Engineering Optimization with Particle Swarm

Xiaohui Hu [1,2] Russell C. Eberhart [2] Yuhui Shi [3]

[1] Department of Biomedical Engineering
Purdue University, West Lafayette, Indiana, USA
hux@ecn.purdue.edu

[2] Department of Electrical and Computer Engineering
Purdue School of Engineering and Technology, Indianapolis, Indiana, USA
reberhar@iupui.edu

[3] EDS Embeded Systems Group
Kokomo, Indiana, USA
Yuhui.Shi@eds.com

Abstract- This paper presents a modified particle swarm optimization (PSO) algorithm for engineering optimization problems with constraints. PSO is started with a group of feasible solutions and a feasibility function is used to check if the newly explored solutions satisfy all the constraints. All the particles keep only those feasible solutions in their memory. Several engineering design optimization problems were tested and the results show that PSO is an efficient and general approach to solve most nonlinear optimization problems with inequity constraints.

I. INTRODUCTION

One of the most difficult parts encoutered in practical engineering design optimizations is constraint handling. Real-world limitations frequently introduce multiple, non-linear and non-trivial contraints on a design. Constraints often limit the feasible solutions to a small subset of the design space. A general engineering optimization problem can be defined as follows:

Minimize $f(X)$, $X = \{x_1, x_2, ..., x_n\} \in R$

subject to $g_i(X) \le 0, \quad i = 1,2,...,p$

and $h_i(X) = 0, \quad i = 1,2,...,m$

where $x_i^{(L)} \le x \le x_i^{(U)}, \quad i = 1,2,...,n$

Due to the complexity and unpredictability of constaints, a general deterministic solution is hard to find. In recent years, several evolutionary algorithms have been proposed for constrained engineering optimization problems [1]. Different kinds of methods were proposed for handling constraints, which is the key point of the optimization process [2]. Here a new kind of particle swarm optimization (PSO) algorithm is developed to solve the nonlinear engineering optimization problems with constraints.

II. PARTICLE SWARM

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Kennedy and Eberhart [3,4]. It exhibits some evolutionary computation attributes: 1. It is initialized with a population of random solutions. 2. It searches for optima by updating generations. 3. Updating is based on previous generations. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

The updates of the particles are accomplished according to the following equations. Equation (1) calculates a new velocity for each particle (potential solution) based on its previous velocity (V_{id}), the particle's location at which the best fitness has been achieved (p_{id}, or *pBest*) so far, and the best particle among the neighbors (p_{nd}, or *nBest*) at which the best fitness has been achieved so far. Equation (2) updates each particle's position in the solution hyperspace. The two random numbers ($rand()$ and $Rand()$) are independently generated, and c_1 and c_2 are learning factors. The use of the inertia weight w has provided improved performance in a number of applications [5].

$$V_{id} = w \times V_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times Rand() \times (p_{nd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + V_{id} \quad (2)$$

III. CONSTRAINT HANDLING

There are some studies reported in the literature that extended PSO to constrained optimization problems. Various constraint handling techniques were employed to facilitate the optimization process.

Parsopoulos *et al.* [6] converted the constrained optimization problem into a non-constrained optimization problem by adopting a non-stationary multi-stage assignment penalty function and then applying PSO to the converted problems. Several benchmark problems are tested and the author claimed it outperformed other different evolutionary algorithms, such as Evolution Strategies and Genetic Algorithms.

Ray *et al.* [7] employed a Pareto ranking scheme to handle constraints, which is a concept in multiobjective optimization scenarios. The authors proposed a swarm metaphor with a multilevel information sharing strategy. In a swarm, there are some better performers (leaders) that set the direction of search for the rest of the individuals. An individual that is not in the better performer list (BPL) improves its performance by deriving information from its closest neighbor in BPL. The constraints are handled by a constraint matrix. A multilevel Pareto ranking scheme is implemented to generate the BPL based on the constraint matrix. It should be noted that the updates of particles use a simple generational operator instead of the regular PSO update formula. The simulation on test cases showed much faster convergence and much less time for function evaluations.

Here, a simple but effective method is introduced to solve constrained optimization problems [8]. The preserving feasibility strategy is employed to deal with constraints. Two modifications were made to PSO algorithms. 1. When updating the memories (*pBest* and *nBest*), all the particles only keep feasible solutions in their memory, 2. During the initialization process, all particles are started with feasible solutions. The algorithm can be stated as follows:

```
For each particle {
    REPEAT initialize particle until it satisfies all the constraints
}

Do {
    For each particle {
        Calculate fitness value
        If the fitness value is better than the best fitness value (pBest)
        in history AND the particle is in the feasible space
            set current value as the new pBest
    }
    For each particle {
        Choose the particle with the best pBest value among all the
        neighbors as the nBest
        Calculate particle velocity according equation (1)
        Update particle position according equation (2)
    }
} While maximum iterations or minimum criteria is not attained
```

Figure 1: Modified PSO algorithm

Compared to other constraint handling techniques, this approach has the following advantages:

1. It is simple. There is no preprocessing to the constraints and there is no complicated manipulation, either. Fitness function and constraints are handled seperatedly, thus there are no limitaions to the constraints.
2. It is faster. The only part of the algorithm dealing with contraints is to check if a solution satisfies all the constraints. This will reduce the computation time when handling multiple or complicated constraints.

IV. EXPERIMENTS AND RESULTS ANAYLSIS

Several examples takne from the optimization literature are used to demonstrate the perfromance of the proposed approach. These examples have linear or non-linear constraints and have been solved using a variety of other techniques [9-13].

In all the experiments, the population size is 20, and the maximum generations is 10,000. Since the particles will be repeatedly intialized until they meet all the constraints. A small population size is preferred, espicially when there is a smaller feasible space. There are two versions of PSO algorithm, local version and global version. The global version is faster while the local version is better in avoiding local optima [3]. Here a local version is used with the neighborhood size set to three. The other parameters are set as follows: the inertia weight is [0.5 + (Rnd/2.0)]; the learning rates are 1.49445; the maximum velocity V_{MAX} was set to the dynamic range of the particle on each dimension [14].

A. Design of a Pressure Vessel

The objective of the problem is to minimize the total cost of the material, forming and welding of a cylindrical vessel. There are four design variables: x_1 (T_s, thickness of the shell), x_2 (T_h, thickness of the head), x_3 (R, inner radius) and x_4 (L, length of the cylindrical section of the vessel), T_s and T_h are integer multiples of 0.0625 inch, which are the available thicknesses of rolled steel plates, R and L are continuous. The problem can be specified as follows:

Minimize

$$f(\mathbf{X}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$g_1(\mathbf{X}) = -x_1 + 0.0193x_3 \le 0$$
$$g_2(\mathbf{X}) = -x_2 + 0.00954x_3 \le 0$$
$$g_3(\mathbf{X}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$
$$g_4(\mathbf{X}) = x_4 - 240 \le 0$$

The following ranges of the variables were used [9]:

$1 \le x_1 \le 99,\ \ 1 \le x_2 \le 99,\ \ 10.0 \le x_3 \le 200.0,\ \ 10.0 \le x_4 \le 200.0$

In PSO, each parameter is coded independently. When dealing with integers, the velocity and position of particles in dimension x_1 and x_2 are truncated to integers.

In order to compare the results, eleven runs were executed [9]. The best solution found by PSO is better than any solution previously reported in the literature as shown in Table 1. The worst solution found by PSO is 6632.5. It seems that the PSO is not stable, and the possible reason is the integer variable.When the velocities are limited to integer values, particles are easily trapped into local optima and fail to explore new areas. Further investigation is needed to solve this issue.

Table 1: Comparison of the results for pressure vessel design problem

Design Variables	Best solutions found		
	This paper	Coello [9]	GeneAS [11]
$x_1(T_s)$	0.8125	0.8125	0.9375
$x_2(T_h)$	0.4375	0.4375	0.5000
$x_3(R)$	42.09845	40.3239	48.3290
$x_4(L)$	176.6366	200.0000	112.6790
$g_1(X)$	0.0	-0.034324	-0.004750
$g_2(X)$	-0.03588	-0.052847	-0.038941
$g_3(X)$	-5.8208E-11	-27.105845	-3652.876838
$g_4(X)$	-63.3634	-40.0000	-127.321000
$f(X)$	**6059.131296**	**6288.7445**	**6410.3811**

B. Welded Beam Design

The objective is to minimize the cost of a welded beam subject to constraints on shear stress, bending stress in the beam, bucking load on the bar, end deflection of the beam, and side constraints. The problem can be stated as follows:

Minimize $f(\mathbf{X}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$

Subject to

$$g_1(\mathbf{X}) = \tau(\mathbf{X}) - \tau_{max} \le 0$$
$$g_2(\mathbf{X}) = \sigma(\mathbf{X}) - \sigma_{max} \le 0$$
$$g_3(\mathbf{X}) = x_1 - x_4 \le 0$$
$$g_4(\mathbf{X}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0$$
$$g_5(\mathbf{X}) = 0.125 - x_1 \le 0$$
$$g_6(\mathbf{X}) = \delta(\mathbf{X}) - \delta_{max} \le 0$$
$$g_7(\mathbf{X}) = P - P_c(\mathbf{X}) \le 0$$

where

$$\tau(\mathbf{X}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$

$$\tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right)$$

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

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

$$\sigma(\mathbf{X}) = \frac{6PL}{x_4x_3^2}, \quad \delta(\mathbf{X}) = \frac{4PL^3}{Ex_3^3x_4}$$

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

$$P = 6000lb, \quad L = 14in, \quad E = 30 \times 10^6 psi$$
$$G = 12 \times 10^6 psi, \quad \tau_{max} = 13600psi$$
$$\sigma_{max} = 30000psi, \quad \delta_{max} = 0.25in$$

The following ranges for the variables were used:

$0.1 \le x_1 \le 2.0, \quad 0.1 \le x_2 \le 10.0, \quad 0.1 \le x_3 \le 10.0, \quad 0.1 \le x_4 \le 2.0$

Eleven runs were executed. PSO found the same solution in all runs. The result is better than any other solutions previously reported as shown in Table 2.

Table 2: Comparison of the results for the welded beam desisgn problem

Design Variables	Best solutions found		
	This paper	Coello [9]	Deb [12]
$x_1(h)$	0.20573	0.2088	0.2489
$x_2(l)$	3.47049	3.4205	6.1730
$x_3(t)$	9.03662	8.9975	8.1739
$x_4(b)$	0.20573	0.2100	0.2533
$g_1(X)$	0.0	0.337812	-5758.603777
$g_2(X)$	0.0	-353.902604	-255.576901
$g_3(X)$	-5.5511151E-17	-0.00120	-0.004400
$g_4(X)$	-3.432983785	-3.411865	-2.982866
$g_5(X)$	-0.0807296	-0.08380	-0.123900
$g_6(X)$	-0.2355403	-0.235649	-0.234160
$g_7(X)$	-9.094947E-13	-363.232384	-4465.270928
$f(X)$	**1.72485084**	**1.74830941**	**2.43311600**

C. Minimization of the Weight of a Tension/Compression Spring

The problem consists of minimizing the weight of a tension/compression spring subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the mean coil diameter D, the wire diameter d and the number of active coils N. The problem can be expressed as follows:

Minimize $f(\mathbf{X}) = (N + 2)Dd^2$

Subject to

$$g_1(\mathbf{X}) = 1 - \frac{D^3N}{71785d^4} \le 0$$
$$g_2(\mathbf{X}) = \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \le 0$$
$$g_3(\mathbf{X}) = 1 - \frac{140.45d}{D^2N} \le 0$$
$$g_4(\mathbf{X}) = \frac{D + d}{1.5} - 1 \le 0$$

The following ranges for the variables were used:

$0.05 \le x_1 \le 2.0, \quad 0.25 \le x_2 \le 1.3, \quad 2.0 \le x_3 \le 15.0$

The mean value from 11 runs was $f(\mathbf{X}) = 0.012718975$ with a standard deviation of 6.446×10^{-5}. The best solution is better than previously reported results in the literature.

Table 3: Comparison of the results for the minimization of the weight of a tension/compression spring

Design Variables	Best solutions found		
	This paper	Coello [9]	Arora [13]
$x_1(d)$	0.051466369	0.051480	0.053396
$x_2(D)$	0.351383949	0.351661	0.399180
$x_3(N)$	11.60865920	11.632201	9.185400
$g_1(X)$	-0.003336613	-0.002080	0.000019
$g_2(X)$	-1.0970128E-4	-0.000110	-0.000018
$g_3(X)$	-4.0263180998	-4.026318	-4.123842
$g_4(X)$	-0.7312393333	-0.731239	-0.698283
$f(X)$	**0.0126661409**	**0.0127047834**	**0.127302737**

D. Himmelblau's Nonlinear Optimization Problem

This problem was proposed by Himmelblau and it has been used before as a benchmark for several evolutionary algorithm based technqies. In this problem, there are five design variables, six nonlinear inqueality constraints and ten boundary conditions. The problem can be stated as follows:

$$\text{Minimize } f(\mathbf{X}) = \begin{array}{l} 5.3578547x_3^2 + 0.8356891x_1x_5 \\ + 37.2932239x_1 - 40792.141 \end{array}$$

Subject to

$$0 \le 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5 \le 92$$

$$90 \le 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 110$$

$$20 \le 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 25$$

where

$$78 \le x_1 \le 102,\quad 33 \le x_2 \le 45,\quad 27 \le x_3 \le 45$$

$$27 \le x_4 \le 45,\quad 27 \le x_5 \le 45\ .$$

Table 4: Comparison of the results for the Himmelblau nonlinear optimization problem

Design Variables	Best solutions found		
	This paper	Coello [9]	Homaifar [10]
x_1	78.0	78.0495	78.0000
x_2	33.0	33.0070	33.0000
x_3	27.070997	27.0810	29.9950
x_4	45.0	45.0000	45.0000
x_5	44.96924255	44.9400	36.7760
$g_1(X)$	92.0	91.997635	90.714681
$g_2(X)$	100.4047843	100.407857	98.840511
$g_3(X)$	20.0	20.001911	19.999935
$f(X)$	**-31025.56142**	**-31020.859**	**-30665.609**

PSO found the best solution in eleven runs without execption. The result is listed in Table 4. It should be noted that Koziel and Michalewicz [2] listed a similar test case which is slightly different than the one listed here. The second coefficient in the first constraint is 0.0006262 instead of 0.00026. In that case, the optimal solution is known to be minus 30665.5398 [2,8].

The above experiments showed that the proposed PSO algorithm is an effective method for handling constraints. However, it should be noted the it requires a group of feasible solutions to start the search. In some problems the feasible spaces are extremely small and it is hard to find a group of feasible solutions. The following experiments were designed to investigate the cost of initialization. The first experiments were to check the size of the feasible space. 100,000 solutions were randomly generated, and the number of feasible solutions were counted. The percentages of feasible solutions are listed in the second column (x) of Table 5. The second experiment was to check the average trials needed to get feasible solutions. Random solutions were repeatedly generations until 100,000 feasible solutions were found and the total number of random solutions needed was counted. The average trials needed to get a feasible solution are listed in the fouth column (y) of Table 5. It is showed that the two groups of numbers are reciprocal to each other.

Altough extra loops are needed to find feasible solutions, the time complexity is not high as expected. A feasible solution has to satisfy all the constraints. Once a constriant is not satisfied, it is not necessary to test other constraints. Thus the overall time complexity is not proportional to the number of loops needed and the computation time will be much lower.

Table 5: Percentage of feasible solutions (x) v. s. mean trials need to get a good solution (y)

	x	1/x	y
A	39.76%	2.515	2.5177
B	2.707%	36.94	37.505
C	0.78%	128.2	133.61
D	51.827%	1.929	1.9200

VI. CONCLUSIONS

It is well known that practical engineering optimization envolves multiple, non-linear and non-trivial constaints due to real world limitations. From an engineering standpoint, a better, faster, cheaper solution is always desired.

Compared with other methods, PSO has the following advantages:

- *Faster*. PSO can get the same quality results in significantly fewer fitness evaluations and constraint evaluations.
- *Better*. From demonstration, PSO found better results than others reported in the literature.
- *Cheaper*. The algorithm is intuitive and does not need specific domain knowledge to solve the problem. There is no transformation or any other manipulations needed

to handle the constraints. Furthermore, there is no need to adjust parameter settings for different problems.

However, there are still some limitations to the application of PSO to engineering problems. First, due to the random origin of evolutionary algorithms, it is difficult to deal with equity constraints. It is almost impossible to find a group of initial solutions in the feasible space. This also applies to those problems with extremely small feasible space. Second, for some engineering optimization problems involving integer variables, the algorithm is not stable occasionally and can be trapped in local minima . Further investigation is needed to improve PSO's performance in solving those types of problems.

REFERENCES

[1] Michalewicz, Z. and Schoenauer, M., "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary Computation*, vol. 4, no. 1, pp. 1-32, 1996.

[2] Koziel, S. and Michalewicz, Z., "Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization," *Evolutionary Computation*, vol. 7, no. 1, pp. 19-44, 1999.

[3] Eberhart, R. C. and Kennedy, J. A new optimizer using particle swarm theory. Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995.

[4] Kennedy, J. and Eberhart, R. C. Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995.

[5] Shi, Y. and Eberhart, R. C. A modified particle swarm optimizer. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1998), Piscataway, NJ. pp. 69-73, 1998.

[6] Parsopoulos, K. E. and Vrahatis, M. N. Particle swarm optimization method for constrained optimization problems. Proceedings of the Euro-International Symposium on Computational Intelligence 2002, 2002.

[7] Ray, T. and Liew, K. M. A swarm with an effective information sharing mechanism for unconstrained and constrained single objective optimization problem. Proceedings of IEEE Congress on Evolutionary Computation (CEC 2001), Seoul, Korea. pp. 75-80, 2001.

[8] Hu, X. and Eberhart, R. C. Solving constrained nonlinear optimization problems with particle swarm optimization. Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics 2002 (SCI 2002), Orlando, USA. 2002.

[9] Coello Coello, C. A., "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113-127, Jan. 2000.

[10] Homaifar, A. A., Lai, S. H. Y., and Qi, X., "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242-254, 1994.

[11] Deb, K., "GeneAS: a robust optimal design technique for mechanical component design," in Dasgupta, D. and Michalewicz, Z. (eds.) *Evolutionary Algorithms in Engineering Applications* Berlin: Springer-Verlag, pp. 497-514, 1997.

[12] Deb, K., "Optimal design of a welded beam via genetic algorithms," *AIAA Journal*, vol. 29, no. 11, pp. 2013-2015, Nov. 1991.

[13] Arora, J. S., *Introduction to optimum design* New York: McGrow-Hill, 1989.

[14] Eberhart, Russell C. and Shi, Yuhui. Tracking and optimizing dynamic systems with particle swarms. Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2001), Seoul, Korea. pp. 94-97, 2001.