INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2011; 2:341-355

PSEUDO-RANDOM DIRECTIONAL SEARCH: A NEW HEURISTIC FOR OPTIMIZATION

M. Shahrouzi*, †
Department of Engineering, Tarbiat Moallem University, Tehran, Iran

ABSTRACT

Meta-heuristics have already received considerable attention in various fields of engineering optimization problems. Each of them employes some key features best suited for a specific class of problems due to its type of search space and constraints. The present work develops a *Pseudo-random Directional Search*, PDS, for adaptive combination of such heuristic operators. It utilizes a short term memory via indirect information share between search agents and the directional search inspired by natural swarms. Treated numerical examples illustrate the PDS performance in continuous and discrete design spaces.

Received: 20 January 2011; Accepted: 5 October 2011

KEY WORDS: Pseudo-random directional search, ant system, particle swarm optimization, characteristic graph

1. INTRODUCTION

Up to date, several heuristic and metaheuristic procedures have been investigated for optimization problems. They can be classified to gradient based methods, guided and greedy search algorithms and also stochastic sampling methods. Some of them are mathematical programming [1], intiuitive techniques like optimality criteria [2], single-agent heuristics such as simulated annealing [3], multi-agent stochastic methods including genetic and evolutionary search, ant colony algorithms, and most recently harmony search [4-7], swarm algorithms [8,9], charged system search[10], gravitational search [11] and colonial competition [12].

Deterministic methods mostly are affected by dependence to the starting point assumption, neighborhood structure definition and derivative computations while many real-

---

* Corresponding author: M. Shahrouzi Department of Engineering, Tarbiat Moallem University, Tehran, Iran
† E-mail address: shahruzi@tmu.ac.ir

world problems have discrete or multimodal complex search spaces with several local optima. Despite of their local search capability, they generally suffer from lack of providing sufficient diversity to capture the global and true optimum of the problem.

The second class including single-agent methods use stochastic procedures to overpass the local optima, however, their efficiecny in practicle problems is relatively low because of their dependence to single starting point assumption and difficulties in tuning parameters before sampling the design space.

The most successive class are parallel or multi-agent stochastic search methods, for which the balance between diversification and intensification has a critical rule. Some methods in this class like ant colony algorithms make the soultion part-by-part while the others sample complete design vectors among the search space.

Every such algorithm employs its own method of search space decomposition in order to move in a neighborhood structure or to sample the candidate solutions for further fitness evaluation and selection. Genetic jumps are suited in a space of genotypes coded from corresponding main phenotypes. Simulated annealing and harmony search use bandwidth guided perturbation of the design variables. Particle swarm algorithms and provide a directional decomposition of the search space for further vector sum alteration of solution candidates. Almost no single method is generally efficient for all types of search spaces and problem complexities. The present research provides an adpative method to combine various procedures in searching the deaign space. It is based on a pseudo-random memory based selection of artificial states whose directions dynamically changes with the search progress. The proposed *Pseduo-random Directional Search* is further treated in a variety of illustrative test functions with different shape and modality of their search spaces in order to evaluate its performance for optimization problems.

2. DIRECTIONAL SEARCH IN SWARM ALGORITHMS

Swarm algorithms are a class of stochastic search dealing with design variable changes in a vector sum manner. *Particle swarm optimization*, PSO, is the first in this class introduced by Kennedy and Eberhart [8,9] to simulate natural swarm behaviours such as fish schools and bird flocks. They extracted the following factors for each particle to move around:

- Inertial term representing tendency of a particle to move in its previous-i direction
- Cognitive factor amplifies movement toward the best position of a particle in its memory of all previous steps
- Social term that identifies the best position found by the entire swarm up to the current search iteration
- Randomized bandwidths to add further stochastic property for partial movement in the above-mentioned directions

Consequently, any search agent called *particle* take its new poition by a sum of movement vectors in each distinct search direction according to the following relation:

$$\underline{V}_i^{k+1} = c_i \underline{V}_i^k + r_1 c_c (\underline{P}_i^k - \underline{X}_i^k) + r_2 c_s (\underline{B}^k - \underline{X}_i^k) \tag{1}$$

Where as c_i, c_c, c_s stand for *inertial* , *cognitive* and *social* factors and r_1, r_2 are random numbers uniformly distributed in range [0,1]. $\underline{P}_i^k$ denotes the best pervious position that a particle has already experienced while $\underline{B}^k$ is the global best position of the entire swarm up to now. $\underline{X}_i^{k+1}$, position of the i^{th} particle at the new iteraion $k+1$ is calculated by adding the velocity term to its previous position vector. Time interval dt is usually taken 1 in the algorithmic formulations:

$$\underline{X}_i^{k+1} = \underline{X}_i^k + dt.\underline{V}_i^k \tag{2}$$

In PSO terminology, the current position vector of a particle, $\underline{X}_i$, corresponds to the evaluated design variables vector of the problem in its decision space. All PSO variants are based on and recognized with such an especialized method of design vector pertubation, here-in-after, called the *directional search*. It associates the vector sum variation of $\underline{X}_i$ with some baseline directions. Two arbitrary vectors $\underline{q}_i$ and $\underline{q}_j$ are defined to be in the same direction if any of them can be obtained by scaling the other with a scalar product, b; e.g., $\underline{q}_i = b\underline{q}_j$. Hence, the ratio of corresponding components to the first is similar in both the vectors; that is a specific decomposition of the search space.

As an instance of PSO variants, *Particle Swarm Optimization with Random Direction*, has added an extra move direction to the relation (1) as:

$$\underline{V}_i^{k+1} = c_i\underline{V}_i^k + r_1c_c(\underline{P}_i^k - \underline{X}_i^k) + r_2c_s(\underline{B}^k - \underline{X}_i^k) + r_3c_r(\underline{R}_i^k - \underline{X}_i^k) \tag{3}$$

in which, $\underline{R}_i^k$ introduces a search direction randomly chosen at iteration k for particle i, amplified by the corresponding randomized bandwidth, r_3c_r .

3. PSEUDO-RANDOM DIRECTIONAL SEARCH

Search directions in PSO variants, are mixed through vector sum in every iteration of the search. However, it is not the only way to combine such terms. For any particle, *i*, in the swarm a search direction can be selected due to a stochastic procedure, instead. It will not limit the particle movement because different directions can be selected by a single particle during consequent iterations. Let's call any such term a *state* as a candidate option to be chosen by the i^{th} search agent.

A characteristic bi-partite graph is thus defined so that a vertice *i* in its first part associates with the particle *i* adjecent to any vertice *j* in the second part of the graph that corresponds to the j^{th} state of search directions . Any edge of such a graph, thus, relates a vertice in its first part to a second part vertice (Figure 1). Therefore, a subgraph of it with *N* edges represent *N* states each one selected by a distinct particle.

In the newly developed PDS, an i^{th} particle of the swarm selects the j^{th} term (state) of search directions using the pseduo-random relation:

$$j = \begin{cases} \arg\max\limits_{l}(P_{i,l}) & if \quad r \le q_0 \\ j^P & if \quad q_0 < r \le q_1 \\ j^R & otherwise \end{cases} \tag{4}$$

Where as j^R stands for randomly chosen state and j^P is determined using *Roulette Wheel* selection based on the $P_{i,l}$; the probability of the l^{th} direction term to be selected by the i^{th} particle as:

$$P_{i,l} = \frac{\tau_{i,l}}{\sum_{h=1}^{NumStates} \tau_{i,h}} \tag{5}$$

Similar to ant algorithms, $\tau_{i,l}$ is defined the remained amount of artificial pheromone trail on each edge of the characteristic graph. It is used as a short-term memory to indirectly extend the previous steps experiences to the current. The trail matrix is initiated by 1 in all the graph edges. Once the j^{th} direction term is assigned with the i^{th} particle, $\tau_{i,l}$ is increased by a predetermined value, $\Delta\tau$:

$$\tau_{i,j}^{(k+1)} = \tau_{i,j}^{(k)} + \Delta\tau \tag{6}$$

the amount of trail in any graph edge is gradually decreased using a predetermined evaporation ratio, ρ , according to the relation:

$$\tau_{i,j}^{(k+1)} = (1-\rho)\tau_{i,j}^{(k)} \tag{7}$$

Trail evaporation in further iteration may cause it to tend zero in some graph edges and consequently eliminate the chance of such edges to be selected in further iterations. Thus, normalizing the trail and limiting its lower bound to τ^{LB} the trial update strategy is completed as:

$$\tau_{i,j}{}^{(k)} = \begin{cases} \dfrac{\tau_{i,j}{}^{(k)}}{\min\limits_h(\tau_{i,h}{}^{(k)})} & if \quad \min\limits_h(\tau_{i,h}{}^{(k)}) \geq \tau^{LB} \\ \\ 1 & otherwise \\ \end{cases} \tag{8}$$

Hence, the trail deposit, evaporation and round-off truncation provides an artificial short-term memory for the algorithm in order not to get trapped in local optima during state transition. Once a direction *j* is selected by the particle *i*, the $\underline{X}^{k}_{State(j)}$ is used as a target for the movement velocity vector, $\underline{V}_i^{k+1}$:

$$\underline{V}_i^{k+1} = r\, c_{(j)} (\underline{X}^{k}_{State(j)} - \underline{X}_i^{k}) \tag{9}$$

The corresponding bandwidth for the j^{th} state is a predetermined coefficient, $c_{(j)}$, multiplied by *r*; taken unity for inertial term or chosen a uniformly generated random number in the range 0 to1 for other states. New position of particle, *i*, is then obtained by:

$$\underline{X}_i^{k+1} = \underline{X}_i^{k} + \underline{V}_i^{k+1} \tag{10}$$

As an instance of such general methodology, in this paper four states are taken into account as $\underline{X}_i^{k}$; previous position of the particle representing inertial term, $\underline{P}_i^{k}$; the previous best position of the i^{th} particle being the cognitive term, $\underline{B}^{k}$ the global best position or social term and $\underline{R}_i^{k}$; a randomly oriented particle for higher search diversity:

$$\underline{X}^{k}_{State(\{1,2,3,4\})} = \{\underline{X}_i^{k}, \underline{P}_i^{k}, \underline{B}^{k}, \underline{R}_i^{k}\} \tag{11}$$

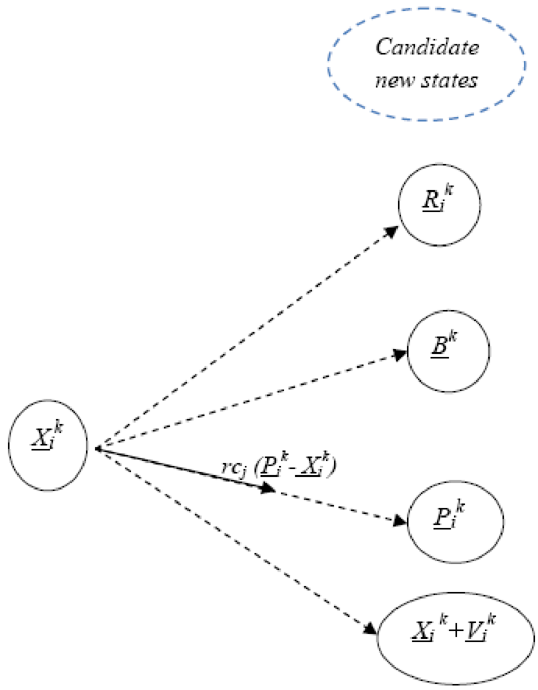

Figure 1. Sample chosen direction state (bold) of the PDS characteristic subgraph (dash arrows)

4. NUMERICAL EXAMPLES

A number of problems are consider to test PSO and PDS as follows. The employed version of PSO has a random additional term with respect to standard particle swarm optimizaton as described in Eq.3. The same values for the 4 coefficients are used in both the methods. Table 1 reveals the employed control parameters taken similar in all the examples for true comparison. The other control parameters are selected regarding evaporation ratio, ρ, as $\tau^{LB} = \rho$, $\Delta\tau = \rho$, $q_0 = \rho$ and $q_1 = 1$.

A number of well-nown test functions in literature are arranged and employed due to complexity of their search spaces [13,14]. For the treated test functions, a unified problem formulation is used as:

$$\begin{aligned} &Maximize \quad Fitness(\underline{X}) = -f(\underline{X}) \\ &Subject.to \\ &\qquad x^{LB} \le x_i \le x^{UB} \qquad i = 1,\ldots,N \end{aligned} \tag{12}$$

where $\underline{X} = < x_i >$ is the vector of N design variables constrained to be in the continuous domain of [x^{LB}, x^{UB}]. These lower and upper bounds are selected according to Table 2. In this study the dimension N is taken 2 for illustration purposes.

Table 1. Control parameters for the PSO and PDS

ρ	C_i	Cc	C_s	C_r	**Number of Particles**	**Number of Iterations**
0.1	2.0	2.0	2.0	2.0	15	500

Performance of PDS and PSO for each problem is treated for a number of trials and the resulted statistical parameters are given in Table 2. For the sake of true comparison, the randomly initiated population in the first method is saved and identically used for the second so that the resulting convergence curves have the same starting point. Thus, a fitness improvement factor *F.I.* is defined for each run of the optimization algorithm as:

$$FI = \frac{Fitness^{NI} - Fitness^{1}}{Fitness^{1}} \tag{13}$$

in which, *NI* denotes the total number of iterations in the search. This way, the *F.I.* values can be combared between PDS and PSO by the following relation:

$$FI_{Ratio} = \frac{FI^{PDS}}{FI^{PSO}} \tag{14}$$

Such a dimensionless ratio will be a measure of how better is the PDS performance over PSO. Sorting the trial runs are in ascending order of their FI_{Ratio}, the median results are extracted and ploted as the Present Works; PW-1~4 for treated test problems in the consequent Figures. A brief reiew of each function is given, followed by its obtained results in both the methods.The results' statistics for all test functions are summarized in Table 2.

Table 2. Comparison of PDS and PSO results for continuous test problems

Test Function	**Domain** $[x^{LB}, x^{UB}]$	**Statistical Item**	**F.I.ratio PDS/PSO**	**F.I. PSO**	**F.I. PDS**
DeJong	[-5.12,5.12]	*Mean.Value*	1.004	0.995	0.999
		Min	1.000	0.998	1.000
		Max.	1.010	0.989	0.999
		PW-1(Median)	1.004	0.996	1.000
Rastrigin	[-5.12,5.12]	*Mean.Value*	1.125	0.864	0.964
		Min	1.017	0.936	0.953
		Max.	1.243	0.749	0.987
		PW-2(Median)	1.116	0.834	0.930

Griewangk	[-10,10]	*Mean.Value*	1.314	0.822	0.996
		Min	1.015	0.984	1.000
		Max.	2.207	0.452	0.999
		PW-3(Median)	1.138	0.878	1.000
Ackley	[-32.7,32.7]	*Mean.Value*	1.215	0.798	0.964
		Min	1.472	0.668	0.982
		Max.	1.109	0.874	0.967
		PW-4(Median)	1.179	0.796	0.939

4.1. De Jong's first function

The first example is *De Jong's first function* having one of the simplest search spaces. It is a unimodal and convex test function given by the following general relation. There is no local optima but one global optimum 0 at $\underline{X} = \underline{0}$ for this minimization problem (Figure 2).

$$f(\underline{X}) = \sum_{i=1}^{N} x_i^2 \tag{15}$$

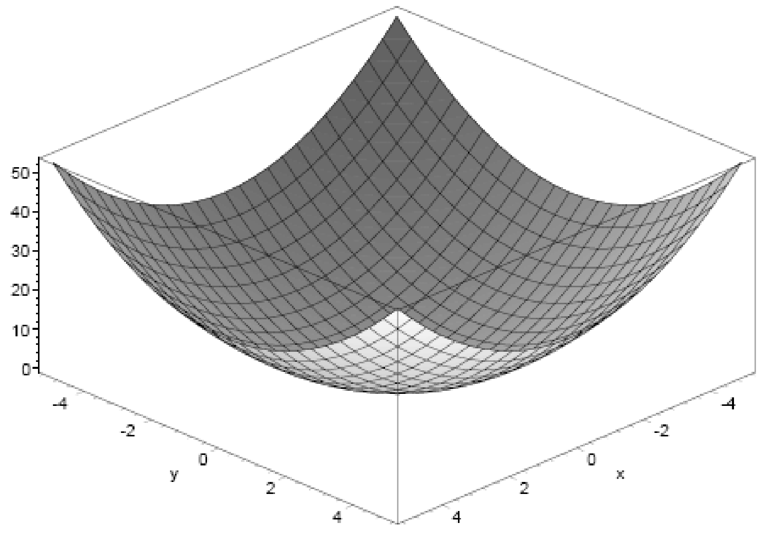

Figure 2. De Jong 2-dimensional test function for $\underline{X} = < x, y >$ [14]

Figure 3 shows the result of present work-1 for this example, in which both methods got close to global optimum but PDS efficiency has been more than PDS. It is confirmed by other statistical achievements in Table 2.

4.2. Rastrigin's function

In the second example, *Rastrigin's* formula is considered as a multimodal test function:

$$f(\underline{X}) = 10N + \sum_{i=1}^{N} \left[x_i^2 - 10\cos(2\pi . x_i) \right] \tag{16}$$

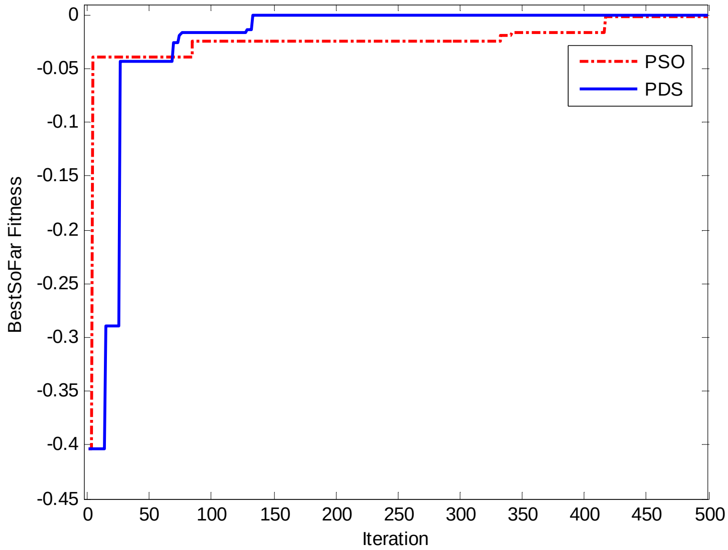

Figure 3. Convergence curves for De Jong's function in PW-1

The cosine term has added several local optima to the search space with respect to De Jong's first function while one global optimum yet exists at at $\underline{X} = \underline{0}$ (Figure 4).

According to the result of PW-2 in Figure 5, the PDS has stood considerably higher than PSO for this example with more rapid convergence.

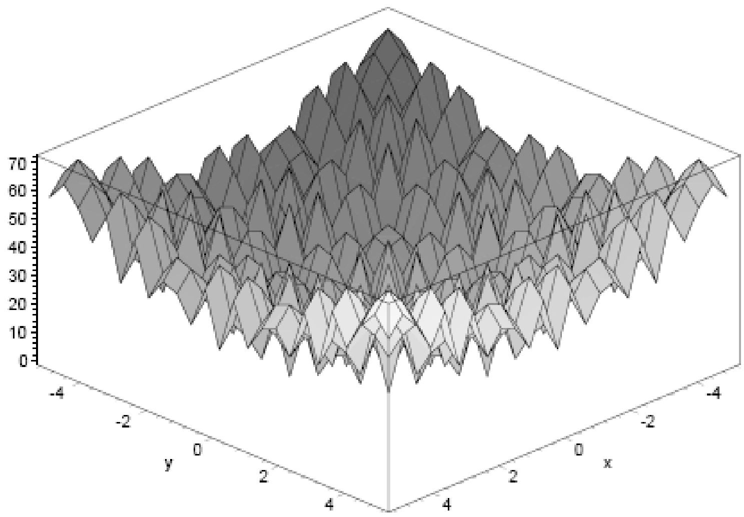

Figure 4. A view of Rastrigin's test function [14]

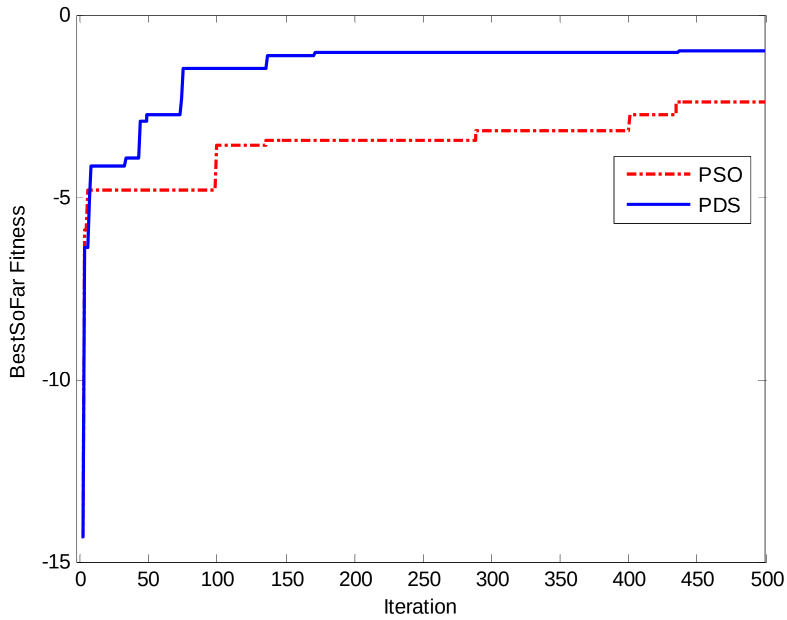

Figure 5. Convergence curves for Rastrigin's test function in PW-2

4.3. Griewangk's function

Optimization methods are tested by *Griewangks's function* in this example with the relation:

$$f(\underline{X}) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{j=1}^{N}\cos\left(\frac{x_j}{\sqrt{j}}\right) + 1 \qquad (17)$$

The function has several local optima in detail scale with one global optimum of $f(\underline{0}) = \underline{0}$ (Figure 6). Despite the PDS, in this example (for PW-3) the PSO is trapped in local optima or premature convergence as shown in Figure 7.

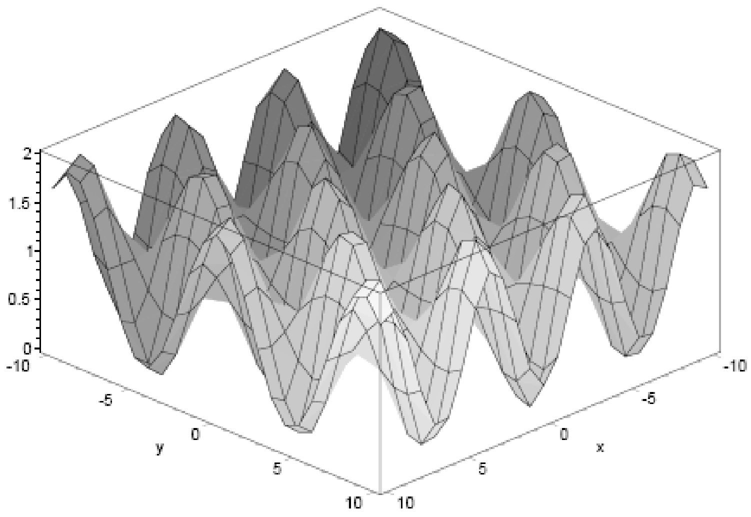

Figure 6. Griewangk's test function [14]

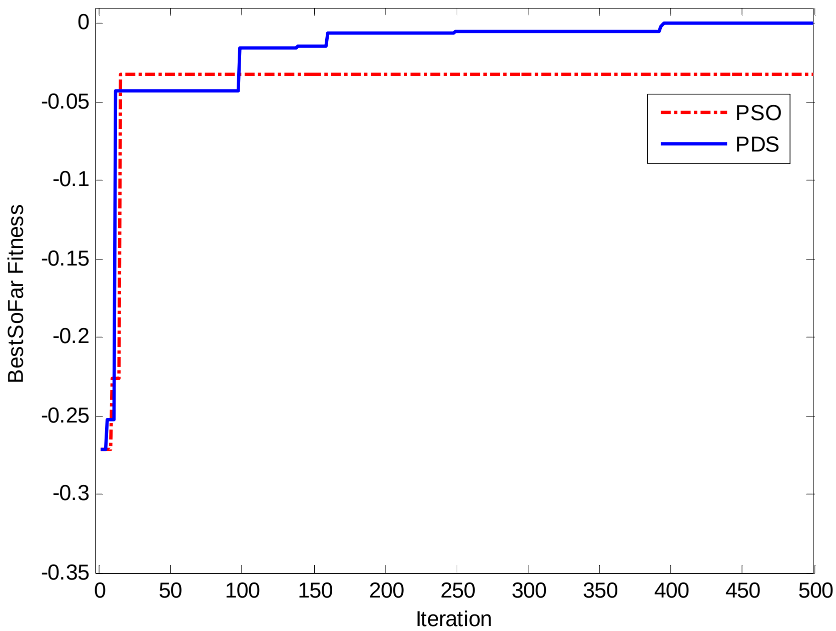

Figure 7. Convergence curves for Griewangk's test function in PW-3

4.4. Ackley's function

For this example, *Ackley's test function*, with a multimodal search space is selected, by the following definition:

$$f(\underline{X}) = -20\exp(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} - \exp(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi.x_i)) + 20 + e \tag{18}$$

This function has several local optima in detail scale with one global optimum of $f(\underline{0}) = \underline{0}$, but the local minima is far abow the global minimum in a relatively narrow region (Figure 8). Result of the present work PW-4 shows that using PSO,the *best-so-far* fitness has fallen well below the proposed PSD even in earlier iterations of the search (Figure 9).

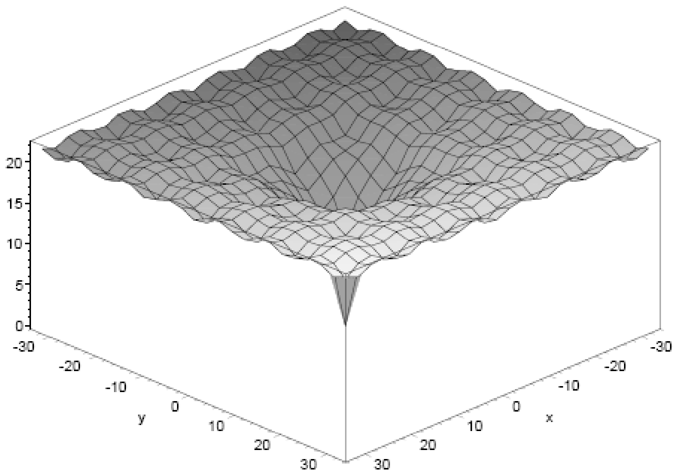

Figure 8. Ackley's test function [14]

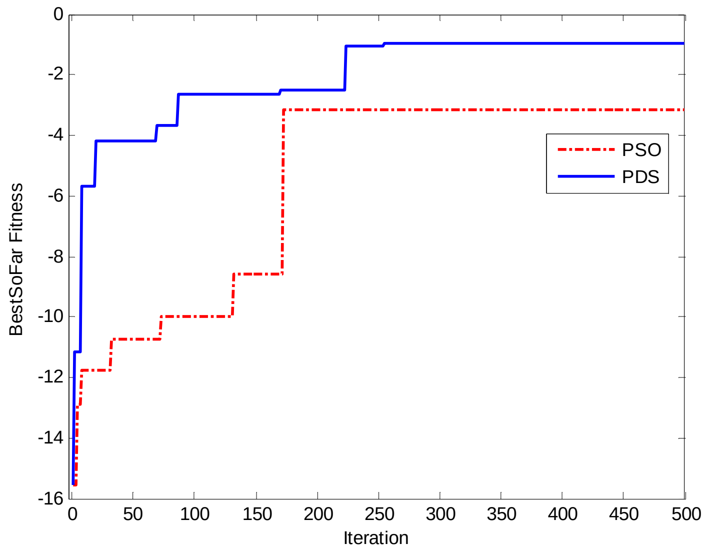

Figure 9. Convergence curves for Ackley's test function in PW-4

As can be realized the mean FI^{Ratio} has increased in last more complex test functions. While its the maximum has reached about 2 times in the studied cases, the minimal ratio is obtained at least 1 or greater denoting superirity of the proposed method.

4.5. discrete Ackley's function

In this example, the domain of which x_i can be peaked is discrete; that is integer numbers in range {-32,-31,...,31,32} for the Ackley's 2-dimensional test function. According to Figure 10, the PSO may result in much slower convergence speed than the present PDS for such a discrete problem, however, might capture the global optimum in further runs (Table 3).

Table 3. Comparison of PDS and PSO results for discrete test problem

Test Function	**Domain** **[x^{LB} , x^{UB}]**	**Statistical Item**	**F.I.ratio PDS/PSO**	**F.I. PSO**	**F.I. PDS**
Ackley	{-32,...,32}	*Mean Value*	1.034	0.971	1.000
		Min	1.000	1.000	1.000
		Max (PW-5)	1.168	0.855	1.000

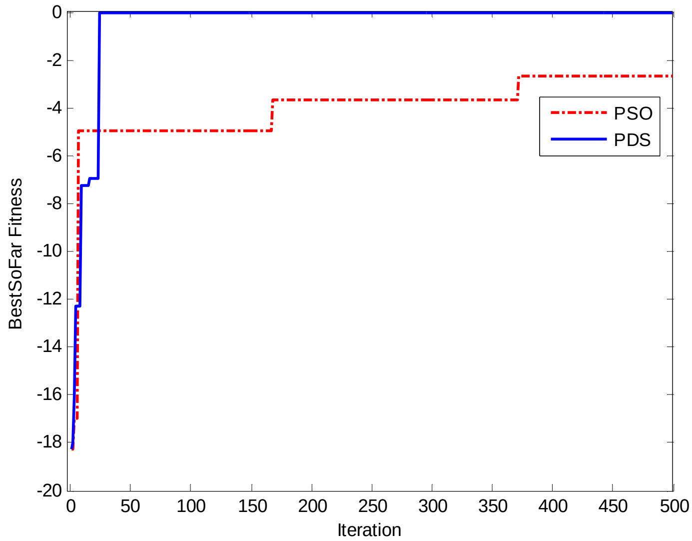

Figure 10. Convergence curves for discrete Ackley's test function in PW-5

4.6. sizing of a 26-bar truss example

A 14-node 26-bar truss example is considered for this example. Boundary conditions and loading are depicted in Figure 11. For all truss members material properties are taken as $E = 68.95 GN/m^2$, $\rho = 2712 kg/m^3$ while the allowable stress is $\sigma = \pm 172 MN/m^2$. The displacement constraint is given as $|\Delta| \leq 50.8 mm$. Member sections may be selected from Table 4. Since each truss member will be assigned a section separately, it is a 26-fold (number of variables) problem. Thus, size of the discrete search space is 6^{26}; that is of order 10^{20}. In this example, some of the search space points corresponds truss sizing models which are infeasible because of their constraint violation.

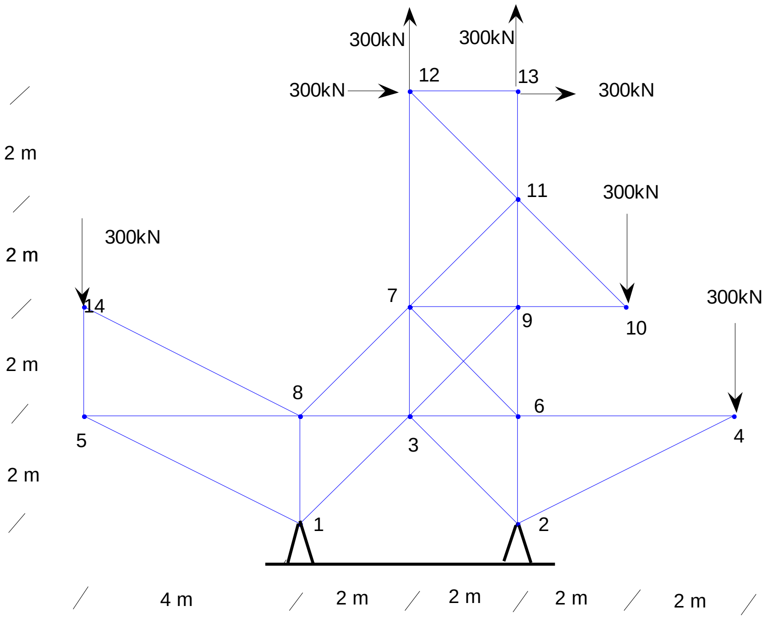

Figure 11. The 26-bar truss

Table 4. Section list for sizing 26-bar truss

Section area (mm^2)	1045	5142	9161	14193	14774	21613

The algorithms are run for 200 iterations and even with 5 particles the results are as good as in Table 5. The objective function is the total truss weight being minimized under the stress and displacement constraints. Hence the penalized objective function is utilized in the fitness as:

$$Fitness(\underline{X}) = -W * (1 + Kp * \sum_{i=1}^{m} C_i) \quad (19)$$

Where W denotes the total structural weight, Kp, is the penalty coefficient desired by the user and C_i stands for the amount of violation in any violated constraint. The design variable X_j can take its value as a section index between 1 and 6 to, assigned for the corresponding truss member. The equilibrium is satisfied for evaluation of member stresses and nodal displacements.

Table 5. Comparison of PDS and PSO results for 26-bar truss

Statistical Item	**F.I.ratio PSO**	**F.I.ratio PDS**	**F.I.ratio PDS/PSO**	**PSO Elitist TrussWeigh t** (kg)	**PDS Elitist TrussWeigh t** (kg)
Mean.Value	0.52	0.52	1.00	2280	2280

Min	0.38	0.25	0.65	2001	2130
Max. (PW-6)	0.67	0.63	1.45	2560	2480

According to Table 5, the achieved mean values in this example are similar for both the methods. However, arising more infeasible designs in the elitist results of the PSO is a drawback of it against PDS when using the penalty coefficient, *Kp*, of 10.

6. CONCLUSION

A class of metaheuristic procedures were reviewed which decompose the search space using search vector directions. Well known particle swarm optimization and its variants take benefits of such a directional search approach using a vector sum within each iteration.

A variety of test functions were then selected form literature with various search spaces to test performance of the developed method vs. particle swarm regarding similar states. In this study the states are taken as inertial, cognitive, social and an additional random direction in the search space.

Superiority of the developed PDS over PSO in the treated examples was declared as the search space altered from uni-modal to multi-modal with additional several local optima. In the Griewangks' function global and local optima are difficult to identify because of their close fitness values and the search space shape; for which PSO led to premature convergence while PDS successfully overcame such a challenge. PDS also showed reasonable higher effectiveness and efficiency even in the Ackley's test function with realtively narrow global optimum region and steep fitness variation. In addition, testing both methods with a discrete search space confirms that PDS can lead to more effectiveness than PSO.

The search effectiveness achieved in the results confirmed that the proposed directional state-tranistion rule can work even if each state addresses a formula rather than its particle position that is dynamically changing via iterations of the search.

It is worth mentaining that only one direction of movement is accomplished in every step of PDS while a sum of different move vectors is used in any search iteration of PSO. Considering this matter, the convergence speed of PDS might be expected lower, however, the treated examples showed at least comparable performance of the proposed PDS with respect to PSO with a minor parameter tuning and specially for unconstrained problems.

REFERENCES

1. Topping BHV. Mathematical programming techniques for shape optimization of skeletal structures, Shape & Layout Opt of Struct Sys & Opt Criteria Meth; Rozvany GIN (ed.), Springer, Vienna, 1992.
2. Rozvany GIN. Structural design via optimality criteria, Kluwer, Dordrecht, 1989.
3. Kirkpatrick S, Gelatto CD, Vecchi MP. Optimization by simulated annealing, *Science* 1983; **220**: 671–80.

4. Holland J. Adaptation in natural and artificial systems, an introductory analysis with application to biology, control and artificial intelligence, Ann Arbor, The University of Michigan Press, 1975.
5. Coello CA. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, *Comp Meth in Appl Mech Eng* 2002; **191**: 1245-87.
6. Dorigo MA, Colorni, A, Maniezzo V. The ant system: optimization by a colony of cooperating agents, *IEEE Trans Sys Man Cybernetics* 1996; 26(1):1-13.
7. Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search, Simulation 2001; **76**(2): 60-8.
8. Kennedy J, Eberhart R. Particle swarm optimization, Proc 1995 Int Conf Neural Networks, Perth, Australia, 1995, pp. 1942-1948.
9. Kennedy J, Eberhart R. Swarm Intelligence, Academic Press, London, 2001.
10. Kaveh A, Talatahari S. Novel heuristic optimization method: charged system search, *Acta Mech* 2010; **213**(3-4): 267-86.
11. Rashedi E, Nezamabadipour H., Saryazdi S. GSA: a gravitational search algorithm. *Info Sciences* 2009; **179** (13): 2232–48.
12. Atashpaz-Gargari E, Hashemzadeh F, Rajabioun R, Lucas C. Colonial competitive algorithm: A novel approach for PID controller design in MIMO distillation column process, *Int J Intelligent Comput Cybernetics* 2008; **1**(3): 337-55.
13. Yang XS. Firefly algorithm stochastic test functions and design optimization, *Int J Bio-inspired Comput* 2010; **2**(2): 78-84.
14. Molga M, Smutnicki C. Test functions for optimization needs, 1995; www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf