INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2015; 5(2):151-165

RELIABILITY-BASED DESIGN OPTIMIZATION OF COMPLEX FUNCTIONS USING SELF-ADAPTIVE PARTICLE SWARM OPTIMIZATION METHOD

B. Dizangian[1] and M.R. Ghasemi[2*,†]
Department of Civil Engineering, University of Sistan and Baluchestan, Zahedan, Iran
Department of Civil Engineering, University of Sistan and Baluchestan, Zahedan, Iran

ABSTRACT

A Reliability-Based Design Optimization (RBDO) framework is presented that accounts for stochastic variations in structural parameters and operating conditions. The reliability index calculation is itself an iterative process, potentially employing an optimization technique to find the shortest distance from the origin to the limit-state boundary in a standard normal space.

Monte Carlo simulation (MCs) is embedded into a design optimization procedure by a modular double loop approach, which the self-adaptive version of particle swarm optimization method is introduced as an optimization technique. Double loop method has the advantage of being simple in concepts and easy to implement. First, we study the efficiency of self-adaptive PSO algorithm in order to solve the optimization problem in reliability analysis and then compare the results with the Monte Carlo simulation. While computationally significantly more expensive than deterministic design optimization, the examples illustrate the importance of accounting for uncertainties and the need for regarding reliability-based optimization methods and also, should encourage the use of PSO as the best of evolutionary optimization methods to more such reliability-based optimization problems.

Received: 20 January 2015; Accepted: 1 March 2015

KEY WORDS: Reliability- Optimization- Particle swarm optimization- Monte Carlo simulation.

1. INTRODUCTION

[*]Corresponding author: Department of Civil Engineering, University of Sistan and Baluchestan, Zahedan, Iran
[†]E-mail address: mrghasemi@hamoon.usb.ac.ir (M.R. Ghasemi)

The main objectives in the design of structural systems are to design systems which have satisfactory reliabilities and are as inexpensive as possible. In deterministic structural optimization problems the objective function is usually the volume or the weight of the structure and the constraints are related to code requirements for stresses, displacements or e.g. eigenfrequencies.

For the rational design it is crucial to account for uncertain properties of material, loading and geometry as well as the mathematical model of the system. Moreover, reliability performances should be introduced as the most rational safety measures. Deterministic optimization enhanced by reliability performances and formulated within the probabilistic frameworkis called Reliability-Based stractural Design and Optimization (RBDO).

These contradictory requirements are usually fulfilled using code-based, deterministic design procedures. In this paper it is described how reliability-based structural optimization problems using the evolutionary algorithm PSO, can be formulated and used as a robust tool in the overall reliability based design optimization process [1 -4].

A large number of numerical procedures have been developed to solve this type of problem. Most of the numerical algorithms used in deterministic structural optimization are based on sequential linear programming and dual methods that these methods are time consuming and usually don't gain a exact results.

In the last few decades, meta-heuristics on the basis of life evolution and swarm intelligence have been widely researched. They have been applied to various optimization problems and effective solutions were obtained for certain problems. For example, genetic algorithm (GA), evolutionary programming (EP), and ant colony optimization (ACO) are applied instead of strict optimization methods for the optimization of complex systems [5,6].

PSO has been successfully applied in many different application areas due to its robustness and simplicity [7 and 8]. In comparison with other stochastic optimization techniques, PSO have fewer complicated operations and fewer defining parameters, and can be coded in just a few lines.

2. ENGINEERING OPTIMIZATION

Most engineers when using optimization for design purposes, assume that the design variables in the problem are deterministic. This is referred to as deterministic design optimization. A deterministic design optimization does not account for the uncertainties that exist in modeling, simulationand manufacturing processes. A general deterministic optimization problem is defined in a mathematical form as

Minimize: The objective function $f(\underline{X})$
Subject to:

$$\begin{aligned} &g_j(\underline{X}) \le 0 \qquad j=1,\dots,m \qquad \text{inequality constraints} \\ &h_k(\underline{X}) = 0 \qquad k=1,\dots,s \qquad \text{equality constraints} \\ &b_i^l \le X_i \le b_i^u \quad i=1,\dots,n \qquad \text{bound constraints} \end{aligned} \tag{1}$$

Where $b = \{b_1, b_2, ..., b_n\}^T$ are design variables. The optimizer searches for the best design within the design space defined by the above problem statement. There are many algorithms available that can solve this problem, such as mathematical optimization schemes, iteration algorithms(Sequential quadratic programming, Quasi-Newton methods, Conjugate gradient methods, Gradient descent and Reduced gradient method) or Heuristic algorithms (Particle swarm optimization, Simulated annealing, Genetic Algorithm, Ant Colony Optimization, etc.).

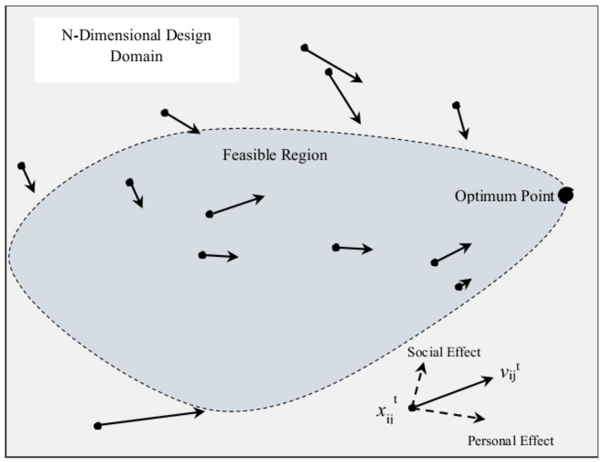

Figure 1. Each particle forms avelocity vector to continue searching base on his own and swarm history of behavior

3. PARTICLE SWARM OPTIMIZATION

PSO is one of the latest evolutionary optimization techniques, that is one of the nature-inspiredmeta-heuristics which is based on a metaphor of social interaction such as bird flocking and fish schooling [1]. PSO is the only evolutionary algorithm that does not implement survival of the fittest. As simple and economic in concept and computational cost, PSO has been shown to successfully optimize a wide range of continuous optimization problems. The swarm consists of *NP* particles, and each particle has a position $X_i = \{x_{i1}, x_{i2}, ..., x_{in}\}$ between lower bound $L_i = \{l_{i1}, l_{i2}, ..., l_{in}\}$ and upper bound $U_i = \{u_{i1}, u_{i2}, ..., u_{in}\}$, $V_i = \{v_{i1}, v_{i2}, ..., v_{in}\}$ between V_{min} *and* V_{max}, where $i = 1, 2, ..., NP$ and moves through a n-dimensional search space. According to the global variant of the PSO algorithm, each particle moves towards its best previous position and towards the best particle *G* of the current swarm. Let us denote the best previously visited position of the *i*th particle that gives the best fitness value as $P_i = \{p_{i1}, p_{i2}, ..., p_{in}\}$ and the best previously visited position of the swarm that gives the best fitness as $G_i = \{g_{i1}, g_{i2}, ..., g_{in}\}$. The change of position of each particle each iteration can be computed according the distance between the current position (X_i) and its previous best position (P_i) and the distance between the current position and the best position of the hole swarm (G). Then the updating of velocity and particle position can be obtained as follow [1].

$$v_{ij}^{t} = v_{ij}^{t-1} + c_1 r_1 (p_{ij}^{t-1} - x_{ij}^{t-1}) + c_2 r_2 (G_j^{t-1} - x_{ij}^{t-1}) \; x_{ij}^{t} = x_{ij}^{t-1} + v_{ij}^{t} \tag{2}$$

where $t = 1,2,\ldots,T_{max}$, represents the iteration number, j denotes the design variable number, c_1 is the cognition learning factor, c_2 is the social learning factor and r_1 and r_2 are random numbers uniformly distributed in [0, 1]. Shi and Eberhart (1998) [7] proposed a modification in which a constant inertia weight ω is used to control how much a particle tends to follow its current direction as compared to the memorized P_i and the G. In this version, the velocity update is given as Eq. 3.

$$v_{ij}^{t} = \omega^{t} \, v_{ij}^{t-1} + c_1 r_1 (p_{ij}^{t-1} - x_{ij}^{t-1}) + c_2 r_2 (G_j^{t-1} - x_{ij}^{t-1}) \tag{3}$$

3.1. Fitness of Particles

For engineering problems we regularly look for the designs that minimize the cost (e.g. minimizing the volume of materials such as area of bars in trusses), so in these problems we consider the fitness as the smallest value of objective function by considering the amount of violation of the design constraints as they design for the structure. We consider the net objective function value as $f(\underline{X})$ and the penalized objective function $f'(\underline{X})$ in this paper as

$$f'(\underline{X}) = f(\underline{X}) \times \mathrm{P}(\underline{X}) \tag{4}$$

That,

$$\mathrm{P}(\underline{X}) = (1 + \bar{r} \times \gamma(\underline{X})) \tag{5}$$

and;

$$\bar{r} = \max\,(100,\, 20 \times (1 + 0.2 \times (t-1))) \tag{6}$$

Where; t and $\underline{X}$ are current iteration of algorithm and design vector of each particle respectively; $\mathrm{P}(\underline{X})$ is quadratic penalty function, $\bar{r}$ is the exterior penalty parameter and γ denotes the net value of penalazation due to the vaiolations.

$$P(\underline{X})_i^t = \sum_{j=1}^{m} \{\max[0, g_j(\underline{X})]\}^2 + \sum_{k=1}^{s} [h_k(\underline{X})]^2 + \sum_{i=1}^{n} \{(\max[0, (\frac{X}{b_i^u} - 1)])^2 + (\min[0, (\frac{X}{b_i^l} - 1)])^2\} \tag{7}$$

3.2. Self-adaptive particle swarm optimization

To improve the global searching capability by escaping the lcal solutions, we consider the following modification.

In the most published papers about the use of PSO algorithm, they consider the inertia weight factor linearly changes with the current iteration. As using this concept, at the start of

PSO ω must has a great value (close to 1) and then linearly decrease to value close to zero. For the problems with complexity of objective function we couldn't sure about the behavior of particles that how and when particles enter to feasible region and how to tune the ω to avoid diverging of solution. To solve this, we consider the inertia weight factor ω as a function of success of particles that could find a better fitness than the self previous design. We here use the exponensial function to define ω in each iteration as follow:

$$\omega^t = \begin{cases} \exp(-(1+Ps)) & \text{if Ps} \geq 0.5 \\ \exp(-Ps) & \text{if Ps} < 0.5 \end{cases} \tag{8}$$

Where, exp(.) denotes the exponential function and PS is the percent of particles that succeed to find the better solution in the iteration *t* compare to the previous iteration *t-1*.

Using the feasible function to manage the movement of each particle and don't let it to go outside of feasible region after it enters this domain:

$$feasible_i^t = \begin{cases} 1 & \text{if } \mathrm{P}(\underline{X})_i^t = 0 \\ 0 & \text{else} \end{cases} \tag{9}$$

That, *t* stands for current iteration and *i* for particle number. Based on this tracking method, if each particle finds the solution that locate in the feasible region, then for all remaining iterations, it couldn't move to the outside of this region and restores the previous position.

4. CONSTRAINTS HANDLING

While searching the design space, particles may violate either the problem constraints or the bound limits of design variables. In the current work, a modified feasible-based mechanism is used to handle the problem specific constraints based on the following four rules [9]

Rule 1: Any feasible solution is preferred to any infeasible solution.

Rule 2: infeasible solutions containing small violation of the constraints are considered as feasible solution (from 0.01 in the first iteration to 0.001 in the last iteration).

Rule 3: Between two feasible solutions, the one having the better objective function value is preferred.

Rule 4: Between two infeasible solutions, the one having the smaller sum of constraint violation is preferred.

5. RELIABILITY ANALYSIS

Reliability is understood as the probability of a component or system performing required functions over its lifetime. The evaluation of the probabilistic design criteria requires a stochastic analysis of the system. Today, several reliability methods are available for calculating the probability of an event occurringwhile accounting for uncertainties.

The Monte Carlo simulation method has been the most widely used probabilistic analysis method due to its generality, simplicity, and effectiveness on problems that are highly non-linear with respect to uncertainty parameters [10].

Statistical moment methods provide a computationally less costly alternative for evaluating the reliability of systems but may lead to large approximation errors. The mean value First-Order Second Moment (MVFOSM) method, for example, is inaccurate and sensitive to the mathematical formulation of the limit state function if the mean value point is not on the failure surface [12].

First- and Second-Order Reliability Methods (FORM and SORM) are more accurate and robust. Both FORM and SORM require a search for the Most Probable Point (MPP) on the failure surface. FORM employs a first-order approximation of the failure function at the MPP for evaluating the probability of failure. Therefore, FORM is considered accurate if the curvature of the failure surface in the standard normal space of the random variables is not too large at the MPP.

a) Monte Carlo simulation

Monte Carlo Simulation (MCS) is known as a simple random sampling method or statistical trial method that make realizations based on randomly generated sampling sets for uncertain variables according to their distributions. The computation procedure of MCS is quite simple:

1. Select a distribution type for the random variable;
2. Generate a sampling set from the distribution;
3. Conduct simulations using the generated sampling set;
4. Compute the performance function base on the sampling set;
5. check if the computed performance function is less than zero or not;

In each trial, sample values can be digitally generated and analyzed. If N trials are conducted, the probability of failure is given approximately by

$$P_f = \frac{N_f}{N} \tag{10}$$

where N_f is the number of trials for which $g(.)$ is violated out of the N experiments conducted.

6. BASIC RBDO FORMULATION

Reliability based design optimization (RBDO) deals with obtaining optimal designs characterized by a low probability of failure. Integration of reliability analysis into the design optimization process as a constraint, or even an objective, has been widely accomplished in the design of structural systems [14 and 15].

Fig. 2, illustrates such a case, where the chance that the deterministic solution fails due to uncertainties in design variable settings. The reliable solution is characterized by a slightly higher function value and is located inside the feasible region.

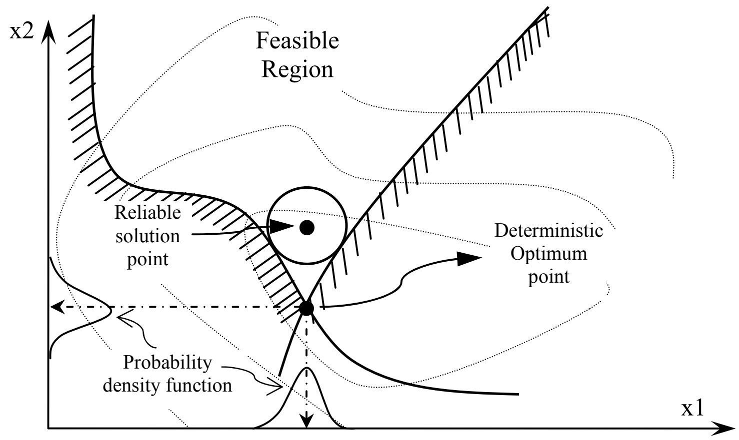

Figure 2. The concept of reliability base design optimization procedure

The probability distributions of the random variables are obtained using statistical models. We consider here a reliability-based single-objective optimization problem of the following type

$$\min_{\mathrm{d}}: \quad C(\mathbf{d})$$
$$s.t.: \begin{cases} \Pr[G_i(\mathbf{d},\mathbf{X}) \le 0] \le P_{f_i}^T & i = 1,\dots,m \\ h_j(\mathbf{d}) \le 0 & j = m+1,\dots,M \end{cases} \tag{11}$$

Here, C is the cost or objective function to be minimizedby variation of determinstic design variables **d, X** is a set of design variables which are uncertain (random variables), G_i is the i th performance function, h_j are deterministic constraints, Pr [·] is the probability operator, $P_{f_i}^T$ is the allowable failure probability, m is the number of performance functions and M isthe total number of constraints. It is to be noted that the design variables dmay be either independent deterministic variables or probability distribution parameters (e.g. the mean m_X of random variables X). The deterministic constraints h_j are typically the upper and lower bounds of the design variables. In the above model, the probabilistic constraints define the feasible region by restricting the probability of violating the limit state G_i to the admissible probability $P_{f_i}^T$. To find whether a given probability of constraint G_i is satisfied at a design point, weneed to compute the following probability of the complementary failure event:

$$\Pr[G_i(\mathbf{d},\mathbf{X}) \le 0] = \int \dots \int_{G_i(\mathbf{d},\mathbf{X}) \le 0} f_X(x)dx \qquad (12)$$

where $f_X(x)$ is the joint probability density function of **X**. It is usually impossible to find an analytical expression for the above integral.The only difficulty the above problem poses is to compute the probability,Pr[.]. The existing reliability-based design optimization procedures can be classified into four classes [15], mainly based on the way the probability term Pr[.] iscomputed:

1. Simulation methods; 2. Double-loop methods;
3. Decoupled methods; 4. Single-loop methods.

Here, in this paper we discuss Double-loop method and use it as the direct solution of RBDO problems in our examples.

6.1. Double-Loop Method

In the double-loop method, a nested optimization is considered. To compute the probability of success of each constraint, an optimization procedure (an inner-level optimization) is used. The outer loop optimizes the original objective function and the inner loop ends an equivalent deterministic version of each probabilistic constraint by formulating and solving an optimization problem. In the present study, PSO algorithm is used in outer loop for finding the best solution of the objective function and governs the MCs method to compute the failure probability of each design that was chosen before with particles. Flowchart of Fig. 3 illustrates this procedure finding the best reliable point base on constraints.

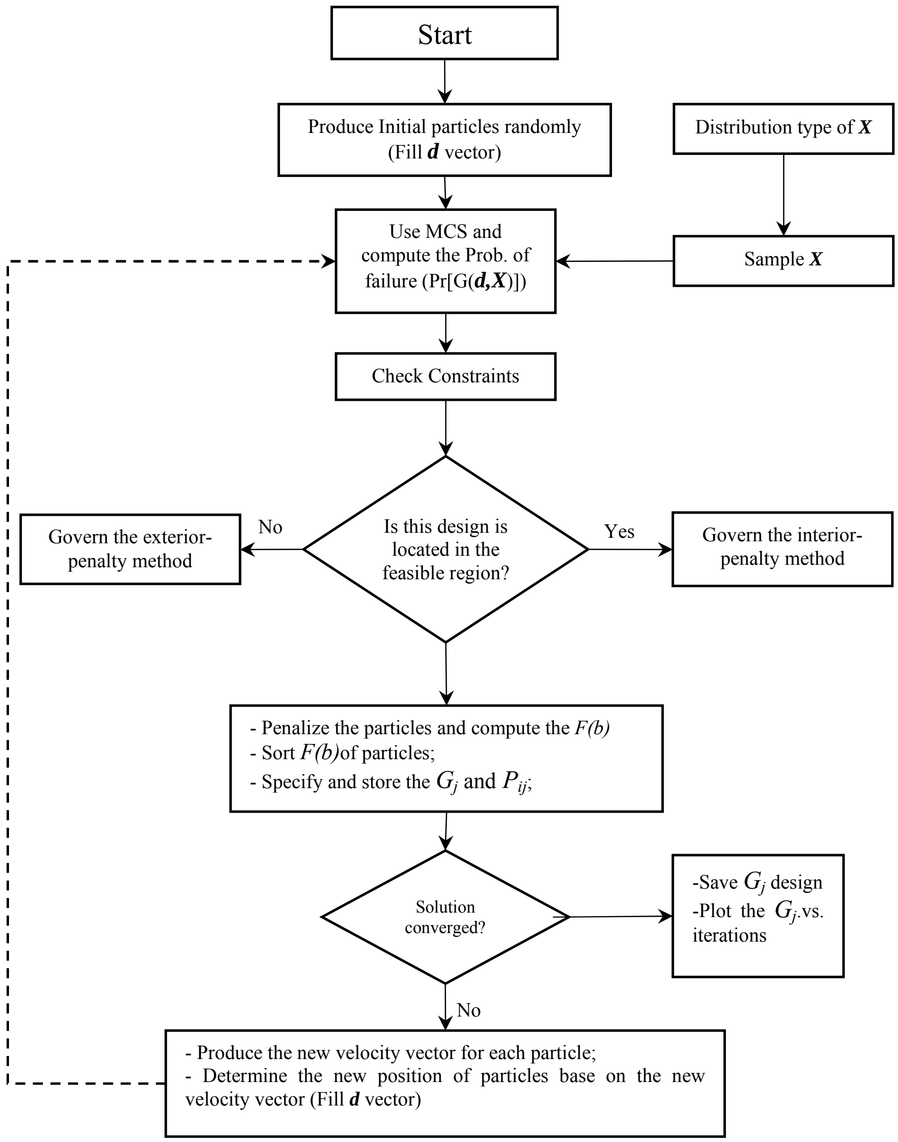

Figure 3. Flowchart of double-loop approach of RBDO base on PSO-MCS

7. NUMERICAL EXAMPLES

This section focuses on the efficiency of using the PSO algorithm in both inner and outer loops of RBDO approache as tested against several benchmark functions with continuous variables. Calibrated parameters of PSO For all examplesare listed in Table 1.

Table 1. Calibrated parameters of PSO

No. of particles	80
Maximum iteration	50-100
Social parameter	2
Cognitive parameter	2

7.1. Function Optimization

In this Section, to show the efficiency of the proposed self-adaptive PSO algorithm, a non-linear unconstrained benchmark performance functions is attempted for the reliability-index computation.

Consider two normally distributed random variable x_1 and x_2 with the highly nonlinear performance function as [15]

$$g(X) = g(x_1, x_2) = x_1^3 + x_2^3 - 18$$
$$x_1 \rightarrow \begin{cases} \mu = 10 \\ \sigma = 5 \end{cases} \qquad x_2 \rightarrow \begin{cases} \mu = 10 \\ \sigma = 5 \end{cases} \tag{13}$$

The goal is to compute the reliability index. The convergence history of the reliability index are illustrated in Fig. 4.

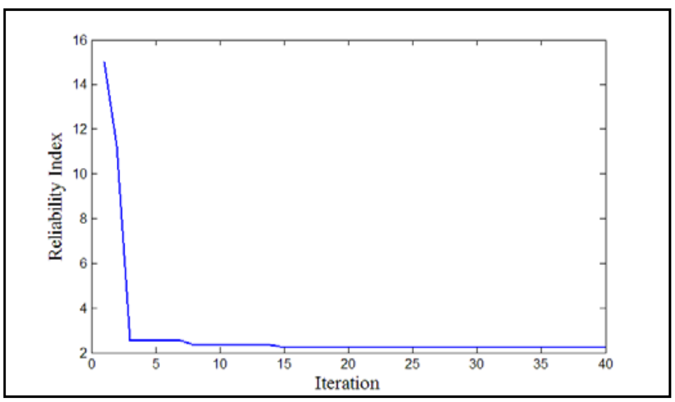

Figure 4. Convergence history of reliability index

Table 2. Estimated reliability index, Example 1

	x_1	x_2	u_1	u_2
MPFP	1.9959	2.1579	-1.6008	-1.5684
$G(\mathbf{X})$	0.0000293			
Reliability Index (P_f)	**2.2411 (0.01251)**			

Table 2 also shows the optimum results obtained by the proposed technique. Comparing the results and the reliability index obtained in [15] (β=2.2401), the PSO shows fast convergence and accurate results which it could find the best solution after 14 iterations.

7.2. Numerical examples of RBDO functions

7.2.1. Non-linear benchmark RBDO problem

This is an analytical multidisciplinary problem. Even though the problem is just two-dimensional, it is sufficiently nonlinear and has the attributes of a general multidisciplinary problem. This problem has two deterministic design variables; d_1 and d_2, and there are two random variables, x_1 and x_2. This problem involves a coupled system analysis and has two constraints. One of the constraints is deal with the upper and lower bond limitation of the deterministic design variables; another constraint is the probabilistic type and should use the stochastic analysis procedure for bound the probability computed from the performance function, to the target probability. The desired value of allowable probability is chosen as 0.01. This benchmark RBDO example is [16] as

$$\begin{aligned}
&\min_d : d_1^2 + d_2^2 \\
&subject\ to : \begin{cases} \Pr[\frac{1}{5} d_1 d_2 x_2^2 - X_1 \le 0] \le P_f^T \\ 0 \le d_i \le 15 \end{cases}, x_1 \rightarrow \begin{cases} \mu = 5 \\ \sigma = 1.5 \end{cases} \quad x_2 \rightarrow \begin{cases} \mu = 3 \\ \sigma = 0.9 \end{cases} \\
&P_f^T = 1\% \ (corresponding\ to\ \beta^T = 2.32)
\end{aligned} \tag{14}$$

At first glance, this can quickly found that regardless of the probabilistic constraint in this problem, the optimum solution is zero; however, by considering the probabilistic constraint and order a greater safety margin, we should use the RBDO formulation to minimize the objective function.

The results and convergence history of this example are presented in Table 3 and Fig. 5.

Table 3. Results of benchmark RBDO function

	f^*	P_f^* (β^*)	d_1^*	d_2^*	Constraint Violation	Run Time (sec.)
Current Study	<u>58.7611</u>	0.00994 (2.3286)	5.3340	5.5053	0.0	143
Aoues and Chateauneuf [16]	<u>63.88</u>	0.01	5.65	5.65	--	--
Improvement of objective function = **8.711** %						

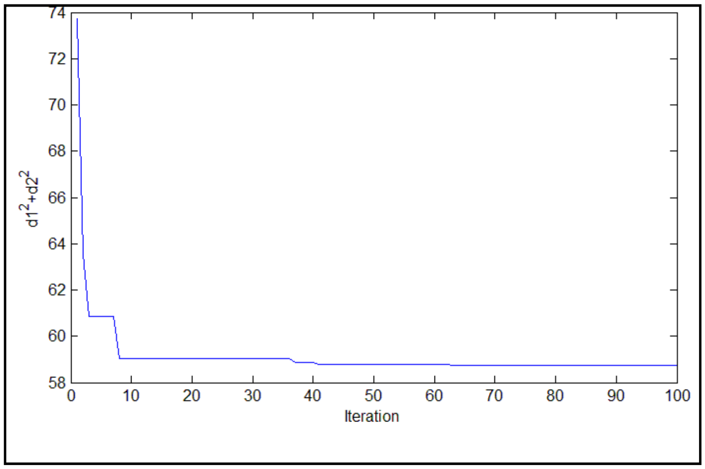

Figure 5. Convergence history of reliability index of Example 2

According to the best objective function achieved, this improvement preposition to the results of reference [16] is very salient**.**

7.2.2. Highly non-linear benchmark RBDO problem

In order to investigate the robustness of the proposed RBDO method, a highly nonlinear limit state function is considered with high curvatures in the neighborhood of the optimum point [16]

$$\min : f(d) = d_1^2 + d_2^2$$

$$\textit{subject to}: \begin{cases} \Pr[G(\underline{d},\underline{X}) \le 0] \le P_f^T \\ 0 \le d \le 15 \end{cases}, x_1 \rightarrow \begin{cases} \mu = 5 \\ \sigma = 1.5 \end{cases} \quad x_2 \rightarrow \begin{cases} \mu = 3 \\ \sigma = 0.9 \end{cases} \tag{15}$$

$$G(\underline{d},\underline{X}) = d_1 \times d_2 \times x_2 - \ln(x_1)$$

$$P_f^T = 1\% \; (\textit{corresponding to } \beta^T = 2.32)$$

In the following, the results and the convergence history of the objective function is shown as in Table 4 and Fig. 6.

Table 4. Results of benchmark RBDO function; Example 3

	f^*	P_f^* (β^*)	d_1^*	d_2^*	Constraint Violation	Run Time (sec.)
Current Study	3.488	0.0097 (2.3378)	1.3406	1.3029	0.0	212
Aoues and Chateauneuf [16]	3.67	0.01	1.35	1.35	--	--
Improvement of objective function = **5.218**%						

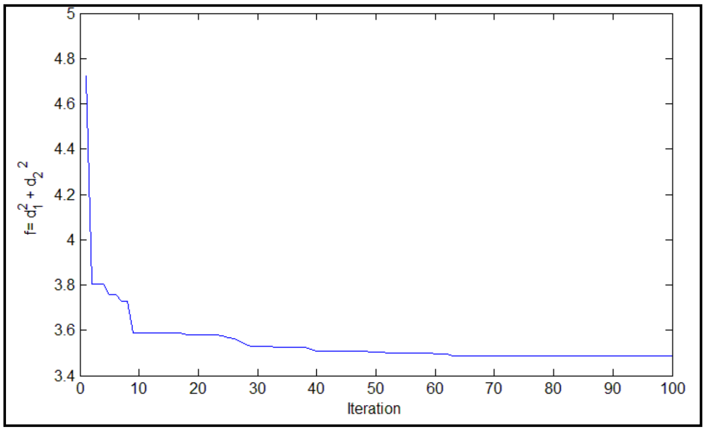

Figure 6. Convergence history of reliability index of Example 3

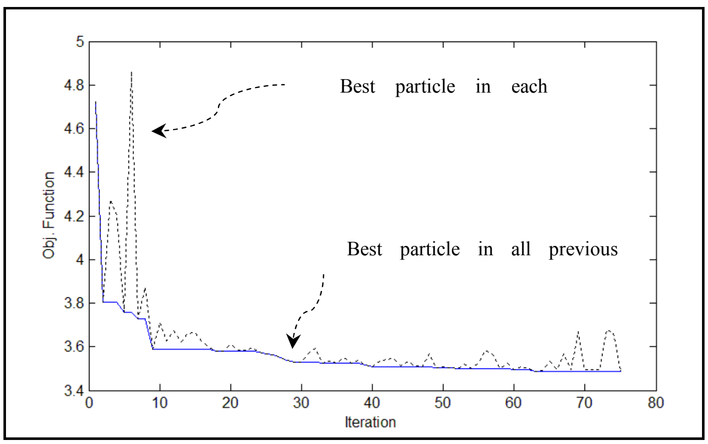

Figure 7. Convergence history of reliability index of Example 3

With respect to the Fig. 7 we can also see the difference between the best solution at the end of each iteration and the global best solution (G_i) of all particles up to current iteration.

It is obvious that the cited difference is due to random behavior of particles in design space. Gradually, by increasing the number of iterations and decreasing the amount of corresponding penalties, the particles tend to the best global design point and two diagrams are match on the same.

8. CONCLUSIONS

In this paper a fast self-adaptive particle swarm optimization method was proposed to solve reliability-based design optimization of complex functions using exterior-interior penalty method in dealing with the design constraints. In this regard, when a design is allocated in the in feasible region, the exterior penalty method is embedded to encourage speeding up the particle movement to the feasible domain. The interior penalty formulation however, emphasizes governing the design within the feasible rejoin, towards the optimum solution.

In order to prove the adequacy and accuracy of self-adaptive PSO algorithm, we first review a function to calculate the reliability index .In the second example, which is an oscillating function, this method was successful in the accurate calculation of reliability index.

In examples 2 and 3, trying to solve two highly nonlinear functions that are well known in the field of RBDO. Finally, the excellent improvement from solving these problems with the new method presented in this paper was reached.

REFERENCES

[1] Kennedy J, Eberhart R. Particle Swarm Optimization, *IEEE International Conference on Neural Networks*, 1995.

[2] Kennedy J, Eberhart RC. A new optimizer using particle swarm theory, in: *Proceedings of the Sixth International Sympusiom on Micro Machine and Human Science (MHS'95)*, Nagoya, Japan, 1995, pp. 39-43.

[3] Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computers and Structures* 2009, **87**: 267-83.

[4] Eberhart RC, Shi YH. Tracking and optimizing dynamic systems with particle swarms, in: *Proceedings of the IEEE Congress on Evolutionary Computation*, San Francisco, USA, 2001, pp. 94–100.

[5] Kaveh A, Talatahari S. A hybrid particle swarm and ant colony optimization for design of truss structures, *Asian Journal of Civil Engineering* 2008, **9**(4): 325-44.

[6] Kaveh A, Talatahari S. A discrete particle swarm ant colony optimization for design of steel frames, *Asian Journal of Civil Engineering* 2008, **9**(6): 531-42.

[7] Kaveh A, Hassani M. Simultaneous analysis, design and optimization of structures using force method and ant colony algorithms, *Asian Journal of Civil Engineering* 2009, **10**(4): 381-96.

[8] Eberhart RC, Shi YH. Tracking and optimizing dynamic systems with particle swarms, in: *Proceedings of the IEEE Congress on Evolutionary Computation*, San Francisco, USA, 2001, 94-100.

[9] Zhang YL, Ma LH, Zhang LY, Qian JX. On the convergence analysis and parameter selection in panicle swarm optimization, in: *Proceedings of the International Conference on Machine Learning and Cybernetics*, Zhejiang University, Hangzhou, China, 2003, pp. 1802-1807.

[10] Li AG. Particle swarms cooperative optimizer, *J. Fudan Univ. (Natural Science)* 2004, ShangHai, China, **43**: 923-5.

[11] Enevoldsen I, Sørensen J. Reliability-based optimization in structural engineering, *Struct Safety* 1994; **15**:169-96.

[12] Enevoldsen I, Sorensen J. Reliability-based optimization as an information tool, *Mech Struct Mach* 1994; **22**(1): 117-35.

[13] Chandu S, Grandhi R. General purpose procedure for reliability based structural optimization under parametric uncertainties, *Adv Eng Software* 1995; **23**: 7-14.

[14] Nowak AS, Collins K. *Reliability of Structures*, McGraw-Hill, Science/Engineering/Math, 1 edition, January 14, 2000.

[15] Grandhi R, Wang L. Reliability-based structural optimization using improved two-point adaptive nonlinear approximations, *Finite Elem Anal Design* 1998: 35-48.

[16] Choi S, Grandhi RV, Robert A. *Canfield, Reliability-based Structural Design*, Springer-Verlag London Limited, 2007.

[17] Aoues Y, Chateauneuf A. Benchmark study of numerical methods for reliability-based design optimization, *Struct Multidisc Optim* 2010, Springer, Verlag, **41**: 277-294,