A PARTICLE SWARM OPTIMIZATION ALGORITHM FOR MIXED VARIABLE NONLINEAR PROBLEMS

H. Nahvi and I. Mohagheghian*

Department of Mechanical Engineering, Isfahan University of Technology
P.O. Box 84156-83111, Isfahan, Iran
hnahvi@cc.iut.ac.ir - imaniut81@me.iut.ac.ir

*Corresponding Author

(Received: August 26, 2008 – Accepted: July 2, 2009)

Abstract Many engineering design problems involve a combination of both continuous and discrete variables. However, the number of studies scarcely exceeds a few on mixed-variable problems. In this research Particle Swarm Optimization (PSO) algorithm is employed to solve mixed-variable nonlinear problems. PSO is an efficient method of dealing with nonlinear and non-convex optimization problems. In this paper, it will be shown that PSO is one of the best optimization algorithms for solving mixed-variable nonlinear problems. Some changes are performed in the convergence criterion of PSO to reduce computational costs. Two different types of PSO methods are employed in order to find the one which is more suitable for using in this approach. Then, several practical mechanical design problems are solved by this method. Numerical results show noticeable improvements in the results in different aspects.

Keywords Particle Swarm Optimization, Mixed-Variable, Engineering Design

چکیده در بسیاری از مسائل طراحی مهندسی ترکیبی از متغیرهای پیوسته و گسسته به کار می‌رود. با این وجود مطالعات کمی بر روی این گونه مسائل انجام پذیرفته است. در این تحقیق از روش حرکت دسته جمعی ذرات (PSO) برای حل مسائل بهینه‌سازی غیر خطی دارای متغیرهای ترکیبی استفاده شده است. PSO روشی است که می‌تواند برای یافتن جواب بهینه مطلق مسائل بهینه‌سازی غیرخطی و غیرمحدب به‌کار رود. در این مقاله نشان داده می‌شود که PSO یکی از بهترین روش‌ها برای حل مسائل با متغیرهای ترکیبی است. با تغییراتی در این روش زمان اجرای الگوریتم کاهش یافته است. دو نوع مختلف از روش PSO به‌کار رفته و از روش دقیق‌تر در حل تعدادی از مسا ئل طراحی مهندسی استفاده شده است.

1. INTRODUCTION

In many engineering optimization problems, the variables cannot accept arbitrary values. That is to say, for practical reasons, some or all of the variables must be selected from a list of integer or discrete values. For example, structural members such as sheets or springs may have to be selected from sections available as standard sizes. Also, the numbers of reinforcement rods in concrete members, bolts in connections, or gear teeth are all integers.

In recent years, considerable interest has been expressed by researchers in the area of mixed-variable optimization. Schmit, et al [1] used dual method for discrete-continuous variables. Sandgrern [2] and Hajela, et al [3] proposed nonlinear branch and bound algorithms where a solution is first obtained by ignoring the discrete conditions. Typical approach to the solution of optimization problems composed of discrete variables includes sequential linear approach [4], penalty function approach [5], simulated annealing [6] and fuzzy programming [7].

Genetic algorithms (GA) are recently developed optimization techniques and rapidly become popular. Many researchers, including Lin, et al [8], Wu, et al [9], Cheung, et al [10] and Rao, et al [11] considered genetic algorithm to deal with such problems.

Another novel evolutionary computational technique, Particle Swarm Optimization (PSO), has been proposed by Kennedy, et al [12]. The development of PSO was based on observations of animals, social behavior such as birds flocking and fish schooling. PSO is initialized with a population of random solutions. Each individual is assigned a random velocity according to both its own flying experience and that of its companions. The individual, namely particle, is then flown through hyperspace. PSO has memory so, knowledge of good solution is retained within all particles; whereas for example in GAs, previous knowledge of the problem will be destroyed once the population changes. In PSO there is a mechanism of constructive cooperation and information sharing between particles. Due to its simple concept, easy implementation and quick convergence, PSO has gained much attention and has been successfully applied in a variety of fields.

Several researchers used PSO as a basic algorithm for solving mixed variable problems. Guo, et al [13] presented a hybrid swarm intelligence approach (HSIA) for solving problems containing integer, discrete, zero-one and continuous variables. Zhang, et al [14], and He, et al [15] proposed a co-evolutionary particle swarm optimization approach and a hybrid particle swarm with differential evolution operator, respectively. They also solved some benchmark problems comprised of mixed variable problems.

Elbeltagi, et al [16] compared five evolutionary-based search methods consisting of genetic algorithms, memetic algorithms, particle swarm, ant-colony and shuffled frog leaping. They showed that PSO method performs better than the other algorithms in terms of success rate and solution quality while is preceded by ACO with respect to processing time.

In this paper, two common PSO algorithms are compared and advantages and disadvantages of these algorithms when dealing with mixed discrete nonlinear problems are expressed. Then, by manipulating the convergence criterion, efficiency of the algorithm is improved. Also, two rounding techniques are employed to treat discrete values and the results are compared. Finally, by applying the proposed PSO to numerical examples, it is shown that particle swarm optimization is one of the best methods of solving mixed discrete nonlinear problems.

2. PARTICLE SWARM OPTIMIZATION

The particle swarm optimization (PSO) was inspired by the observations of birds flocking and fish schooling. It differs from other well-known Evolutionary Algorithms (EA) as in EA a population of potential solutions is used to probe the search space; but, no operators, inspired by evolution procedure, are applied on the population to generate a new promising solution. Instead, in PSO, each individual (named particle) of the population (called swarm), adjusts its trajectory towards its own previous best solution (called pbest) and the previous best solution attained by any member of its topological neighborhood. There are different kinds of sharing information between particles. In the global variant of PSO, the whole swarm is considered as the neighborhood. Thus, global sharing of information takes place and the particles benefit from the discoveries and the previous experiences of all other companions during the search for promising regions of the landscape [17]. Alternatively, there are some local variants of PSO wherein particles only make use of their own information and that of the best of their adjacent neighbors.

Each particle in PSO has two main characteristics: its position and its velocity. Assume that the current position and velocity vector of the i-th particle in the d-dimensional search space are denoted as $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ and $V_i = (v_{i1}, v_{i2}, ..., v_{id})$, respectively. The best earlier position of the i-th particle is represented as $Pbest_i = (pbest_{i1}, pbest_{i2}, ..., pbest_{id})$.

There are different kinds of PSO including global vision of PSO with inertia weight (GWPSO), local vision of PSO with inertia weight (LWPSO), global vision of PSO with constriction factor (GCPSO), and local vision of PSO with constriction factor (LCPSO) [18].

In GWPSO, which is very popular among researchers, there are two methods for updating position and velocity of each particle. The best position of entire group at k-th iteration is used in the first method while in the second method; the best position of entire group up to the current search is employed.

In the first method, the position x_{id}^{k} and velocity v_{id}^{k} of particle i in the k-th iteration are updated as follows:

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \qquad (1)$$

$$v_{id}^{k+1} = \\ wv_{id}^{k} + c_1 r_1 (pbest_{id}^{k} - x_{id}^{k}) + c_2 r_2 (gbest - x_{id}^{k}) \qquad (2)$$

In Equation 2 w is the inertia weight, c_1 and c_2 are positive constants called cognitive and social parameters, respectively, and r_1 and r_2 are random numbers selected in the interval [0 1]. The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle towards pbest and gbest positions and usually are set $c_1 = c_2 = 2$.

In the second method gbest is replaced by $gbest^k$. As will be shown later, in the numerical examples of mixed-variables or in the problems that only have discrete variables, usage of $gbest^k$ is more suitable compared to the use of gbest. In other words, the success rate of $gbest^k$ is higher than that of gbest. The reason is firstly due to the fast convergence of gbest and secondly, the inability of particles to escape from local minima in gbest method. In other words, since the discrete variables are rapidly converged the continuous variables will be obliged to search in a limited specific area which might not be the optimum area.

The role that inertia weight w plays in the convergence behavior of PSO is very important. The inertia weight is employed to control the effect of the previous velocities on the current velocity. This way, the parameter w makes a compromise between global and local exploration abilities of the swarm. In PSO, when the search continues, the inertia term decreases linearly as:

$$w = w_{max} - (\frac{w_{max} - w_{min}}{k_{max}})k \qquad (3)$$

where w_{max} and w_{min} are the maximum and minimum values of the inertia term, respectively, and k_{max} is the maximum number of iterations. In this paper, these parameters are assumed to be:

$$w_{max} = 1, \ w_{min} = 0 \qquad (4)$$

Sometimes as particle oscillations become wider, the system will gain tendency to explode [12]. The usual means of preventing explosion is simply to define a parameter v_{max} and curb the velocity of every individual i from exceeding that velocity on each dimension d. In the case that velocity violates, it will be modified as follows:

$$\text{If } v_{id} > v_{max} \text{ then } v_{id} = v_{max}$$

$$\text{If } v_{id} < -v_{max} \text{ then } v_{id} = -v_{max} \qquad (5)$$

The effect of this is to allow particles to oscillate within the bounds [12].

2.1. Mathematical Formulation of Constraint Problems

Problem definition:

Minimize f(x)

Subject to

$$x_i^L \le x_i \le x_i^U \quad i = 1,2,.....,n \qquad (6)$$

$$g_h(x) \le 0 \quad h = 1,2,.....,m \qquad (7)$$

where n is the number of continuous variables, x_i^L and x_i^U are the lower and upper bounds of continuous variables, respectively, $g_h(x)$ are applied constraints and m is the number of constraints.

Penalty function is defined as:

$$F(x) = f(x) + r \sum_{h=1}^{m+2n} \max\{0, g_h(x)\} \qquad (8)$$

where r is the penalty parameter, m + 2n is the number of all inequality constraints including $g_h(x)$ and two supplemental constraints, related to the upper and lower bounds of each continuous variable.

2.2. Treatment of Discrete Variables

Assuming

the number of discrete variables in the problem is s:

$$x_{n+i} \in D_i$$

$$D_i = \{d_{i,1}, d_{i,2}, \ldots\ldots, d_{i,f}\} \quad i = 1,2,\ldots,s \qquad (9)$$

At first these variables are treated as continuous and similar to above, the following supplemental constraints are applied for the upper and lower bounds

$$d_{i,1} \le x_{n+i} \le d_{i,f} \qquad i = 1,2,\ldots\ldots\ldots\ldots,s \qquad (10)$$

The new objective function incorporates the behavior constraints as:

$$F(x) = f(x) + r \sum_{h=1}^{m+2n+2s} \max\{0, g_h(x)\} \qquad (11)$$

At any iteration, after finding the new position of particles (x^{k+1}), appropriate variables can be made discrete using two methods. The first method is to find the nearest discrete value to the current continuous variable as follows:

$$\min\left(\left|x_{n+i} - d_{i,1}\right|, \left|x_{n+i} - d_{i,2}\right|, \ldots\ldots., \left|x_{n+i} - d_{i,f}\right|\right) \rightarrow \left|x_{n+i} - d_{i,q}\right| x'_{n+i} = d_{i,q} \qquad (12)$$

The second method is to check the upper and lower discrete values of the current continuous variable and determine which one is the best. It is obvious that second method imposes additional fitness calls to the solution and accordingly increases computational costs. However, when the range of discrete variables is vast, usage of second method leads more accurate results and better convergence [19].

Using these two methods, at any iteration, all of discrete variables that are treated as continuous become discrete. These methods are very simple and efficient and do not have the complexities of the penalty approach. They also surmount shortcomings existing in the methods that adopt discrete values near the optimal solution or use discrete values obtained by rounding off.

2.3. Convergence Criterion In practice, after some iterations all of the particles converge to a specific part of problem's space and with subsequent iterations, particles only oscillate in that region. These oscillations only increase the precision of results. Most of the times such high precision is unnecessary so, a convergence criterion is introduced based on the required accuracy. When the results attained such accuracy, iterations will be terminated and do not need to reach k_{max}. This can decrease the number of fitness calls and as a result reduce the computational costs. This reduction is more significant in the cases of mixed or discrete variable problems. The convergence criterion is determined as:

$$v_{min} = \min\left(\left|V_{ij}\right|\right) \quad i = 1,2\ldots.,c \quad j = 1,2,\ldots.,d$$

$$\text{Terminate if } v_{min} < e \qquad (13)$$

where c and d are the number of variables and particles, respectively, and e is the required accuracy.

2.4. Optimization Algorithm of Mixed Discrete Variables Problems The algorithm of the proposed method is as follows:

1. The number of particles and maximum number of iterations are determined. The iteration number k is set to k = 1.
2. The position and velocity for every particle are set at random.
3. The due variables are made discrete.
4. The values of $pbest^k$, $gbest^k$ and $gbest$ are determined.
5. The value of inertia weight is calculated.
6. The position and velocity of every particle are updated by $pbest^k$ and $gbest^k$.
7. If the velocity of particles violate v_{max} and $-v_{max}$ they are modified.
8. If the minimum value of the particle's absolute velocity is lower than what was chosen; jump to 11.
9. k = k + 1
10. The number of iterations is checked. If $k < k_{max}$, then return to 3. Otherwise:
11. The value of $gbest$ is displayed.

3. NUMERICAL EXAMPLES

To demonstrate the effectiveness of the proposed approach, several applications of optimum design, each corresponds to a particular class, are considered. In the welded beam design the entire variables are continuous while, in the speed reducer design, all of the variables are discrete. In the pressure vessel problem variables are combination of discrete and continuous. In the coil compression spring design problem continuous, discrete and integer variable are exist.

3.1. Example 1. Welded Beam Design The welded beam shown in Figure 1 is designed for minimum cost subject to constraint on shear stress in weld, τ, bending stress in beam, σ, buckling load on the bar, P_c, end deflection of the beam, δ, and side constraints. This problem involves four continuous variables; the fillet weld leg size, h, the fillet weld length, l, the bar's width, t, and the bar's thickness, b. Using the design vector $X = [x_1, x_2, x_3, x_4]^T = [h, l, t, b]^T$, the objective function and constraints are

$$f(x) = 1.10471 x_1^2 x_2 + 0.0481 x_3 x_4 (14 + x_2) \tag{14}$$

$$g_1(x) = \tau(x) - \tau_{max} \le 0 \tag{15}$$

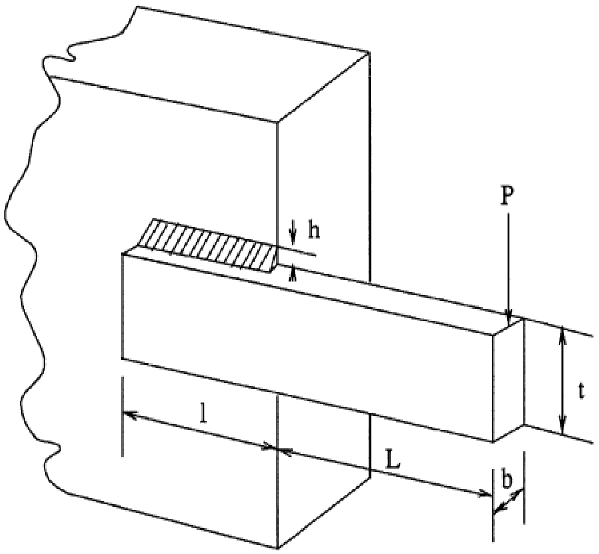

Figure 1. Welded beam design problem.

$$g_2(x) = \sigma(x) - \sigma_{max} \le 0 \tag{16}$$

$$g_3(x) = x_1 - x_4 \le 0 \tag{17}$$

$$g_4(x) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14 + x_2) - 5 \le 0 \tag{18}$$

$$g_5(x) = 0.125 - x_1 \le 0 \tag{19}$$

$$g_6(x) = \delta(x) - \delta_{max} \le 0 \tag{20}$$

$$g_7(x) = P - P_c(x) \le 0 \tag{21}$$

$$g_8(x) \text{ to } g_{11}(x): \ 0.1 \le x_i \le 2.0 \quad i = 1,4 \tag{22}$$

$$g_{12}(x) \text{ to } g_{15}(x): \ 0.1 \le x_i \le 10.0 \quad i = 2,3 \tag{23}$$

where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2} \tag{24}$$

$$\tau' = \frac{P}{\sqrt{2} x x_2}, \quad \tau'' = \frac{MR}{J} \tag{25}$$

$$M = P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \tag{26}$$

and

$$J = 2\left\{\sqrt{2} x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \tag{27}$$

$$\sigma(x) = \frac{6PL}{x_4 x_3^2} \tag{28}$$

$$\delta(x) = \frac{4PL^3}{E x_3^3 x_4} \tag{29}$$

$$P_c(x) = \frac{4.013\sqrt{E(x_3^2 x_4^6 / 36)}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{30}$$

The parameters and material properties are assumed as: P = 6000 lb, L = 14 in, E = 30×10^6 psi, G = 12×10^6 psi, $\tau_{max} = 13600$ psi, $\sigma_{max} = 30000$ psi and $\delta_{max} = 0.25$ in.

In this example, the number of particles, maximum iterations number and convergence criterion value are assumed to be 30, 1000 and 10^{-6}, respectively. The results are obtained through 30 independent runs. Table 1 lists the welded beam design results obtained by different researchers.

As can be seen, the objective function value gained in this research is the best compare to the results of other researches. The statistical results are shown in Table 2. It is evident that not only the best feasible solution obtained in this paper is the best among other researches also, the average searching quality and the worst solution is better than those of other techniques. The standard deviation of the results of this research is very small.

3.2. Example 2. Pressure Vessel Design

Optimum design of a pressure vessel is one of the most famous benchmarks for mixed discrete nonlinear programming (MDNLP) problems and several studies of it are available. The pressure vessel is an air storage tank (as shown in Figure 2) with a working pressure of 3000 psi and a minimum volume of 750 ft^3. The cylindrical pressure vessel is capped at both ends by hemispherical heads. Using rolled steel plate, the shell is to be made in two halves that are joined by two longitudinal welds to form a cylinder. Each head is forged and then welded to the shell. Four design variables are considered as $X=[x_1,x_2,x_3,x_4]^T=[T_s,T_h,R,L]^T$ which indicate thickness of the shell, T_s, thickness

TABLE 1. Comparison of the Best Solution for Example 1 Found by Different Methods.

	Ragsedell	Deb	Coello	He	
$x_1(h)$	0.245500	0.248900	0.208800	0.202369	0.20572276
$x_2(l)$	6.196000	6.173000	3.420500	3.544214	3.4707008
$x_3(t)$	8.273000	8.178900	8.997500	9.048210	9.03681306
$x_4(b)$	0.245500	0.253300	0.210000	0.205723	0.20572878
$g_1(x)$	-5743.8265	-5758.6037	-0.337812	-12.839796	-0.4287188
$g_2(x)$	-4.715097	-255.5769	-353.902604	-1.247467	-1.13037
$g_3(x)$	0.000000	-0.004400	-0.001200	-0.001498	-0.0000006
$g_4(x)$	-3.020289	-2.982866	-3.411865	-3.429347	-3.4329389
$g_5(x)$	-0.120500	-0.123900	-0.083800	-0.079381	-0.0807227
$g_6(x)$	-0.234208	-0.234160	-0.235649	-0.235536	-0.2355411
$g_7(x)$	-3604.275	-4465.2709	-363.23238	-11.681355	-0.0088613
$f(x)$	2.385937	2.433116	1.748309	1.728024	1.7248965

TABLE 2. Statistical Results of Different Methods for Example 1.

Ragsdell [22]	2.385937	N/A	N/A	N/A
Deb [23]	2.433116	N/A	N/A	N/A
Coello [24]	1.748309	1.771973	1.785835	0.011220
He [15]	1.728024	1.748831	1.782143	0.012926
This Research	1.724897	1.730813	1.767000	0.010281

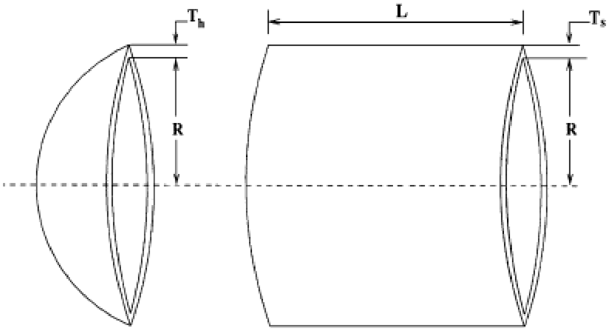

Figure 2. Center and end section of pressure vessel.

of the head, T_h, the inner radius, R and length of the cylindrical section, L. Among the variables R and L are continuous while, T_s and T_h are discrete and required to be a standard size (multiple of 0.0625 in). This problem is formulated according to the ASME boiler and pressure vessel code. The objective function is to minimize the total cost of material used, forming, and welding of the pressure vessel. The problem may be mathematically stated as:

Minimize

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \qquad (31)$$

Subject to the following behavior and side constraints

$$g_1(x) = 0.0193\frac{x_3}{x_1} - 1 \le 0 \qquad (32)$$

$$g_2(x) = 0.00954\frac{x_3}{x_2} - 1 \le 0 \qquad (33)$$

$$g_3(x) = \frac{750 \times 1728 - \frac{4}{3}\pi x_3^3}{\pi x_3^2 x_4} - 1 \le 0 \qquad (34)$$

$$g_4(x) = \frac{x_4}{240} - 1 \le 0 \qquad (35)$$

$$0.0625 \le x_i < 1.25 \quad i = 1,2$$

$$x_i = 0.0625k \quad k = 1,2...,20 \qquad (36)$$

$$25 \le x_3 \le 150 \qquad (37)$$

$$25 \le x_4 \le 240 \qquad (38)$$

Different techniques are employed by researchers for solving this problem. Table 3 indicates a number of results obtained by researchers using particle swarm optimization as the basic solution algorithm while Table 4 shows the results gained by using other methods.

The results of this research, shown in Table 3, are obtained by considering 50, 500, 10^{-6} as the number of particles, maximum iterations number and convergence criterion, respectively.

As shown in Table 4, the best objective function value obtained by methods excluding particle swarm is 5850.77, credited to adaptive range genetic algorithm (ARGA) [20]. The presented approach found a slightly better optimal solution (5850.385) of the pressure vessel problem. Additionally, it is the best solution among the methods that implemented PSO. On the other hand, the average of function calls in the present solution is 22200 while this value is 30000, and 49000 according to binary method [21] and DEPSO [14], respectively. The number of function calls in CPSO [13] is 200000 and in GA-base is 900000 [9]. The number of function calls is a criterion of measuring the computational cost. Since this value is not reported by all researchers the computational cost can not be properly compared in all the researches.

The statistical results of different methods are shown in Table 5. Significant improvement can be seen in three fields of best solution, worst solution and the average search quality as a result of the work done in this study. As discussed in the previous sections, when the problem consists of a combination of both discrete and continuous variables the usage of $gbest^k$ is more efficient than use of $gbest$. This fact can be concluded from Table 6. It is evident from the table that by increasing the number of particles or the maximum iterations value, the improvement will be more significant in attempts that use $gbest^k$.

3.3. Example 3. Coils Spring Design

This example deals with the design of a coil spring under constant axial load. The objective is to design

TABLE 3. Results of Example 2 using Particle Swarm Optimization as the Basic Solution Technique.

	Guo	He	He	Kitayama	Zhang	
R	58.29	42.091	42.1	36.684	42.1	38.860
L	43.7	176.747	176.64	224.096	176.64	221.365
T_s	1.125	0.8125	0.8125	0.75	0.8125	0.75
T_h	0.625	0.4375	0.4375	0.375	0.4375	0.375
g_1	0	0	0	-0.004	0	0
g_2	-0.0689	-0.036	-0.08	-0.016	-0.08	-0.0014
g_3	-69.24	0.036	--0.26	-0.066	-0.26	0
g_4	-196.3	-63.26	0	0.00	0	-0.0776
Cost	7197.9	6061.077	6059.7	5875.254	6059.7	5850.385

TABLE 4. Results of Example 2 Obtained by Different Methods Excluding Particle Swarm Optimization Technique.

	Sandgren	Fu	Wu	Coello	Rao	Lewis	Arakawa
Method	Penalty		MGA		MDHGA	RS + NLP	ARGA
R	47.000	48.38	58.19	40.324	61.4483	38.76	38.858
L	117.70	111.75	44.29	200	27.4037	223.299	221.402
T_s	1.125	1.125	1.125	0.8125	1.1875	0.75	0.75
T_h	0.625	0.625	0.625	0.4375	0.625	0.375	0.375
g_1	-0.194	-0.191	-0.0017	-0.0421	-0.0013	-0.003	0
g_2	-0.283	-0.163	-0.0698	-0.1207	-0.0621	-0.014	-0.011
g_3	-0.510	-75.875	-974.58	0	-0.003	-0.07	-0.078
g_4	0.054	-128.26	-195.7	-0.1667	-0.8858	-1.519	0
Cost	8129.8	8048.6	7207.5	6288.74	7284.02	5980.95	5850.77

TABLE 5. Statistical Results of Different Methods for Example 2.

Zhang (DE) [14]	6062.74	6126.658	6408.792
Zhang (PSO) [14]	6059.714	6332.784	6820.410
Zhang (DEPSO) [14]	6059.714	6108.177	6410.087
Guo (CPSO) [13]	6061.077	6147.133	6363.804
Coello [24]	6288.745	6293.843	6308.15
Coello [27]	6059.946	6177.25	6469.32
This Research	5850.385	5921.195	6090.526

TABLE 6. Comparison of gbest **and** gbest^{k} **for Different Solutions of Example 2.**

gbest^{k}	10	2000	5850.38	6009.7	6090.6	5/30
gbest^{k}	15	2000	5850.38	6002.0	6163.9	7/30
gbest^{k}	10	4000	5850.38	5991.7	6090.7	9/30
gbest^{k}	50	500	5850.38	5921.95	6090.5	16/30
gbest	10	2000	5850.38	6146.6	6424.1	4/30
gbest	15	2000	5850.38	6125.3	6726.5	4/30
gbest	10	4000	5850.38	6082.1	6820.4	7/30
gbest	50	500	5850.38	6083.6	6411.0	5/30

a helical compression spring (Figure 3) with minimum volume of material. There are three different kinds of variables in this example including the integer variable N that is the number of coils of the spring, the wire diameter d that is a discrete value chosen as per ASME code and the outside spring diameter D that is a continuous variable. Using the design vector $X=[x_1,x_2,x_3]^T=[N,d,D]^T$, the objective function can be obtained as:

$$f(x)=\frac{\pi^2 x_3 x_2^2(x_1+2)}{4} \tag{39}$$

The following constraints, defined in [2], are considered

$$g_1(x)=\frac{-S+8K_s P_{max} x_3}{\pi x_2^3}\le 0 \tag{40}$$

$$g_2(x)=(\delta+1.05(x_1+2)x_2)-l_{max}\le 0 \tag{41}$$

$$g_3(x)=d_{min}-x_2\le 0 \tag{42}$$

$$g_4(x)=x_3-D_{max}\le 0 \tag{43}$$

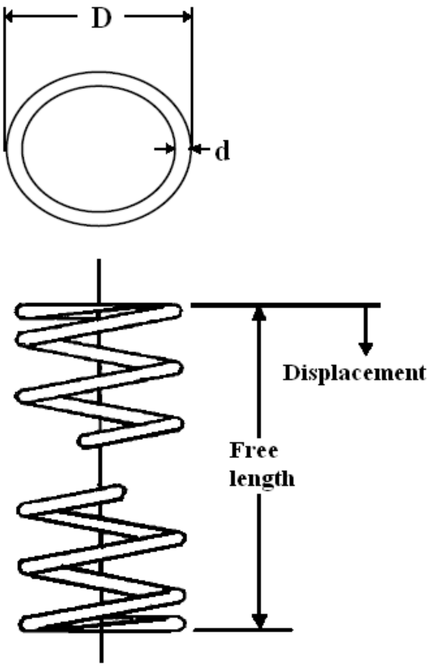

Figure 3. Coil compression spring.

$$g_5(x)=3-C\le 0 \tag{44}$$

$$g_6(x)=\delta-\delta_{pm}\le 0 \tag{45}$$

$$g_7(x)=l_f-\delta-\frac{(P_{max}-P_{load})}{K}-1.05(x_1+2)x_2\le 0 \tag{46}$$

$$g_8(x)=\frac{(P_{max}-P_{load})}{K_S}-\delta_w\le 0 \tag{47}$$

where

$$\delta=\frac{8P_{max}x_3^3 x_1}{Gx_2^4} \tag{48}$$

$$C=\frac{x_3}{x_2} \tag{49}$$

$$K_s=\frac{(4C-1)}{(4C-4)}+\frac{0.615}{C} \tag{50}$$

$$K=\frac{Gx_2{}^4}{8x_1x_3{}^3} \tag{51}$$

In the above relations, S, l_{max}, d_{min}, D_{max}, δ_{pm} and δ_w are the maximum shear stress, the maximum spring free length, the minimum wire diameter, the maximum outside coil diameter, the maximum initial deflection, and the allowable deflection from preload to maximum load, respectively.

Also

$$5\le x_1\le 20; x_1=5+k; k=0,1,\ldots,15 \tag{52}$$

$$0.207\le x_2\le 0.5 \tag{53}$$

$$x_2\in\{0.207, 0.225, 0.244, 0.263, 0.283, 0.307, 0.331, 0.362, 0.394, 0.4375, 0.5\} \tag{54}$$

$$1\le x_3\le 3 \tag{55}$$

The values of pre-assigned parameters are chosen as: $P_{max}=1000$ lb, $S=1.89\times10^5$ psi, $G=1.15\times10^7$ psi, $l_{max}=14$ in, $d_{mni}=0.2$ in, $D_{max}=3$ in, $\delta_{pm}=6$ in, $P_{load}=300$ lb, $l_f=6.6$ in and $\delta_w=1.25$ in.

In this example, the number of particles, maximum iterations value and convergence criterion are assumed to be 10, 1000, 10^{-6}, respectively.

The optimum results obtained by the present approach are compared with the results reported in the literature, presented in Table 7. Results of this research is similar to those achieved by Guo, et al [13] using HSIA and is the best result compare to the other researches. Among 30 independent runs, 24 runs find the global optimum (2.6586). The average of function calls in this problem is 7738.

3.4. Example 4. Speed Reducer (Gear Train)

In this example optimum design of a the speed reducer, shown in Figure 4, is considered.

Design variables are face width, b, the teeth module, m, the number of pinion teeth, n, length of shaft 1 between bearings, l_1, length of the shaft 2 between bearings, l_2, diameter of shaft 1, d_1 and diameter of shaft 2, d_2. The design vector is defined as: $X=[x_1,x_2,x_3,x_4,x_5,x_6,x_7]^T=[b,m,n,l_1,l_2,d_1,d_2]^T$ where x_3 is an integer variable, x_1, x_2, x_4 and x_5 are defined as integer multiples of 0.1, x_6 and x_7 are defined as integer multiples of 0.01. The objective is to minimize the total weight of speed reducer. The constraints include limitation on the bending stress and surface stress of the gear teeth, and transverse deflection of the shafts 1 and 2 [11]. The mathematical formulation of the problem is:

Minimize

$$f(x)=0.7854x_1x_2^2(3.3333x_3^2+14.9334x_3-43.0934)- \\ 1.508x_1(x_6^2+x_7^2)+7.477(x_6^3+x_7^3)+0.7854(x_4x_6^2++x_5x_7^2) \tag{56}$$

Subject to the following constraints

$$g_1(x)=\frac{27}{x_1x_2^2x_3}-1\le 0 \tag{57}$$

$$g_2(x)=\frac{397.5}{x_1x_2^2x_3^2}-1\le 0 \tag{58}$$

$$g_3(x)=\frac{1.93x_4^3}{x_2x_3x_6^4}-1\le 0 \tag{59}$$

$$g_4(x)=\frac{1.93x_5^3}{x_2x_3x_7^4}-1\le 0 \tag{60}$$

$$g_5(x)=\{[(\frac{745\,x_4}{x_2x_3})^2+(16.9)10^6]^{0.5}\Big/0.1x_6^3\} \\ -1100\le 0 \tag{61}$$

$$g_6(x)=\{[(\frac{745x_5}{x_2x_3})^2+(157.5)10^6]^{0.5}\Big/0.1x_7^3\} \\ -850\le 0 \tag{62}$$

$$g_7(x)=x_2x_3-40\le 0 \tag{63}$$

$$g_8(x)=5-\frac{x_1}{x_2}\le 0 \tag{64}$$

$$g_9(x)=\frac{x_1}{x_2}-12\le 0 \tag{65}$$

$$g_{10}(x)=\frac{(1.5x_6+1.9)}{x_4}-1\le 0 \tag{66}$$

$$g_{11}(x)=\frac{(1.1x_7+1.9)}{x_5}-1\le 0 \tag{67}$$

$$2.6\le x_1\le 3.6 \tag{68}$$

$$0.7\le x_2\le 0.8 \tag{69}$$

$$17\le x_3\le 28 \tag{70}$$

$$7.3\le x_4\le 8.3 \tag{71}$$

$$7.3\le x_5\le 8.3 \tag{72}$$

$$2.9\le x_6\le 3.9 \tag{73}$$

$$5.0\le x_7\le 5.5 \tag{74}$$

The problem is solved with 10 particles and 500 maximum iterations. As mentioned in the previous sections, when the problem only consists of discrete variables, using the second approach is more efficient, so this method is used to solve this problem. The optimum solution obtained is $X^*=[3.5,0.7,17,7.3,7.8,3.36,5.29]$ and the optimum objective function value is 3000.8. This solution is

TABLE 7. Best Results Obtained by Other Researches for Example 3.

	Sandgren	Fu	Wu	Rao	Guo	
N	10	9	9	9	9	9
d	1.1807	1.2287	1.2274	1.2253	1.223	1.223
D	0.283	0.283	0.283	0.283	0.283	0.283
g_1	-543.09	-415.97	-550.993	-772.22	-1008.8	-1008.8
g_2	-8.8187	-8.9207	-8.9264	-8.9357	-8.9458	-8.9458
g_3	-0.083	-0.083	-0.083	-0.083	-0.083	-0.083
g_4	-1.189	-1.77	-1.7726	-1.7747	-1.777	-1.777
g_5	-1.172	-1.342	-1.3371	-1.3297	-1.322	-1.322
g_6	-5.464	-5.4568	-5.4585	-5.4613	-5.464	-5.464
g_7	0.0	0.0	0.0	0.0	0.0	0.0
g_8	0.0	-1.0174	0.0134	-0.007	0.0	0.0
f(x)	2.7995	2.6709	2.6681	2.6634	2.6586	2.6586

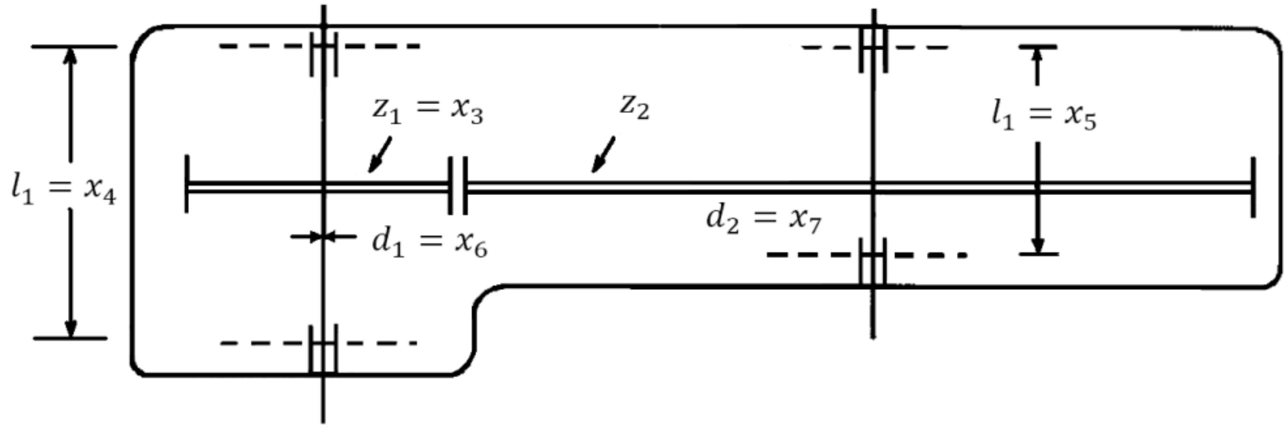

Figure 4. Speed reducer.

identical to the solution reported by Rao, et al [11], obtained by MDHGA method. The constraints values $g_1(x)$ to $g_{11}(x)$ are -0.0739, -0.198, -0.5050, -0.9017, -9.5826, -1.5978, -28.1, 0, -7, -0.0493, -0.0104, respectively. This implies that the optimum solution is in the feasible region.

4. CONCLUSION

In this paper the particle swarm optimization algorithm is employed to solve mixed-variable nonlinear problems. Two common PSO algorithms were compared and results of engineering design problems show that for mixed-discrete problems $gbest^k$ has better performance. Also some changes are performed in the convergence criterion. Simulation results for four constrained engineering design problems are compared with the previously reported results. Noticeable improvements are observed in the results.

5. REFERENCES

1. Schmit, L.A. and Fleury, C., “Discrete-Continuous Variable Structural Synthesis Using Dual Method”, ***AIAA Journal***, Vol. 18, (1980), 1515-1524.
2. Sandgren, E., “Nonlinear Integer and Discrete Programming in Mechanical Design Optimization”, ***ASME, Journal of Mechanical Design***, Vol. 112, (1990), 223-229.
3. Hajela, P. and Shih, C., “Multiobjective Optimum Design in Mixed-Integer and Discrete Design Variable Problems”, ***AIAA Journal***, Vol. 28, (1989), 670-675.
4. Loh, H.T. and Papalambros, P.Y., “A Sequential Linearization Approach for Solving Mixed-Discrete Nonlinear Design Optimization Problems” ***ASME, Journal of Mechanical Design***, Vol. 113, (1991a), 325-334.
5. Fu, J.F., Fenton, R.G. and Cleghorn, W.L., “A Mixed Integer-Discrete-Continuous Programming Method and its Application to Engineering Design Optimization”, ***Eng. Optimization***, Vol. 17, (1991), 263-280.
6. Zhang, C. and Wang, H.P., “Mixed-Discrete Nonlinear Optimization with Simulated Annealing”, ***Eng. Optimization***, Vol. 21, (1993), 277-291.
7. Shih, C.J. and Lai, T.K., “Mixed-Discrete Fuzzy Programming for Nonlinear Engineering Optimization”, ***Eng. Optimization***, Vol. 23, (1995), 187-199.
8. Lin, C.Y. and Hajela, P., “Genetic Algorithms in Optimization Problems with Discrete and Integer Design Variables”, ***Eng. Optimization***, Vol. 19, (1992), 309-327.
9. Wu, S.J. and Chow, P.T., “Genetic Algorithms for Nonlinear Mixed Discrete-Integer Optimization Problems via Meta-Genetic Parameter Optimization”, ***Eng. Optimization***, Vol. 24, (1995), 137-159.
10. Cheung, B.K.-S., Langevin, A. and Delmaire, H., “Coupling Genetic Algorithm with s Grid Search Method to Solve Mixed Integer Nonlinear Programming Problems”, ***Comput. Math. Appl.***, Vol. 34, (1997), 13-23.
11. Rao, S.S. and Xiang, Y., “A Hybrid Genetic Algorithm for Mixed-Discrete Design Optimization”, ***ASME, Journal of Mechanical Design***, Vol. 127, (1992), 1100-1112.
12. Kennedy, J., Eberhart, R.C. and Shi, Y. (Eds.), “Swarm Intelligence”, Morgan Kaufmann, San Francisco, U.S.A., (2001).
13. Guo, X., Hu, J.-X., Ye, B. and Cao, Y.-J., “Swarm Intelligence for Mixed-Variable Design Optimization”, ***J. Zhejiang Univ. SCI***., Vol. 5, (2004), 851-860.
14. Zhang, W.J. and Xie, X.F., “DEPSO: Hybrid Particle Swarm with Differential Evolution Operator”, ***IEEE Int. Conference on Systems and Cybernetics***, (2003), 3816-3821.
15. He, Q. and Wang, L., “An Effective Co-Evolutionary Particle Swarm Optimization for Constrained Engineering Design Problems”, ***Engineering Applications of Artificial Intelligence***, Vol. 20, (2007), 89-99.
16. Elbeltagi, E., Hegazy, T. and Grierson, D., “Comparison Among Five Evolutionary-Based Optimization Algorithms”, ***Advanced Engineering Informatics***, Vol. 19, (2005), 43-53.
17. Laskari, E.C., Parsopoulos, K.C. and Vrahatis, M.N., “Particle Swarm Optimization for Integer Programming”, ***Proceedings IEEE Congress on Evolutionary Computation***, Hawaii, U.S.A., (2002), 1576-1581.
18. Yu, B., Yuan, X. and Wang, J., “Short-Term Hydro-Thermal Scheduling using Particle Swarm Optimization Method”, ***Energy Conversion and Management***, Vol. 48, (2007), 1902-1908.
19. Kitayama, S. and Yasuda, K., “A Method for Mixed Integer Programming by Particle Swarm Optimization”, ***Electrical Engineering in Japan***, Vol. 157, No. 2. (2006), 40-49.
20. Arakawa, M., Ishikawa, H. and Miyashita, T., “Genetic Range Genetic Algorithm to Obtain Quasi-Optimum Solutions Proc ASME/DETC/DAC”, Paper No. 48800, in CDROM, (2003).
21. Kitayama, S., Arakawa, M. and Yamazaki, K., “Penalty Function Approach for the Mixed Discrete Problems by Particle Swarm Optimization”, ***Struc. Multidisciplinary Optimization***, Vol. 32, (2006), 191-202.
22. Ragsdell, K.M. and Phillips, D.T., “Optimal Design of a Class of Welded Structures using Geometric Programming”, ***ASME, Journal of Engineering for Industry***, Vol. 98, No. 3, (1976), 1021-1025.
23. Deb, K., “Optimal Design of a Welded Beam via Genetic Algorithms”, ***AIAA Journal***, Vol. 29, No. 11, (1991), 2013-2015.
24. Coello, C.A.C., “Use of a Self-Adaptive Penalty Approach for Engineering Optimization Problems”, ***Computers in Industry***, Vol. 41, (2000), 113-127.

25. He, S., Prempian, E. and Wu, Q.H., “An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems”, ***Eng. Optimization***, Vol. 36, No. 5, (2004), 585-605.

26. Lewis, K. and Mistree, F., “Foraging-Directed Adaptive Linear Programming: An Algorithm for Solving Nonlinear Mixed Discrete/Continuous Design Problems” Proc. ASME/DETC/DAC, Paper No. 1601, in CDROM, (1996).

27. Coello, C.A.C., “Theoretical and Numerical Constraint Handling Techniques used with Evolutionary Algorithms: A Survey of the State of the Art”, ***Computer Methods in Applied Mechanics and Engineering***, Vol. 191, No. 11-12, (2002), 1245-1287.