ASIAN JOURNAL OF CIVIL ENGINEERING (BUILDING AND HOUSING) VOL. 11, NO. 6 (2010)
PAGES 777-793

OPTIMUM DESIGN OF STRUCTURES BY AN IMPROVED PARTICLE SWARM ALGORITHM

S. Gholizadeh*
Department of Civil Engineering, Urmia University, Urmia, Iran

ABSTRACT

In the present study, an efficient optimization algorithm is proposed to optimal design of structures. The proposed algorithm is an improved particle swarm optimization (PSO) which its global search performance is enhanced by employing the concept of cellular automata (CA). In the so-called improved particle swarm optimization (IPSO) algorithm a new cellular automata based term is added to the conventional velocity equation. Also, the real-values of design variables are used and the artificial evolution is evolved on a small dimensioned grid. To show the computational advantages of the IPSO two numerical examples are presented. Using the new IPSO, not only the algorithm converges to a better solution but also the number of structural analyses is significantly reduced compared with the other existing variants of PSO algorithm.

Keywords: Optimization; cellular automata; particle swarm optimization; cellular velocity updating equation

1. INTRODUCTION

Optimum design of structures is usually achieved by selecting the design variables such that an objective function is minimized while all of the design constraints are satisfied. The great development of structural optimization took place in the early 1960s, when programming techniques were used in the minimization of structure weight [1-2]. From then on, various general approaches have been developed and adopted to structural optimization. In the last years, advanced algorithms are developed and widely used for solving structural optimization problems. Some evolutionary algorithms such as genetic algorithms (GA) [3–6], particle swarm optimization (PSO) [7-9] and ant colony optimization (ACO) [10–11] have been used for optimization of structures. These optimization algorithms are popular and widely used due to their high potential for modeling engineering problems and simple programming in computers. These optimization algorithms have many similarities. All of them explore the design space by a population of potential designs using some artificial evolution operators with random nature. They have also some differences in terms of the

* E-mail address of the corresponding author: s.gholizadeh@urmia.ac.ir (S. Gholizadeh)

type of operations used to create new solutions and the mechanism of selecting new population. The main drawback of the mentioned advanced algorithms is a slow rate of convergence which increases the computational burden of the optimization process.

An efficient structural optimization algorithm should possess two important characteristics: the ability of finding the global optimum without trapping into local optima and requiring fewer structural analyses to be performed during the optimization process. Some researchers [7,12,13] have developed structural optimization algorithms that can find only the global optimum. But they have not considered too much about the structural analysis reduction in their works. Therefore, the studies related to analysis reduction in structural optimization are comparatively fewer.

The main objective of the present study is to propose an improved PSO algorithm for structural optimization that simultaneously possesses the two mentioned important characteristics. For this purpose, the concepts of cellular automata (CA) [14] and PSO are hybridized and the resulted hybrid algorithm is termed as improved particle swarm optimization (IPSO) algorithm. Basically, CA represents simple mathematical idealizations of physical systems in which space and time are discrete and physical quantities are taken from a finite set of discrete values. Models based on CA provide an alternative and more general approach to physical modeling rather than an approximation. In the proposed IPSO, the particles are distributed on a small dimensioned grid and the artificial evolution is evolved by a novel velocity updating equation. In the novel equation, the velocity is defined by adding a new cellular automata based term to the conventional equation. The original PSO can not control the balance between exploration (global investigation of the search place) and exploitation (the fine search around a local optimum) [15]. The proposed IPSO eliminates this difficulty and also reduces the number of required structural analyses during the optimization process compared with the PSO algorithm.

To demonstrate the computational advantages of the proposed IPSO optimization algorithm two benchmark examples, a 10-bar planner truss and a 72-bar space truss structures subject to static displacement and stress constraints, are optimized. The numerical results imply that the proposed IPSO is an efficient and powerful algorithm for optimal design of structures.

The organization of the present paper is as follows: In Section 2, the formulation of optimization problem is presented. The standard PSO algorithm is briefly described in Section 3. The concept of CA is introduced in section 4. In Section 5, IPSO algorithm is explained. In Section 6, the numerical results are presented. And finally the concluding remarks are mentioned in section 7.

2. FORMULATION OF OPTIMIZATION PROBLEM

An optimization problem is usually formulated in the following form:

$$\begin{aligned} &\text{Minimize} && f(X) \\ &\text{Subject to} && g_i(\mathrm{X}) \le 0 \quad , \quad i = 1,2,\cdots,m \\ & && X_j \in R^d \quad , \quad j = 1,2,\cdots,n \end{aligned} \tag{1}$$

where $f(X)$ represents the objective function, $g(X)$ is the behavioral constraint, m and n are the number of constraints and design variables, respectively. A given set of discrete values is expressed by R^d.

In constrained optimization problems, constraints are mostly handled by using the concept of penalty functions, which penalize infeasible solutions as follows:

$$f_s(X)=\begin{cases} W(X) & \text{if } X\in\tilde{\Delta} \\ W(X)+f_p(X) & \text{otherwise} \end{cases},\ f_p(X)=r_P\left[\sum_{i=1}^{m}\left(\max(g_i(X),0)\right)^2\right] \tag{2}$$

where $f_s(X)$ and $f_p(X)$ are supplemental and penalty functions, respectively. Also, $\tilde{\Delta}$ and r_p are the feasible search space and an adjusting coefficient, respectively.

3. PARTICLE SWARM OPTIMIZATION ALGORITHM

The PSO has been proposed by Kennedy [16] to simulate the graceful motion of bird swarms as a part of a socio-cognitive study. The PSO involves a number of particles, which are randomly initialized in the search space. These particles are referred to as swarm. Each particle of the swarm represents a potential solution of the optimization problem. The particles fly through the search space and their positions are updated based on the best positions of individual particles and the best of the swarm in each iteration. The objective function is evaluated for each particle at each grid point and the fitness values of particles are obtained to determine the best position in the search space [17]. In iteration k, the swarm is updated using the following equations:

$$V_i^{k+1}=\omega^k V_i^k+c_1 r_1(P_i^k-X_i^k)+c_2 r_2(P_g^k-X_i^k) \tag{3}$$

$$X_i^{k+1}=X_i^k+V_i^{k+1} \tag{4}$$

where X_i and V_i represent the current position and the velocity of the ith particle, respectively; P_i is the best previous position of the ith particle (called *pbest*) and P_g is the best global position among all the particles in the swarm (called *gbest*); r_1 and r_2 are two uniform random sequences generated from interval [0, 1]; c_1 and c_2 are the cognitive and social scaling parameters, respectively. Each component of V_i is constrained to a maximum value defined as V_i^{max} and a minimum value defined as V_i^{min}. The inertia weight used to discount the previous velocity of particle preserved is expressed by ω.

Due to the importance of ω in achieving efficient search behavior the optimal updating criterion is taken as:

$$\omega=\omega_{max}-\frac{\omega_{max}-\omega_{min}}{k_{max}}.k \tag{5}$$

where ω_{max} and ω_{min} are the maximum and minimum values of ω, respectively. Also, k_{max}, and k are the number of maximum iterations and the number of present iteration, respectively.

4. CELLULAR AUTOMATA

Cellular automata (CA) were firstly introduced by von Neumann [14] and subsequently developed by other researchers in many fields of science. Basically, CA represents simple mathematical idealizations of physical systems in which space and time are discrete, and physical quantities are taken from a finite set of discrete values. Models based on CA provide an alternative and more general approach to physical modeling rather than an approximation. The CA shows a complex behavior analogous to that associated with complex differential equations, but in this case complexity emerges from the interaction of simple entities following simple rules.

In its basic form, a cellular automaton consists of a regular uniform grid of sites or cells with a discrete variable in each cell which can take on a finite number of states. The state of the cellular automaton is then completely specified by the values $s_i = s_i(t)$ of the variables at each cell i. During time, cellular automata evolve in discrete time steps according to a parallel state transition determined by a set of local rules: the variables $s_i^{k+1} = s_i(t_{k+1})$ at each site i at time t_{k+1} are updated synchronously based on the values of the variables $s_{n_c}^k$ in their n_c neighborhood at the preceding time instant t_k. The neighborhood n_c of a cell i is typically taken to be the cell itself and a set of adjacent cells within a given radius r; $i-r \le n_c \le i+r$. Thus, the dynamics of a cellular automaton can be formally represented as follows [18]:

$$s_i^{k+1} = \theta(s_i^k, s_{n_c}^k), \quad i-r \le n_c \le i+r \tag{6}$$

where the function θ is the evolutionary rule of the automaton.

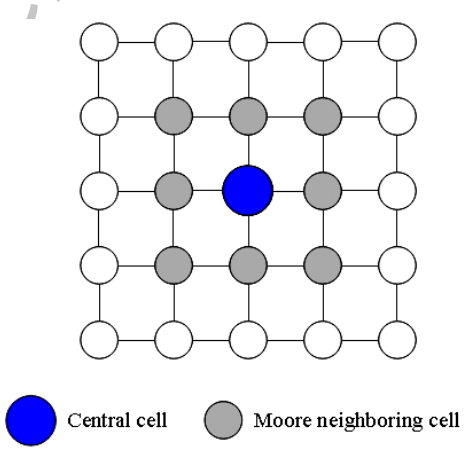

Figure 1. Cellular Automata Moore neighborhood

One of the most important features of CA is the neighborhood structure. For updating the value of a cell, its own value and the values of neighboring cells should be considered. Configuration of the neighborhood structure is highly problem dependent and depends on the nature of the physical phenomenon that should be modeled. Clearly, a proper choice of the neighborhood plays a crucial role in determining the effectiveness of such a rule. In this paper, the widely used Moore neighborhood of interaction [18], by r=1, is adopted and shown in Figure 1 for a 2D grid.

5. IMPROVED PARTICLE SWARM OPTIMIZATION ALGORITHM

In the field of structural optimization, a number of researchers tried to enhance the performance of PSO algorithm. The followed computational strategies in this regard include two main classes. In the first class, researchers have combined PSO with the other optimization algorithms. Such hybrid optimization algorithms may be found in Refs. [8,19,20]. In the second class, basic velocity equation of the standard PSO has been enhanced by adding some terms to it. In Ref. [7] such modified equation has been proposed. In the present paper, following the idea of the second class to discover a novel optimization mechanism through simulation of a social model, a new term is added to Eq. (3) based on concepts of CA. The CA technique can be combined with the evolutionary algorithms to solve numerical optimization problems. In the field of structural optimization some of researchers [21-22] have combined the concepts of CA and GA to create cellular genetic algorithms (CGA) but combination of CA with the other types of the evolutionary algorithms have not been yet reported.

In the present paper CA and PSO are integrated and a cellular automata-based PSO algorithm is proposed to optimal design of structures. In the proposed IPSO algorithm, particles are set on discrete locations of a 2D grid. The state variables associated with each cell site are simply the design variables of the optimization problem. In the traditional PSO, the evolution is accomplished by applying Eqs. (3) and (4). In the IPSO, this type of evolution is substituted by an evolutionary rule of the automaton or the local rules of interaction among neighboring members of the grid which are simultaneously applied to all particles of the swarm. In other words, in the IPSO the evolution process is accomplished locally, with probabilistic interaction rules applied synchronously to each central site, and using information from members of its Moore neighborhood. When the swarm has been updated, the evolutionary rules of the automaton are repeated until one of the stopping criteria is met. In both the PSO and IPSO, the objective function of the optimization problem is employed to define the fitness of each design vector. In this paper, a new velocity updating equation the so-called cellular velocity updating equation is presented. Also, the real-values of design variables, instead of their binary codes, are considered. In the proposed IPSO algorithm, the evolutionary rule of the automaton includes the cellular velocity updating equation and after this the Eq. (4) are applied to the sites.

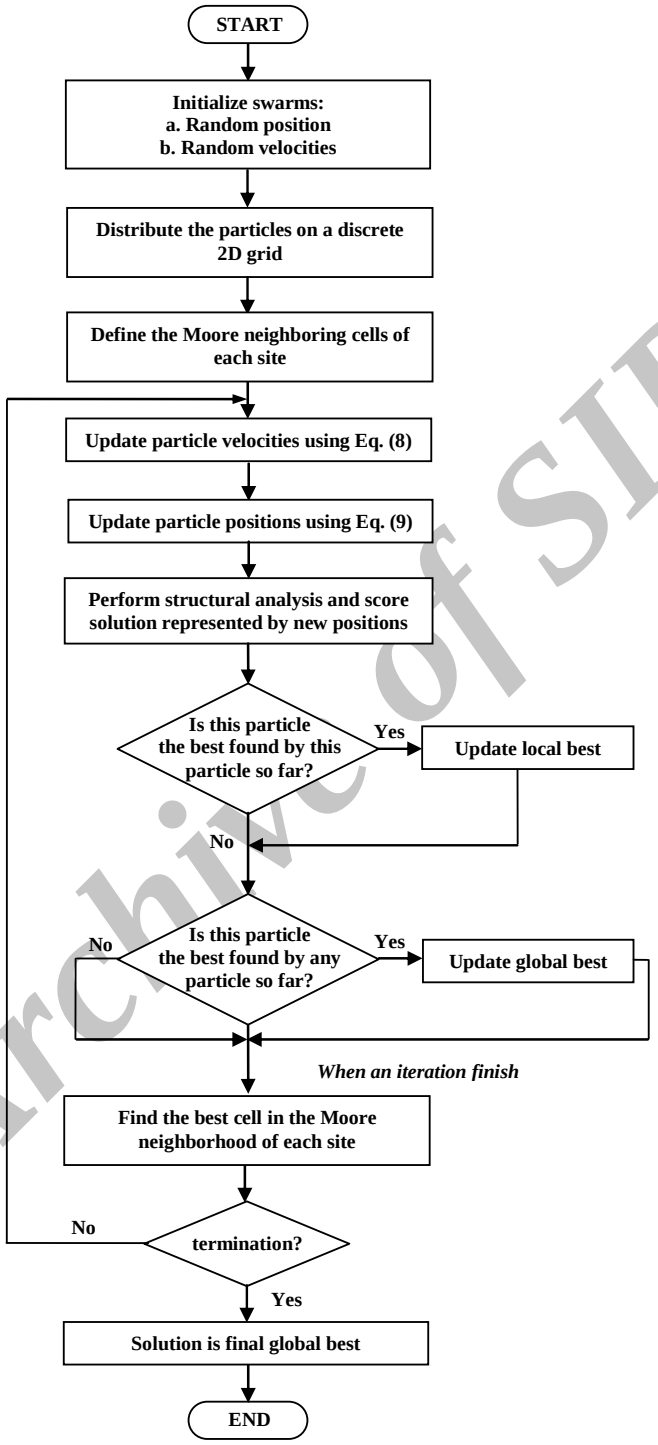

Figure 2. Flowchart of the proposed IPSO algorithm

5.1 Cellular Velocity Updating Equation

In the proposed IPSO algorithm, a swarm of potential designs is structured on a 2D grid. In this case, each site contains a real-valued vector describing of a design and therefore the state of the cellular automaton in each site is a design vector of design variables.

$$s_i \to X_i = \{x_1, x_2, ..., x_n\}^{\mathrm{T}}, i = 1,2,...,n_c \tag{7}$$

The proposed cellular velocity updating equation acts on the design variables and combines the information available at the central site and its immediate neighbors. In this study, in each discrete time step, the proposed cellular velocity updating equation produces a new design at each site using existing information in immediate neighbors of each central site as follows:

$$V_i^{k+1} = \omega^k V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + \frac{c_3 r_3}{n_c} (\sum_{j=1}^{n_c} (X_i^{\text{best}} - X_{i,j})) \tag{8}$$

where r_3 is a uniform random number generated from interval [0, 1]; c_3 is a scaling parameter. X_i^{best} is the best particle in immediate neighbors of ith central cell. $X_{i,j}$ is the jth particle in immediate neighbors of ith central cell.

In each iteration or in each discrete time step, the proposed equation produces a new design at each site according to the following equation:

$$s_i^{k+1} = \theta(s_i^k, s_{n_c}^k) \to X_i^{k+1} = X_i^k + \omega^k V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) + \frac{c_3 r_3}{n_c} (\sum_{j=1}^{n_c} (X_i^{\text{best}} - X_{i,j})) \tag{9}$$

In comparison with Eq. (3) of standard PSO algorithm, Eq. (9) in IPSO uses more information to update the velocity of particles. Therefore it is expected that the IPSO possesses better performance than the standard PSO. The flowchart of the proposed IPSO algorithm is shown in Figure 2.

6. NUMERICAL RESULTS

In the present paper two numerical examples, a 10-bar planar truss and a 72-bar space truss structures, are optimized by PSO and IPSO to investigate the computational advantages of the proposed optimization algorithm. The selected test examples have been optimized by other researchers and the results obtained in this paper are compared with their results. For both examples, a small dimensioned grid of 5×5 (25 particles) is considered and the maximum iterations are 500. This means that the maximum number of analyses is 12500. The value of ω_{max} and ω_{min} are considered to be 0.9 and 0.4, respectively. Also the value of c_1 and c_2 are set 0.5 and c_3 is taken as 0.65. To achieve optimization processes a personal Pentium IV 3000MHz is employed.

6.1 Example 1: 10-bar planar truss structure

The 10-bar truss structure is shown in Figure 3. The loads applied to the truss are $P_1 = 10^5$ lbs, $P_2 = 0$. The material density is 0.1 lb/in^3 and the modulus of elasticity is 10,000 ksi. The members are subjected to stress limitations of ±25 ksi. All nodes in both directions are subjected to displacement limitations of ±2.0 in [7].

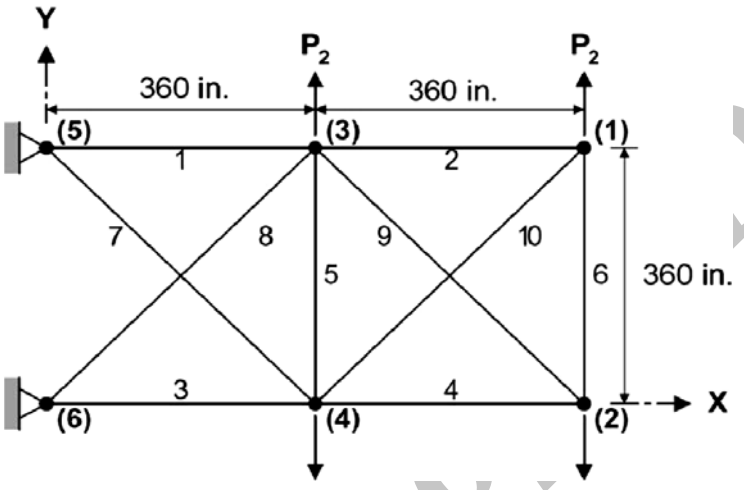

Figure 3. 10-bar truss structure

There are 10 design variables and two design cases in this example are considered. In case 1: the discrete variables are selected from the set D = {1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.50, 13.50, 13.90, 14.20, 15.50, 16.00, 16.90, 18.80, 19.90, 22.00, 22.90, 26.50, 30.00, 33.50} (in^2);

In case 2: the discrete variables are selected from the set D = {0.1, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0, 11.5, 12.0, 12.5, 13.0, 13.5, 14.0, 14.5, 15.0, 15.5, 16.0, 16.5, 17.0, 17.5, 18.0, 18.5, 19.0, 19.5, 20.0, 20.5, 21.0, 21.5, 22.0, 22.5, 23.0, 23.5, 24.0, 24.5, 25.0, 25.5, 26.0, 26.5, 27.0, 27.5, 28.0, 28.5, 29.0, 29.5, 30.0, 30.5, 31.0, 31.5} (in^2).

The comparison of optimal designs for the 10-bar planar truss structure under two load cases are given in Tables 1 and 2, respectively. In these tables the maximum nodal deflection and element stress are shown by $|d_{\max}|$ and $|\sigma_{\max}|$, respectively. To investigate the computational performance of the proposed IPSO in this example, 50 independent runs are implemented and statistical results of these runs are given in Table 3 for both cases.

Table 1: Comparison of optimal designs for the 10-bar truss structure (case 1)

Design variables No.	Li et al. [7]			Present study	
	PSO	PSOPC	HPSO	PSO	IPSO
1	30.00	30.00	30.00	30.00	33.50
2	1.62	1.80	1.62	1.62	1.62
3	30.00	26.50	22.90	26.50	22.90
4	13.50	15.50	13.50	13.50	14.20
5	1.62	1.62	1.62	1.62	1.62
6	1.80	1.62	1.62	1.62	1.62
7	11.50	11.50	7.97	11.50	7.97
8	18.80	18.80	26.50	19.90	22.90
9	22.00	22.00	22.00	22.00	22.00
10	1.80	3.09	1.80	3.09	1.62
Weight	5581.76	5593.44	5531.98	5570.96	5490.74
Number of analyses	50000	50000	50000	10000	4512
$\lvert d_{\max}\rvert$	1.999	1.995	1.999	2.000	1.999
$\lvert \sigma_{\max}\rvert$	11534.40	11482.38	14871.33	11383.05	14196.93

Table 2: Comparison of optimal designs for the 10-bar truss structure (case 2)

Design variables No.	Li et al. [7]			Present study	
	PSO	PSOPC	HPSO	PSO	IPSO
1	24.5	25.5	31.5	31.5	29.5
2	0.1	0.1	0.1	0.1	0.1
3	22.5	23.5	24.5	29.0	24.0
4	15.5	18.5	15.5	13.5	15.0
5	0.1	0.1	0.1	0.1	0.1
6	1.5	0.5	0.5	0.1	0.5
7	8.5	7.5	7.5	8.0	7.5
8	21.5	21.5	20.5	25.5	21.5
9	27.5	23.5	20.5	16.5	21.5
10	0.1	0.1	0.1	0.1	0.1
Weight	5243.71	5133.16	5073.51	5225.47	5067.33
Number of analyses	50000	50000	50000	12500	5600
$\lvert d_{\max}\rvert$	1.999	2.000	1.999	2.000	1.999
$\lvert \sigma_{\max}\rvert$	24843.65	24713.85	24393.65	23570.62	24604.07

Table 3: Investigation on the performance of IPSO for the 10-bar truss in 50 runs

	Case 1	**Case 2**
Best weight	5490.74	5067.33
Worst weight	5513.32	5098.96
Average weight	5492.09	5074.38
Standard deviation	3.66	8.67
Minimum number of analyses	3968	5600
Maximum number of analyses	8000	8000
Average number of analyses	5408	7715

It can be observed from the tables that the best weight of the IPSO and HPSO [7] algorithms in Case 1 are 5490.74 lb and 5531.98 lb, respectively and in Case 2 are 5067.33lb and 5073.51 lb, respectively. Also the best number of required structural analyses during the optimization processes by IPSO in Cases 1 and 2 are 4512 and 5600, respectively while in the case of HPSO [7] this number is equal to 50000. Therefore, IPSO algorithm considerably reduces the computational burden of the optimization process in Case 1 (4512/50000 ≈ 0.09) and Case 2 (5600/50000 ≈ 0.11) with respect to the other variants of PSO.

6.2 Example 2: 72-bar Space truss structure

The 72-bar spatial truss structure is shown in Figure 4. The material density is 0.1 lb/in^3 and the modulus of elasticity is 10,000 ksi. The members are subjected to stress limitations of ±25 ksi. The uppermost nodes are subjected to displacement limitations of ±0.25 in both in x and y directions [7]. Two load cases are listed in Table 4. There are 72 members, which are divided into 16 groups, as follows: (1) A1–A4, (2) A5–A12, (3) A13–A16, (4) A17–A18, (5) A19–A22, (6) A23–A30 (7) A31–A34, (8) A35–A36, (9) A37–A40, (10) A41–A48, (11) A49–A52, (12) A53–A54, (13) A55–A58, (14) A59–A66 (15) A67–A70, (16) A71–A72.

Two optimization cases are considered. Case 1: The discrete variables are selected from the set D = {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8,0.9,1.0, 1.1,1.2, 1.3,1.4, 1.5,1.6, 1.7, 1.8, 1.9, 2.0, 2.1,2.2, 2.3,2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0, 3.1,3.2} (in^2). Case 2: The discrete variables are selected from Table 5.

Optimal design results for the 72-bar space truss structure are compared in Tables 6 and 7. In these tables the maximum nodal deflection and element stress are shown by $|d_{max}|$ and $|\sigma_{max}|$, respectively.

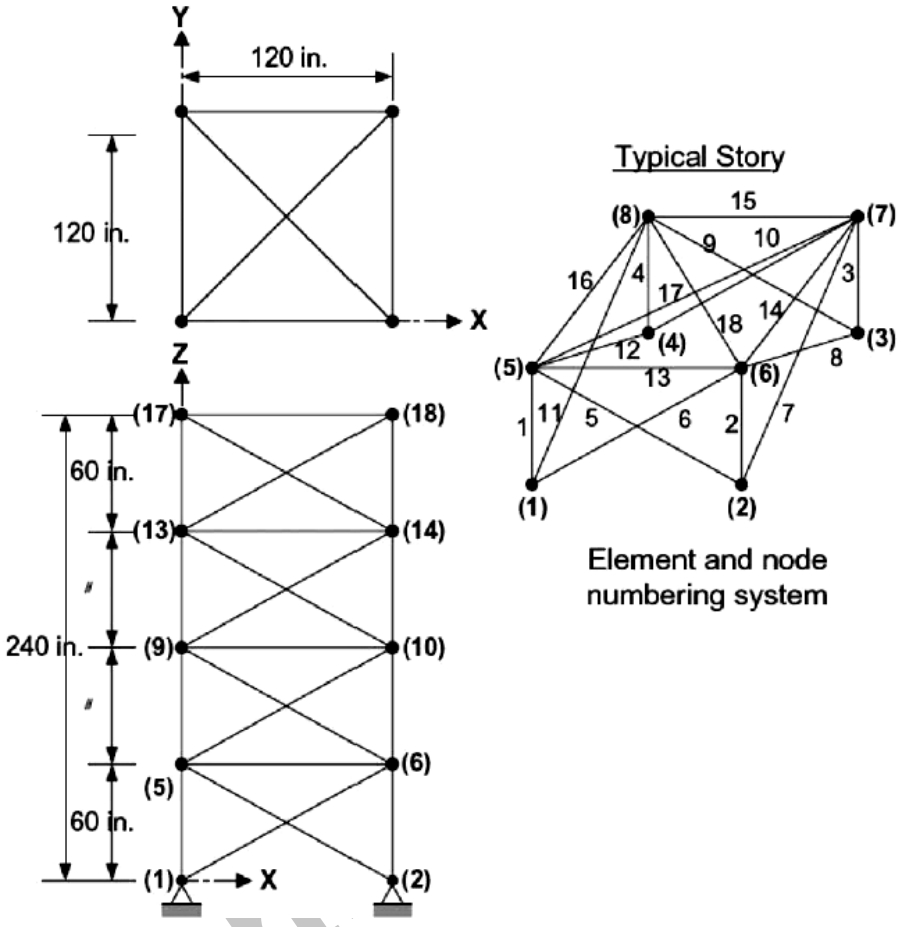

Figure 4. 72-bar space truss structure

Table 4: The load cases for the 72-bar space truss structure

Nodes	Load Case 1			Load case 2		
	P_x (kips)	P_y (kips)	P_z (kips)	P_x (kips)	P_y (kips)	P_z (kips)
17	5.0	5.0	–5.0	0.0	0.0	–5.0
18	0.0	0.0	0.0	0.0	0.0	–5.0
19	0.0	0.0	0.0	0.0	0.0	–5.0
20	0.0	0.0	0.0	0.0	0.0	–5.0

Table 5: The available cross-section areas of the ASIC code for example 2

No.	Cross-sectional area (in^2)	No.	Cross-sectional area (in^2)
1	0.111	33	3.840
2	0.141	34	3.870
3	0.196	35	3.880
4	0.250	36	4.180
5	0.307	37	4.220
6	0.391	38	4.490
7	0.442	39	4.590
8	0.563	40	4.800
9	0.602	41	4.970
10	0.766	42	5.120
11	0.785	43	5.740
12	0.994	44	7.220
13	1.000	45	7.970
14	1.228	46	8.530
15	1.266	47	9.300
16	1.457	48	10.850
17	1.563	49	11.500
18	1.620	50	13.500
19	1.800	51	13.900
20	1.990	52	14.200
21	2.130	53	15.500
22	2.380	54	16.000
23	2.620	55	16.900
24	2.630	56	18.800
25	2.880	57	19.900
26	2.930	58	22.000
27	3.090	59	22.900
28	1.130	60	24.500
29	3.380	61	26.500
30	3.470	62	28.000
31	3.550	63	30.000
32	3.630	64	33.500

Table 6: Comparison of optimal designs for the 72-bar truss structure (case 1)

Design variables No.	Li et al. [7]			Present study	
	PSO	PSOPC	HPSO	PSO	IPSO
1	2.6	3.0	2.1	3.2	2.0
2	1.5	1.4	0.6	0.5	0.5
3	0.3	0.2	0.1	0.1	0.1
4	0.1	0.1	0.1	0.1	0.1
5	2.1	2.7	1.4	1.4	1.2
6	1.5	1.9	0.5	0.4	0.5
7	0.6	0.7	0.1	0.1	0.1
8	0.3	0.8	0.1	0.1	0.1
9	2.2	1.4	0.5	0.7	0.6
10	1.9	1.2	0.5	0.5	0.5
11	0.2	0.8	0.1	0.1	0.1
12	0.9	0.1	0.1	0.1	0.1
13	0.4	0.4	0.2	0.2	0.2
14	1.9	1.9	0.5	0.5	0.6
15	0.7	0.9	0.3	0.4	0.4
16	1.6	1.3	0.7	0.7	0.6
Weight	1089.88	1069.79	388.94	403.47	385.54
Number of analyses	50000	50000	50000	8000	4176
$\lvert d_{\max}\rvert$	0.1	0.1	0.25	0.25	0.25
$\lvert \sigma_{\max}\rvert$	5463.48	5726.65	3293.67	20595.03	20368.78

Table 7: Comparison of optimal designs for the 72-bar truss structure (case 2)

Design variables No.	Li et al. [7]			Present study	
	PSO	PSOPC	HPSO	PSO	IPSO
1	7.220	4.490	4.970	2.130	1.800
2	1.800	1.457	1.228	0.391	0.563
3	1.130	0.111	0.111	0.111	0.111
4	0.196	0.111	0.111	0.111	0.111
5	3.090	2.620	2.880	1.266	1.228
6	0.785	1.130	1.457	0.442	0.442
7	0.563	0.196	0.141	0.250	0.111
8	0.785	0.111	0.111	0.111	0.111
9	3.090	1.266	1.563	0.785	0.563
10	1.228	1.457	1.228	0.602	0.563
11	0.111	0.111	0.111	0.250	0.111
12	0.563	0.111	0.196	0.141	0.111
13	1.990	0.442	0.391	0.141	0.196
14	1.620	1.457	1.457	0.602	0.563
15	1.563	1.228	0.766	0.391	0.442
16	1.266	1.457	1.563	0.602	0.602
Weight	1209.48	941.82	933.09	403.21	388.56
Number of analyses	50000	50000	50000	12500	5968
$\lvert d_{\max}\rvert$	0.1	0.1	0.1	0.25	0.25
$\lvert \sigma_{\max}\rvert$	2373.795	9491.242	10272.405	24996.14	20709.51

The computational performance of the proposed IPSO in this example is investigated through the 50 independent runs and the results are given in Table 8.

Table 8: Investigation on the performance of IPSO for the 72-bar truss in 50 runs

	Case 1	**Case 2**
Best weight	385.54	388.56
Worst weight	398.81	398.75
Average weight	387.23	391.51
Standard deviation	2.79	1.95
Minimum number of analyses	4000	4832
Maximum number of analyses	9800	12500
Average number of analyses	5468	7002

The results given in Tables 6 to 8 indicate that the best weight of the IPSO and HPSO [7] algorithms in Case 1 are 385.56 lb and 388.94 lb, respectively and in Case 2 are 388.56 lb and 933.09 lb, respectively. Also the best number of required structural analyses during the optimization processes by IPSO in Cases 1 and 2 are 4176 and 5968, respectively while in the case of HPSO [7] this number is equal to 50000. Therefore, IPSO algorithm considerably reduces the computational burden of the optimization process in Case 1 ($4176/50000 \approx 0.08$) and Case 2 ($5968/50000 \approx 0.12$) compared with the other variants of PSO.

The given results in both the numerical examples indicate that employing the IPSO results in better solution by spending lower computational costs.

7. CONCLUSIONS

An efficient optimization algorithm is proposed for optimal design of structures. The new developed optimization algorithm, called IPSO, has capability of finding global optima using few structural analyses. In the IPSO algorithm the advantages of the CA and PSO are combined. In this case a cellular automata-based new term is added to the conventional velocity updating equation to employ more information in the evolution process. In the IPSO algorithm, particles are distributed on a 2D grid. The state variables associated with each cell site are simply the design variables of the optimization problem. The evolution process is accomplished locally, using more information from members of Moore neighborhood of each site. The process is continued until the algorithm converges. As the

IPSO employs more information than the standard PSO, it increases the probability of finding the global optimum while requiring lower structural analyses. In order to assess the effectiveness of the developed IPSO algorithm two benchmark structural optimization examples are presented. The numerical results demonstrate the efficiency and computational advantages of the IPSO.

REFERENCES

1. Vanderplaats GN. Numerical optimization techniques for engineering design: with application. 2nd ed. New York, McGraw-Hill, 1984.
2. Haftka R and Grdal Z. *Elements of Structural Optimization*. 3rd ed. Dordrecht, The Netherlands, KluwerAcademic, 1992.
3. Rajeev S, Krishnamoorthy CS. Discrete optimization of structures using genetic algorithms, *Journal of Structural Engineering*, **118**(1992) 1233–50.
4. Erbatur F, Hasancebi O, Tutuncu I, Kitic H. Optimal design of planar and space structures with genetic algorithms, *Computers and Structures*, **75**(2000) 209–24.
5. Kaveh A, Kalatjari V. Genetic algorithm for discrete sizing optimal design of trusses using the force method, *International Journal for Numerical Methods in Engineering*, **55**(2002) 55–72.
6. Salajegheh E, Gholizadeh S. Optimum design of structures by an improved genetic algorithm using neural networks, *Advances in Engineering Software*, **36**(2005) 757–67.
7. Li LJ, Huang ZB, Liu F, Wu QH. A heuristic particle swarm optimizer for optimization of pin connected structures, *Computers and Structures*, **85**(2007) 340–9.
8. Kaveh A, Talatahari S. Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures, *Computers and Structures*, **87**(2009) 267–83.
9. Gholizadeh S, Salajegheh E. Optimal design of structures for time history loading by swarm intelligence and an advanced metamodel, *Computer Methods in Applied Mechanics and Engineering*, **198**(2009) 2936–49.
10. Camp CV, Bichon BJ. Design of space trusses using ant colony optimization, *Journal of Structural Engineering*, **130** (2004) 741–51.
11. Camp CV, Bichon BJ, Stovall SP. Design of steel frames using ant colony optimization, *Journal of Structural Engineering*, **131**(2005) 369–79.
12. Lee KS and Geem ZW. A new structural optimization method based on the harmony search algorithm. *Computers and Structures*, **82**(2004) 781–98.
13. Perez RE, Behdinan K. Particle swarm approach for structural design optimization, *Computers and Structures*, **85**(2007) 1579–88.
14. Von Neumann J. *Theory of Self-reproducing Automata*, A. W. Burks, (Eds.), University of Illinois Press, Champaign, Ill, 1966.
15. Angeline P. Evolutionary optimization versus particle swarm optimization: philosophy and performance difference. *Proceeding of the Evolutionary Programming Conference*, San Diego, USA; 1998.
16. Kennedy J. The particle swarm: social adaptation of knowledge, *Proceedings of the*

International Conference on Evolutionary Computation, Piscataway, NJ: IEEE, 1977, pp. 303–308.
17. Kennedy J, Eberhart RC. Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942–1945.
18. Biondini F, Bontempi F, Frangopol DM, Malerba PG. Cellular automata approach to durability analysis of concrete structures in aggressive environments, *Journal of Structural Engineering*, **130**(2004) 1724–37.
19. Gholizadeh S, Salajegheh E. Optimal seismic design of steel structures by an efficient soft computing based algorithm, *Journal of Constructional Steel Research*, **66**(2010) 85-95.
20. Seyedpoor SM, Gholizadeh S, Talebian SR. An efficient structural optimization algorithm using a hybrid version of particle swarm optimization with simultaneous perturbation stochastic approximation, *Civil Engineering and Environmental Systems*, 2010. (in press)
21. Canyurt OE, Hajela P. A cellular framework for structural analysis and optimization, *Computer Methods in Applied Mechanics and Engineering*, **194** (2005) 3516-34.
22. Rajasekaran S. Optimization of large scale three dimensional reticulated structures using cellular genetics and neural networks, *International Journal of Space Structures*, **16**(2001) 315-24.