

An Improved Constrained Engineering Optimization Design Algorithm

Yuxin Sun^{*1}, Qinghua Wu², Xuesong Yan³

^{1,2}Hubei Provincial Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, China

^{1,2}School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan, China

³School of Computer Science, China University of Geosciences, Wuhan, China

*Corresponding author, e-mail: yuxin_sun@263.net¹, 1622375249@qq.com²

Abstract

Many engineering optimization problems can be state as function optimization with constrained, intelligence optimization algorithm can solve these problems well. Particle Swarm Optimization (PSO) algorithm was developed under the inspiration of behavior laws of bird flocks, fish schools and human communities. In this paper, aim at the disadvantages of standard Particle Swarm Optimization algorithm like being trapped easily into a local optimum, we improves the standard PSO and proposes a new algorithm to solve the overcomes of the standard PSO. The new algorithm keeps not only the fast convergence speed characteristic of PSO, but effectively improves the capability of global searching as well. Experiment results reveal that the proposed algorithm can find better solutions when compared to other heuristic methods and is a powerful optimization algorithm for engineering optimization problems.

Keywords: engineering optimization problems, particle swarm optimization, constrained optimization, evolutionary computation

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Candidate solutions to some problems are not simply deemed correct or incorrect but are instead rated in terms of quality and finding the candidate solution with the highest quality is known as optimization. Optimization problems arise in many real-world scenarios. Take for example the spreading of manure on a cornfield, where depending on the species of grain, the soil quality, expected amount of rain, sunshine and so on, we wish to find the amount and composition of fertilizer that maximizes the crop, while still being within the bounds imposed by environmental law.

Several challenges arise in optimization. First is the nature of the problem to be optimized which may have several local optima the optimizer can get stuck in, the problem may be discontinuous, candidate solutions may yield different fitness values when evaluated at different times, and there may be constraints as to what candidate solutions are feasible as actual solutions to the real-world problem. Furthermore, the large number of candidate solutions to an optimization problem makes it intractable to consider all candidate solutions in turn, which is the only way to be completely sure that the global optimum has been found. This difficulty grows much worse with increasing dimensionality, which is frequently called the curse of dimensionality, a name that is attributed to Bellman, see for example [1]. This phenomenon can be understood by first considering an n-dimensional binary search-space. Here, adding another dimension to the problem means a doubling of the number of candidate solutions. So the number of candidate solutions grows exponentially with increasing dimensionality. The same principle holds for continuous or real-valued search-spaces, only it is now the volume of the search-space that grows exponentially with increasing dimensionality. In either case it is therefore of great interest to find optimization methods which not only perform well in few dimensions, but do not require an exponential number of fitness evaluations as the dimensionality grows. Preferably such optimization methods have a linear relationship between the dimensionality of the problem and the number of candidate solutions they must evaluate in order to achieve satisfactory results, that is, optimization methods should ideally have linear time-complexity $O(n)$ in the dimensionality n of the problem to be optimized.

Another challenge in optimization arises from how much or how little is known about the problem at hand. For example, if the optimization problem is given by a simple formula then it may be possible to derive the inverse of that formula and thus find its optimum. Other families of problems have had specialized methods developed to optimize them efficiently. But when nothing is known about the optimization problem at hand, then the No Free Lunch (NFL) set of theorems by Wolpert and Macready states that any one optimization method will be as likely as any other to find a satisfactory solution [2]. This is especially important in deciding what performance goals one should have when designing new optimization methods, and whether one should attempt to devise the ultimate optimization method which will adapt to all problems and perform well. According to the NFL theorems such an optimization method does not exist and the focus of this thesis will therefore be on the opposite: Simple optimization methods that perform well for a range of problems of interest.

Many engineering optimization design problems can be formulated as constrained optimization problems. The presence of constraints may significantly affect the optimization performances of any optimization algorithms for unconstrained problems. With the increase of the research and applications based on evolutionary computation techniques [3], constraint handling used in evolutionary computation techniques has been a hot topic in both academic and engineering fields [4, 5]. A general constrained optimization problem may be written as follows:

$$\max f(x) \quad (1)$$

Subject to:

$$\begin{aligned} g_i(x) &= c_i, i = 1, 2, \dots, n, \\ h_j(x) &\leq d_j, j = 1, 2, \dots, m. \end{aligned} \quad (2)$$

Where x is a vector residing in a n -dimensional space, $f(x)$ is a scalar valued objective function, $g_i(x) = c_i, i = 1, 2, \dots, n$ and $h_j(x) \leq d_j, j = 1, 2, \dots, m$ are constraint functions that need to be satisfied.

Evolutionary computation has found a wide range of applications in various fields of science and engineering. Among others, evolutionary algorithms (EA) have been proved to be powerful global optimizers. Generally, evolutionary algorithms outperform conventional optimization algorithms for problems which are discontinuous, non-differential, multi-modal, noisy and not well-defined problems, such as art design, music composition and experimental designs. Besides, evolutionary algorithms are also well suitable for multi-criteria problems.

Particle Swarm Optimization (PSO) algorithm was an intelligent technology first presented in 1995 by Eberhart and Kennedy, and it was developed under the inspiration of behaviour laws of bird flocks, fish schools and human communities [6]. If we compare PSO with Genetic Algorithms (GAs), we may find that they are all manoeuvred on the basis of population operated. But PSO doesn't rely on genetic operators like selection operators, crossover operators and mutation operators to operate individual, it optimizes the population through information exchange among individuals. PSO achieves its optimum solution by starting from a group of random solution and then searching repeatedly. Once PSO was presented, it invited widespread concerns among scholars in the optimization fields and shortly afterwards it had become a studying focus within only several years. A number of scientific achievements had emerged in these fields [7-9]. PSO was proved to be a sort of high efficient optimization algorithm by numerous research and experiments [10]. PSO is a meta-heuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-Newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc. This paper improves the disadvantages of standard PSO being easily trapped into a local optimum and proposed an improved PSO algorithm (IPSO) which

proves to be more simply conducted and with more efficient global searching capability, then use the new algorithm for engineering optimization problems.

2. Particle Swarm Optimization Algorithm

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered. Formally, let $f : R^n \rightarrow R$ be the cost function which must be minimized. The function takes a candidate solution as argument in the form of a vector of real numbers and produces a real number as output which indicates the objective function value of the given candidate solution. The gradient of f is not known. The goal is to find a solution a for which $f(a) \leq f(b)$ for all b in the search-space, which would mean a is the global minimum. Maximization can be performed by considering the function $h = -f$ instead.

PSO was presented under the inspiration of bird flock immigration during the course of finding food and then be used in the optimization problems. In PSO, each optimization problem solution is taken as a bird in the searching space and it is called "particle". Every particle has a fitness value which is determined by target functions and it has also a velocity which determines its destination and distance. All particles search in the solution space for their best positions and the positions of the best particles in the swarm. PSO is initially a group of random particles (random solutions), and then the optimum solutions are found by repeated searching. In every iteration, a particle will follow two bests to renew itself: the best position found for a particle called $pbest$; the best position found for the whole swarm called $gbest$. All particles will determine following steps through the best experiences of individuals themselves and their companions.

For particle id , its velocity and its position renewal formula are as follows:

$$V'_{id} = \omega V_{id} + \eta_1 rand()(P_{idb} - X_{id}) + \eta_2 rand()(P_{gdb} - X_{id}) \quad (3)$$

$$X'_{id} = X_{id} + V'_{id} \quad (4)$$

In here: ω is called inertia weight, it is a proportion factor that is concerned with former velocity, $0 < \omega < 1$, η_1 and η_2 are constants and are called accelerating factors, normally $\eta_1 = \eta_2 = 2$; $rand()$ are random numbers, X_{id} represents the position of particle id ; V_{id} represents the velocity of particle id ; P_{idb} , P_{gdb} represent separately the best position particle id has found and the position of the best particles in the whole swarm.

In formula (3), the first part represents the former velocity of the particle, it enables the particle to possess expanding tendency in the searching space and thus makes the algorithm be more capable in global searching; the second part is called cognition part, it represents the process of absorbing individual experience knowledge on the part of the particle; the third part is called social part, it represents the process of learning from the experiences of other particles on the part of certain particle, and it also shows the information sharing and social cooperation among particles.

The flow of PSO can briefly describe as following: First, to initialize a group of particles, e.g. to give randomly each particle an initial position X_i and an initial velocity V_i , and then to calculate its fitness value f . In every iteration, evaluated a particle's fitness value by analyzing the velocity and positions of renewed particles in formula (3) and (4). When a particle finds a better position than previously, it will mark this coordinate into vector $P1$, the vector difference between $P1$ and the present position of the particle will randomly be added to next velocity vector, so that the following renewed particles will search around this point, it's also called in formula (3) cognition component. The weight difference of the present position of the particle swarm and the best position of the swarm P_{gdb} will also be added to velocity vector for adjusting

the next population velocity. This is also called in formula (3) social component. These two adjustments will enable particles to search around two bests.

The most obvious advantage of PSO is that the convergence speed of the swarm is very high, scholars like Clerc [11] has presented proof on its convergence. Here a fatal weakness may result from this characteristic. With constant increase of iterations, the velocity of particles will gradually diminish and reach zero in the end. At this time, the whole swarm will be converged at one point in the solution space, if gbest particles haven't found gbest, the whole swarm will be trapped into a local optimum; and the capacity of swarm jump out of a local optimum is rather weak.

3. Improved PSO Algorithm

In the standard PSO algorithm, the convergence speed of particles is fast, but the adjustments of cognition component and social component make particles search around P_{gdb} and P_{idb} . According to velocity and position renewal formula, once the best individual in the swarm is trapped into a local optimum, the information sharing mechanism in PSO will attract other particles to approach this local optimum gradually, and in the end the whole swarm will be converged at this position. But according to velocity and position renewal formula (3), once the whole swarm is trapped into a local optimum, its cognition component and social component will become zero in the end; still, because $0 < \omega < 1$ and with the number of iteration increase, the velocity of particles will become zero in the end, thus the whole swarm is hard to jump out of the local optimum and has no way to achieve the global optimum. Here a fatal weakness may result from this characteristic. With constant increase of iterations, the velocity of particles will gradually diminish and reach zero in the end. At this time, the whole swarm will be converged at one point in the solution space, if gbest particles haven't found gbest, the whole swarm will be trapped into a local optimum; and the capacity of swarm jump out of a local optimum is rather weak. In order to get through this disadvantage, in this paper we presents a new algorithm based on PSO.

3.1. Information Sharing Mechanism

In order to avoid being trapped into a local optimum, the new algorithm adopts a new information sharing mechanism. We all know that when a particle is searching in the solution space, it doesn't know the exact position of the optimum solution. But we can not only record the best positions an individual particle and the whole swarm have experienced, we can also record the worst positions an individual particle and the whole swarm have experienced, thus we may make individual particles move in the direction of evading the worst positions an individual particle and the whole flock have experienced, this will surely enlarge the global searching space of particles and enable them to avoid being trapped into a local optimum too early, in the same time, it will improve the possibility of finding gbest in the searching space. In the new strategy, the particle velocity and position renewal formula are as follows:

$$V'_{id} = \omega V_{id} + \eta_1 rand()(X_{id} - P_{idw}) + \eta_2 rand()(X_{id} - P_{gdw}) \quad (5)$$

$$X'_{id} = X_{id} + V'_{id} \quad (6)$$

In here: P_{idw} , P_{gdw} represent the worst position particle id has found and the worst positions of the whole swarm has found.

3.2. Elite Selection Strategy

In standard PSO algorithm, the next flying direction of each particle is nearly definite, it can fly to the best individual and the best individuals for the whole swarm. From the above conclusion we may easily to know it will be the danger for being trapped into a local optimum. In order to decrease the possibility of being trapped into the local optimum, the improved PSO introduces elite selection strategy. Traditional genetic algorithm is usually complete the selection operation based on the individual's fitness value, in the mechanism of elite selection, the population of the front generation mixed with the new population which generate through

genetic operations, in the mixed population select the optimum individuals according to a certain probability. The specific procedure is as follows:

Step 1: Using crossover and mutation operations for population P1 which size is N then generating the next generation of sub-populations P2;

Step 2: The current population P1 and the next generation of sub-populations P2 mixed together form a temporary population;

Step 3: Temporary population according to fitness values in descending order, to retain the best N individuals to form new populations P1.

The characteristic of this strategy is mainly in the following aspects. First is robust, because of using this selection strategy, even when the genetic operations to produce more inferior individuals, as the results of the majority of individual residues of the original population, does not cause lower the fitness value of the individual. The second is in genetic diversity maintaining, the operation of large populations, you can better maintain the genetic diversity of the population evolution process. Third is in the sorting method, it is good to overcome proportional to adapt to the calculation of scale. This process of this strategy in improve PSO like this: To set particle number in the swarm as m , father population and son population add up to $2m$. To select randomly q pairs from m ; as to each individual particle i , if the fitness value of i is smaller than its opponents, we will win out and then add one to its mark, and finally select those particles which have the maximum mark value into the next generation. The experiment result shows that this strategy greatly reduces the possibility of being trapped into a local optimum when solving certain functions.

4. Constrained Engineering Optimization Problems

In this section, we will carry out numerical simulation based on some well-known constrained engineering optimization design problems to investigate the performances of the proposed IPSO. The selected problems have been well studied before as benchmarks by various approaches, which is useful to show the validity and effectiveness of the proposed algorithm. For each testing problem, the parameters of the IPSO are set as follows: the number of particle is 100, $c_1=c_2=2.0$ and the number of iteration is 500.

4.1. Tension/Compression String Problem

This problem is described by Arora [12], Coello and Montes [13] and Belegundu [14]. It consists of minimizing the weight ($f(x)$) of a tension/compression string subject to constraints on shear stress, surge frequency and minimum deflection as shown in Figure 1. The design variables are the mean coil diameter $D(=x_1)$; the wire diameter $d(=x_2)$ and the number of active coils $N(=x_3)$. The problem can be stated as:

Minimize:

$$f(x) = (x_3 + 2)x_2x_1^2 \quad (7)$$

Subject to:

$$\begin{aligned} g_1(x) &= 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0, \\ g_2(x) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\ g_3(x) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\ g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0. \end{aligned} \quad (8)$$

This problem has been solved by Belegundu using eight different mathematical optimization techniques [14], Arora also solved this problem using a numerical optimization technique called GA constraint correction at constant cost [12], Additionally, Coello solved this problem using GA-based method [15] and a feasibility-based tournament selection scheme [13],

He solved this problem using co-evolutionary particle swarm optimization method [3]. In this paper, the IPSO is run 50 times independently. Table 1 presents the best solution of this problem obtained using the IPSO algorithm and compares the IPSO results with solutions reported by other researchers. It is obvious from the Table 1 that the result obtained using IPSO algorithm is better than those reported previously in the literature.

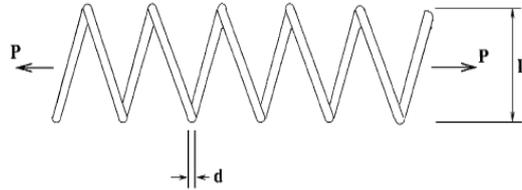


Figure 1. Tension/compression String Problem

Table 1. Comparison of the Best Solution for Tension/compression String Problem

Design variables	IPSO	Belegundu (1982)	Arora (1989)	Coello (2000)	Coello (2002)	He (2007)
$x_1(d)$	0.051154	0.050000	0.053396	0.051480	0.051989	0.051728
$x_2(D)$	0.349871	0.315900	0.399180	0.351661	0.363965	0.357644
$x_3(N)$	12.076432	14.250000	9.185400	11.632201	10.890522	11.244543
$g_1(x)$	0.000000	-0.000014	0.000019	-0.002080	-0.000013	-0.000845
$g_2(x)$	0.000007	-0.003782	-0.000018	-0.000110	-0.000021	-1.2600e-05
$g_3(x)$	4.027840	-3.938302	-4.123832	-4.026318	-4.061338	-4.051300
$g_4(x)$	0.736572	-0.756067	-0.698283	-4.026318	-0.722698	-0.727090
$f(x)$	0.0126706	0.0128334	0.0127303	0.0127048	0.0126810	0.0126747

4.2. Pressure Vessel Problem

A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure 2. The objective is to minimize the total cost, including the cost of material, forming and welding. There are four design variables: T_s (thickness of the shell, x_1), T_h (thickness of the head, x_2), R (inner radius, x_3) and L (length of cylindrical section of the vessel, not including the head, x_4). T_s and T_h are integer multiples of 0.0625 inch, which are the available thickness of rolled steel plates, and R and L are continuous.

Using the same notation given by Coello [16], the problem can be stated as follows:

Minimize:

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (9)$$

Subject to:

$$\begin{aligned} g_1(x) &= -x_1 + 0.0193x_3 \leq 0, \\ g_2(x) &= -x_2 + 0.00954x_3 \leq 0, \\ g_3(x) &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1,296,000 \leq 0, \\ g_4(x) &= x_4 - 240 \leq 0. \end{aligned} \quad (10)$$

This problem has been solved before by Sandgren using a branch and bound technique [17], by Kannan and Kramer using an augmented Lagrangian Multiplier approach [18], by Deb and Gene using Genetic Adaptive Search [19], by Coello using GA-based co-evolution model [15] and a feasibility-based tournament selection scheme [13], and by He using co-evolutionary

particle swarm optimization method [3]. In this paper, the IPSO is run 50 times independently. The comparisons of results are shown in Table 2. The results obtained using the IPSO algorithm, were better optimized than any other earlier solutions reported in the literature.

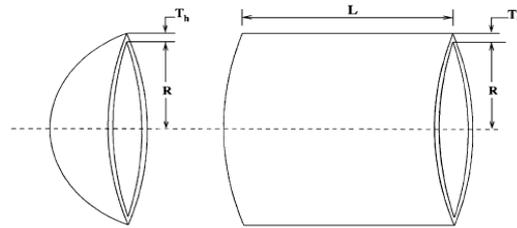


Figure 2. Pressure Vessel Problem

Table 2. Comparison of the Best Solution for Pressure Vessel Problem

Design variables	IPSO	Sandgren (1988)	Kannan (1994)	Deb (1997)	Coello (2000)	Coello (2002)	He (2007)
$x_1(T_s)$	0.812500	1.125000	1.125000	0.937500	0.812500	0.812500	0.812500
$x_2(T_h)$	0.437500	0.625000	0.625000	0.500000	0.437500	0.437500	0.437500
$x_3(R)$	38.860100	47.700000	58.29100	48.329000	40.323900	42.097398	42.091266
$x_4(L)$	221.365000	117.701000	43.690000	112.679000	200.000000	176.654050	176.746500
$g_1(x)$	-0.000000	-0.204390	0.000016	-0.004750	-0.034324	-0.000020	-0.000139
$g_2(x)$	-0.004300	-0.169942	-0.068904	-0.038941	-0.052847	-0.035891	-0.035949
$g_3(x)$	-0.000000	54.226012	-21.220104	3652.876838	-27.105845	-27.886075	116.382700
$g_4(x)$	-18.63500	122.299000	196.310000	-127.321000	-40.000000	-63.345953	-63.253500
$f(x)$	5850.3800	8129.1036	7198.0428	6410.3811	6288.7445	6059.9463	6061.0777

4.3. Welded Beam Problem

The welded beam structure, shown in Figure 3, is a practical design problem that has been often used as a benchmark for testing different optimization methods. The objective is to find the minimum fabricating cost of the welded beam subject to constraints on shear stress (τ), bending stress (σ), buckling load (P_c), end deflection (δ), and side constraint. There are four design variables: $h(=x_1)$; $l(=x_2)$; $t(=x_3)$ and $b(=x_4)$.

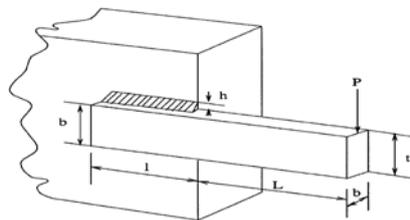


Figure 3. Welded Beam Problem

The mathematical formulation of the objective function $f(x)$, which is the total fabricating cost mainly comprised of the set-up, welding labor, and material costs, is as follows:

Minimize:

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (11)$$

Subject to:

$$\begin{aligned}
 g_1(x) &= \tau(x) - 13000 \leq 0, \\
 g_2(x) &= \sigma(x) - 30000 \leq 0, \\
 g_3(x) &= x_1 - x_4 \leq 0, \\
 g_4(x) &= 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0, \\
 g_5(x) &= 0.125 - x_1 \leq 0, \\
 g_6(x) &= \delta(x) - 0.25 \leq 0, \\
 g_7(x) &= 6000 - P_c(x) \leq 0,
 \end{aligned} \tag{12}$$

Where:

$$\begin{aligned}
 \tau(x) &= \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}, \\
 \tau' &= \frac{6000}{\sqrt{2x_1x_2}}, \\
 \tau'' &= \frac{MR}{J}, \\
 M &= 6000(14 + \frac{x_2}{2}), \\
 R &= \sqrt{\frac{x_2^2}{4} + (\frac{x_1 + x_3}{2})^2}, \\
 J &= 2 \left\{ \sqrt{2x_1x_2} \left[\frac{x_2^2}{12} + (\frac{x_1 + x_3}{2})^2 \right] \right\}, \\
 \sigma(x) &= \frac{504000}{x_4x_3^2}, \\
 \delta(x) &= \frac{2.1952}{x_3^4x_4}, \\
 P_c(x) &= 64746.022(1 - 0.0282346x_3)x_3x_4^3.
 \end{aligned} \tag{13}$$

The approaches applied to this problem include geometric programming [20], genetic algorithm with binary representation and traditional penalty function [21], a GA-based co-evolution model [15] and a feasibility-based tournament selection scheme inspired by the multi-objective optimization techniques [13], and co-evolutionary particle swarm optimization method [3]. In this paper, the IPSO is run 50 times independently. The comparisons of results are shown in Table 3. The results obtained using the IPSO algorithm, were better optimized than any other earlier solutions reported in the literature.

Table 3. Comparison of the Best Solution for Welded Beam Problem

Design variables	IPSO	Ragsdell (1976)	Deb (1991)	Coello (2000)	Coello (2002)	He (2007)
$x_1(h)$	0.205730	0.245500	0.248900	0.208800	0.205986	0.202369
$x_2(l)$	3.470490	6.196000	6.173000	3.420500	3.471328	3.544214
$x_3(t)$	9.036620	8.273000	8.178900	8.997500	9.020224	9.048210
$x_4(b)$	0.205730	0.245500	0.253300	0.210000	0.206480	0.205723
$f(x)$	1.724800	2.385937	2.433116	-1.748309	1.728226	1.728024

5. Conclusion

This paper introduce a new algorithm based on the standard PSO algorithm, for the standard PSO algorithm the new algorithm has done two improvements: 1) By introducing a new information sharing mechanism, make particles moved on the contrary direction of the worst individual positions and the worst whole swarm positions, thus enlarge global searching

space and reduce the possibility of particles to be trapped into a local optimum; 2) By introducing elite selection strategy, decreased the possibility of being trapped into a local optimum. Compared with the standard PSO algorithm, the new algorithm enlarges the searching space and the complexity is not high. Experiment results based on some well-known constrained engineering optimization problems and comparisons with previously reported results demonstrate the effectiveness, efficiency and robustness of the IPSO.

References

- [1] R Bellman. Dynamic Programming. Princeton: Princeton University Press. 1957.
- [2] DH Wolpert, WG Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*. 1997; 1(1): 67-82.
- [3] Qie He, Ling Wang. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*. 2007; 20: 89-99.
- [4] Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*. 2002; 16: 193-203.
- [5] Michalewicz Z. Evolutionary Algorithms in Engineering Applications. Berlin: Springer. 1995: 497-514.
- [6] J Kennedy, RC Eberhart. Particle Swarm Optimization. *IEEE International Conference on Neural Networks*. 1995: 1942-1948.
- [7] Clare M, Kennedy J. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Trans. on Evolutionary Computation*. 2002; 6(1): 58-73.
- [8] Coello CAC, MS Lechuga, Mopso. A proposal for multiple objective particle swarm optimization. *In IEEE Proceedings World Congress on Computational Intelligence*. 2002; 1051-1056.
- [9] J.Kennedy. The particle swarm: social adaptation of knowledge. *Proc.IEEE int.Conf.on evolutionary computation*. 1997; 3003-3008.
- [10] E Oscan, CK Mohan. Analysis of A Simple Particle Swarm Optimization System. *Intelligence Engineering Systems Through Artificial Neural Networks*. 1998; 253-258.
- [11] Clare M, Kennedy J. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Trans. on Evolutionary Computation*. 2002; 6(1): 58-73.
- [12] Arora JS. Introduction to Optimum Design. New York: McGraw-Hill. 1989.
- [13] Coello CAC, Montes EM. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. *Advanced Engineering Informatics*. 2002; 16: 193-203.
- [14] Belegundu AD. A study of mathematical programming methods for structural optimization. *Department of Civil and Environmental Engineering University of Iowa, Iowa City, Iowa*. 1982.
- [15] Coello, C.A.C. Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*. 2000: 41; 113-127.
- [16] Coello CAC. 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*. 2002: 191 (11/12), 1245-1287.
- [17] Sandgren E. Nonlinear integer and discrete programming in mechanical design. *In Proceedings of the ASME Design Technology Conference*. Kissimmee, FL. 1988: 95-105.
- [18] Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Transactions of the ASME, Journal of Mechanical Design*. 1994: 116; 318-320.
- [19] Deb K. GeneAS: a robust optimal design technique for mechanical component design. *Evolutionary Algorithms in Engineering Applications*. 1997: 497-514.
- [20] Ragsdell KM, Phillips DT. Optimal design of a class of welded structures using geometric programming. *ASME Journal of Engineering for Industries*. 1976: 98(3); 1021-1025.
- [21] Deb K. Optimal design of a welded beam via genetic algorithms. *AIAA Journal*. 1991: 29(11); 2013-2015.
- [22] Xue Song Yan, Qing Hua Wu, Cheng Yu Hu, Qing Zhong Liang. Circuit Design Based on Particle Swarm Optimization Algorithms. *Key Engineering Materials*. 2011: 474-476: 1093-1098.
- [23] X.S.Yan, Qing Hua Wu. A New Optimizaition Algorithm for Function Optimization. *Proceedings of the 3rd International Symposium on Intelligence Computation & Applications*. 2009: 144-150.
- [24] Xuesong Yan, Wei Li, Wei Chen, Wenjing Luo, Can Zhang, Hanmin Liu. Cultural Algorithm for Engineering Design Problems. *International Journal of Computer Science Issues*. 2012: 9(6): 53-61.