

PARTICLE SWARM OPTIMIZATION FOR GLOBAL OPTIMIZATION PROBLEMS

Hsin-Chuan Kuo*, Jiang-Ren Chang**, and Ching-Hsiang Liu***

Key words: optimization problems, grouping wisdom, particle swarm optimization.

ABSTRACT

In this paper, the particle swarm optimization (PSO), originated as a simulation of a simplified social system with swarm intelligence and having exploring and exploiting characteristics of the particle swarm, is adopted to deal with the global optimization problems. To begin with, four non-linear two-dimensional functions are adopted as benchmark examples for determining the suitable ranges of parameters in the PSO. It is found that uses of the PSO can make the search quickly converge to the global optimum. Then, several benchmark functions below ten-dimensions are adopted especially for comparisons of search efficiency on the PSO and other algorithms. Results show that the proposed approach is not only superior to other algorithms but also has the higher success rate. Finally, the PSO is also applied to deal with the optimization problem of a grillage structure. Optimization solutions of the success rate and the used average generation show that the PSO also has the better performance for the optimal design of structures with constraints.

INTRODUCTION

In natural world, many kinds of species exhibit unbelievable wisdoms of social behaviors and self-organization, such as the ant swarm, bird flocking or hunting for food, and fish schooling, etc. In these different animal groups composed of simple individuals, neither leaders nor managements of a central control mechanism exist and they only "run" their lives depending on some self-specific simple rules which govern those interactive behaviors of individuals. Nevertheless,

it appears some unpredictable swarm intelligence from these individuals. In the past, most general-purpose optimization software used in industrial applications made use of gradient-based algorithms mainly due to their computation efficiency; however, in recent years, non-gradient-based, probabilistic search algorithms have attracted much attention in the research community and those algorithms generally mimic some natural phenomena [28].

Particle swarm optimization (PSO), originated by Kennedy and Eberhart [9, 16] in 1995 and known as an optimizer, is a population-based, self-adaptive search optimization technique. Through a process of trial and error, a number of parameters in the algorithms extraneous to optimization are eliminated from its algorithms and thus, result in the very simple original implementation [10]. The PSO is, basically, quite similar to a genetic algorithm (GA), in which the system is initialized with a population of random solutions; however, unlike the GA, the PSO assigns a randomized velocity to each potential solution, called the particle, and then fly through the problem space. Moreover, the PSO belongs to a single way of information flow without complex genetic operators of the GA, such as the crossover and mutation, and more differently, the PSO adopts the current optimal solution as the mechanism for renewing the whole search process in contrast to the GA such that the PSO has the ability to quickly converge to a reasonably good solution [14].

Simulating simplified animal social behaviors leads to the motivation for developing the PSO algorithm, which works on the social behavior of particles in the swarm. In fact, the original concept was to graphically simulate the graceful but unpredictable choreography of birds flocking and thereafter, such initial simulations were modified to incorporate nearest-neighbor velocity matching, climate ancillary variables, and incorporate multidimensional search and acceleration by distance [10]. In this regard, the PSO finds the global optimum by simply adjusting the trajectory of each particle toward its own best solution and toward the best particle of the entire swarm at each time step as the basic search

Paper Submitted 11/15/05, Accepted 03/13/06. Author for Correspondence: Hsin-Chuan Kuo. E-mail: khc@mail.ntou.edu.tw

**Associate Professor, Institute of Systems Engineering & Naval Architecture, National Taiwan Ocean University, 2, Pei-Ning Road, Keelung, Taiwan 202, R.O.C.*

***Professor, Institute of Systems Engineering & Naval Architecture, National Taiwan Ocean University, 2, Pei-Ning Road, Keelung, Taiwan 202, R.O.C.*

****Graduate Student, Institute of Systems Engineering & Naval Architecture, National Taiwan Ocean University, 2, Pei-Ning Road, Keelung, Taiwan 202, R.O.C.*

concept [7, 8, 16]. During the search process, each particle moving in the problem space has its own fitness value with respect to its mapping from the objective function with a velocity determining its search direction and position. Particles swarm fly through the problem space with the best experience of every particle and that of the best particle in the swarm [17]. During the particle's search process, the individual's previous best solution will be kept and recorded in its memory. That is, each particle memorizes its individual's previous best solution, in which the particle can further adopt to change the velocity of next generation and thus, it is called the cognition-only model. Comparisons of the fitness value of the previous best solution between the individual and the swarm can modify the previous best fitness value for the swarm and each particle also takes the updated memory to change its velocity of the next generation. Thus, it is called the society-only model. By using such a methodology in the change of the velocity of each particle for generations, the PSO can find the optimal solution of the optimization problem [11, 27].

The PSO has been proposed for more than ten years and in contrast to some available tools, such as genetic algorithms or evolutionary algorithms, it apparently shows its deficiency in the mathematical formulations of the evolution algorithms and its stability of the search effectiveness. Although several applications with the PSO on magnetic system design [2], electrical power system [13], traffic and transportation [26], and data analysis [28] can be found, how to develop a more effective search mechanism for the space problem is unavailable. In addition, notwithstanding its recent popularity, one drawback for the PSO is the presence of problem dependent parameters and some workers began to find "universal" values for the PSO parameters [22]. Thus, it loses its robustness. A further drawback of the original algorithm proposed by Kennedy and Eberhart [16] lies therein that the algorithm is known to quickly converge to the approximate region of the global minimum. However, the algorithm does not maintain this efficiency when entering the stage where a refined local search is required to pinpoint the global minimum exactly. This has led to a number of variations on the original PSO being proposed to overcome this shortcoming [22].

For improving the robustness of the PSO, Shi and Eberhart [23] proposed the concept of the inertia weight, which can be used to change the velocity of the particle in order to attain the balancing point between the global and local search. By using this scheme, the PSO can quickly find the global optimum. It is found that in the inertia weight ranges of 0.9 to 1.2, the search performance is better while the computational time does need much more. Shi and Eberhart [24] further pointed out

that introducing the linearly decreasing concept into the inertia weight can reduce the initial inertia weight value of 0.9 down to 0.4 as the generation increases. Moreover, an inclusion of dynamic inertia weight concept was also proposed to render the PSO relatively insensitive to the values of the cognitive and social scaling factors for global optimization [21, 22]. Clerc [7] proposed the constriction factor to adjust the velocity of the particle for obtaining the better convergence. In references of [15, 25], different frameworks were also proposed to analyze the characteristics of the PSO, such as an analysis for particle swarm effect [15] and the study on parameter selection in PSO [25].

Aimed at the developments and applications of the PSO, an article [12] for reviewing the improvements of the PSO and its resources was proposed and focused the discussions on the inertia weight, constriction factors and tracking dynamic system. On the other hand, Ratnaweera [20] and Suganthan [27] applied the concept of the time-varying inertia weight. It was shown that different ranges of the inertia weight considerably influence the performance of the particle itself and the swarm effect. In the above-mentioned papers, influences on the search for parameters in the PSO have been carefully discussed; however, most of the testing problems center on the moderate-scaled design problems (20 to 50 dimensions) and even on the large-scaled problems. Little attention has been given to the study and discuss for the low dimensional problems. In this regard, the purpose of this paper is to study the search performance of the PSO in the low dimensional problems. First of all, we will begin by adopting four benchmark functions under ten dimensions for finding the optimal ranges of parameters needed in the PSO and propose the more suitable parameter setting for low dimensional problems. Then, aimed at the computation efficiency, some other algorithms will be selected to make a comparison with the PSO. Finally, a practical engineering optimization of the grillage design problem will be examined. This paper, apart from the Introduction, is organized as follows. Section 2 details the PSO algorithm and its core elements of the operation model. The strategies for the testing cases are presented in Section 3 while several benchmark functions are listed in appendix for the reader's convenience. Besides, the practical engineering example is also included in this section for completeness. Section 4 presents the testing parameters setting and discussions on results. Finally, some concrete conclusions are drawn in Section 5.

THE PSO MODELLING

The initial particles of the swarm for the PSO are randomly generated and one can find the optimal solu-

tion of the problem through the generation. In each generation, the particle changes its velocity by way of two kinds of search memories. One is the particle's best memory, called *pbest*, and the other is the group's best memory, called *gbest*. After generations, the PSO can find the best solution according to the best solution memories. The changing scheme for the velocity of the particle is graphically illustrated in Figure 1 and its algorithm is presented in the following.

- Step 1:** The position and velocity of the particle in the initial swarm for n -dimensional space are both randomly generated.
- Step 2:** Aimed at the fitness function defined, the fitness value of the particle is well evaluated.
- Step 3:** The fitness value of the particle is compared with that of the previous best one and then, the new velocity of the particle is modified according to the best positions of the particle and swarm.
- Step 4:** After comparing the fitness of the particle and swarm, if the best fitness of the particle is superior to that of the swarm, then it modifies the memory of the swarm's best fitness and at the same time, every particle modifies the particle's velocity of the next generation.
- Step 5:** The new velocities and positions of the particles for the next generation are shown in Figure 1 and determined according to the following equations.

$$v_i^{k+1} = wv_i^k + c_1 \times \text{rand}(\cdot) \times (s_i^{pbest} - s_i^k) + c_2 \times \text{rand}(\cdot) \times (s^{gbest} - s_i^k), \quad (1)$$

$$s_i^{k+1} = s_i^k + v_i^{k+1}, \quad (2)$$

in which v_i^k and v_i^{k+1} represent the velocities of the particle i at the k^{th} and $k+1^{th}$ generation respectively, w is the inertia weight, c_1 and c_2 are constants known as learning coefficients, and $\text{rand}(\cdot)$ denotes a generated uniformly distributed random number in the range of 0 to 1. Besides, s_i^{pbest} represents the best position of the particle i , s^{gbest} represents the best position of the swarm and s_i^{k+1} represents the new position of the particle i .

- Step 6:** If the search satisfies the termination condition then it stops; otherwise, it returns to step 2.

In Eq. (1), the second part is the cognition-only model and the third part is the so-called social-only model. It should be noted that the velocity, or called the step size per generation, and position of the particle are

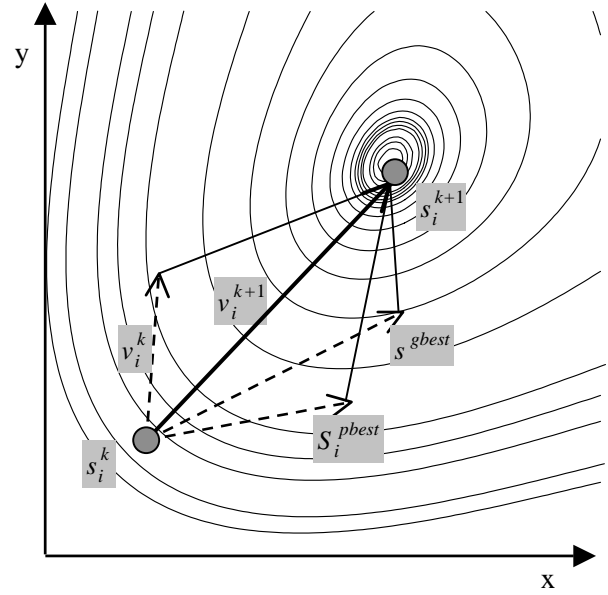


Fig. 1. Velocity changing scheme of a particle.

limited to the maximum values of v_{\max} and s_{\max} , respectively. If the new velocity and the position of the particle in the problem space are over the limitations, then the velocity and the position should be designated as v_{\max} and s_{\max} , respectively [23].

TESTING EXPERIMENTS

1. Strategy description for setup of testing parameters

Four two-dimensional non-linear benchmark functions of Rosenbrock ($R_2 - f_1$), Zakharov ($Z_2 - f_5$), MH - f_2 and Shaffer - f_{11} are respectively adopted to examine three parameters in the PSO and to find more suitable ranges of parameters. Aimed at the inertia weight, two types of the fixed-value and linearly decreasing are given to investigate the influence on the search of the swarm. As for the study of learning factors, c_1 and c_2 , their impacts on the search are also conducted in detail and once the optimal ranges of c_1 and c_2 are suggested and then, they will further be applied to low dimensional problems with several benchmark functions listed in appendix for validation. On the other hand, other six evolutionary algorithms proposed in references of [1, 3-6, 19] are adopted for comparing the search efficiency with the PSO.

2. Optimization problem of the grillage structure

A typical grillage structure shown in Figure 2 and usually appeared in naval architecture is adopted as a target problem and the minimum uses of the material is

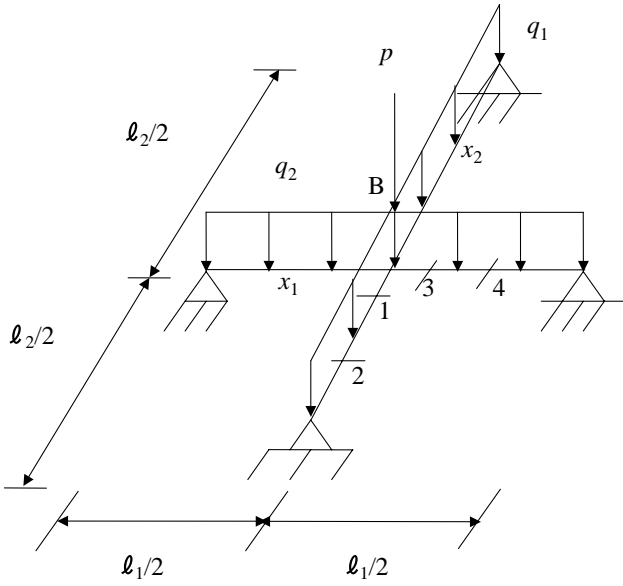


Fig. 2. Optimization problem of the grillage structure.

usually considered for naval architects due to the cost; therefore, the minimum volume of the grillage structure is taken as the objective function. With the stress constraint conditions, this optimization problem can be described as [18]:

$$\text{Minimize } f(x) = \ell_1 x_1 + \ell_2 x_2 \quad (3)$$

Subject to:

$$-\begin{Bmatrix} 20.0 \\ 20.0 \end{Bmatrix} \leq \frac{1}{S_1} \begin{Bmatrix} M_1 \\ M_2 \end{Bmatrix} \leq \begin{Bmatrix} 20.0 \\ 20.0 \end{Bmatrix}, \quad (4)$$

$$-\begin{Bmatrix} 20.0 \\ 20.0 \end{Bmatrix} \leq \frac{1}{S_2} \begin{Bmatrix} M_3 \\ M_4 \end{Bmatrix} \leq \begin{Bmatrix} 20.0 \\ 20.0 \end{Bmatrix}, \quad (5)$$

where the design variables x_1 and x_2 represent the cross-sectional areas of the first stiffener and the second stiffener, respectively, with both ranges of $1 \leq x_1, x_2 \leq 30$. The symbols, ℓ_1 and ℓ_2 , denote lengths of the first and second stiffeners, respectively. M_1, M_2, M_3 and M_4 represent the bending moments of locations, respectively, with their section modulus of $S_i = (x_i/1.48)^{1.82}$, $i = 1, 2$. For the reader's convenience, symbols appeared in Figure 2 also illustrate here. Symbols, q_1 and q_2 , represent the uniform distributed forces of the first and second stiffeners, respectively and p is the concentrated load acting on the joint B.

For investigating the PSO for the optimization problem with constraints, this testing problem is sub-

jected to two different loading cases.

Case 1: $q_1 = q_2 = 0.1, p = 1.0, \ell_1 = 100, \ell_2 = 300$

This case, basically, has two local minima and their values are $x_1^* = [2.2, 18.8]^T, f(x_1^*) = 5860$ and $x_2^{opt} = [9.70, 6.06]^T, f(x_2^{opt}) = 2790$, respectively. The global optimum is denoted as x_2^{opt} .

Case 2: $q_1 = q_2 = 0.1, p = 1.0, \ell_1 = 100, \ell_2 = 120$

In case 2, there are three local minima and their values are $x_1^* = [5.5, 2.56]^T, f(x_1^*) = 3620$; $x_2^* = [13.3, 18.8]^T, f(x_2^*) = 3580$; $x_3^* = [23.44, 7.19]^T, f(x_3^{opt}) = 3200$, respectively. The global optimum is denoted as x_3^{opt} .

RESULTS AND DISCUSSIONS OF TESTING EXPERIMENTS

First of all, the PSO model shown in Eq. (1) is conducted with different search strategies for a systematic discussion. After the adequate ranges of parameter setting for the PSO have been determined, those benchmark functions listed in the appendix are adopted for testing and validation. Besides, comparisons of the performance of the PSO and other algorithms are made. Finally, the experimental results of the grillage design using the PSO are also discussed.

During the search, when the optimal solution, OBJ_{PSO} , meets the termination criterion

$$|OBJ_{PSO} - OBJ^*| \leq 10^{-4}, \quad (6)$$

where OBJ^* represents the known best solution, and the search is taken as a success one. For study of the reliability of the PSO, the success rate, SR, measured by counting the number of successful searches from all 100 runs, is recorded. In addition, the computation efficiency of the PSO, AG, defined as the used average number of generations, i.e., the summation of the used generations among all the successful searches divided by the total number of successful searches, is also conducted.

1. Search efficiency of the PSO under different inertia weights

For evaluating the inertia weight, two 2-dimensional single-minimum problems, $R_2 - f_1$ and $Z_2 - f_5$, and two 2-dimensional multi-minimum problems, HM - f_2 and Shaffer - f_{11} , are all selected with parameters settings of swarm size = 20, $v_{\max} = 20$, $S_{\max} = 20$, and the learning factors c_1 and c_2 both equal to 2. Table 1 is the optimization results at several fixed values of the inertial weight and linearly decreasing expressions. Two concrete conclusions can be drawn as follows.

(1) *To deal with single-minimum problems, the search efficiency of the PSO is better as the inertia weight is given a fixed value*

As shown in Table 1, for single-minimum problems, the inertia weight does not influence the search and further, when the inertia weight is equal to 0, the success rate can nearly approach to 100%. Besides, it is also shown that when the inertia weight is given a smaller value, the success rate and the search efficiency are both well satisfied. However, when the fixed value of the inertia weight increases up to 0.8, the success rate dominantly reduces and the search efficiency also shows the decrease tendency. It should be noted that when the inertia weight linearly decreases from 0.9 to 0.4, although the success rate can approach up to over 90%, the calculation effort is at last over five times of the fixed-value type.

For the mechanism of preserving the previous velocity, it has little influence on the search of the swarm when dealing with single-minimum problems. The possible reason may come from the only one minimum solution of these testing functions and that with the particle activeness and the interactive guidance existing in particles of the swarm the PSO can have the higher performance by itself. In general, when considering the indexes of success rate and computation efficiency, one can conclude that for dealing with a single-minimum problem, the suitable inertia weight for the PSO are suggested in the ranges of 0.2 to 0.6.

(2) *To deal with multi-minimum problems, the search efficiency of the PSO is better when the inertia weight decreases from 0.9 down to 0.4*

As shown in Table 1, when the inertia weight is given a fixed value, the PSO although can find the

global optimum, yet the success rate is less satisfied. It is apparently meant that when the PSO with the fixed-value inertia weight deals with the multi-minimum problems, once the distribution of the initial swarm approaches near the global optimum, it can quickly meet the termination criterion. However, when the particle is far away the global optimum, it might lead to the earlier convergence or be trapped into the local optimum owing to the insufficient activeness of the particle and the lack of swarm's diversity.

In contrast to the fixed-value inertia weight, if one adopts the linearly decreasing inertia weight of 0.9 down to 0.4 and asks the initial particle to reserve 90% of the previous velocity, thus, the diversity of swarm becomes wider. Besides, as the generation increases, it will gradually decrease the influence of the previous velocity. Such a mechanism makes the PSO quickly converge to the global optimum because the particle can hold the best solutions of the individual and the population simultaneously. For example, as shown in Table 1, for the MH – f_2 function, the success rate (SR) of the PSO can approach to 64% and the used average number of generations (AG) is only 339. In contrast to the fixed-value inertia weight at 0.85, both of them have nearly the same success rate; however, the former is one-third of the latter in the used average generations. While for the Shaffer – f_{11} function, if the PSO adopts the linearly decreasing inertia, weight the success rate can approach to 96% with generations only 512.

2. Influence analysis of the learning factors, c_1 and c_2 , on search

Aimed at evaluating the learning factors of c_1 and c_2 , the swarm size is set as 20, v_{\max} and S_{\max} are both set as 10. From results of the previous subsection, the inertia weight value is reset as 0.4 for the single-minimum problems and the linearly decreasing inertia weight

Table 1. Performance of different inertia weights in PSO

Function		w									
		0.0	0.2	0.4	0.6	0.8	0.85	0.9	0.95	1	0.9~0.4
Rosenbrock	SR	100	100	100	100	93	37	0	0	–	90
	AG	301	169	153	215	813	2674	–	–	–	867
Zakharov	SR	100	100	100	100	100	70	10	7	0	91
	AG	29	28	42	76	382	1893	2356	3656	–	906
MH – f_2	SR	43	47	43	53	53	70	13	0	0	64
	AG	274	228	147	143	303	1016	2465	–	–	339
Shaffer – f_{11}	SR	40	47	37	43	50	67	83	47	40	96
	AG	212	266	285	305	526	680	1514	2229	2076	512

Note: SR: Success rate(%); AG: Average generation; w: Inertia weight.

of 0.9 down to 0.4 is adopted for the multi-minimum problems. Figure 3 illustrates the contours of the success rates for four benchmark functions when the learning factors, c_1 and c_2 , are both set from 1 to 3. Figure 4 shows the contours of the used average number of generations. Although the more suitable bandwidth regions, as shown in Figure 3, are different, their search efficiency are satisfied and especially in single-minimum functions like $R_2 - f_1$ and $Z_2 - f_5$, the success rates of most cases are up to 100% when the learning factors are set up in ranges of 1 to 3. As for multi-minimum functions such as $MH - f_2$ and $\text{Shaffer} - f_{11}$, each function has its own suitable range of learning factor

and as shown in Figure 3, their success rates can reach from 79% to 95%. However, when c_1 and c_2 are setup over 2.6 and 2.8, respectively, success rates of functions $R_2 - f_1$ and $MH - f_2$ both decrease. It reveals that the higher the learning factors are given, the more negative effects the PSO will be.

As can be seen in Figure 4, the used average generation numbers are much less for the four functions when the learning factors are smaller; while as the learning factors become larger, average number of generations used gradually increase. It means that activeness of the learning factors for the particle increases and this leads the particle to have more diversity and such a

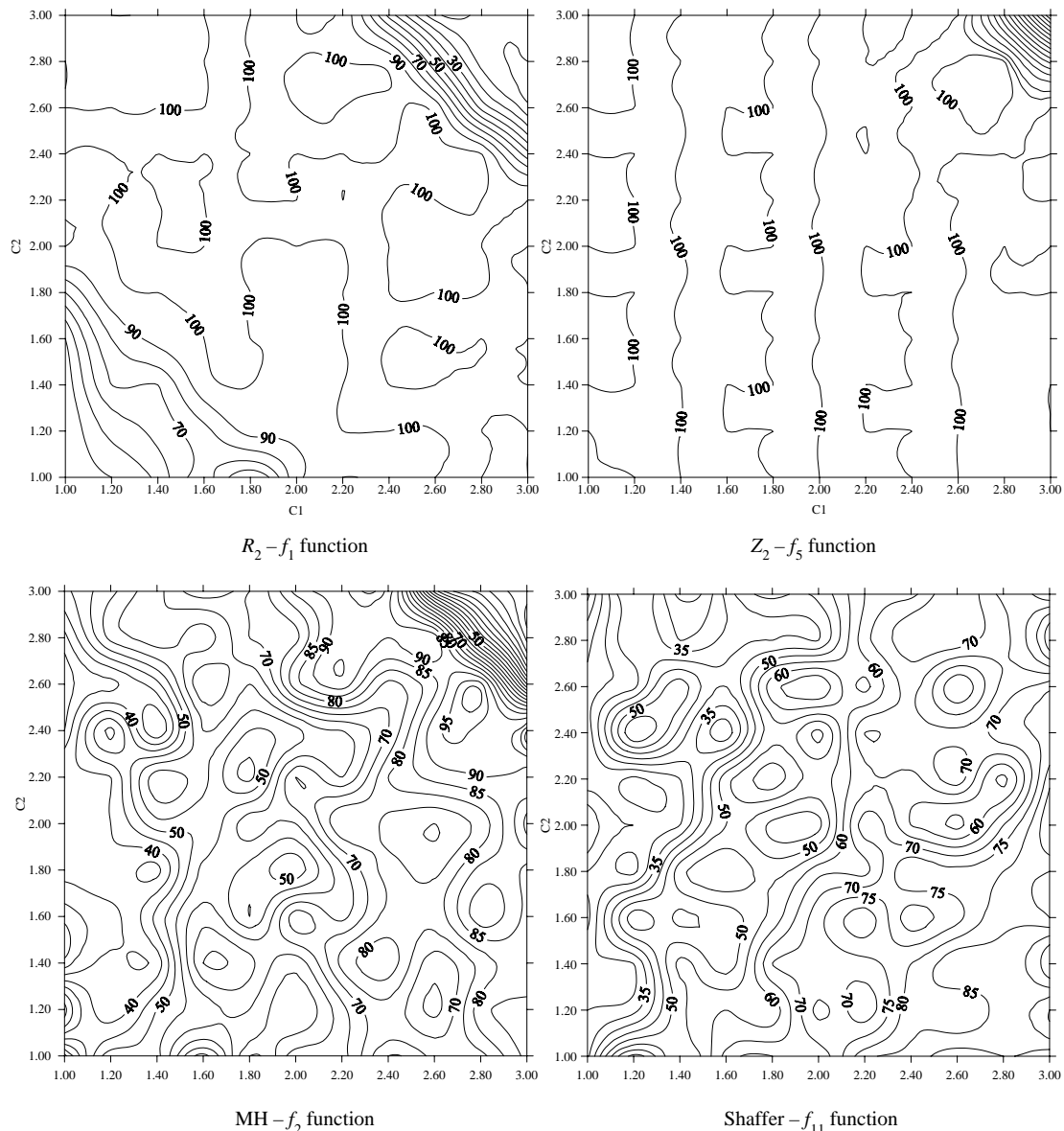


Fig. 3. The success rate, SR, of the PSO with different learning factors c_1 and c_2 for four benchmark functions.

rates can reach to 85% and it indicates that the PSO also has satisfied search ability for the multi-minimum problems. As for the 10-dimensional functions of $R_{10} - f_1$, and $Z_{10} - f_5$, as the used average number of generation increases, the success rates are still over 70%.

Tables 3 and 4 respectively show results of success rates and the number of function evaluations, N_{fe} , using the PSO and other algorithms, which include Real-Coding Genetic Algorithm, (RCGA) [1], Continuous

Genetic Algorithm (CGA) [3], Continuous Hybrid Algorithm (CHA) [5], Enhanced Continuous Tabu Search (ECTS) [4], Continuous Tabu Simplex Search (CTSS) [6] and Simplex and Evolution Algorithm (SEA) [19]. It should be noted that the RCGA, CGA and CHA are all categorized into the genetic algorithms with different improvement strategies for local search, while the ECTS and TCSS belong to the improving types of the Tabu search, and the SEA is basically a hybrid algorithm of the simplex method with evolutionary algorithm. As can be seen in Table 3, the PSO actually has a dominant performance in the search ability with robustness and completeness because other algorithms still use some improving strategies for a better local search. Table 4 shows a comparison on the number of function evaluations for the PSO and other algorithms. It is found that for functions of $R_n - f_1$ and $S_{4,n} - f_{12}$ types, their testing results in N_{fe} are less satisfied because the global optima of such types of functions are difficult to be found. If we further refer to Figure 4, it can be found that the reason may come from high sensitivities of these two learning factors, c_1 and c_2 . Moreover, it is also found that the PSO is not so better as compared with other algorithms in the computation efficiency and it indicates that the PSO still needs some effective strategies for improving its local search ability.

4. Optimization results of the grillage structure

In this subsection, the PSO with the following parameters setting is used to deal with the optimization problem of the grillage structure. The swarm size is

Table 2. Optimization results of the PSO for seventeen benchmark functions

Function	Dimensions	SR(%)	AG
ES - f_3	2	100	37
B2 - f_6	2	100	29
SH - f_8	2	100	40
GP - f_9	2	100	24
MZ - f_7	2	100	18
$Z_2 - f_5$	2	100	19
$R_2 - f_1$	2	100	83
RC - f_4	2	100	37
MH - f_2	2	85	133
Shaffer - f_{11}	2	100	251
DJ - f_{10}	3	100	24
$S_{4,7} - f_{12}$	4	84	1459
$S_{4,10} - f_{12}$	4	87	1508
$Z_5 - f_5$	5	100	51
$Z_{10} - f_5$	10	100	124
$R_5 - f_1$	5	84	1655
$R_{10} - f_1$	10	74	2674

Table 3. A comparison of the SR(%) for several different functions with PSO and other algorithms

Method /Function	PSO [present]	RCG A [1]	CGA [3]	ECTS [4]	CHA [5]	CTSS [6]	SEA [19]
ES - f_3	100	100	100	100	100	100	97
$B_2 - f_6$	100	—	100	—	100	100	100
SH - f_8	100	100	100	100	100	100	100
GP - f_9	100	100	100	100	100	100	—
$Z_2 - f_5$	100	100	100	100	100	100	97
$R_2 - f_1$	100	100	100	100	100	100	97
RC - f_4	100	100	100	100	100	100	100
DJ - f_{10}	100	100	100	100	100	100	100
$S_{4,7} - f_{12}$	84	70	83	80	85	77	—
$S_{4,10} - f_{12}$	87	58	81	75	85	74	—
$Z_5 - f_5$	100	100	100	100	100	—	—
$Z_{10} - f_5$	100	100	100	100	100	—	—
$R_5 - f_1$	84	60	100	100	100	—	—
$R_{10} - f_1$	71	70	77	75	83	—	—

Table 4. A comparison on for several different functions with PSO and other algorithms

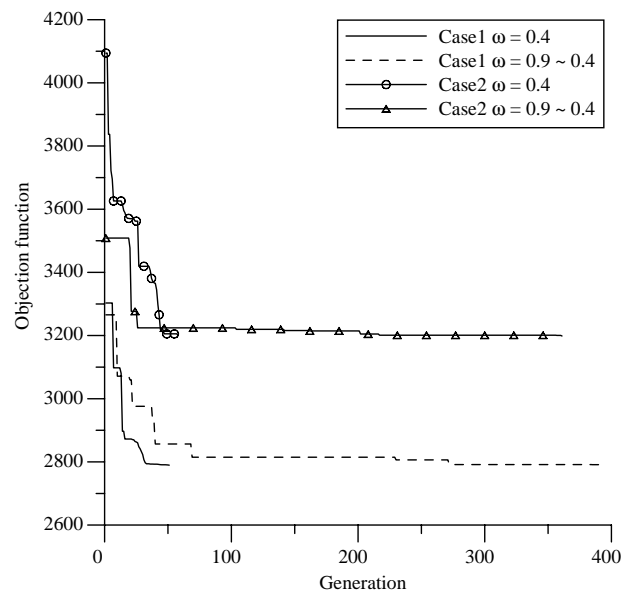
Method/ Function	PSO [present]	RCGA [1]	CGA [3]	ECTS [4]	CHA [5]	CTSS [6]	SEA [19]
ES - f_3	740	642	1504	1284	952	325	197
$B_2 - f_6$	580	—	430	—	132	98	258
SH - f_8	800	946	575	370	345	283	420
GP - f_9	480	270	410	231	259	119	—
$Z_2 - f_5$	380	437	620	195	215	78	90
$R_2 - f_1$	1660	596	960	480	459	369	266
RC - f_4	740	490	620	254	295	125	272
DJ - f_{10}	500	449	750	338	371	155	—
$S_{4,7} - f_{12}$	29180	1143	680	910	620	590	—
$S_{4,10} - f_{12}$	30160	1235	650	898	635	555	—
$Z_5 - f_5$	1530	1115	1350	2254	950	—	—
$Z_{10} - f_5$	7440	2190	6991	4630	100	—	—
$R_5 - f_1$	33100	4150	3990	2142	3290	—	—
$R_{10} - f_1$	106960	8100	21563	15720	14563	—	—

Table 5. Optimization results of the PSO for the typical grillage structure

	w	SR (%)	AG
Case 1	0.4	100	43
	0.9~0.4	100	416
Case 2	0.4	85	65
	0.9~0.4	100	269

given as 20, both v_{\max} and S_{\max} are 100, and the learning factors, c_1 and c_2 , are given as 2. As can be seen in Table 5, the PSO can find the global optimum satisfies all the constraints. In case 1, with two types of inertia weights, both their success rates can approach to 100%. In case 2, there are three local minima in the feasible domain. When the fixed-value of inertial weight is given as 0.4, the success rate can only approach to 85%; however, when the linearly decreasing inertia weight, which is better for dealing with this multi-minimum problem, is adopted with the PSO, the success rate can approach to 100%. As for the comparison of the computation efficiency, when the fixed value of inertia weight is 0.4, the PSO can quickly find the global optimum. Especially in the case 1, as shown in Table 5, the computation efficiency of the fixed-value type is superior to that of the linearly decreasing type.

As shown in Figure 5, which illustrates the search convergence of the generation, the PSO with given two types of inertia weights can have the robust search ability for the grillage problem especially in the initial stage. It indicates the PSO can find a better solution in

**Fig. 5. The search convergence of the generation for the grillage structure.**

a short time and more importantly, it has a good explorative ability with the exploitive search simultaneously to reach the global optimum. It should be noted that both cases 1 and 2 of using the fixed-value inertia weight, as shown in Figure 5, finish their searches at about the 50th generation. The same tendency also occurs in the linearly decreasing type of inertia weight and more specifically, as the particle searches in the final stage, its exploring ability has gradually reduced. It means its finding result is quite approaching to the global optimum in this stage.

CONCLUSIONS

In this paper, seventeen optimization problems of single-minimum and multi-minimum types have been adopted for evaluating the PSO and for determining the suitable ranges of parameter setting. For the settings of the inertia weight, the linearly decreasing inertia, which is commonly adopted in the literature, has advantages of increasing the particle's diversity in the early generations. Such a property makes the PSO not convergent too early and not easily trapped into the local optimum when dealing with the multi-minimum problem. In contrast, when treating the single-minimum problem, it will interfere with the velocity of the particle and thus influence the computation efficiency. Based on the testing results, for the single-minimum problem, we suggest a use of a fixed value in ranges of 0.2 to 0.6 for setting the inertia weight; while for the multi-minimum problem, the linearly decreasing inertia weight at ranges of 0.9 down to 0.4 is suggested. Both of them show satisfactory success rate and computation efficiency. As for setting the learning factors, they are usually dependent on the characteristics of testing problems; however, they are, in general, shown in a region of a specified bandwidth and a constant value of 2 for both c_1 and c_2 , as suggested in references, can be adopted.

Comparisons of the PSO and other algorithms show that the PSO inherent with the concept of the particle swarm intelligence can have satisfactory robustness and completeness. As shown in the seventeen testing cases, all their success rates can approach to 95%; however, for needed numbers of objective function evaluations, the PSO has shown its computation efficiency but not the best as compared with other algorithms. It means that the PSO still needs to be improved in its local search ability. Finally, due to its reliability and high success rate, the PSO can further be applied to deal with the optimization problem with constrained conditions, like the structural design of the grillage problem, which is usually encountered in naval architecture.

REFERENCES

1. Bessaou, M. and Siarry, P., "A Genetic Algorithm with Real-Value Coding to Optimize Multimodal Continuous Functions," *Structural and Multidiscipline Optimization*, Vol. 23, pp. 63-74 (2001).
2. Brandstatter, B. and Baumgartner, U., "Particle Swarm Optimization- Mass-Spring System Analogon," *IEEE Transactions on Magnetics*, Vol. 38, pp. 997-1000 (2002).
3. Chelouah, R. and Siarry, P., "A Continuous Genetic Algorithm Designed for the Global Optimization of Multimodal Functions," *Journal of Heuristics*, Vol. 6, pp. 191-213 (2000).
4. Chelouah, R. and Siarry, P., "Tabu Search Applied to Global Optimization," *European Journal of Operational Research*, Vol. 123, pp. 256-270 (2000).
5. Chelouah, R. and Siarry, P., "Genetic and Nelder-Mead Algorithms Hybridized for a More Accurate Global Optimization of Continuous Multimodal Functions," *European Journal of Operational Research*, Vol. 148, pp. 335-348 (2003).
6. Chelouah, R. and Siarry, P., "A Hybrid Method Combining Continuous Tabu Search and Nelder-Mead Simplex Algorithms for The Global Optimization of Multimodal Functions," *European Journal of Operational Research*, Vol. 161, pp. 636-654 (2005).
7. Clerc, M., "The Swarm and the Queen: Toward a Deterministic and Adaptive Particle Swarm Optimization," *Proceedings of IEEE International Congress of Evolutionary Computation*, Washington DC, Vol. 3, pp. 1951-1957 (1999).
8. Clerc, M. and Kennedy, J., "The Particle Swarm-Explosion, Stability, and Convergence in a Multi-Dimensional Complex Space," *IEEE Transactions of Evolutionary Computation*, Vol. 6, pp. 58-73 (2002).
9. Eberhart, R.C. and Kennedy, J., "A New Optimizer Using Particle Swarm Theory," *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Japan, pp. 39-43 (1995).
10. Eberhart, R.C., Simpson, P.K., and Dobbins, R.W., *Computational Intelligence PC Tools*, Academic Press Professional, Boston, MA (1996).
11. Eberhart, R.C. and Shi, Y., "Comparison between Genetic Algorithms and Particle Swarm Optimization," *Proceedings of Annual Conference on Evolutionary Programming*, San Diego, CA (1998).
12. Eberhart, R.C. and Shi, Y., "Particle Swarm Optimization: Developments, Applications and Resources," *Proceedings of IEEE of International Conference on Evolutionary Computation*, FL, Vol. 1, pp. 81-86 (2001).
13. Fukuyama, Y. and Yoshida, H., "A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power System," *Proceedings of IEEE Congress on Evolutionary Computation*, Korea, Vol. 1, pp. 93-97 (2001).
14. Hu, X., "Introduction of Particle Swarm Optimization," <http://web.ics.purdue.edu/~hux/tutorials.shtml> (2001).
15. Kennedy, J., "Stereotyping: Improving Particle Swarm Performance with Cluster Analysis," *Proceedings of IEEE of International Conference on Evolutionary Computation*, CA, Vol. 2, pp. 1507-1512 (2000).
16. Kennedy, J. and Eberhart, R.C., "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, Vol. 4, pp. 1942-1948 (1995).
17. Kennedy, J. and Eberhart, R.C., *Swarm Intelligence*,

Morgan Kaufmann Press, CA (2001).

18. Kirsh, U., *Optimum Structural Design*, McGraw-Hill, New York (1981).
19. Kuo, H.C., Chang, J.R., and Shyu, K.S., "A Hybrid Algorithm of Evolution and Simplex Methods Applied to Global Optimization," *Journal of Marine Science and Technology*, Vol. 12, pp. 280-289 (2004).
20. Ratnaweera, A., Halgamuge, S.K., and Watson, H.C., "Self-Organizing Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients," *Evolutionary Computation, IEEE Transactions*, Vol. 8, pp. 240-255 (2004).
21. Schutte, J.F. and Groenwold, A.A., "Sizing Design of Truss Structures Using Particle Swarms," *Structural and Multidisciplinary Optimization*, Vol. 25, pp. 261-269 (2003).
22. Schutte, J.F. and Groenwold, A.A., "A Study of Global Optimization Using Particle Swarm," *Journal of Global Optimization*, Vol. 31, pp. 93-108 (2005).
23. Shi, Y. and Eberhart, R.C., "A Modified Particle Swarm Optimizer," *Proceedings of the 7th International Conference on Evolutionary Programming*, CA, pp. 69-73 (1998).
24. Shi, Y. and Eberhart, R.C., "Parameter Selection in Particle Swarm Optimization," *Proceedings of the 7th International Conference on Evolutionary Programming*, CA, Vol. 4, pp. 591-600 (1998).
25. Shi, Y. and Eberhart, R.C., "Empirical Study of Particle Swarm Optimization," *Proceedings of the Evolutionary Computation 1999 Congress*, Vol. 3, pp. 1945-1950 (1999).
26. Srinivasan, D., Loo, W.H., and Cheu, R.L., "Traffic Incident Detection Using Particle Swarm Optimization," *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, IN, pp. 144-151 (2003).
27. Suganthan, P.N., "Particle Swarm Optimizer with Neighborhood Operator," *Proceedings of IEEE of Congress on Evolutionary Computation*, Washington DC, Vol. 3, pp. 1958-1962 (1999).
28. Venter, G. and Sobieszcanski-Sobieski, J., "Particle Swarm Optimization," *AIAA Journal*, Vol. 41, pp. 1583-1589 (2003).

APPENDIX: LIST OF BENCHMARK FUNCTIONS

1. f_1 : Rosenbrock, (R_n), (n variables)

$$f_1(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$$

search domain: $-5 < x_i < 5$, $i = 1, 2, 3, \dots, n$;
one global minimum: $x^* = (1, \dots, 1)$, $f_1(x^*) = 0$.

2. f_2 : Modified Himmelblau Function, (MH), (2 variables)

$$f_2(x) = (x_2 + x_1^2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + x_1$$

search domain: $-5 < x_i < 5$, $i = 1, 2$;
number of local minima = 3; one global minimum: $x^* = (-3.788, -3.246)$, $f_2(x^*) = -3.78$.

3. f_3 : Easom, (ES), (2 variables)

$$f_3(x) = -\cos(x_1) \cos(x_2) \exp(-((x_1 - \pi)^2 + (x_2 - \pi)^2))$$

search domain: $-100 < x_i < 100$, $i = 1, 2$;
several local minima (exact number unspecified in the usual literature);
one global minimum: $x^* = (-3.788, -3.246)$, $f_3(x^*) = -1$.

4. f_4 : Branin RCOS, (RC), (2 variables)

$$f_4(x) = x_2 - (5/(4\pi^2))x_1^2 + (5/\pi)x_1 - 6)^2 + 10(1 - (1/8\pi))\cos(x_1) + 10$$

search domain: $-5 < x_i < 10$, $i = 1, 2$;
three global minimum: $x^* = (-\pi, 12.275)$, $(\pi, 2.275)$, $(9.42478, 2.475)$; $f_4(x^*) = 0.39788$.

5. f_5 : Zakharov, (Zn), (n variables)

$$f_5(x) = \left(\sum_{i=1}^n x_i^2\right) + \left(\sum_{i=1}^n 0.5i \cdot x_i\right)^2 + \left(\sum_{i=1}^n 0.5i \cdot x_i\right)^4$$

search domain: $-5 < x_i < 10$, $i = 1, 2, \dots, n$;
several local minima (exact number unspecified in the usual literature);
one global minimum: $x^* = (0, \dots, 0)$, $f_5(x^*) = 0$.

6. f_6 : B2 (2 variables)

$$f_6(x) = x_1^2 + 2x_2^2 - 0.3 \cos(3\pi x_1) - 0.4 \cos(4\pi x_2) = 0.7$$

search domain: $-100 < x_i < 100$, $i = 1, 2$;
several local minima (exact number unspecified in the usual literature);
one global minimum: $x^* = (0, 0)$, $f_6(x^*) = 0$.

7. f_7 : Michalewicz, (MZ), (2 variables)

$$f_7(x) = -\sum_{i=1}^n \sin(x_i) [\sin((i \cdot (x_i)^2)/\pi)]^{2m}$$

$m = 10$, search domain: $-\pi \leq x_i \leq \pi$, $i = 1, 2$;
one global minimum: $x^* = (2.25, 1.57)$, $f_7(x^*) = -1.8$.

8. f_8 : Shubert, (SH), (2 variables)

$$f_8(x) = \left\{ \sum_{j=1}^5 j \cdot \cos[(j+1)x_1 + j] \right\} \times \left\{ \sum_{j=1}^5 j \cdot \cos[(j+1)x_2 + j] \right\}$$

search domain: $-10 < x_i < 10$, $i = 1, 2$;

760 local minima; 18 global minima: $f_8(x^*) = -186.7309$.

9. f_9 : Goldstein and Price, (GP), (2 variables)

$$f_9(x) = [1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

search domain: $-2 < x_i < 2, i = 1, 2$;
4 local minima; one global minimum: $x^* = (0, -1)$,
 $f_9(x^*) = 3$.

10. f_{10} : De Jong, (DJ), (3 variables)

$$f_{10}(x) = x_1^2 + x_2^2 + x_3^2$$

search domain: $-5.12 < x_i < 5.12, i = 1, 3$;
one global minimum: $x^* = (0, 0, 0)$, $f_{10}(x^*) = 0$.

11. f_{11} : Shaffer's f_6 , Shaffer, (2variables)

$$f_{11}(x) = 0.5 - \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))^2}$$

search domain: $-100 < x_i < 100, i = 1, 2$;

several local minima (exact number unspecified in the usual literature);

one global minimum: $x^* = (0, 0)$, $f_{11}(x^*) = 0$.

12. f_{12} : Shekel, ($S_{4,n}$), (4 variables)

$$f_{12}(x) = -\sum_{i=1}^n \left[(x - a_i)^T (x - a_i) + c_i \right]^{-1};$$

$$x = (x_1, x_2, x_3, x_4)^T; a_i = (a_i^1, a_i^2, a_i^3, a_i^4)^T;$$

$S_{4,7}, n = 7$, 7 minima with one global minimum:
 $S_{4,7}(x^*) = -10.40294$
 $S_{4,10}, n = 10$, 10 minima with one global minimum:
 $S_{4,10}(x^*) = -10.53641$.

i		a_i^T			c_i
1	4.	4.	4.	4.	.1
2	1.	1.	1.	1.	.2
3	8.	8.	8.	8.	.2
4	6.	6.	6.	6.	.4
5	3.	7.	3.	7.	.4
6	2.	9.	2.	9.	.6
7	5.	5.	3.	3.	.3
8	8.	1.	8.	1.	.7
9	6.	2.	6.	2.	.5
10	7.	3.6	7.	3.6	.5