An Improved Particle Swarm Optimization Algorithm for Global Numerical Optimization

Bo Zhao

Jiangsu Electric Power Research Institute Corporation Limited Nanjing 210063, Jiangsu, China zhaobozju@163.com

Abstract. This paper presents an improved particle swarm optimization algorithm (IPSO) for global numerical optimization. The IPSO uses more particles' information to control the mutation operation. A new adaptive strategy for choosing parameters is also proposed to assure convergence of the IPSO. Meanwhile, we execute the IPSO to solve eight benchmark problems. The results show that the IPSO is superior to some existing methods for finding the best solution, in terms of both solution quality and algorithm robustness.

1 Introduction

Particle swarm optimization (PSO) is one of the evolutionary computation techniques based on the social behavior metaphor [1]. It is initialized with a population of random solutions, conceptualized as particles. Each particle in PSO flies through the search space with a velocity according to its own and the whole population's historical behaviors. The particles have a tendency to fly toward better search areas over the course of a search process. In recent years, PSO has been widely used for numerical optimization and many other engineering problems. Global optimization problems arise in almost every field of science, engineering, and business. Now, the PSO is becoming one of the popular methods to address them due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. But, in global optimization problems, the major challenge is that the POS may be trapped in the local optima of the objective function. The issue is particularly challenging when the dimension is high and there are numerous local optima. Therefore, some improved methods have been proposed, such as the fully informed particle swarm [2], a hybrid of GA and PSO [3], a cooperative approach to PSO [4], a hierarchical particle swarm optimizer [5], a multiagent-based PSO [6], and so on.

In the standard PSO algorithm, the neighborhood of a particle consist of all particles, so that the global best position, i.e., the best solution found so far, directly influences its behavior. Hence, for social science context, a PSO system combines a social-only component model and a cognition-only model. The social-only model component suggests that individuals ignore their own experience and adjust their behavior according to the successful beliefs of individuals in the neighborhood. On the other hand, the cognition-only model component treats individuals as isolated beings. The real strength of the particle swarm derives from the social interactions

among

particles as they search the space collaboratively. So, in this paper, an improved adaptation strategy with enhanced social influence is proposed for PSO algorithm. This adaptation strategy is similar to the social society in that a group of leaders, i.e. several fittest individuals in a population, play a major role in reproduction process. This is different from the standard PSO with only one leader, the fittest individual in a generation, is selected to the reproduction process. The improved particle swarm optimization (IPSO) has been tested with some benchmark functions. The experimental results show that the IPSO achieves a good performance for test functions, which illustrate that the IPSO overcomes the problem of premature convergence of the standard PSO method in some degree.

2 Improved Particle Swarm Optimization (IPSO)

2.1 Standard Particle Swarm Optimization Approach (PSO)

PSO is developed through simulation of bird flocking in two-dimension space. According to the research results for a flock of birds, birds find food by flocking. The observation leads the assumption that information is shared inside flocking. Moreover, according to observation of behavior of human groups, behavior of each individual (agent) is also based on behavior patterns authorized by the groups such as customs and other behavior patterns according to the experiences by each individual. The position of each agent is represented by XY-axis position and the velocity is expressed by v_x (the velocity of X-axis) and v_y (the velocity of Y-axis). Modification of the agent position is realized by using the position and the velocity information [7].

Searching procedures by PSO based on the above concept can be described as follows: a flock of agents optimizes a certain objective function. Each agent knows its best value so far (*pbest*) and its position. The information is corresponding to personal experiences of each agent. Moreover, each agent knows the best value so far in the group (*gbest*) among *pbests*. The information is corresponding to knowledge of how the other agents around them have performed. Namely, each agent tries to modify its position using the following information:

- The distance between the current position and *pbest*, p_t .
- The distance between the current position and *gbest*, \hat{p}_{t} .

The modification can be represented by the concept of velocity. Velocity of each agent can be modified by the following equation:

$$v_{t+1} = c_0 v_t + c_1 r_1(t) \times (p_t - x_t) + c_2 r_2(t) \times (\hat{p}_t - x_t), \qquad (1)$$

where c_0 , c_1 and c_2 are positive constant coefficients, r_1 and r_2 are uniformly distributed random numbers in [0,1], v_t is the current velocity of the particle at iteration t, x_t is current position of the particle at iteration t, v_{t+1} is the modified velocity at iteration t+1.

Using the above equation (1), a certain velocity that gradually gets close to *pbests* and *gbest* can be calculated. The current position (searching point in the solution space) can be modified by the following equation:

$$x_{t+1} = x_t + v_{t+1}, (2)$$

where x_{t+1} is the modified position at iteration t+1.

2.2 Improved Particle Swarm Optimization Approach (IPSO)

Due to the shortage of the social interactions among particles in the standard PSO algorithm, an improved adaptation strategy with enhanced social interactions is proposed for particle swarm optimization (PSO) algorithm in this section. This adaptation strategy uses more particles' information to control the mutation operation and extends the original formulas of the PSO method, which can search the global optimal solution more effectively.

The third term added to the right-hand-side of the velocity equation (1) is derived from the successes of the others; it is considered a "social influence" term. It is found that when this effect is removed from the algorithm, performance is abysmal. So the social interaction is an important factor to improve the PSO performance. To enhance the social interactions in the algorithm, this paper proposes a new method of improved PSO using some fittest particles' information to modify particle's position and velocity. Namely, at *i*th iteration, we rearrange the particles in descending order according to their fitness and select the last *n* particles to modify particle's position and velocity. Let $\hat{p}_{i,t}$ denote current position of the particle *i* in these particles at iteration *t*. The updating equations of IPSO method can be described in the following:

$$v_{i+1} = c_0 v_i + c_1 r_1(t) \times (p_i - x_i) + \sum_{i=1}^n c_{2,i} r_{2,i}(t) \times (\hat{p}_{i,i} - x_i),$$
(3)

$$x_{t+1} = x_t + v_{t+1}, (4)$$

Each particle of IPSO method modifies its position and velocity using the best solution particle achieved and several *gbest* of neighborhood particles. It is similar to the social society in that the group of leaders could make better decisions. However, in standard PSO, only one *gbest* of neighborhood particles is employed. This process using some neighborhood particles can be called 'intensifying' and 'enhancing' the social influence. Based on this understanding, we should intensify these particles that could lead individuals to better fitness. This reinforces the exploitation and exploration of PSO. As a particle swarm population searches over time, individuals are drawn toward one another's successes, with the usual result being clustering of individuals in optimal regions of the space.

2.3 IPSO Algorithm's Convergence

The form of the recurrence relation of particle's position can be derived as follows: substituting equation (3) into (4) and from equation $v_t = x_t \cdot x_{t-1}$, we have the following equation:

$$x_{t+1} = \left(1 + c_0 - c_1 r_1(t) - \sum_{i=1}^n c_{2,i} r_{2,i}(t)\right) x_t - c_0 x_{t-1} + c_1 r_1(t) p_t + \sum_{i=1}^n c_{2,i} r_{2,i}(t) \hat{p}_{i,t},$$
(5)

which is a non-homogeneous recurrence relation that can be solved using standard recursive techniques. This recurrence relation can be written as a matrix-vector product,

$$\begin{bmatrix} x_{t+1} \\ x_t \\ 1 \end{bmatrix} = \begin{bmatrix} 1+c_0 -\eta_0 -\eta_1 & -c_0 & \eta_0 p_t +\eta_1 p \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ x_{t-1} \\ 1 \end{bmatrix},$$
 (6)

where $\eta_0 = c_1 * r_1(t)$, $\eta_1 = \sum_{i=1}^n c_{2,i} * r_{2,i}(t)$, $\eta_1 p = \sum_{i=1}^n c_{2,i} * r_{2,i}(t) * \hat{p}_{i,i}$. The characteristic polynomial of the equation (6) is,

$$(\lambda-1)(\lambda^2-(1+c_0-\eta_0-\eta_1)\lambda+c_0),$$

which has a trivial root of $\lambda_1 = 1.0$, and two other solutions (7) and (8),

$$\lambda_2 = \frac{(1+c_0 - \eta_0 - \eta_1) + \gamma}{2}, \tag{7}$$

$$\lambda_3 = \frac{(1+c_0 - \eta_0 - \eta_1) - \gamma}{2}, \qquad (8)$$

where

$$\gamma = \sqrt{\left(1 + c_0 - \eta_0 - \eta_1\right)^2 - 4c_0} , \qquad (9)$$

Note that λ_1 , λ_2 and λ_3 are both eigenvalues of the equation (6). The explicit form of the recurrence relation (5) is then given by equation,

$$x_{t} = k_{1}\lambda_{1}^{t} + k_{2}\lambda_{2}^{t} + k_{3}\lambda_{3}^{t} = k_{1} + k_{2}\lambda_{2}^{t} + k_{3}\lambda_{3}^{t}, \qquad (10)$$

where k_1 , k_2 and k_3 are constants determined by the initial conditions of the system at each iteration.

An important aspect of the behavior of a particle concerns whether its trajectory (specified by x_t) converges or diverges. The conditions under which the sequence $\{x_t\}_{t=0}^{+\infty}$ will converge are determined by the magnitude of the values λ_2 and λ_3 as computed using equations (7) and (8). From equation (9), it is clear that there are two cases:

1) Case A:
$$(1+c_0-\eta_0-\eta_1)^2 < 4c_0$$

In this case, *r* will be a complex number with a non-zero imaginary component. A complex *r* results in λ_2 and λ_3 , being complex numbers with non-zero imaginary components as well. Consider the value of x_t in the limit, thus equation (10) becomes,

$$x_{t} = k_{1} + k_{2} \cdot \|\lambda_{2}\|' \cdot e^{j\theta t} + k_{3} \cdot \|\lambda_{3}\|' \cdot e^{j\sigma t}, \qquad (11)$$

where $\|\lambda_2\|' \cdot e^{j\theta_t}$ and $\|\lambda_3\|' \cdot e^{j\sigma_t}$ are the exponent expression of the trivial roots. Clearly, equation (11) explains the sequence $\{x_t\}_{t=0}^{+\infty}$ will converge when $\max(\|\lambda_2\|, \|\lambda_3\|) < 1$, so, $\lim_{t \to +\infty} x_t = k_1 + k_2 \cdot \lambda_2^t + k_3 \cdot \lambda_3^t = k_1$.

2) Case B: $(1+c_0-\eta_0-\eta_1)^2 \ge 4c_0$

In this case, r, λ_2 and λ_3 , will be real numbers, from equation (10), if $\max(\|\lambda_2\|, \|\lambda_3\|) < 1$, then the sequence $\{x_t\}_{t=0}^{+\infty}$ converges.

From the analysis of above two cases, we obtain the convergence condition of the sequence $\{x_i\}_{i=0}^{+\infty}$ is max $(\|\lambda_2\|, \|\lambda_3\|) < 1$.

One popular choice of updating parameters is $c_0 = 0.7298$, $c_1 = 1.49618$ and $\sum_{i=1}^{n} c_{2,i} = 1.49618$ [8]. On account of $\eta_0 = c_1 * r_1(t)$ and $\eta_1 = \sum_{i=1}^{n} c_{2,i} * r_{2,i}(t)$, so $\eta_0 \in (0, 1.49618)$, $\eta_1 \in (0, 1.49618)$ and $\eta_0 + \eta_1 \in (0, 2.9924)$. When $\eta_0 + \eta_1 \in (0, 0.0212]$, from equation (9), it will imply a real-valued γ , which corresponds to case B, then $\gamma \ge 0$ and $\max(\|\lambda_2\|, \|\lambda_3\|) = \frac{1.7298 - \eta_0 - \eta_1 + \gamma}{2} < 1$. Similarly, $\eta_0 + \eta_1 \in (0.0212, 2.9924)$ will result in a complex *r* value, which corresponds to case A and from equations (7) and (8) we can assure $\|\lambda_2\| = \|\lambda_3\| < 1$. The above analysis shows that the choices of parameters satisfy the convergence conditions and will assure convergence of the sequence $\{x_t\}_{t=0}^{+\infty}$.

IPSO method differs from PSO method is that η_1 is the sum of the coefficients of *n gbest* particles. The degree of importance of each particle is weighed through particle's fitness value. The better the particle's fitness value is, the more important the particle's influence is. So the paper proposed the rule of updating parameters chosen as follows:

$$c_{2,i} = \frac{\frac{1}{\hat{f}_i}}{\sum_{i=1}^{n} \frac{1}{\hat{f}_i}} \cdot 1.49618 , \qquad (12)$$

where \hat{f}_i is the *gbest* particle's value. This adaptive strategy of updating parameters in IPSO can assure convergence of IPSO method and enhance the global convergence capability of IPSO method.

3 Numerical Experiments and Results

In order to verify the effectiveness and efficiency of the proposed IPSO method, eight benchmark functions have been used in Table 1.

Function Name	Function Expression	Dimensions (N)	Initial Range
Schwefel	$F_1(x) = \sum_{i=1}^{N} \left(-x_i \sin \sqrt{ x_i } \right)$	30	(-500,500)
Rastrigin	$F_{2}(x) = \sum_{i=1}^{N} \left[x_{i}^{2} - 10\cos(2\pi x_{i}) + 10 \right]$	30	(- 5.12,5.12)
Ackley	$F_{3}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{N} x_{i}^{2}}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^{N} \cos(2\pi x_{i})\right)$	30	(-32,32)
Crievent	$F(x) = \frac{1}{N} \sum_{i=1}^{N} \frac{x^2}{i} = \prod_{i=1}^{N} \cos\left(\frac{x_i}{i}\right) \pm 1$	20	(600 600)
Griewank	$r_4(x) = \frac{1}{4000} \sum_{i=1}^{k} x_i = \prod_{i=1}^{k} \cos(\sqrt{1} \sqrt{i}) + 1$	30	(-000,000)
Rosenbrock	$F_{5}(x) = \sum_{i=1}^{N-1} \left[100(x_{i+1} - x_{i})^{2} + (1 - x_{i})^{2} \right]$	30	(-30,30)
Sphere	$F_6(x) = \sum_{i=1}^{N} x_i^2$	30	(-100,100)
Schwefel	$F_{7}(x) = \sum_{i=1}^{N} x_{i} + \prod_{i=1}^{N} x_{i} $	30	(-10,10)
Schwefel	$F_8(x) = \sum_{i=1}^{N} \left(\sum_{j=1}^{i} x_j\right)^2$	30	(-100,100)

Table 1. Comparisons of generalization capability with published results

Some parameters must be assigned to before the IPSO method is used to solve problems in following experiments. Since the test problem dimensions are high, a moderate population size is set to 70 and the maximal generation is set to 1000. To evaluate uncertain value combinations of n of the IPSO method, we have been executed 50 times to solve the above test function problem under various value combinations. The results show that the best solution can be obtained by the IPSO method when n=4. Hence, in the following study, we always choose n=4.

Owing to the randomness in IPSO, the algorithm is executed 50 independent times on each test function. Table 2 shows the optimal results of IPSO. From Table 2, we see that the mean function values are equal or close to the optimal ones, and the standard deviations of the function values are relatively small, except for function F_2 . These results indicate that the proposed IPSO method can find optimal or close-tooptimal solutions, and its solution quality is quite stable.

Table 3 summarizes the optimal results as obtained by some existing methods. These existing algorithms include standard particle swarm optimization algorithm (PSO), conventional genetic algorithm (CGA), orthogonal genetic algorithm with quantization (QGA/Q), fast evolutionary programming (FEP), evolutionary optimization (EO), enhanced simulated annealing (ESA) and conventional evolutionary programming with Gaussian mutation operator (CEP/GMO). Since each of these existing algorithms was executed to solve some of the above-mentioned functions,

Table 3 has included all of the available results for comparison. From Table 3, these results show that the optimal solutions determined by the IPSO lead to high quality solutions, which confirms that the IPSO is well capable of determining the global or close-to-optimal solutions. Firstly, Table 3 compares IPSO with PSO and CGA. For all numerical optimization functions, IPSO gives better solutions than PSO and CGA. Compared IPSO with PEP and OGA/Q, we see that FEP and OGA/Q obtain good performance on numerical optimization problem, even better solutions in some optimization functions. But the IPSO method is simple and has not complicated operator, not as PEP and OGA/Q. However, in eight benchmark functions, except for F_2 , the optimal solutions obtained by the IPSO are nothing less than these solutions obtained by PEP and OGA/Q. It indicates that IPSO can, in general, give good mean solution quality. At the same time, compared IPSO with EO and ESA, as can been seen, IPSO obtains better solutions in the available results for comparison, and displays a good performance in solving these global numerical optimization problems.

	Mean function	Standard deviation	Globally minimal
Test function	value	of function value	function value
F_{I}	-12569.487	7.056×10 ⁻⁶	-12569.5
F_2	11.5312	0.1242	0
F_{3}	8.253×10 ⁻¹⁶	7.413×10 ⁻¹⁷	0
F_4	0	0	0
F_5	9.584×10 ⁻¹	2.761×10 ⁻¹	0
F_6	4.096×10 ⁻⁹⁶	1.721×10 ⁻⁹⁶	0
F_7	1.654×10 ⁻⁸	7.357×10 ⁻⁹	0
F_8	0	0	0

Table 2. Optimal results of IPSO

Table 3. Comparison between IPSO and some existing algorithms

	Mean function value									
	IPSO	PSO	CGA	OGA/Q	FEP	EO	ESA	CEP/GMO		
F_{I}	-12569.487	-9903.8	-9094.75	-12569.45	-12554.5					
F_2	11.5312	26.8639	22.967	0	$4.6 imes 10^{-2}$	46.47		120.00		
F_3	$8.25 imes 10^{-16}$	7.99×10^{-15}	2.697	$4.44 imes 10^{-16}$	1.8×10^{-2}			9.10		
F_4	0	0.022	1.258	0	$1.6 imes 10^{-2}$	0.40		2.52×10^{-7}		
F_5	$9.58 imes 10^{-1}$	16.770	150.79	$7.52 imes 10^{-1}$		1911.59	17.10	86.70		
F_6	0	4.83×10^{-48}	4.96	0	$5.7 imes 10^{-4}$	9.88		3.09×10^{-7}		
F_7	5.40 × 10 ⁻²²	1.65×10^{-8}	7.93×10^{-1}	0	8.1 × 10 ⁻³			1.99×10^{-3}		
F_8	$2.70 imes 10^{-11}$	0.324	18.82	0	$1.6 imes 10^{-2}$			17.60		

Figure 1 shows the evolution process of F_1 obtained using IPSO, PSO and CGA respectively. It is clear for the figure that the solution by IPSO is converged to high quality solutions at the early iterations. Similar results are obtained for F_2 to F_8 . According to the above analysis, considering together more particles' information to control the mutation operation, the IPSO method performs better than the PSO, both in the quality of the solution discovered and in the velocity of convergence, and simulation results show that IPSO outperforms CGA and PSO.



Fig. 1. Optimization procedure by IPSO, PSO and CGA

4 Conclusions

An improved particle swarm optimization algorithm (IPSO) has been developed for determination of the optimal or close-to-optimal solutions of global numerical optimization problem. The IPSO approach uses more particles' information to control the mutation operation. The convergence property of IPSO is analyzed using standard results from the dynamic system theory and guidelines for proper algorithm parameter selection are derived. A new adaptive strategy for choosing parameters is also proposed to assure convergence of IPSO. Meanwhile, it was found from experimental results that IPSO could find higher quality solutions reliably with the faster converging characteristics than some existing methods on the problem studies.

References

- Kennedy, J., Eberhart, R. C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. (1995) 1942-1948
- Mendes, R., Kennedy, J.: The fully informed particle swarm: simpler, maybe better. IEEE Trans. Evolutionary Computation. 3 (2004) 204-210
- 3. Juang, C.F.: A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans. System, Man and Cybernetics-Part B. 2 (2004) 997-1006
- 4. van den Bergh, F., Engelbrecht, P.: A cooperative approach to particle swarm optimization,. IEEE Trans. Evolutionary Computation. 3 (2004) 225-239
- Janson, S., Middendorf, M.: A hierarchical particle swarm optimizer and its adaptive variant. IEEE Trans. System, Man and Cybernetics-Part B, 6 (2005) 1272-1282
- 6. Zhao, B., Guo, C.Y., Cao, Y.J.: A multiagent-based particle swarm optimization approach for optimal reactive power dispatch. IEEE Trans. Power Systems. 2 (2005) 1070-1078
- 7. Kennedy, J., Eberhart, R.C.: Swarm Intelligence. Morgan Kaufmann Publishers (2001)
- Clerc, M., Kennedy, J.: The Particle Swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. on Evolutionary Computation. 1 (2002) 58-73