved in these design problems, make them difficult to be solved by enumerative strategies or calculus based methods.

Various numerical Optimization methods like Simplex method[16], Lagrange multiplier method[8], Genetic adaptive search[4], Branch and bound technique[17], etc. have been previously used to solve such engineering design problems. However, such methods can only be applied to simpler models of the problems as the researchers found it difficult to solve the complex real-world problems using them. It is due to this failure of the mathematical optimization techniques, that the artificial intelligence techniques found their use in such problems. The techniques used include Genetic Algorithm (GA)[3], Particle Swarm Optimization (PSO)[12, 13] and Co-evolutionary PSO[6], Ant Colony Optimization (ACO)[9], Artificial Bee Colony Optimization (ABC)[1, 2], Harmony Search (HS)[11, 14], Glowworm Swarm Optimization[22], Cuckoo Search[7] etc. In [20], authors have used backtracking search algorithm and coupled it with three different constraint handling techniques to obtain the results.

In 2013, Xiao et al.[19] proposed a membrane algorithm named BIAMC, i.e., Bio-inspired algorithm based on membrane computing, to solve engineering design problem. In the algorithm, the rule of PSO based

* The first author would like to express her appreciation to Ministry of Human Resource Development, Government of India for funding this research work under Grant No. MHRD02-23-200-304. The authors are also thankful to reviewers for their valuable comments.

† Corresponding author. *E-mail address*: kusumdfma@iitr,ac.in, kusumdeep@gmail.com

on Gaussian distribution was used within the parallel and distributed framework of cell-like P system. However, the used technique was not stated in the paper. In 2013 itself, Zhang et al.[21] also presented a membrane algorithm using five different Differential Evolution variants hybridized with tissue P system and applied it to twenty-one benchmark manufacturing optimization problems including the design problems solved in [19]. However, the obtained results were not compared with any of the state-of-art membrane algorithms in any of these papers.

In this paper, the same design problems along with their different versions are solved using all the existing PSO based membrane algorithms described in [18]. All these membrane algorithms solve these problems using two different constraint handling techniques, i.e., using Brute force method and Deb's Constraint handling technique [5]. The obtained results are used to compare the behavior of algorithms in optimizing constrained problems besides proving their efficiency in solving such problems.

In the present paper, Section 2 gives the description of the studied engineering design problem. Section 3 briefly describes the used algorithms and constraint handling techniques. The numerical results are presented in Section 4. Section 5 concludes the objective and the outcome of the analysis drawn.

2 Engineering design problems

In this paper, three different structural design problems are solved. All these are constrained optimization problems. A standard constrained optimization problem is given as:

$$\min\ f(x)$$
$$s.t. \begin{cases} g_i(x) \leq 0, i = 1, 2, \cdots, l, & \text{(inequality constraints)} \\ h_j(x) = 0, j = 1, 2, \cdots, m, & \text{(equality constraints)}, \end{cases}$$

where, $x_{L_k} \leq x_k \leq x_{U_k}$, $k = 1, 2, \cdots, n$ are bounded variables.

x_k here are the decision variables defining the objective function $f(x)$. The feasible solutions of the above constrained problem are all the sets of decision variables defined within their respective bounds, i.e., and , and satisfying all the equality and inequality constraints. The feasible solution with the minimum objective function value gives the optimal solution of the problem.

There are several approaches to handle the constraints of the problem such as:

- Penalty function approaches
- Approaches based on searching of feasible solution or preserving feasibility of solutions
- Hybrid methods

The approach used in the present work is the searching of feasible solutions and finding the optimal solution among them.

The structural problems studied in the paper are well known welded beam, pressure vessel and tension/compression string design problems. A brief description of the problems is given below:

Welded Beam Design Problem: The objective of the problem to minimize the expense involved in the fabrication of the welded beam such that constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ) and the side constraints are satisfied. The design variables involved in the problem are length of the welded joint (l), width of the beam (t), thickness of the weld (h) and thickness of the beam (b). The diagrammatic representation of the problem is shown in Fig. 1 below.

In the literature two different mathematical formulation of welded beam design problem is presented. First version given in [16] is as follows:

Version 1:

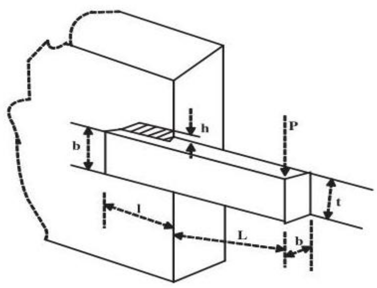

Fig. 1: Design of Welded Beam

$$\min\ f(X) = 1.10471h^2 l + 0.04811tb(14 + l)$$

$$s,t,\ \begin{cases} g_1(X) = \tau(X) - \tau max \le 0 \\ g_2(X) = \sigma(X) - \sigma max \le 0 \\ g_3(X) = h - b \le 0 \\ g_4(X) = 0.125 - h \le 0 \\ g_5(X) = \delta(X) - 0.25 \le 0 \\ g_6(X) = P - Pc(X) \le 0. \\ 0.1 \le h \le 2, \\ 0.1 \le l \le 10, \\ 0.1 \le t \le 10, \\ 0.1 \le b \le 2, \end{cases}$$

here, $X = (h, l, t, b)$ is the decision variable. τ is the shear stress in the weld while the maximum allowed shear stress for the weld is given by $\tau\max = 13600$ psi. The maximum bending stress allowed for the beam material is $\sigma max = 30000$ psi and P is the load ($= 6000lb$). The shear stress τ is given by

$$\tau(X) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{l}{2R}\right) + \tau_2^2},$$

where, the primary stress $\tau_1 = \frac{P}{hl\sqrt{2}}$ and secondary stress $\tau_2 = \frac{MR}{J}$. Here , $M = P\left(L + \frac{l}{2}\right)$ is called moment and $J(X) = 2\left\{\frac{lh}{\sqrt{2}}\left[\frac{l^2}{12} + \left(\frac{h+t}{2}\right)^2\right]\right\}$ is called polar moment of inertia. The other terms are defined as:

$$R = \sqrt{\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2}; \sigma(X) = \frac{6PL}{bt^2}; \delta(X) = \frac{4PL^3}{Ebt^3}; P_c(X) = \frac{4.013\sqrt{\frac{EGt^2b^6}{36}}}{L^2}\left(1 - \frac{t}{2L}\sqrt{\frac{E}{4G}}\right),$$

$$G = 12 \times 10^6 psi, E = 30 \times 10^6 psi, P = 6000lb, L = 14in.$$

Version 2:

In second version, which can be found in [3, 6, 11], another constraint given as

$$g_7(X) = 0.10471h^2 + 0.04811tb(14 + l) - 5 \le 0.$$

is included besides the change in deflection (σ), buckling load (P_c) and polar moment of inertia $J(X)$. The new terms are now given as:

$$\delta(X) = \frac{6PL^3}{Ebt^3}; P_c(X) = \frac{4.013\sqrt{\frac{t^2b^6}{36}}}{L^2}\left(1 - \frac{t}{2L}\sqrt{\frac{E}{4G}}\right) \quad and \quad J(X) = 2\left\{lh\sqrt{2}\left[\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2\right]\right\},$$

Pressure Vessel Design Problem: A pressure vessel is made up of a cylindrical vessel having hemispherical caps at both its end. The cylinder is formed by welding two halves of a shell made out of rolled steel

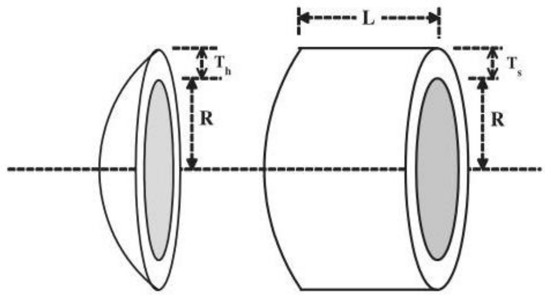

Fig. 2: Design of pressure vessel

plates. The objective of this problem, hence, is to minimize the cost of designing such pressure vessel, where the cost include the cost of material, forming and welding. The diagrammatic representation of design of the pressure vessel is given in Fig. 2 below.

In the diagram, T_s is the thickness of pressure vessel, T_h is the thickness of vessel head, R = radius of the vessel and L is the length of vessel without including its head. These four variables form the decision variable of the optimization problem whose mathematical formulation is given as:

$$\min\ f(X) = 0.6224T_sRL + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_s^2R$$
$$s.t. \begin{cases} g_1(X) = -T_s + 0.0193R \leq 0 \\ g_2(X) = -T_h + 0.00954R \leq 0 \\ g_3(X) = -\pi R^2 L - \frac{4}{3}\pi R^3 + 1296000 \leq 0 \\ g_4(X) = L - 240 \leq 0 \end{cases}$$

There exist two different version of this problem solved in literature on the basis of the different variable range used in which the problem has been solved. In [3, 4, 8] the variable region used is given as:

Version 1: $0.0625 \leq T_s, T_h \leq 99 \times 0.0625$; $10 \leq R, L \leq 200$.

However, in [1], the different variable range has been used given as: Version 2: $0.0625 \leq T_s, T_h \leq 99 \times 0.0625$; $10 \leq R \leq 200$, $10 \leq L \leq 240$.

Tension/Compression String Design Problem: The objective of this problem is to minimize the weight of a tension/compression spring such that constraints on shear stress, minimum deflection, surge frequency, outer diameter limits are satisfied along with the bounds on the decision variables. Fig. 3 below shows the design of tension/compression spring.

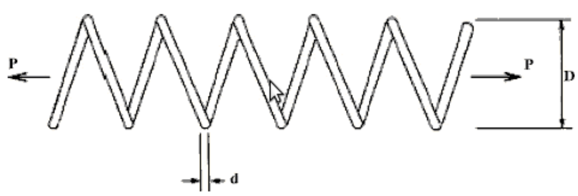

Fig. 3: Design of tension/compression string

Here, the objective function or the weight to be minimized depends on the mean coil diameter (d), the wire diameter (D) and the number of active coil (P). Hence, the decision variable of the problem is. The mathematical form of the problem is given as:

$$\min\ f(x) = (N+2)Dd^2$$

$$s.t.\ \begin{cases} g_1(X) = 1 - \frac{D^3N}{71785d^4} \le 0 \\ g_2(X) = \frac{4D^2-dD}{12566(Dd^3-d^4)} + \frac{1}{5108d^2} - 1 \le 0 \\ g_3(X) = 1 - \frac{140.45d}{D^2N} \le 0 \\ g_4(X) = \frac{d+D}{1.5} - 1 \le 0, \\ 0.05 \le d \le 2, \\ 0.25 \le D \le 1.3, \\ 2 \le N \le 15. \end{cases}$$

3 Preliminaries

3.1 PSO based membrane algorithms

Membrane algorithms are parallel and distributed computing algorithms based on living cell/tissues functioning. In these algorithms computing model, called P system, which has a defined membrane structure, either of hierarchical or single membrane connected through different channels, are used. These models were introduced by Paun in 1998[15]. These membrane structures have certain pre-defined set of rules, for both evaluation as well as communication, to carry out the functioning over certain set of variables which can be either real, integer, Boolean or strings. Different P systems are designed based on differences in the membrane structure used, set of rules performed, way in which communication is carried out and the type of objects on which process is performed.

However, if the evolution rules used within the membranes of P systems are of different types of Particle Swarm Optimization (PSO), the algorithms are called PSO based membrane algorithm. PSO is a concept proposed by Kennedy and Eberhart[10] inspired by swarm behavior of birds. The general basic equation for PSO is given as:

$$v_{ij}(t+1) = v_{ij}(t) + \varphi_1 \times r^t_{1(ij)} \times (pbest_{ij}(t) - x_{ij}(t)) + \varphi_2 \times r^t_{2(ij)} \times (gbest_j(t) - x_{ij}(t)),$$
$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1),$$

where, first equation is called the velocity update equation and second one is called the position update equation of any j^{th} dimension of the i^{th} particle of the swarm or the population in any t^{th} iteration. and are acceleration constants while and are uniform random numbers in the range [0,1]. The pseudo code of general PSO based Membrane algorithm is shown below in Fig. 4.

In this paper we have used five PSO based membrane algorithms which are briefly described along with their pseudo codes in [18]. All these algorithms use cell-like P systems, i.e., P systems having a hierarchical membrane structure. These five algorithms are Particle Swarm Optimization Based on P system (PSOPS), Membrane Computing based Particle Swarm Optimization (MCBPSO), Hybrid Particle Swarm Optimization based P system (HPSOPS), Bio-Inspired Algorithm based on Membrane Computing (BIAMC), and Membrane Optimization Algorithm based on Mutated Particle Swarm Optimization (MOMPSO). The pseudo codes of these algorithms are presented in Figs. 5–9. In [18] these algorithms were compared on the basis of their behavior for unconstrained function optimization.

3.2 Constraint handling techniques

In literature various constraint handling techniques can be found each having their own merits and demerits. However, Deb's constraint handling technique[5] has been found as one of the most popular technique existing in literature. This technique is based on tournament selection operator and is implemented by separating constraints and objective function. The approach used a set of three feasibility criteria, also known as feasibility and dominance rules, coupled with binary tournament selection in GA. The rules are:

i.The solution with better fitness value is selected between two feasible solutions. ii.The feasible solution is preferred over the infeasible solution. iii.The solution with minimum constraint violation is preferred among the two infeasible solutions.

```
Begin
generation ← 1
    Initialize membrane structure with specified number of membranes, say m
    Generate n individuals for each membrane.
    while (not termination condition) do
    –  Perform PSO in each membrane independently and find global best of each of the m
       membranes.
    –  Execute communication rules
    –  generation = generation + 1
       end
       global best of the skin membrane is the optimized result
end
```

Fig. 4: Pseudo code of general PSO based membrane algorithm

```
Begin
    t ← 1
    Initialize membrane structure with specified number of membrane
    Generate n individuals in skin membrane and assign them to m membranes randomly
     while (not termination condition) do
   –  Perform PSO in each membrane independently and find gbest within each of the m membrane
   –  Execute communication rules in which gbest from all membranes are sent to skin membrane and
      global gbest is computed and is sent back to all membranes
   –  t =   t + 1
     end
     global gbest is the optimized result
end
```

Fig. 5: Pseudo code of PSOPS

This approach has an advantage as user-defined parameters are not required.

On other hand, there is always brute force approach to solve any problem. In optimizing a constrained problem, the basic brute force approach can be finding all the feasible solutions from the generated population set. The objective function of only feasible solution is calculated while for infeasible it is assumed to be infinity. The best solution at any iteration is the one having the minimum objective function value. In this paper, the established Deb's constraint handling technique is compared against the very basic brute force method.

Begin

t ← 1

Initialize membrane structure with specified number of membranes

Generate specified number of individuals and compute their pbest and gbest in each membrane separately

while (not termination condition) do

- Perform PSO updating in each of the m membranes
- Execute communication rules (named here as cooperation in which each membrane sends its n best particles to its outer membrane and receives n best particles from it)
- Mutation is performed in each membrane on n worst particles by reinitializing them
- Evaluate all the objects
- t = t + 1

end

end

Fig. 6: Pseudo code of MCBPSO

Begin

t ← 1

Initialize membrane structure with specified number of membranes

Generate n individuals in skin membrane and assign them to m membranes randomly

while (not termination condition) do

- Perform PSO in each membrane independently and find gbest within each of the m membranes
- Wavelet mutation is applied to the swarm in each membrane
- Execute communication rules in which gbest from all membranes are sent to skin membrane and global gbest is computed and is sent back to all membranes
- t = t + 1

end

global gbest in skin is the optimized result

end

Fig. 7: Pseudo code of HPSOPS

4 Experimental results

4.1 Procedure

To demonstrate the application of PSO based Membrane Algorithms in Constrained Optimization Problems, all the five algorithms are applied to the engineering design problems described in Section 2. To obtain the results, all the algorithms are coded in MPI-C on the cluster. The parameters used in the algorithms are same as that used in [19]. The population size is kept fixed at 49 irrespective of the algorithm or the problem. This population size was so chosen with an idea that it can be equally divided among the seven membranes as

Begin

t ← 1

Initialize membrane structure with specified number of membranes

Generate n individuals in each membrane randomly

while (not termination condition) do

- Perform PSO based on Gaussian distribution in each membrane independently
- Send-out communication i.e. each membrane sends all its particles to skin membrane
- In skin membrane local and global neighborhood searches are applied on each particle respectively
- Calculate fitness for each particle
- Send-in communication is performed where one best and n-1 worst particles of the particles present in skin membrane at a time are sent to each membrane one by one till there are no particles left in skin
- t = t + 1

end

the best value in the first elementary membrane is the global minima

end

Fig. 8: Pseudo code of BIAMC

Begin

t ← 1

Initialize membrane structure with specified number of membranes

Generate some individuals and assign them to one and only one membrane randomly with each membrane containing at least one particle

Perform PSO updating in each of the membrane

while (not termination condition) do

- Calculate mutant power for each particle in each membrane and decide if the particle has to mutate
- Evaluate the fitness values for particle and find local and global minima of the membrane
- Send out the global minima of each membrane to skin, calculate best among them and send in the best value to each membrane as new global minima
- t = t + 1

end

global minima in skin gives the optimized result

end

Fig. 9: Pseudo code of MOMPSO

all the used algorithms have seven membranes. For each problem, 50 runs are performed where the stopping criteria of a run was kept as maximum iteration completed. The maximum iteration limit was 104 function evaluations. The obtained results are presented in tabular form below.

4.2 Computational time required

The average time taken to complete 104 function evaluations by the algorithms using both Deb's technique and brute force method has been recorded and is presented in Tab. 1. The table shows that for three problems PSOPS attains the minimum and BIAMC attains the maximum computational time. However, for remaining two problems MOMPSO attains the minimum and MCBPSO attains the maximum computational time with PSOPS and BIAMC attaining very close position. The other fact to be noticed is that higher time complexity is attained by an algorithm when it is coupled with Deb's technique as compared to Brute force method. The highlighted values in Tab. 1 clearly prove the fact.

Table 1: Average computational time (in secs) for 104 function evaluations

Algorithms	Technique	Welded beam (Version 1)	Welded beam (Version 2)	Pressure vessel Design (Version 1)	Pressure vdessel design (Version 2)	Tension/ compression string design
PSOPS	Brute Force	1.022159^a	0.99161^a	0.859725^a	0.942318	0.845303
	Deb's Tech.	1.086551	1.139383	0.907616	0.990209	1.019908
MCBPSO	Brute Force	2.714302	2.625981	2.541696	2.353988	2.48539
	Deb's Tech.	2.867829	2.938997	2.586111	3.004618^b	3.215778^b
HPSOPS	Brute Force	1.592532	1.635091	1.409166	1.412108	1.266458
	Deb's Tech.	1.712778	1.669447	1.434127	1.442104	2.031082
BIAMC	Brute Force	2.851957	2.860454	2.206078	2.259418	2.39843
	Deb's Tech.	2.948309^b	2.97675^b	2.624394^b	2.571357	3.18955
MOMPSO	Brute Force	1.133472	1.197758	0.919424	0.931936^a	0.837728^a
	Deb's Tech.	1.15989	1.213826	1.028128	0.976448	1.164213

Note: ^{a} means max, ^{b} means min.

4.3 Convergence analysis

The manner in which an algorithm converges shows its efficiency in exploring the given search space without getting trapped in local minima or avoiding premature convergence. The algorithm with higher convergence is considered the better algorithm. To analyze the convergence behavior of all the algorithms under consideration, a single run on the same population set of a problem is conducted by each algorithm with both constraint handling techniques separately. A run here means completion of 104 function evaluations or iteration. The obtained function value after each iteration is recorded and is plotted as graph shown in Fig. 10–14.

Fig. 10 shows the convergence behavior of the algorithms for Welded Beam Problem (Version 1). The graph shows that HPSOPS with Deb's technique has worst convergence behavior followed by HPSOPS with brute force method while MOMPSO has best convergence behavior irrespective of the technique used. The convergence of all the algorithms is better when coupled with brute force algorithm as compared to their convergence when Deb's technique is used. The high convergence by any algorithm is noticed only till 5 x 104 function evaluations however after this attained values remains almost constant.

The convergence graph for Welded Beam Problem (Version 2) is illustrated in Fig. 11 which again concludes that MOMPSO has best converging property while HPSOPS has worst. However, for this problem MCBPSO with Deb's technique has a better convergence as compared to MCBPSO with brute force method.

Fig. 12 presents the convergence plots for Pressure Vessel Design Problem (Version 1). For this problem convergence for PSOPS with Deb's technique is worst. However, PSOPS performance with Brute Force method is considerably good. Similarly, there is a huge difference in the performance of the BIAMC algorithm with different technique. However, the performance of MOMPSO is best irrespective of the technique used.

The convergence graphs for Pressure Vessel Design Problem (Version 2) are presented in Fig. 13. The plots show that MOMPSO has best convergence. However, it is MOMPSO with Brute Force Method that is preferred over its counterpart. Even for this problem worst convergence is presented by HPSOPS with Deb's technique.

The convergence behavior of HPSOPS and PSOPS with Deb's technique is found to be worst for Tension/Compression String Design Problem as shown in Fig. 14. Although, MOMPSO with Brute Force method

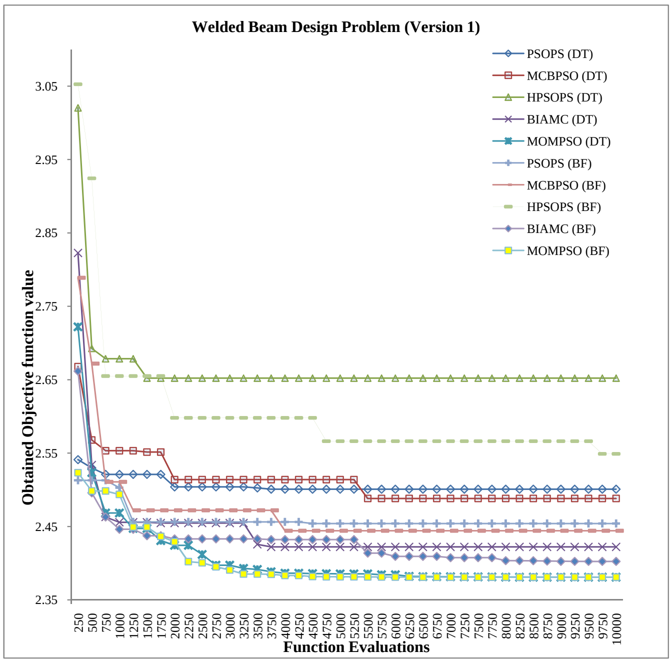

Fig. 10: Convergence graph for Welded Beam Design Problem (Version 1). The initial data point for all the curves in the figure is 9.742996, i.e., at FE = 0 objective function value is 9.742996

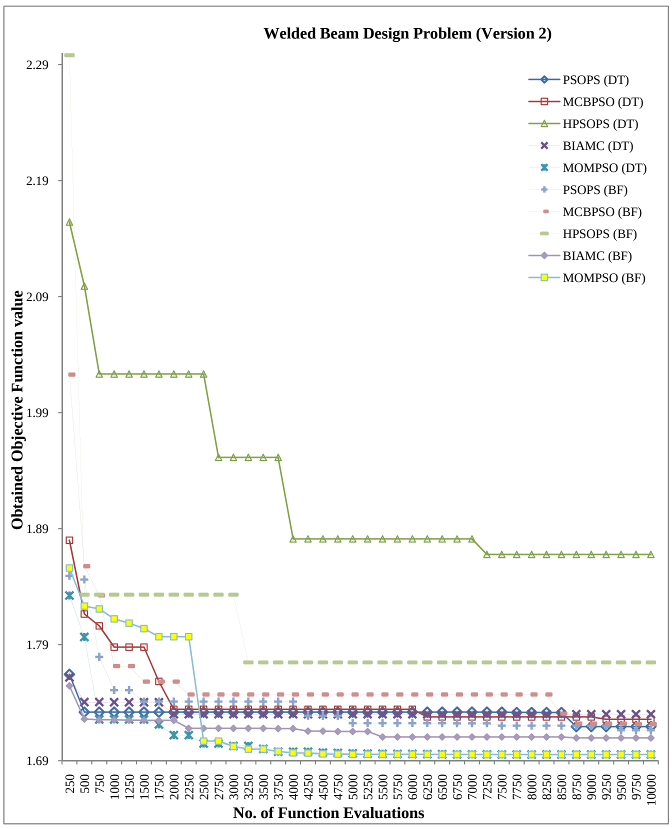

Fig. 11: Convergence graph for Welded Beam Design Problem (Version 2). The initial data point for all the curves in the figure, i.e., objective function value at FE = 0 is 3.21101

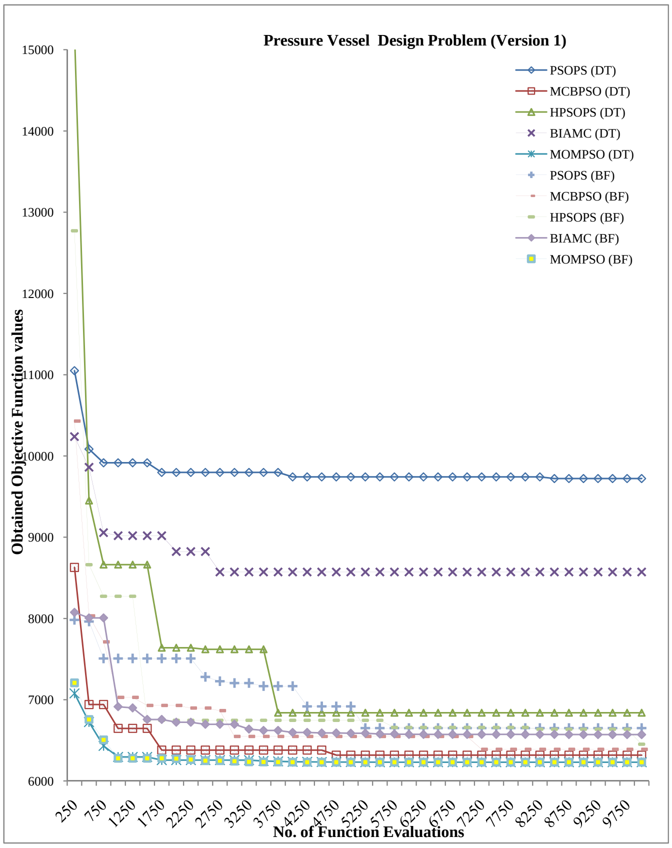

Fig. 12: Convergence graph for Pressure Vessel Design Problem (Version 1). The initial data point for all the curves in the figure, i.e., objective function value at FE = 0 is 54540.12983

presents the best convergence, the behavior of MOMPSO with Deb's technique for this problem is not that satisfactory.

4.4 Analysis of obtained objective function value

To further analyze the behavior of the algorithms, coupled two different constraint handling methods to solve engineering design problems under consideration, the statistics of the obtained objective function value over 50 different runs is presented in Tab. 2–6.

Tab. 2 shows the statistical results for Welded Beam Problem (Version 1). The obtained results show that MOMPSO is the algorithm that performs best for this problem. The performance of HPSOPS and PSOPS using Deb's technique is found to be worst for this problem. However, if only results of algorithms with brute technique are compared it is MCBPSO which has the worst performance. It can be noticed that in most cases the performance of algorithms coupled with Deb's technique is worse as compared to its counterpart using Brute force method except for MCBPSO.

The obtained statistics for Welded Beam Problem (Version 2) is shown in Tab. 3. The depicted results clearly show the dominance of MOMPSO over other algorithms. Its performance remains almost constant irrespective of the constraint handling technique used. However, PSOPS coupled with Deb's technique is the worst performer for the problem followed by HPSOPS coupled with Deb's technique. MCBPSO is the worst performer if the algorithms are compared only on the basis of results obtained using brute force method. The performance of HPSOPS and PSOPS using Brute force method is also poor but it is better as compared its counterpart using Deb's Techniques.

The algorithms when applied to Pressure Vessel Design Problem (Version 1) the obtained results, as shown in Tab. 4, again proves the dominance of MOMPSO. However, here the use of Brute force method

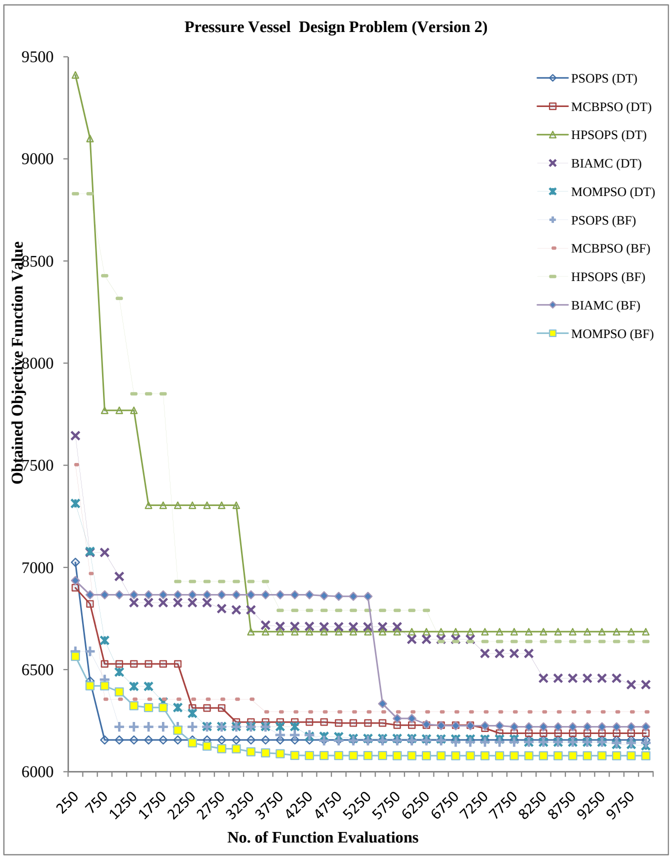

Fig. 13: Convergence graph for Pressure Vessel Design Problem (Version 2). The initial data point for all the curves in the figure, i.e., objective function value at FE = 0 is 28742.64469

Table 2: Statistical results of objective function value for Welded Beam Problem (Version 1)

Algorithms	Technique	Best	Worst	Mean	Median	Std. Dev
PSOPS	Brute Force	2.416923	2.682339	2.461399	2.435194	0.067317
	Deb's Tech.	2.402414	7.663271^b	3.419493^b	2.501988	2.00419^b
MCBPSO	Brute Force	2.444199	2.847656	2.672767	2.6915555	0.087961
	Deb's Tech.	2.412255	2.711196	2.515763	2.498791	0.079044
HPSOPS	Brute Force	2.501486	2.681661	2.585481	2.5813065	0.053774
	Deb's Tech.	2.584547^b	2.827444	2.716917	2.712714^b	0.057074
BIAMC	Brute Force	2.387657	2.464206	2.416728	2.418063	0.018553
	Deb's Tech.	2.389246	2.491294	2.430805	2.428661	0.026611
MOMPSO	Brute Force	2.380959^a	2.380989^a	2.38097^a	2.380966^a	0.00001^a
	Deb's Tech.	2.380959^a	2.381821	2.381071	2.381012	0.000168

Note: ^{a} means max, ^{b} means min.

proves to be better than MOMPSO using Deb's technique. Also, the results again show that the worst performances are given by PSOPS and HPSOPS algorithms using Deb's techniques.

Tab. 5 shows the obtained results for Pressure Vessel Design Problem (Version 2). The displayed results prove the efficiency of MOMPSO in attaining the best results. On other hand it proves HPSOPS with Deb's Technique to be the worst choice.

The results for Tension/Compression String Design Problem are presented in Tab. 6. Even for this problem MOMPSO with Brute force technique is the best choice but the worst performance here is given by BIAMC. Although, again the technique involved is Deb's technique.

The overall analysis of the obtained results can be concluded as:

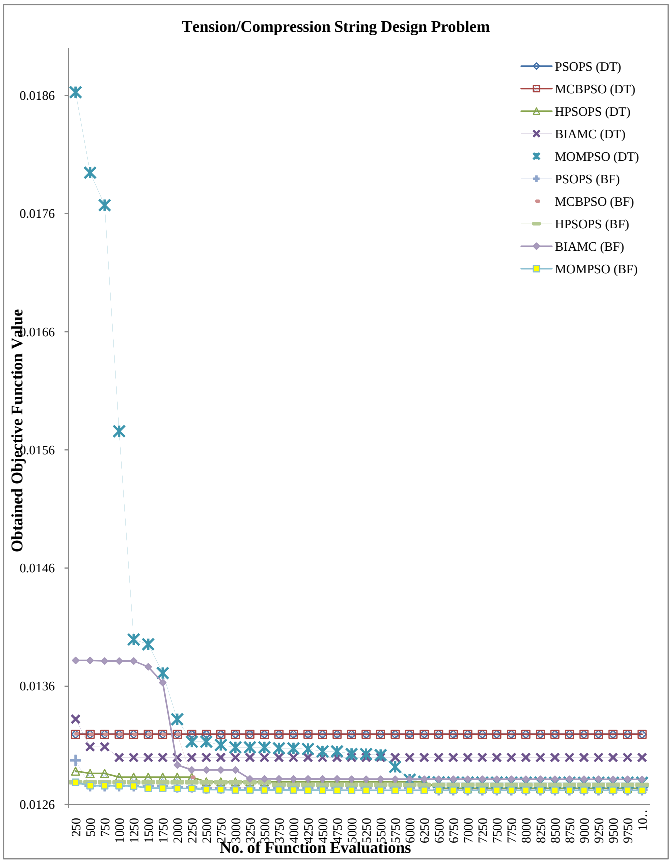

Fig. 14: Convergence graph for Tension/Compression String Design Problem. The initial data point for all the curves in the figure, i.e., objective function value at FE = 0 is 0.076805.

Table 3: Statistical results of objective function value for Welded Beam Problem (Version 2)

Algorithms	Technique	Best	Worst	Mean	Median	Std. Dev
PSOPS	Brute Force	1.712758	1.806279	1.750393	1.730562	0.038167
	Deb's Tech.	1.709942	5.39621^b	1.866779^b	1.732658	0.534265^b
MCBPSO	Brute Force	1.721496	2.080633	1.850451	1.96315^b	0.080716
	Deb's Tech.	1.71396	2.327434	1.7741	1.746362	0.087175
HPSOPS	Brute Force	1.742025	1.895844	1.837473	1.836627	0.040271
	Deb's Tech.	1.747417^b	1.947505	1.8551	1.857796	0.041758
BIAMC	Brute Force	1.698599	1.749738	1.716629	1.718315	0.012228
	Deb's Tech.	1.702414	1.802702	1.727031	1.720208	0.020974
MOMPSO	Brute Force	1.695247a	1.69526^a	1.695251^a	1.695251	0.000004^a
	Deb's Tech.	1.695247^a	1.695268	1.695251^a	1.695249^a	0.000004^a

Note: ^{a} means $\max$, ^{b} means $\min$.

(1) MOMPSO is best PSO based membrane algorithm to solve the structural engineering design problems. However, MOMPSO in conjunction with Brute force method is the preferred option for some problems while for other it performs best irrespective of the constraint handling technique used.

(2) PSOPS and HPSOPS using Deb's Constraint handling Technique are the worst membrane algorithm to be chosen.

(3) If the technique to be used is Brute force Method, the worst choice is MCBPSO.

(4) BIAMC although attains better results as compared to other membrane algorithms except MOMPSO, but has a very high computational time.

(5) The standard deviation of the algorithms incorporating Deb's technique is very high proving it to be an unstable method.

Table 4: Statistical results of objective function value for Pressure Vessel Design Problem (Version 1)

Algorithms	Technique	Best	Worst	Mean	Median	Std. Dev
PSOPS	Brute Force	5917.1	7452.25	6139.93	5979.47	449.92
	Deb's Tech.	6360.99	10700.17[b]	7156.92	6798.51	1100.95[b]
MCBPSO	Brute Force	6389.12	10376.51	7407.58	7285.94	931.23
	Deb's Tech.	6338.11	10128.98	7271.72	7184.68	679.2
HPSOPS	Brute Force	6188.11	7616.55	6730.33	6775.31	361.69
	Deb's Tech.	6525.08[b]	9067.38	7578.38[b]	7547.13[b]	609.78
BIAMC	Brute Force	5893.07	7341.24	6352.79	6371.53	391.05
	Deb's Tech.	6249.68	9020.12	6771.94	6598.72	593.14
MOMPSO	Brute Force	5885.33[a]	7319.05[a]	6008.55[a]	5889.38[a]	322.45[a]
	Deb's Tech.	6230.73	9479.77	6300.39	6230.73	459.61

Note: [a] means max, [b] means min.

Table 5: Statistical results of objective function value for Pressure Vessel Design Problem (Version 2)

Algorithms	Technique	Best	Worst	Mean	Median	Std. Dev
PSOPS	Brute Force	5878.89	7380.37	6119.35	5963.26	440.09
	Deb's Tech.	6149.14	7455.57	6569.75	6489.08	270.19
MCBPSO	Brute Force	6192.81	7408.66	6363.14	6261.01	256.97
	Deb's Tech.	6154.88	7634.32	6951.67	6962.89	353.6
HPSOPS	Brute Force	6179.71	7637.12	6694.98	6534.07	430.8
	Deb's Tech.	6240.63[b]	8986.11[b]	7443.86[b]	7348.53[b]	602.63[b]
BIAMC	Brute Force	5839.72	7024.11	6284.4	6235.79	328.8
	Deb's Tech.	6086.3	7774.24	6530.04	6300.3	510.85
MOMPSO	Brute Force	5804.38[a]	6659.43	5880.02[a]	5804.38[a]	194.23
	Deb's Tech.	6078.31	6392.92[a]	6089.52	6078.31	47[a]

Note: [a] means max, [b] means min.

Table 6: Statistical results of objective function value for tension/compression string design problem

Algorithms	Technique	Best	Worst	Mean	Median	Std. Dev
PSOPS	Brute Force	0.012685	0.012856	0.012721	0.012723	0.000046
	Deb's Tech.	0.012723	0.030455[b]	0.013873	0.012834	0.002913[b]
MCBPSO	Brute Force	0.012704	0.012874	0.01273	0.012727	0.000042
	Deb's Tech.	0.012739	0.01868	0.013944	0.013193	0.001849
HPSOPS	Brute Force	0.012727	0.012839	0.012774	0.012772	0.000035
	Deb's Tech.	0.012725	0.019165	0.013062	0.012852	0.001029
BIAMC	Brute Force	0.01271	0.015956	0.013738	0.013392	0.001005
	Deb's Tech.	0.012747[b]	0.018362	0.014599[b]	0.014069[b]	0.001813
MOMPSO	Brute Force	0.012667[a]	0.01282[a]	0.012718[a]	0.012721[a]	0.000041[a]
	Deb's Tech.	0.012668	0.017773	0.013041	0.012729	0.000965

Note: [a] means max, [b] means min.

(6) The performance of all the algorithms, except MCBPSO, is better when coupled with Brute force Method.

In Deb's constraint handling technique feasible solutions are always given a preference over infeasible solution, thus difficulty is faced in finding the solution to the problem which has their global optimum lying on the common boundary of the feasible and infeasible regions. However, as brute force method can be computationally expensive in context of both time and space, Deb's technique is preferred in general. However, as the results shows that when PSO based membrane algorithms are used even with brute force method the efficient results can be obtained.

5 Conclusion

In this paper five state-of-art membrane algorithms using rules of PSO defined within framework of cell-like P systems are analyzed on the basis of their performance in solving constrained engineering design problems. For handling the constraints two popular approaches, namely, Deb's constraint handling technique and Brute Force Method are used and the impact of their usage on the performance of the algorithms is studied. Based on the extensive numerical and graphical analysis it is concluded that MOMPSO is highly recommended choice while HPSOPS and PSOPS should be given the least preference in order to solve these problems. Moreover, while solving constraint problems using such membrane algorithms Brute Force Method should be preferred over Deb's technique for the purpose of handling constraint. The reason behind such outcome is that the when brute force method is coupled with PSO based membrane algorithms drawback of high time and space complexity associated with brute force method is conquered.

References

[1] B. Akay, D. Karaboga. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, 2012, **23**(4): 1001–1014.

[2] B. Akay, D. Karaboga. Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, 2012, **23**(4): 1001–1014.

[3] C. A. C. Coello. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, 2000, **41**(2): 113–127.

[4] K. Deb. Geneas: A robust optimal design technique for mechanical component design. **in:** *Evolutionary algorithms in engineering applications*, Springer, 1997, 497–514.

[5] K. Deb. An efficient constraint handling method for genetic algorithms. *Computer methods in applied mechanics and engineering*, 2000, **186**(2): 311–338.

[6] Q. He, L. Wang. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, 2007, **20**(1): 89–99.

[7] J. Huang, L. Gao, X. Li. A teaching–learning-based cuckoo search for constrained engineering design problems. **in:** *Advances in Global Optimization*, Springer, 2015, 375–386.

[8] B. Kannan, S. N. Kramer. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of mechanical design*, 1994, **116**(2): 405–411.

[9] A. Kaveh, S. Talatahari. Engineering optimization with hybrid particle swarm and ant colony optimization. *Asian journal of civil engineering*, 2009, **10**(6): 611–628.

[10] J. Kennedy. Particle swarm optimization. **in:** *Proceedings of International Conference on Neural Networks, Perth, Australia*, 1995, 942–1948.

[11] M. Mahdavi, M. Fesanghary, E. Damangir. An improved harmony search algorithm for solving optimization problems. *Applied mathematics and computation*, 2007, **188**(2): 1567–1579.

[12] K. Masuda, K. Kurihara. A constrained global optimization method based on multi-objective particle swarm optimization. *Electronics and Communications in Japan*, 2012, **95**(1): 43–54.

[13] I. Mazhoud, K. Hadj-Hamou, J. Bigeon, P. Joyeux. Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. *Engineering Applications of Artificial Intelligence*, 2013, **26**(4): 1263–1273.

[14] S. Mun, Y.-H. Cho. Modified harmony search optimization for constrained design problems. *Expert Systems with Applications*, 2012, **39**(1): 419–423.

[15] G. Păun. Computing with membranes. *Journal of Computer and System Sciences*, 2000, **61**(1): 108–143.

[16] K. Ragsdell, D. Phillips. Optimal design of a class of welded structures using geometric programming. *Journal of Engineering for Industry*, 1976, **98**(3): 1021–1025.

[17] E. Sandgren. Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, 1990, **112**(2): 223–229.

[18] G. Singh, K. Deep, A. K. Nagar. Cell-like p-systems based on rules of particle swarm optimization. *Applied Mathematics and Computation*, 2014, **246**: 546–560.

[19] J.-h. Xiao, Y.-f. Huang, Z. Cheng. A bio-inspired algorithm based on membrane computing for engineering design problem. *Int. J. Comput. Sci*, 2013, **10**: 580–588.

[20] C. Zhang, Q. Lin, et al. Backtracking search algorithm with three constraint handling methods for constrained optimization problems. *Expert Systems with Applications*, 2015, **42**(21): 7831–7845.

[21] G. Zhang, J. Cheng, et al. A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems. *Applied Soft Computing*, 2013, **13**(3): 1528–1542.

[22] Y. Zhou, G. Zhou, J. Zhang. A hybrid glowworm swarm optimization algorithm for constrained engineering design problems. *Appl. Math. Inf. Sci*, 2013, **7**(1): 379–388.