An Improved PSO-based of Harmony Search for Complicated Optimization Problems

LI Hong-qi[1]， LI Li[1], Tai-hoon Kim[2], XIE Shao-long[3]
1. Department of Computer Science and Technology, China University of Petroleum
2. Hannam University
3. Resource Information School, China University of Petroleum
Hq.li@cup.edu.cnl

Abstract

As an optimization technique, particle swarm optimization (PSO) has obtained much attention during the past decade. It is gaining popularity, especially because of the speed of convergence and the fact that it is easy to realize. To enhance the performance of PSO, an improved hybrid particle swarm optimization (IPSO) is proposed to solve complex optimization problems more efficiently, accurately and reliably. It provides a new way of producing new individuals through organically merges the harmony search (HS) method into particle swarm optimization (PSO). During the course of evolvement, harmony search is used to generate new solutions and this makes IPSO algorithm have more powerful exploitation capabilities. Simulation results and comparisons with the standard PSO based on several well-studied benchmarks demonstrate that the IPSO can effectively enhance the searching efficiency and greatly improve the search quality.

1. Introduction

Function optimization has received extensive research attention, and several optimization algorithm such as neural networks [1], evolutionary algorithms [2], genetic algorithms [3] and swarm intelligence-based algorithms [4-5] have been developed and applied successfully to solve a wide range of complex optimization problems. Most stochastic optimization algorithms including particle swarm optimizer (PSO) [6, 7] and genetic algorithm (GA) [3] have shown inadequate to complex optimization problems, as they rapidly push an artificial population toward convergence. That is, all individuals in the population soon become nearly identical. To improve PSO performance, several methods have been proposed. Many of these methods concerned predefining numerical coefficients, consisting of the maximum velocity, inertia weight, social factor and individual factor, which can affect various characteristics of the algorithm, such as convergence rate or the ability of global optimization. Recently, some hybrid implementations of PSO algorithm with other search methods. NM-PSO (Nelder-Mead-PSO) [8] comprises NM method at the top of level, and PSO at the lower level. CPSO (Chaotic PSO) [9] applies PSO to perform global exploration and chaotic local search to perform local search on the solutions produced in the global exploration process. These methods can equip PSO with extra facilities. In this paper, an improved PSO (IPSO) based of harmony search (HS) [10,11] is proposed to solve complex optimizations.

The PSO algorithm includes some tuning parameters that greatly influence the algorithm performance, often stated as the exploration-exploitation tradeoff: Exploration is the ability to test various regions in the problem space in order to locate a good optimum, hopefully the global one. Exploitation is the ability to concentrate the search around a promising candidate solution in order to locate the optimum precisely. Facing complicated optimizations, it's difficult to explore every possible region of the search space. Recently, harmony search (HS) algorithm imitates the improvisation process of music players and had been very successful in a wide variety of optimization problems [12-15], presenting several advantages with respect to traditional optimization techniques such as the following [12]:(a) HS algorithm imposes fewer mathematical requirements and does not require initial value settings of the decision variables. (b) As the HS algorithm uses stochastic random searches, derivative information is also unnecessary. (c) The HS algorithm generates a new vector, after considering all of the existing vectors, whereas the genetic algorithm (GA) only considers the two parent vectors. These features increase the flexibility of the HS algorithm and produce better solutions. In this study, one of the ways of integrating the concepts of these two optimization algorithms for solving complex optimization problems is explored.

The rest of the paper is organized as follows. Section 2 describes the standard particle swarm optimization (SPSO) and harmony search algorithm. Section3 motivates and describes the improved particle swarm optimizer (IPSO) combined with harmony search (HS) in details and gives the pseudo code of it. Section 4 presents the benchmark optimization problems used for experiment comparison of the algorithms, and the experimental settings for each algorithm. Section 5 contains the concluding remarks and the future work.

2. Related works

2.1. Standard PSO

PSO is a population-based, co-operative search meta-heuristic introduced by Kennedy and Eberhart. The fundament for the development of PSO is hypothesis that a potential solution to an optimization problem is treated as a bird without quality and volume, which is called a particle, coexisting and evolving simultaneously based on knowledge sharing with neighboring particles. While flying through the problem search space, each particle modifies its velocity to find a better solution (position) by applying its own flying experience (i.e. memory having best position found in the earlier flights) and experience of neighboring particles (i.e. best-found solution of the population). Particles update their positions and velocities as shown below:

$$v_{t+1}^i = \omega_t \cdot v_t^i + c_1 \cdot R_1 \cdot \left(p_t^i - x_t^i\right) + c_2 \cdot R_2 \cdot \left(p_t^g - x_t^i\right) \tag{1}$$

$$x_{t+1}^i = x_t^i + v_{t+1}^i \tag{2}$$

Where x_t^i represents the current position of particle i in solution space and subscript t indicates an iteration count; p_t^i is the best-found position of particle i up to iteration count t and represents the cognitive contribution to the search velocity v_t^i . Each component of v_t^i can be clamped to the range $[-v_{max}, v_{max}]$ to control excessive roaming of particles outside the search space; p_t^g is the global best-found position among all particles in the swarm up to iteration count t and forms the social contribution to the velocity vector; r_1 and r_2 are random numbers uniformly distributed in the interval $(0,1)$, where c_1 and c_2 are the cognitive and social scaling parameters, respectively; ω_t is the particle inertia, which is reduced dynamically to decrease the search area in a gradual fashion [15]. The variable ω_t is updated as

$$\omega_t = \left(\omega_{max} - \omega_{min}\right) \cdot \frac{\left(t_{max} - t\right)}{t_{max}} + \omega_{min} \tag{3}$$

Where ω_{max} and ω_{min} denote the maximum and minimum of ω_t respectively; t_{max} is a given number of maximum iterations. Particle i flies toward a new position according to Eq. (1) and (2). In this way, all particles of the swarm find their new positions and apply these new positions to update their individual best p_t^i points and global best p_t^g of the swarm. This process is repeated until termination conditions are met.

2.2. Harmony search

Harmony search (HS) algorithm is based on natural musical performance processes that occur when a musician searches for a better state of harmony, such as during jazz improvisation. The engineers seek for a global solution as determined by an objective function, just like the musicians seek to find musically pleasing harmony as determined by an aesthetic [16]. In music improvisation, each player sounds any pitch within the possible range, together making one harmony vector. If all the pitches make a good solution, that experience is stored in each variable's memory, and the possibility to make a good solution is also increased next time. HS algorithm includes a number of optimization operators, such as the harmony memory (HM), the harmony memory size (HMS, number of solution vectors in harmony memory), the harmony memory considering rate (HMCR), and the pitch adjusting rate (PAR). In the HS algorithm, the harmony memory (HM) stores the feasible vectors, which are all in the feasible space. The harmony memory size determines how many vectors it stores. A new vector is generated by selecting the components of different vectors randomly in the harmony memory.

For example, Consider a jazz trio composed of saxophone, double bass and guitar. There exist certain amount of preferable pitches in each musician's memory: saxophonist, {Do, Mi, Sol}; double bassist, {Si, Sol, Re}; and guitarist, {La, Fa, Do}. If saxophonist randomly plays {Sol} out of {Do, Mi, Sol}, double bassist {Si} out of {Si, Sol, Re}, and guitarist {Do} out of {La, Fa, Do}, that harmony (Sol, Si, Do) makes another harmony (musically C-7 chord). And if the new harmony is better than existing worst harmony in the HM, the new harmony is included in the HM and the worst harmony is excluded from the HM. This procedure is repeated until fantastic harmony is found. When a musician improvises one pitch, usually he (or she) follows any one of three rules: (1) playing any one pitch from his (or her) memory, (2) playing an adjacent pitch of one pitch from his (or her) memory, and (3) playing totally random pitch from the possible sound range. Similarly, when each decision variable chooses one value in the HS algorithm, it follows any one of three rules: (1) choosing any one value from HS memory (defined as memory considerations), (2) choosing an adjacent value of one value from the HS memory (defined as pitch adjustments), and (3) choosing totally random value from the possible value range (defined as randomization). The three rules in HS algorithm are effectively directed using two parameters, i.e., harmony memory considering rate (HMCR) and pitch adjusting rate (PAR).

The steps in the procedure of harmony search are as follows:

Step1. Initialize the problem and algorithm parameters.
Step2. Initialize the harmony memory (HM).
Step3. Improvise a new harmony from the HM.
Step4. Update the HM.
Step5. Repeat Steps 3 and 4 until the termination criterion is satisfied.

3. The realization of IPSO based of HS

This section describes the implementation of proposed improvement in PSO using HS approach. The proposed method, called, IPSO (improved particle swarm optimization) is based on the common characteristics of both PSO and HS algorithms. HS algorithm provides a new way to produce new particles. Different from PSO and GA, HS algorithm generates a new vector after considering all of the existing vectors.

HS algorithm can produce new solution and the parameters of HMCR and PAR are introduced to allow the solution to escape from local optima and to improve the global optimum prediction of the algorithm. Enlightened by this, the HS realization concept has been used in the PSO in this paper to exploration the potential solution space. In summary，the realization of improved PSO can be described as follows:

Step1.Initializing the parameters of PSO and HS;

Step2.Initailizing the particles;

Step3.Evaluating particles according to their fitness then descending sort them;

Step4.Performing HS and generating a new solution;

Step5.If the new solution is better than the worst particle then replacing it with the new one;

Step6.Updating particles according to formula 1,2;

Step7.The program is finished if the termination conditions are met otherwise go to step3.

All particles are looked as HM in IPSO and the dimension of particle is HMS.

Improvising a new harmony from the HM can be realized as follows:

A new harmony vector, $x' = \{x'_1, x'_2, \Lambda\ , x'_N\}$ is generated form the HM based on memory considerations, pitch adjustments, and randomization. For instance, the value of   for the new vector can be chosen from any value in the specified HM rang $\left(x_1^1 - x_1^{HMS}\right)$. Values of the other design variables can be chosen in the same manner. Here, it’s possible to choose the new value using the HMCR parameter, which varies between 0 and 1 as follows:

$$x'_i \leftarrow \begin{cases} x'_i \in \{x_i^1, x_i^2, \Lambda\ , x_i^{HMS}\} & with \quad probality \quad HMCR \\ x'_i \in X_i & with \quad probality \quad 1 - HMCR \end{cases} \tag{4}$$

The HMCR is the probability of choosing one value from the historic values stored in the HM, and (1-HMCR) is the probability of randomly choosing one feasible value not limited to those stored in the HM. For example, an HMCR of 0.95 indicates that the HS algorithm will choose the design variable value from historically stored values in the HM with a 95% probability and from the entire feasible range with a 5% probability. An HMCR value of 1.0 is not recommended because of the possibility that the solution may be improved by values not stored in the HM. This is similar to the reason why the genetic algorithm uses a mutation rate in the selection process.

Every component of the new harmony vector, $x'_i = \{x'_1, x'_2, \Lambda\ , x'_N\}$, is examined to determine whether it should be pitch-adjusted. This procedure uses the parameter that set the rate of adjustment for the pitch chosen from the HM as follows:

Pitch adjusting decision for

$$x'_i \leftarrow \begin{cases} Yes & with \quad probality \quad PAR \\ No & with \quad probality \quad 1 - PAR \end{cases} \tag{5}$$

The Pitch adjusting process is performed only after a value is chosen from the HM. The value (1-PAR) sets the rate of doing nothing. A PAR of 0.3 indicates that the algorithm will choose a neighboring value with   probability. If the pitch adjustment decision for x' is Yes, and x' is assumed to be $x_i(k)$, i.e., the kth element in X_i , the pitch-adjusted value of $x_i(k)$ is:

$$x' = x' + \alpha \tag{6}$$

Where α is the value of $bw \times u(-1,1)$, bw is an arbitrary distance bandwidth for the continuous design variable, and $u(-1,1)$ is a uniform distribution between -1 and 1.

The HMCR and PAR parameters introduced in the harmony search help the algorithm escape from local optima and to improve the global optimum prediction of the HS algorithm.

After improvising a new harmony, evaluating the new one and if it is better than the worst one in the HM in terms of the objective function value, the new one is included in the HM and the existing worst harmony is excluded from the HM. The HM is then sorted by the objective function value.

4. Simulation results and comparisons

4.1. Experimental parameters setting

For the purpose of comparison we have set SPSO and IPSO with the same parameters.

The population size was set at 30. The acceleration constants $c_1 = c_2 = 1.49445$. The inertia weight ω is critical for the convergence behavior of PSO. A suitable value for the inertia weight usually provides a balance between global and local exploration abilities and consequently results in a better optimum solution. In our programming ω is set to 0.729. Both SPSO and IPSO are set to terminate after 3000 fitness evaluations.

The parameters in HS are set as follows: PAR=0.2, HMCR=0.9, HS is terminated after 200 fitness evaluation.

4.2. Test functions

In our experimental studies, a set of 3 benchmark functions was employed to evaluate the IPSO algorithm in comparison with others:

Generalized Rastigrin function

$$f_1(x) = \sum_{i=1}^{N}\left[x_i^2 - 10cos(2\pi x_i) + 10\right] \tag{7}$$

Griewank function

$$f_2(x) = \sum_{i=1}^{N}\left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right] \tag{8}$$

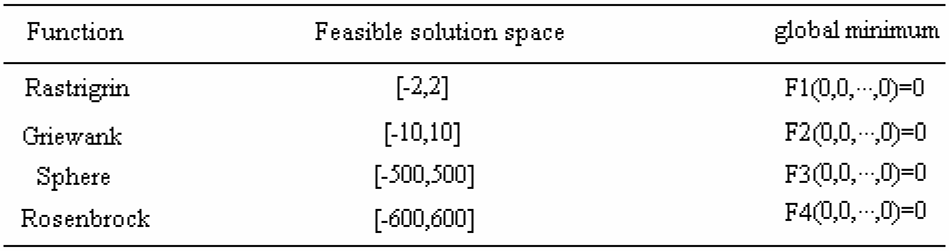
Sphere function

$$f_3(x) = \sum_{i=1}^{N} x_i^2 \tag{9}$$

Rosenbrock function

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} cos\left(\frac{x_i}{\sqrt{i}}\right) \tag{10}$$

Table.1. Basic characters of the test functions



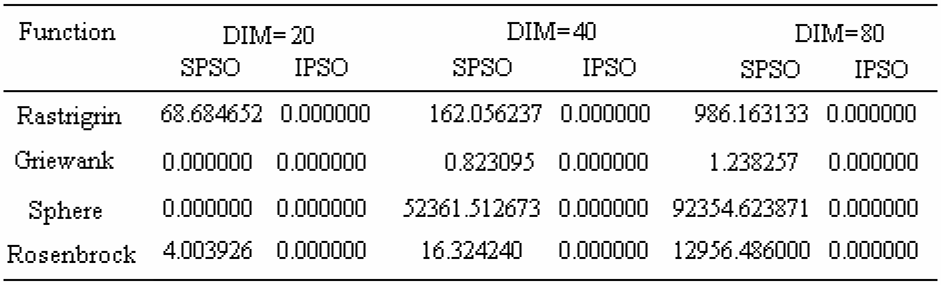
Function	Feasible solution space	global minimum
Rastrigrin	[-2,2]	F1(0,0,···,0)=0
Griewank	[-10,10]	F2(0,0,···,0)=0
Sphere	[-500,500]	F3(0,0,···,0)=0
Rosenbrock	[-600,600]	F4(0,0,···,0)=0

The above benchmark functions were tested widely. They can be grouped as unimodal and multimodal functions where the number of local minima increases exponentially with the problem dimension. It is very difficult to locate their global solution with the increasing dimensions. The characters of the test functions are listed in Table 1.

4.3. Experimental results

To evaluate the performance of the proposed IPSO, the dimensions were set to 20, 40 and 80 respectively for comparison. The experimental results of SPSO and IPSO are given in table 2.

Table.2.The results of PSO algorithms



Function	DIM=20		DIM=40		DIM=80	
	SPSO	IPSO	SPSO	IPSO	SPSO	IPSO
Rastrigrin	68.684652	0.000000	162.056237	0.000000	986.163133	0.000000
Griewank	0.000000	0.000000	0.823095	0.000000	1.238257	0.000000
Sphere	0.000000	0.000000	52361.512673	0.000000	92354.623871	0.000000
Rosenbrock	4.003926	0.000000	16.324240	0.000000	12956.486000	0.000000

Table 2 shows the global minimum found by PSO algorithm as the dimension is set to different values. When dimension is set to 20, the global best solutions of function Griewank and Sphere have been found while Rastrigrin and Rosenbrock do not. As the dimension increasing, it is difficult for SPSO to locate the global best position and the algorithm traps into local point. For example, the global best minima are 162.056237 and 986.163133 when dimensions equal to 40 and 80 for function Rastrigrin. We can see the performance deteriorates as the dimensionality of the search space increases. Facing complex optimization functions SPSO is at the end of its wits. While the improved hybrid PSO-based of harmony search can find the best solutions successfully for these functions at any dimensionalities; the precision of solutions is much higher than SPSO at the same time.

Because of the limitation of pages, we only present the result of Rosenbrock when dimension is 20 in Fig.1~4.

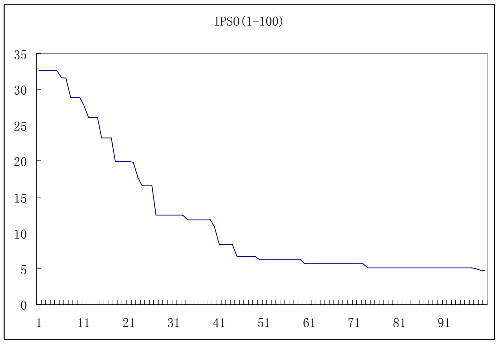


Figure 1. The result of Rosenbrock

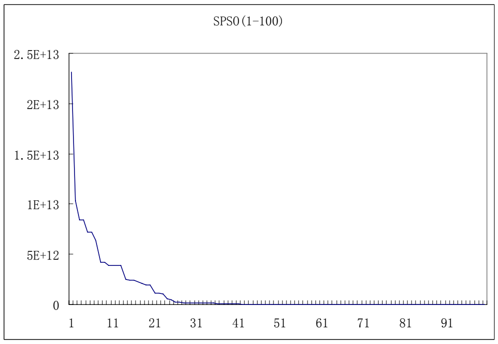


Figure 2. The result of Rosenbrock

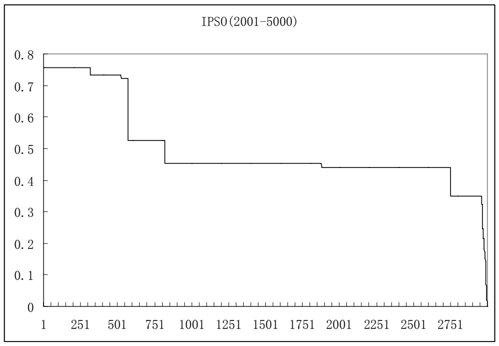


Figure 3. The result of Rosenbrock

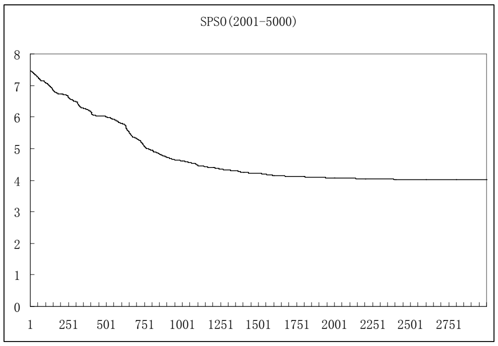


Figure 4. The result of Rosenbrock

Figure 1 presents the performance of IPSO and Figure 2 presents the performance of SPSO when generation is between 1 and 100. From the two figures we can see that IPSO's best solution is much better than SPSO. Figure 3 and 4 give the condition when generation is between 2000 and 5000. From these figures we can draw conclusions that the improved PSO not only can find the global solution but also has a rapid convergence speed than standard PSO.

5. Conclusions

In summary, we show that improved PSO-based of harmony search can be successfully applied to complicated optimization problems. Facing complex problems the stand PSO performance starts to suffer and it is easy to trap into local extremums. For solving this kind of optimizations, a novel hybrid PSO is proposed in this paper. The main course leading the search frustrated is the weak of exploration. Harmony search (HS) provides new methods of producing new individuals. So a novel hybrid algorithm is proposed merging HS into PSO. At last the numerical examples show that the improved algorithm is fairly effective.

Refrenece:

[1] C. M. Bishop. Neural Networks for Pattern Recognitionm. Oxford University Press, 1995.

[2] T. Back, D.B. Foge, A. Michalewicz(Eds.). Handbook of Evolutionary Computation, IOP Publishers and Oxford University Press, 1997.

[3] Holland J.H. Adaptation in Natural and Artificial Systems. MIT Press, 1975.

[4] J. Kennedy, R. C. Eberhart. Swarm Intelligence. Morgan Kaufman, 2001.

[5] A. P. Engelbrecht. Fundamentals of Computational Swarm Intelligence. John Wiley & Sons, 2005.

[6] R. C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan,1995,pp.39-43.

[7] KENNEDY J, EBERHART R C. Particle swarm optimization[R]. Piscataway: Proceedings of IEEE International Conference on Neural Networks, NJ: IEEE Press, 1995.4: 1942-1948.

[8] S.K.S. Fan, Y.C. Liang, E. Zahara. Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. Engineering Optimization 36(4), 2004, pp. 401-418.

[9] B.Liu, L. Wang, Y.H. Jin, F. Tang, D.X. Huang. Improved particle swarm optimization combined with chaos, Chaos Solitons & Fractals 25, 2005, pp. 1261-1271.

[10] Z.W.Geem, J.H.Kim,G.V.Loganathan, A new heuristic optimization algorithm: harmony search,Simulation 76(2)(2001) 60-68.

[11] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continues engineering optimization: harmony search theory and practice, Comput. Meth. Appl. Mech. Eng.194 (2004) 3902-3933.

[12] J.H.Kim,Z.W.Geem, E.S. Kim. Parameter estimation of the nonlinear Muskingum model using harmony search. J.Am. Water Resour. Assoc. 37(5) (2001),pp.1131-1138.

[13]Z.W.Geem,J.H.Kim,G.V.Loganathan. Harmony search optimization: application to pipe network design. Int.J.Model. Simul.22(2) (2002), pp. 125-133.

[14] S.L.Kang, Z.W.Geem. A new structural optimization method based on the harmony search algorithm. Comput.Struct.82(9-10)(2004), PP. 781-798.

[15] Z.W.Geem, C.Tseng, Y.Park. Harmony search for generalized orienteering problem: best touring in China, in: Springer Lecture Notes in Computer Science, vol.3412,2005,pp. 741-750.

[16] R.C.Eberhart,J.Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan,1995,pp.39-43.

Authors

LI Hong-qi Professor and Ph.D. Advisor in the Computer Science and Technology department of the University of Petroleum, Beijing, China. Li ’ s research interests are swarm intelligence, particle swarm optimization, intelligent information processing and application, software engineering.

LI Li Received M.S. degree in Computer application from Beijing institute of technology in 2004. She is now in pursuit for Ph.D. degree in Artificial Intelligence & Swarm Intelligence from China petroleum university (Beijing). Her research interests include swarm intelligence, particle swarm optimization algorithm, genetic algorithm, and intelligent information processing and application.

Tai Hoon Kim Received his M.S. degrees and Ph.D. in Electrics, Electronics & Computer Engineering from the Sungkyunkwan University, Korea. Now he is a professor of Hannam University. He was a program committee of SNPD 2004, IPC 2007 and so on. He is an EIC of JSE and IJSIA Journal. He researched security engineering, the evaluation of information security products or systems with Common Criteria and the process improvement for security enhancement.

XIE Shao-long is a graduate student of School of Recourses and Information at China University of Petroleum. Mainly major in earth exploration and information technology. His research interests include earth resources information evaluating and processing, particle swarm optimization.