Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2020.010138
Article

Study on the Application of an Improved Intelligent Group Algorithm

Fengjuan Wang[1,2,*]

[1]Qiqihar University, Qiqihar, 161006, China
[2]Heilongjiang Province Collaborative Innovation Center for Intelligent Manufacturing Equipment Industrialization, Qiqihar, 161006, China
[*]Corresponding Author: Fengjuan Wang. Email: wfj320110@163.com

Abstract: The Particle swarm optimization algorithm (PSO) and the simulated annealing algorithm (SA) are two well-known stochastic and intelligent methods used for optimization. Both methods have some shortcomings. On the basis of the shortcomings of PSO and SA, this paper offers an enhanced intelligent group algorithm on the basis of the roulette rule to improve the parameter velocity ν of (PSO) and the initial temperature of SA algorithm. This paper gives a detailed introduction to the principle and flow of the new algorithm and introduces the application status of the new algorithm.

Keywords: Intelligent group algorithm; particle swarm optimization; simulated annealing algorithm; application field

1 Introduction

The PSO and SA works on the stochastic approach and both algorithms have capabilities to solve many multi-dimensional problems well, but the particle swarm optimization algorithm (PSO) can easily fall into the problem of local minimum and premature convergence, while the simulated annealing algorithm (SA) has a fast jump out of the local extremum and better global convergence. performance. Therefore, this paper combines the gains of the PSO and SA algorithms to avoid their shortcomings. The fusion principle of this paper begins with roulette rules and takes speed as the fusion point for effective fusion. By improving the initial temperature of the particle swarm algorithm (PSO) and the simulated annealing algorithm (SA), a new efficient algorithm is wished-for, that is, the particle swarm simulated annealing algorithm, referred to as the PSO-SA algorithm.

2 Original Intention of Merging New Algorithms

Both the particle swarm optimization and simulated annealing algorithms can solve many practical problems alone, but they all have their own shortcomings. Based on their respective shortcomings, this paper reconstructs a new intelligent fusion algorithm by using complementary principles.

The PSO algorithm uses a multi-point parallel search method to search all solution space, because of the mechanism of the information exchange between particles and particles. Therefore, in a short time, most of the particles can converge to a certain local extreme point in the solution space [1–2]. At the beginning of the algorithm operation, the particle's flight speed is faster. As the algorithm runs, the particle speed is slower and slower. Finally, when the velocity of all the particles is close to zero, the algorithm falls into the local extreme point. We mistakenly believe that the algorithm is in a state of convergence, and all the particles have converged to this position. This is the typical premature convergence phenomenon. At the same time, the algorithm falls into a local optimal solution [3–5]. At this time, the types of particle swarms are relatively simple, and there is no possibility that they could be out of the local optimal solution. Therefore, the biggest disadvantage of the particle swarm optimization algorithm (PSO) is that it is easier to get into the local optimal

solution. Although expanding the population size can alleviate this phenomenon, it cannot solve such problems fundamentally. Therefore, the research content of this study is to improve the previous algorithm and to improve its working capabilities in the local optimal space.

The SA algorithm can accept the point of the larger energy value in the process of iteration, that is, the poor solution. The algorithm is a global optimization algorithm, and the search range is larger than the Particle swarm algorithm, so it is easier to get the global optimum solution. The main thing is that the simulated annealing algorithm is more likely to jump off the local extremum. The SA algorithm is stress-free and meek to implement and can obtain initial values with good robustness and obtain higher quality solutions [6–7]. The simulated annealing algorithm wants to achieve the purpose of converging to the optimal global solution with a large probability. It must be met as well that the initial temperature is very high, the temperature drops very slowly and the temperature at the end is very low.

However, in practical problems, these conditions are almost impossible to achieve completely, so the solution obtained is not very satisfactory, and sometimes the result is not as good as the result of the intermediate process. Because the convergence rate of the simulated annealing algorithm is slower, if the population size increases or the complexity of the delinquent upsurges, the computation time of the algorithm will be greatly increased [8]. Therefore, the key problem of improving the simulated annealing algorithm is to progress the search adeptness of the algorithm while ensuring the optimal solution of a certain quality.

In summary, in view of the complementary characteristics of the two algorithms, the effective fusion of the two algorithms can improve the performance of the algorithm. It is feasible to transform the disadvantages of the respective algorithms into advantages and form new hybrid algorithms with excellent performance.

3 Optimization of the Flight Velocity V of PSO

Kennedy and Eberhart were the pioneer to propose the particle swarm optimization algorithm. The simple PSO works in the basic principle; and the apiece individual (particle) is considered without volume and mass in the n-dimensional exploration space, and it flies in the search space with firm speed [9]. Moreover, it can adjust its flight speed conferring to its individual flight experience and peer's flight experience [10–11]. In the n-dimensional target search space of a particle swarm composed of m particles, The position of the i-th particle in the n-dimensional space is xi = (xi1,xi2,...,xim). We can get the corresponding fitness value by substituting xi into the corresponding objective function, The particle swarm algorithm measures the pros and cons of the particle's position according to the current fitness value. Assume, f (x) is set as an objective function of fitness.

Xi = (xi1, xi2, xi3…, xin) is set to the contemporary position of particle i.

Vi = (vi1, vi2,…,vin) is set to the contemporary flight speed of particle i.

Pi = (pi1, pi2, …, pin) is set to the finest position.

where particle i has experienced. That is also the position with the best adaptive value of particle i. Thus, it's also called the best position of an individual [12–13].

For this case, the-smaller-the-better approach is utilized. In this approximation, for the values, the enhanced adaptive values are attained [9]. The position of the i-th particle can be determined by the following expression.

If $f(Xi(t+1) \geqq f(Pi(t))$

If $f(Xi(t+1)) < f(Pi(t))$ (1)

$$\mathbf{P}_{\mathbf{i}(\mathbf{t+1})} = \begin{cases} P_i(t) \\ X_i(t+1) \end{cases}$$

Assuming the amount of particles in the group is s, the best global position is set Pg(t), which is called the global superlative position. Then:

$Pg(t) \in P0(t),P1(t),\ldots,Ps(t)\} \mid f(Pg(t))$ (2)

= min {f(P0(t)), f(P1(t)),…, f(Ps(t))

In view of the definition overhead, the evolutionary equation of the basic particle swarm optimization can be design at edas:

$$Vij(t + 1) = vij(t) + c1r1j(t)(pij(t) – xij(t)) + c2r2j(t)(pgi(t) – xij(t)) \quad (3)$$

$$Xij(t + 1) = xij(t)+vij(t+1) \quad (4)$$

where: Subscript j represents the No. j dimension of the particle, i represents the particle i, t represents the No.t generation, c1, c2 are acceleration constants, usually between 0–2, r1～U(0,1), r2～U(0,1） are two independent random functions.

The chief drawback of the particle swarm escalation is that the individual extremum and the group extremum do not change much in the later stage of the algorithm. At the identical time, as the particle correspondence develops higher and higher, the particle velocity will develop Smaller and smaller, in due course constructing record of the particle velocity neighboring to or equivalent to 0, the particle position is not efficient, the algorithm stagnates, and the entire may fall into a local optimum.

Here is the formula for improving flight speed [9]:

$$Vij(t + 1) = vij(t) + c1r1j(t)(pij(t) – xij(t)) + c2r2j(t)(pgi(t) – xij(t)) + r \times cirij \quad (5)$$

$$Xij(t + 1) = xij(t) + vij(t + 1) \quad (6)$$

where r is a small constant in Eq. (5), the other parameters are the same as the basic particle swarm optimization, and the improved algorithm is called velocity particle swarm optimization (PPSO).

4 Particle Swarm Optimization Algorithm

The basic steps of the particle swarm optimization algorithm with improved flight speed are as follows:

STEP l: According to the initialization process, the initial setting of the random position and velocity of the particle swarm is determined by the formulas (1) and (2);

Step 2: Fitness of the particles will be calculated in the second step.

Step 3: Each particle has to be iteratively experienced, and the best position of the history is compared with the current position.

In a condition, if the present state of the particle is better then the corresponding fitness value of the particle is kept as existing the best position.

Step 4: Comparison of fitness values will be performed according to the best position achieve by each particle. In this situation, the maximum the better approach is used to select the current global optimal position

Step 5: Eqs. (5) and (6) are used to regulate the position and velocity of the elements.

Step 6: Customarily an adequately decent adaptive value or reaching a preset concentrated numeral of reiterations can determine whether the end condition is reached, and if not, return to Step 2.

In the initial stage of the algorithm, the algorithm has solid global search ability due to the high particle velocity. In the current situation, the 4th term (r × cirij) in the Eq. (5) is relatively much smaller than the last three terms. Therefore, it has not significant consequence on the ability to pursuit of the algorithm. In the middle and well along stages of the proposed algorithm, when the similarity of the particles is getting higher and higher, the particle speed is getting smaller and smaller, in addition, the 4th term of Eq. (5) also confirms that the exploration speed of the particle does not fall to 0, and the local search can be continued.

5 Principles of the Improved Particle Swarm Simulated Annealing (PSO-SA)

The SAPSO is a new hybrid algorithm, which retain the characteristics of the two algorithms PSO and SA, and this one is a collaborative evolution scheme. On the foundation of the SAPSO algorithm, this paper presents a new intelligent fusion algorithm, namely the particle swarm simulation annealing algorithm (PSO–SA). The algorithm utilizes the search synergy of the SPO and SA algorithms and utilizes the fast-

local search capability from the swarm algorithm, and the global convergence performance of the SA algorithm [14]. Compared with the particle swarm algorithm, the method is easier to jump off the local extremu m point and has a more rapidly convergence rate compared with the SA algorithm. In the process of calculation, the PSO and SA algorithms are iteratively alternated, and the calculated solution is compared to determine whether the calculated solution is optimal. If the SA algorithm preferentially searches for a solution of better quality, then randomly select one of the particles in the particle group, and replace the particle position with the solution, thus increasing the diversity of the particle population; If the PSO algorithm preferentially searches for a better solution [15], the solution is treated as the initial position of the Markov chain at a certain temperature of T of the SA algorithm, which improves the efficiency of the simulated annealing algorithm search.

The PSO-SA algorithm is very useful due to its advanced capabilities to search quickly, and the characteristics of easy implementation and the global convergence of the simulated annealing algorithms to search for problems [16]. In this way, the "premature" phenomenon of the particle swarm optimization can be effectively avoided, and the convergence speed can also be improved.

6 Roulette Rule Introduction

The rudimentary impression of the roulette assortment method is that the specific appropriateness assessment is confidently interrelated with the prospect of being nominated by the roulette. The roulette algorithm can be concise as tracks. If the fitness value of a class A object is known to be A, the fitness value of a class B object is B. the fitness values of C, ...,K objects are C, ...,K, and their probability of the generation is P(A), P(B),...,P(K), respectively, and their probability sum is 1. We need to first divide the length of the 0–1 segment according to the probability size, and randomly generate a number between 0-1. The random number generation result is evenly distributed in 0-1, and then see which probability it falls on section so that an object can be randomly generated in the AK, and its probability of being selected is proportional to its fitness value.

Because this paper is to maximize the problem as the research object, mainly by using the inverse operation to minimize the problem into the maximum problem, but the original roulette strategy needs to be selected and the sequences are positive, so individual fitness value is converted into a positive number, and the order of brightness after the conversion is not changed. The current common practice is to add the fitness values of all individuals and then subtract the individual fitness values for the conversion.

However, the above practices have certain defects, such as widening the gap between different individuals, invisibly magnifying the proportion of superior individuals, narrowing the proportion of disadvantaged individuals, and thus affecting the fairness of choice. Therefore, this paper uses a maximum and minimum summation method to convert, that is, find the maximum and minimum values of all individuals to add, and then subtract the current value for the conversion. Based on this, the roulette strategy can be better embedded in the PSO-SA algorithm under the condition that the original proportion of all individuals is guaranteed.

7 Process of Improved Particle Swarm Simulated Annealing (PSO-SA)

Based on the idea of the co-evolutionary algorithm, this paper proposes a particle swarm simulation annealing algorithm, which is abbreviated as PSO-SA algorithm. The rudimentary phases of the developed algorithm are as tails:

Step 1: Modify the particle swarm algorithm:

Established the group scale N and the maximum time of iteration M;

Initialize the particle swarm's speed v_i and location x_i function of each Particle swarm and the fitness will be calculated and determine the corresponding initial extremum Pi and the group's extreme value Pg based on the fitness of the function;

Step 2: Initialize the simulated annealing algorithm:

Set the initial temperature of T, to generate the initial solution $S(\omega, c_1, c_2)$;

Get the evaluation function $P(i)$. Update the speed v_i and location X_i according to formula (3) and (4). Evaluate the fit function's fitness at the same time. Update Pi and Pg according to the fit function, and take $P(i) = p_g$

Step 3: Generate new $S'(\omega', c_1', c_2')$.

Step 4: Update the speed v_i and location x_i according to formula (1) and (2). For the value of ω, c_1 and c_2, refer to S';

Step 5: Fitness function calculation:

Step 6: Get, $P(i') = \min[fitness\ x(i,:)]$, $\Delta P = P(i') - P(i)$

If $\Delta P < 0$, then $P(i) = P(i')$, $S = S'$.

Meanwhile, accept the speed v_i and location x_i updated by S', $T = T * lamda$;

Else if $\exp(-\Delta P / T) > rand(0,1)$, then $P(i) = P(i')$, $S = S'$, accept the speed v_i and location x_i updated by S', still, $T = T * lamda$;

If the S is at its current status, use only S and rejects the values of S', the values of v_i and location x_i can be updated to determine the fitness values.

Step 7: After getting the fitness values of p_i and p_g the values can be updated:

Step 8: If the final and last condition is satisfactory, the output vales will be the optimal value. If it is not like this, the procedure will start again from the step three.

Based on the above steps, the flow chart of the PSO-SA is as follows.

8 Validation of the Improved PSO-SA Algorithm in Engineering

In this paper, the improved element flock simulated annealing (PSO-SA) algorithm is clearly described. In direction to authenticate the efficiency of the algorithm, this paper applies the improved PSO-SA algorithm to the internationally widely used example, Fibonacci. The prediction was performed on the (Fibonacci) sequence [17].

It can be explained according to following expression.

$S_o = A, S_1 = B,$

$S_{i+1} = S_{i-1} * S_i$

The symbol "*" depicts the connection operation in above equation, e.g.,

In the terms of (S2 = AB), (S3 = BAB, S4 = ABBAB), the two head-to-head expressions make the Si. In the Fibonacci sequence, A represents a hydrophobic residue, which generally appears alone in the arrangement, but B represents a hydrophilic residue, which sometimes appears alone in the series and sometimes appears in pairs [9]. The Fibonacci sequence has a hierarchical expression as follows:

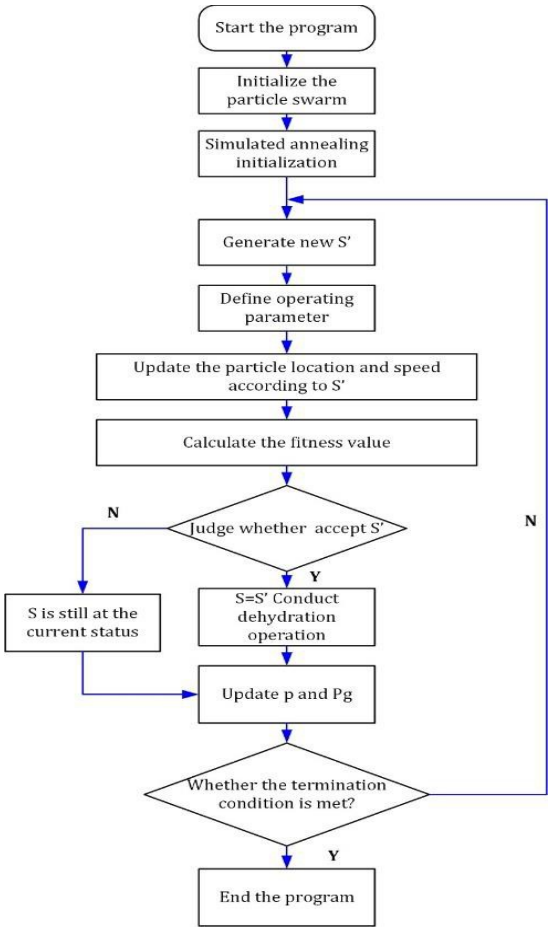

Figure 1: The flow chart of the particle swarm simulated annealing algorithm

$S_i \equiv S_{i-2}*S_{i-3}*S_{i-2} \equiv (S_{i-4}*S_{i-5}*S_{i-4})*(S_{i-5}*S_{i-6}*S_{i-5})*(S_{i-4}*S_{i-5}*S_{i-4}) \equiv \ldots$

In this paper, experiments were performed using typical Fibonacci sequences of lengths 13, 21, 34 and 55. The four artificial protein sequences are shown in the succeeding Tab. 1.

Table 1: Fibonacci sequence $13 \le n \le 55$

A.S.	Len.	Sequence
S13	13	ABBABBABABBAB
S21	21	BABABBABABBABBABABBAB
S34	34	ABBABBABABBABBABABBABABB ABBABABBAB
S55	55	BABABBABABBABBABABBABABB ABBABABBABBABABBABABBABABB ABABBAB

The artificial Fibonacci sequence (Fibonacci) is an artificial protein sequence widely used as an experimental standard [9]. In the Fibonacci sequence, A represents a hydrophobic residue, which generally appears alone in the categorization, whereas B represents a hydrophilic residue that sometimes appears only in the system and sometimes appears in pairs [1,9].

Four Fibonacci sequences of length 13, 21, 34, 55 were tested to test the validity of the algorithm. In our anticipated algorithm termed as PSOSA, the least possible energy of SA algorithm is symbolized by

E_{SA}. Similarly, minutest energy of PSO, and PSOSA are specified by E_S and E_{SSA} correspondingly. In accumulation, all investigational results of are mentioned in Tab. 2.

By comparison, it can be found that the energy value obtained by the particle swarm simulated annealing algorithm is smaller than that obtained by the other two methods, especially for the amino acid sequence with a chain length of 55, and the lowest energy value is significantly reduced. It can also be seen from Tab. 3. that the particle swarm simulated annealing method saves computation time, which indicates that the proposed algorithm is feasible for protein folding prediction.

Table 2: Minimum energy value obtained by different algorithms (2D)

A.S.	Len	ESA	EPSO	EPSOSA
S13	13	−4.9616	−4.7621	−3.2943
S21	21	−11.5238	−10.5578	−6.1978
S34	34	−21.5678	−18.3895	−10.7038
S55	55	−32.8843	−27.6542	−18.6704

Table 3: Calculation time of different algorithms

A.S.	Len(day)	ESA(day)	EPSO(Day)	EPSOSA (Hour)
S13	13	≥2	≥1	≤2
S21	21	≥2	≥1	≤2
S34	34	≥2	≥1	≤4
S55	55	≥2	≥1	≤6

9 Conclusion

In this paper, after studying the parameters of the PSO and the initial temperature of the SA, the problems of the two algorithms in the solution process are discussed. Based on this, an efficient optimization algorithm, particle swarm simulation annealing algorithm (PSO-SA), is proposed. This improved algorithm combines the characteristics of global optimization of PSO and the local optimization of SA. The main findings of this paper are as follows:

(1) The proposed PSO-SA algorithm effectively avoids the shortcomings of PSO, which is stress-free to plunge into local minimum and premature convergence. It also makes full use of the recompenses of SA to jump out of local optimum and global junction. Thence, the PSO-SA algorithm has strong ability to be searched and strong ability to jump off the optimal local elucidation.

(2) For sake of effectiveness of the algorithm to be verified, the PSO-SA is applied to the internationally widely used Fibonacci categorization for prediction. Compared with the operation time obtained by different algorithms, the satisfactory results are obtained.

The PSO-SA algorithm can solve the problem of protein folding prediction well, and the improved algorithm is easier to find the lowest energy value. Moreover, the algorithm can jump out of the local optimal value, can avoid the phenomenon of premature convergence, has better convergence performance, and the experimental results are satisfactory, which proves that the intelligent algorithm after fusion can solve the practical problem better.

Funding Statement: 2018 Annual Heilongjiang Provincial Department of Education Basic Research Business Youth Innovation Talent Project (135309376); 2019 Qiqihar City-level Science and Technology Plan General Project Contract Number (GYGG-201919); 2019 Annual Heilongjiang Provincial Department

of Education Basic Research Business Youth Innovation Talent Project-Intelligent Manufacturing Equipment Innovation Team (135409102).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. N. Tian, S. J. Chen, X. H. Zhang, T. S. Wang and H. Cao, "Sensors deployment optimization in multi-dimensional space based on improved particle swarm optimization algorithm", *Journal of Systems Engineering and Electronics*, vol. 29, no. 5, pp. 969–982, 2018.

[2] H. Luo, "An inertial weight adaptive particle swarm optimization algorithm", *Electronic Technology*, vol. 30, no. 3, pp. 30–36, 2017.

[3] D. Lu, C. M. Tong and W. J. Zhong, "Research on hybrid algorithm based on particle swarm optimization and simulated annealing algorithm", *Computer Engineering and Design*, vol. 32, no. 2, pp. 663–667,2011.

[4] Kennedy, Eberhart, "Particle swarm optimization", in *Proc. of IEEE Int. Conf. on Neural Networks Perth*, no. 2, pp. 1942–1948, 1995.

[5] Q. He, Y. L. Wu and T. W. Xu, "Application of improved genetic simulated annealing algorithm in TSP optimization", *Control and Decision*, vol. 33, no. 2, pp. 219–225, 2018.

[6] L. Wang, J. N. Shen and S. Y. Wang, "Research progress in coevolutionary algorithms", *Control and Decision*, vol. 33, no. 2, pp. 193–202, 2015.

[7] C. Zhang, "The research and application of hybrid swarm intelligence optimization", Ph.D. dissertation, Beijing: University of Science and Technology Beijing, 2017.

[8] C. Huang, D. F. Zhang and Y. W. Si, "Tabu search for the real-word carpooling problem", *Journal of Combinatorial Optimization*, vol. 32, no. 2, pp. 492–512, 2016.

[9] S. G. Wang, F. J. Wang, S. F. Jiang and H. J. Zhang, "Research on particle swarm optimization for protein structure prediction", *Advanced Materials Research*, no. 605, pp. 2497–2501, 2012.

[10] M. Mandloi and V. Bhatia, "A low-complexity hybrid algorithm based on particle swarm and ant colony optimization for large-MIMO detection", *Expert System Application*, no. 50, pp. 66–74, 2016.

[11] B. Zhang, H. Qi and S. C. Sun, "A Novel hybrid ant colony optimization and particle swarm optimization algorithm for inverse problems of coupled radiative and conductive heat transfer", *Thermal Science*, vol. 20, no. 2, pp. 461–472, 2016.

[12] M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia weight for particle swarm optimization", *Applied Soft Computing*, no. 38, pp. 281–295, 2016.

[13] W. Chu, X. Zhang and J. Chen, "Pareto optimal train scheduling for urban rail transit using generalized particle swarm optimization", *Advances in Mechanical Engineering*, vol. 8, no. 10, pp. 1–15, 2016.

[14] H. Zhou and K. Luo, "A K-medoids clustering algorithm based on Particle Swarm Algorithm with Simulated Annealing", *Applied Mechanics and Materials*, no. 325, pp. 1628–1631, 2013.

[15] U. Teschemacher and G. Reinhart, "Enhancing constraint propagation in ACO-based schedulers for solving the job shop scheduling problem," *Procedia Cirp*, no. 41, pp. 443-447, 2016.

[16] H. S.Huang, K. Liu and G. Y. Chu, "Application of improved simulated annealing algorithm in flexible scheduling", *Combined Machine Tool & Automatic Machining Technology*, no. 2, pp. 148–156, 2018.

[17] F. H. Stillinger and T. Head-Gordon, "Collective aspects of protein folding illustrated by a toy mode", *Physical Review*, no. 52, pp. 2872–2877, 1995.