Sensors & Transducers Journal, Vol. 16, Special Issue, November 2012, pp. 116-127

Sensors & Transducers

ISSN 1726-5479
© 2012 by IFSA
http://www.sensorsportal.com

Job-shop Scheduling Problem Based on Particle Swarm Optimization Algorithm

Ying Sun and Hegen Xiong
College of Machinery and Automation,
Box 242, Wuhan University of Science and Technology, Wuhan, 430081, China
Tel.: +86-027-68862283
E-mail: wustsunying@126.com

Received: 11 September 2012 /Accepted: 11 October 2012 /Published: 20 November 2012

Abstract: Production scheduling is a hotspot of manufacturing system and the core of the whole advanced manufacturing system to achieve the development of management technology, optimize technology, automation and computer technology. The research and application of effective scheduling method and optimization technology is the foundation and the key to realize advanced manufacturing and improve production efficiency. And algorithm research is one of the important content of the production scheduling problem. In recent years, various intelligent computation methods have been gradually introduced into the scheduling problem, such as genetic algorithm and simulated annealing algorithm, etc. In view of the standard particle swarm optimization algorithm can not solve the complexity of the production of job-shop scheduling problem. The metropolis sampling criteria is introduced into the PSO algorithm. Other algorithms combined with particle swarm optimization algorithm, three kinds of fusion simulated annealing thoughts of hybrid particle swarm algorithm are constructed respectively. Comparing the results of hybrid PSO with the other algorithms in scheduling the job-shop benchmarking problem, the effectiveness and superiority of the hybrid Particle Swarm algorithm are verified. *Copyright © 2012 IFSA.*

Keywords: Job-shop scheduling, Particle swarm algorithm, Simulated annealing algorithm.

1. Introduction

When standard particle swarm algorithm, which was just effectively applied to solving moderately simple production scheduling problems (such as problems of FT06 and LA01), was utilized to resolve complicated production scheduling problems (such as the problem FT20), it is extraordinarily difficult for the algorithm to converge at the global optimal solution. How to design the PSO algorithm suitable to solve complex the JSP problem becomes the problem which is researched in this paper. Here, the PSO algorithm combined with the simulated annealing algorithm, three kinds of hybrid PSO algorithms are designed so that solution of the complex JSP problem was achieved.

2. Design of PSO Algorithm Combining with Simulated Annealing Concept

A concrete problem optimized, how to make generated candidate solutions pervade the entire solution space is crucial to that the algorithm can effectively converging. The generation mode of initial solution depends on problem nature. This paper adopts the mode of randomly generating initial solution, which makes candidate solutions permeate the overall solution space. Aiming at the specific problem of JSP, the introduced hybrid PSO algorithm still selects the process-based coding mode and discrete-position evolutionary equation. The problem is discussed for the optimization objective of minimizing machining time.

In order to surmount the PSO premature phenomena [1], we can combine PSO with SA. At first, a superior population is obtained by making use of the fast research ability of PSO. And then a part of better individuals are optimized by utilizing the step ability of SA. Simulated annealing manipulation is to integrate the Metropolis sampling criteria into PSO algorithm, combine with the optimized objective function to compare adaptability magnitude between particles and self-particle optimal solution, and accept the solution larger than self-particle optimal solution according to the Metropolis sampling criteria so that it is guaranteed that PSO algorithm could effectively jump out of the local optimal solution. The PSO convergence rate is fast but its accuracy is inferior; SA is of powerful generality and easy to be realized. Nevertheless, its computational time is long and efficiency is lower. Moreover, SA becomes easily stuck in local optimization. Taking complementarities between PSO algorithm and SA algorithm in global search and local search into account, this paper combines with JSP characteristics, considers three kinds of modes which implement the conjunction of PSO and SA, and exerts sufficiently their advantages to construct hybrid discrete PSOSA schedule algorithm.

2.1. PSOSA Hybrid Algorithm-I

Organization of the Text Because population optimal position is adopted in the location update formula, all particles have a tendency to fly into the population optimum position. If the population optimum position is located in local minimal solution, all particles tend to local minimal solution so that it induces search dispersity to become worse and makes global search ability of particles wane. Consequently, in order to boost the ability to avoid algorithm from be stuck in local minimal solution, we try to choose among many a position labeled as to substitute for from the update formula. Therefore, the location update formula changes into:

$$X_i(t+1) = c_i \otimes g(c_i \otimes g(\omega \otimes f_{a,b}(X(t)), pB_i(t)), gB_i^{'}(t)), \tag{1}$$

Apparently, satisfactory performance pB_i should be provided with superior selected probability. Utilizing the SA algorithm mechanism, we consider pB_i which is worse than gB_i as special solution. Accordingly, step probability of pB_i comparing with gB_i is calculated at the temperature t, $e^{-(f_{pB_i} - f_{gB_i})/t}$ namely, where f denotes objective function value. If the step probability is defined as pB_i fitness value, p_i of pB_i replacing gB_i is calculated in accordance with the following equation [2]:

$$p_i = e^{-(f_{pB_i} - f_{gB_i})/t} \Big/ \sum_{i=1}^{N} e^{-(f_{pB_i} - f_{pB_i})/t}, \tag{2}$$

According to the substitution probability above, the roulette strategy is adopted to randomly ascertain which pB_i becomes $gB_i^{'}$, which is good to overcome defects that PSO algorithm convergence rate is too fast and PSO algorithm can not jump out of local optimal solution.

Procedures of the roulette strategy selecting pB_i are depicted as follows:

Step 1: According to substitution probability Pi of n particles we produce next generation $p_i^{'}$. Here, $p_1^{'} = p_1$, $p_i^{'} = p_{i-1}^{'} + p_i$, i=2, 3,…n;

Step 2: Assign i=0;

Step 3: Generate a random number r among $0 \sim p_n^{'}$, r=random(0, 1);

Step 4: Compare r and $p_i^{'}$. If $r < p_1^{'}$, pB_i which p1 corresponds to is selected as $gB_i^{'}$; If $p_{i-1}^{'} < r < p_i^{'}$, which pB_i corresponds to is chosen as $gB_i^{'}$, i=2, 3, … n;

Step 5: If i=n, end; else go to step 6;

Step 6: Assign i=i+1, and go to step 3;

To sum up, the solution flow of PSOSA hybrid algorithm-I is represented as follows:

Step 1: Initialize;

Step 1.1: Generate N initial populations in accordance with the manner of generating initial solution and initialize inertia weight coefficient, acceleration constant, and coefficient of temperature drop;

Step 1.2: Calculate adaptability magnitude of each particle in populations;

Step 1.3: Assign optimal position $pB_i(t)$ of each particle self and its objective value as the current position and its objective value. Assign population optimal position $gB_i(t)$ and its objective value as the optimal position and objective value in all $pB_i(t)$;

Step 1.4: Ascertain initial temperatue.

Step 2: Update particles

Step 2.1: Determine fitness value of each $pB_i(t)$ at the current temperature according to equation (2);

Step 2.2: Adopt the roulette strategy to ascertain N $gB_i^{'}(t)$ among all $pB_i(t)$ and update new position Xi(t) of each particle in accordance with equation (1);

Step 2.3: Calculate fitness value which the new position Xi(t) of each particle corresponds to;

Step 2.4: Update each particle $pB_i(t)$ and its fitness value as well as population $gB_i(t)$ and its fitness value;

Step 2.5: Cooling manipulation.

Step 3: If algorithm end condition is valid, $gB_i(t)$ and its objective value are output; else return to step 2.

Adopting the procedure above to optimize the problem, we implement one selection of $gB_i^{'}$ on each particle. Therefore, different particles maybe utilize different $gB_i^{'}$, which expands search dispersity to a certain extent. Meanwhile, there exists a certain contradiction between global dispersity search and local chemotaxis search. Hybrid algorithm adopts the SA algorithm mechanism to substitute various pB_i for gB_i. Contributing to surmounting premature convergence, it may give rise to the evolutionary process being extended. Difference of selected probability of all pB_i is not obvious because of algorithm controlling temperature and higher temperature during early evolution. Thus, the algorithm emphasizes global dispersity search; as the temperature decreases, selected probability of pB_i with good performance will increase and algorithm will emphasize local search of superior regions. Apparently, the temperature control strategy can automatically exert regulation on algorithm search behavior. The initial temperature t_0 and cooling approach have a certain effect on the algorithm optimal performance. Here, we adopt experience formulae as follows [2]:

$$t_0 = -f_{gB_i} \Big/ \ln(0.2)\,, \tag{3}$$

$$t_{k+1} = \lambda \cdot t_k\,, \tag{4}$$

where f_{eB} is the adaptation value of the optimal particle in the initial population and λ is the cooling rate.

According to above-mentioned analysis of the solution process of PSOSA hybrid algorithm-I, the flow chart of solution of hybrid algorithm is demonstrated as Fig. 1.

2.2. PSOSA Hybrid Algorithm-II

In the standard PSO system, information is a unidirectional flow mode. gB_i transfers information to other particles, and other particles search in the zone close to gB_i. The whole particle population evolves towards the optimal solution under the guide of gB_i. Thus, $\mathrm{gB_i}$ has a significant effect on PSO optimal performance, so inferior search ability of PSO for gB_i is one of main reasons that lead to algorithm prematurity. In order to boost PSO optimal performance, SA sampling process is executed on gB_i after each iterations of the particle population and the attained result is defined as the new gB_i of the PSO system. Thus, application of SA enhances search ability of algorithm for gB_i so that probability of algorithm jumping out of local optimal solution is expanded. The flow chart of solution of PSOSA hybrid algorithm-II is shown as Fig. 2.

2.3. PSOSA Hybrid Algorithm-III

Algorithm above takes implementing sampling on population optimal position into account. Here we consider another combination pattern that sampling process is executed on self optimal position of all particles during the evolutionary process of each generation. Obviously, the population optimal position gets adjusted through enforcing sampling process on self optimal position of all particles. With search proceeding, PSO algorithm inertial coefficient and SA algorithm temperature decrease gradually. And algorithm is gradually located in the region adjacent to optimal solution and carries out fine search. PSO aggregates various particles in the algorithm to gBi and SA ensures population diversity. These optimal manipulations of different effects enrich the solution structure during the optimal process and enhance the whole-space search ability. The flow chart of solution of PSOSA hybrid algorithm-III is shown as Fig. 3.

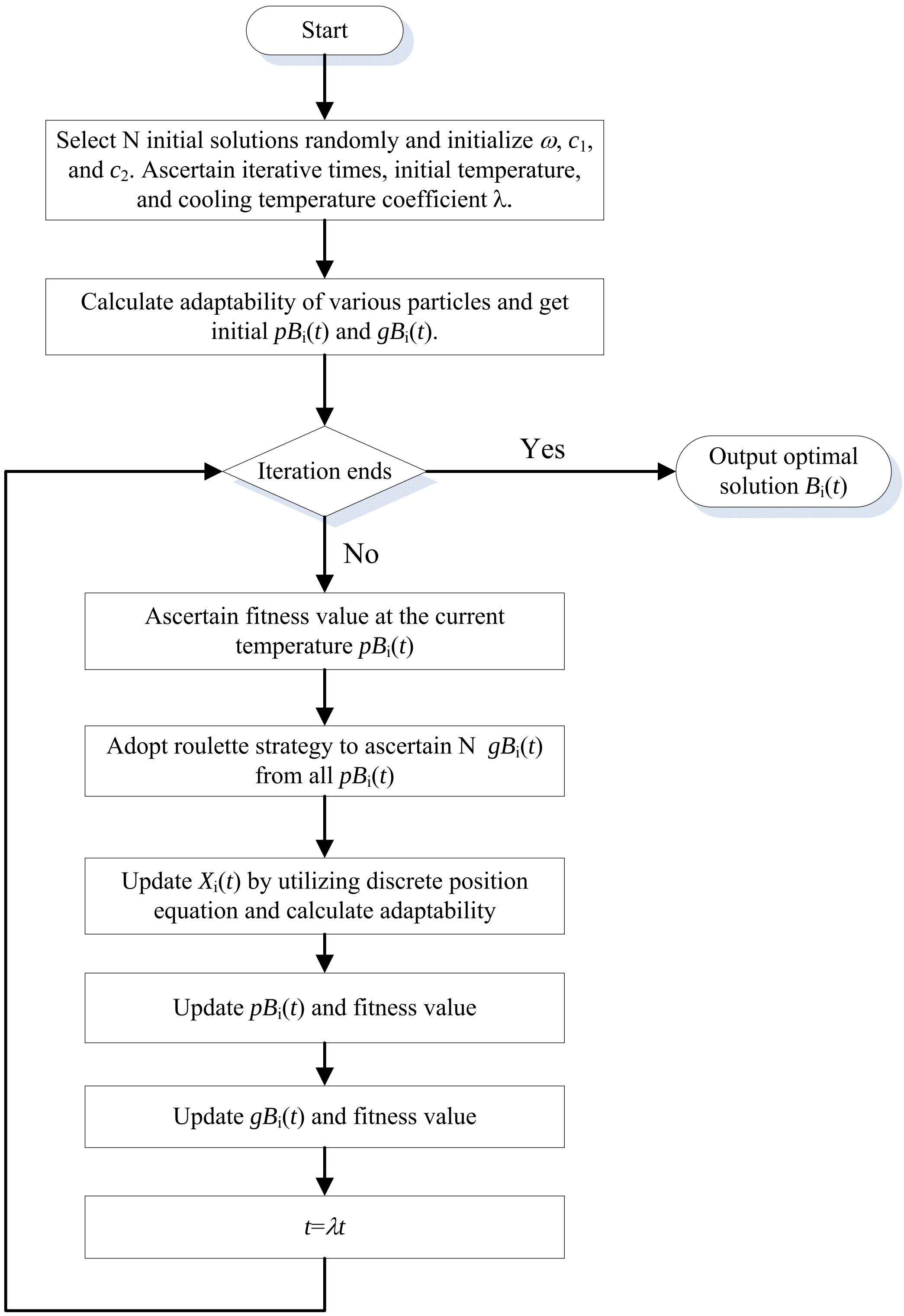

Fig. 1. Solution flow chart of PSOSA-I.

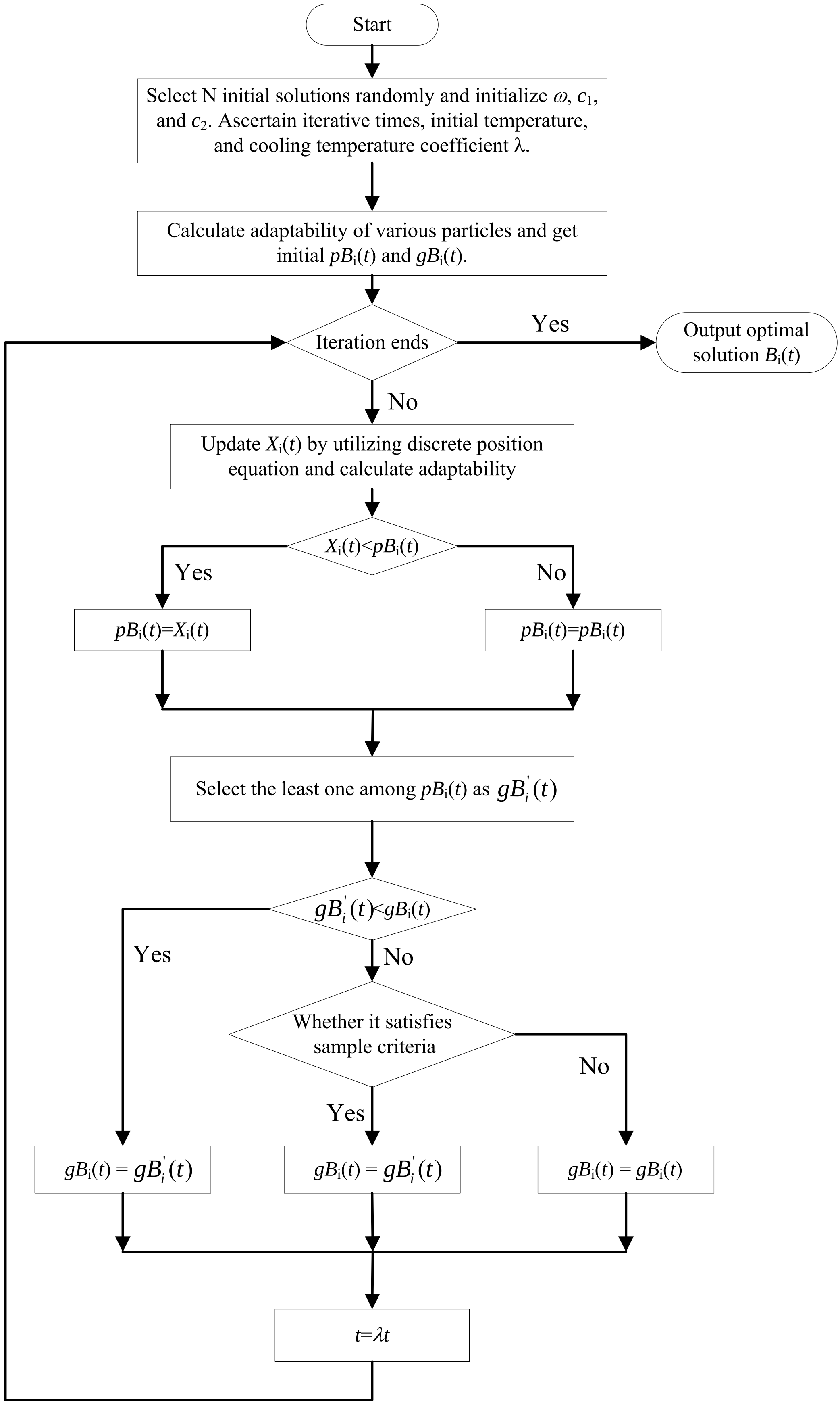

Fig. 2. Solution flow chart of PSOSA-II.

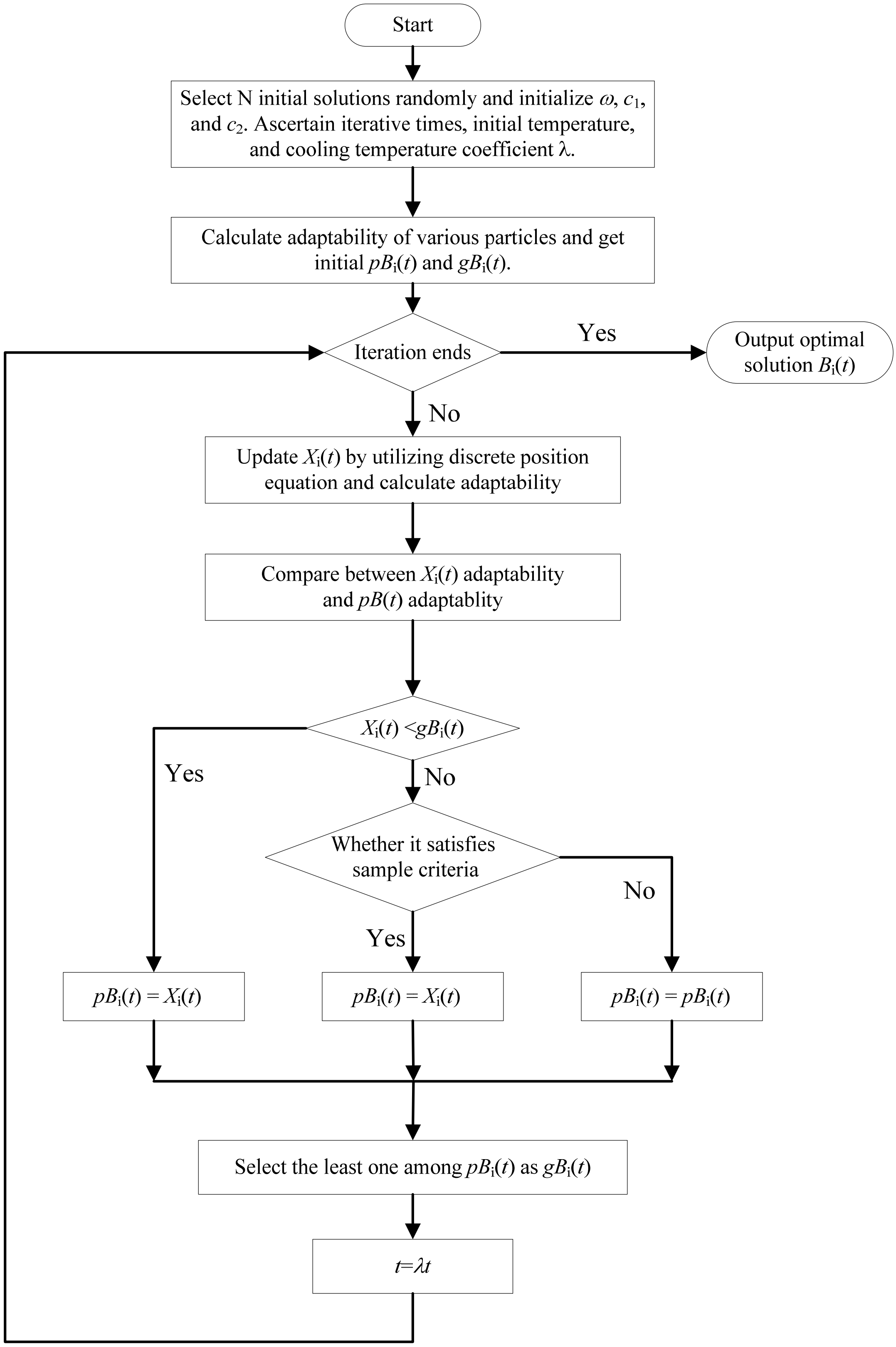

Fig. 3. Solution flow chart of PSOSA-III.

3. Solution Instances of Hybrid PSO Algorithm

Combining discussion of three kinds of hybrid particle swarm algorithm integrating the simulation annealing concept, we achieved three kinds of algorithms above by Java programming. Solving several problems of LA01-LA07 belonging to FT06, FT10, FT20 and LA classes was realized in Jbuilder 2006. Each Benchmark instance was simulated twenty times and the best simulation result was selected as the optimal solution. Moreover, average solutions were attained by calculating them respectively. Deviation (relative error rate) was calculated according to equation (5):

$$Deviation(\%) = \frac{a\lg orithm\ average\ solution\ -\ prescribed\ optimal\ solution}{prescribed\ optimal\ solution} \times 100\% , \tag{5}$$

Parameters of particle swarm algorithm were designed as: inertial weight ω=0.8, inertial weight c1=c2=0.5, annealing coefficient λ=0.9.

Solving Benchmark problems, we independently calculated the population incorporating twenty particles at twenty times as a unit. The calculation results were shown as Table 1, 2, and 3. In tables below, c* denoted prescribed optimal solution, c** indicated the optimal solution introduced by convergence, Ave represents average value of simulating twenty times, % denoted deviation of average value with regard to c*, Number indicated times of prescribed optimal solution introduced from twenty times simulation, $\overline{t}$ represented average convergence time.

Table 1. Solution results of PSOSA-I algorithm.

Problem	c^*	c^{**}	Ave	Number	%	$\overline{t}$ (s)
FT06	55	55	55.5	15/20	0.91	0.253
FT10	930	961	1002	0/20	7.74	8.094
FT20	1165	1223	1258	0/20	7.98	9.109
LA01	666	666	668	18/20	0.30	0.953
LA02	655	655	667	2/20	1.83	0.952
LA03	597	603	615	0/20	3.15	2.358
LA04	590	590	608	0/20	3.05	2.313
LA05	593	593	593	20/20	0	0.972
LA06	926	926	926	20/20	0	1.391
LA07	890	890	907	6/20	1.91	1.390

Table 2. Solution results of PSOSA-II algorithm.

Problem	c^*	c^{**}	Ave	Number	%	$\overline{t}$ (s)
FT06	55	55	55.5	14/20	1.09	0.687
FT10	930	985	1011	0/20	8.82	8.252
FT20	1165	1219	1249	0/20	7.21	9.425
LA01	666	666	674	12/20	1.20	0.955
LA02	655	655	672	1/20	2.59	0.953
LA03	597	597	621	1/20	4.02	2.375
LA04	590	590	612	120	3.73	2.391
LA05	593	593	593	20/20	0	0.969
LA06	926	926	926	20/20	0	1.438
LA07	890	890	902	5/20	1.35	1.453

Table 3. System resulting data of standard experiment.

Problem	c^*	c^{**}	Ave	Number	%	$\bar{t}$ (s)
FT06	55	55	55.5	14/20	0.98	0.719
FT10	930	961	986	0/20	6.02	8.135
FT20	1165	1219	1247	0/20	7.04	9.451
LA01	666	666	672	12/20	0.90	0.950
LA02	655	655	668	4/20	1.98	0.953
LA03	597	597	612	2/20	2.51	2.335
LA04	590	590	610	2/20	3.39	2.375
LA05	593	593	593	20/20	0	1.015
LA06	926	926	926	20/20	0	1.429
LA07	890	890	895	14/20	0.56	1.435

3.1. Analysis of Performance of Hybrid Particle Swarm Algorithm

3.1.1. Comparison between Hybrid PSO Algorithm and Standard PSO Algorithm

From computational results of ten various instances, note that eight instances in PSOSA-II and PSOSA-III got the prescribed optimal solution, seven instances in PSOSA-I achieved the prescribed optimal solution, and the optimal solution of another two instances was also relatively adjacent to the prescribed optimal solution in three kinds of improved hybrid particle swarm algorithms comparing with standard particle swarm algorithm mentioned in chapter three. It was viewed from relative error that the range of deviation of hybrid particle swarm algorithm was regulated within nine percentages and a relatively good effect was achieved. Comparison between relative errors of average solutions produced by four algorithms was shown Fig. 4.

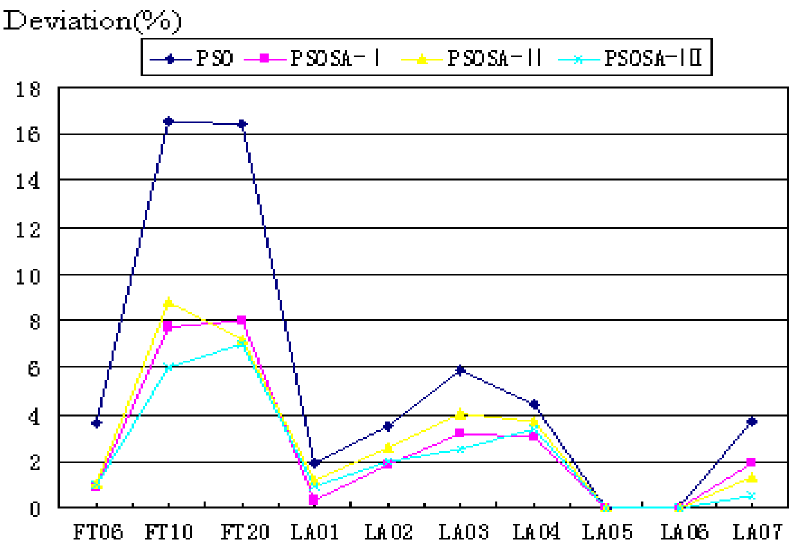

Fig. 4. Deviation comparing standard PSO with three kinds of hybrid algorithms.

From Fig. 4, note that deviation of three kinds of hybrid particle swarm algorithms solving ten types of Benchmark problems above was better than that of standard PSO, which is less than nine percentages, and optimal performance was boosted significantly. Deviation of PSOSA-III was the least and comprehensive performance was the best among three kinds of hybrid PSO algorithms.

3.1.2. Comparison between Hybrid PSO Algorithm and Other PSO Algorithms

Comparison between hybrid PSO algorithm performance and GA algorithm performance in solving FT problems conducted, the statistical result was shown as Table 4.

Table 4. Solution result comparison hybrid PSO algorithm and GA.

Problem	c^*	PSOSA-I			PSOSA-II		
		c^{**}	Ave	%	c^{**}	Ave	%
FT06	55	55	55.5	0.91	55	55.5	1.09
FT10	930	974	1002	7.74	985	1011	8.82
FT20	1165	1223	1258	7.98	1219	1249	7.21
Problem	c^*	**PSOSA-III**			**GA**		
		c^{**}	Ave	%	c^{**}	Ave	%
FT06	55	55	55.5	0.98	55	57	3.27
FT10	930	961	986	6.02	985	1050	12.90

From Table 4, note that instance optimal solution and average value which hybrid PSO algorithm attained were better in comparison with GA algorithm. It was viewed from relative deviation that error was regulated within nine percentages and hybrid PSO was a little better than GA. Thus, the superior result was achieved. In addition, from literature [3], note that time which GA in solving FT problems and LA01-LA07 problems cost was between 68 s and 120 s while time which hybrid PSO algorithm in solving these problems cost was less than 10 s. It could be seen that convergence rate of hybrid PSO algorithm was extraordinarily fast under the circumstances of keeping solving quality.

Comparison between hybrid PSO algorithm in this paper and hybrid PSO algorithm designed by Sha and Hsu [4] conducted, the statistical result was shown as Table 5.

Table 5. Optimal performance compared among various PSO algorithm.

Problem	c^*	Hybrid PSO			PSO-priority[4]
		PSOSA-I	PSOSA-II	PSOSA-III	
FT06	55	55	55	55	55
FT10	930	974	985	961	1007
FT20	1165	1223	1219	1219	1242
LA01	666	666	666	666	666
LA06	1222	926	926	926	926

From Table 5, note that four kinds of hybrid algorithms converged at the optimal solution for simple problems of FT06, LA01 and LA06 whereas hybrid PSO algorithm could not converge at the prescribed optimal solution for FT10 and FT20. Nevertheless, optimal solutions which three kinds of hybrid PSO algorithm designed in this paper attained were better than that of PSO-priority. Thus it could be seen that hybrid PSO algorithm designed in this chapter could achieve a relatively satisfactory effect.

3.1.3. Analysis of Results of LA01 Problem

From Table 1, note that hybrid PSO algorithm realized the effective solution of LA01 problem. The active code which one optimal solution corresponded to was [6 6 8 7 10 4 7 2 1 6 9 2 5 4 1 5 5 9 2 7 9 2

6 10 9 1 4 7 10 9 5 2 3 8 7 8 10 3 5 1 8 4 3 8 6 3 10 4 1 3]. The particle's optical solution attained from active scheduling decoding was 666. According to the optimal solution, the convergence curve (shown in Fig. 5) and the Gantt chart of machining process of various workpieces (shown as Fig. 6) were plotted.

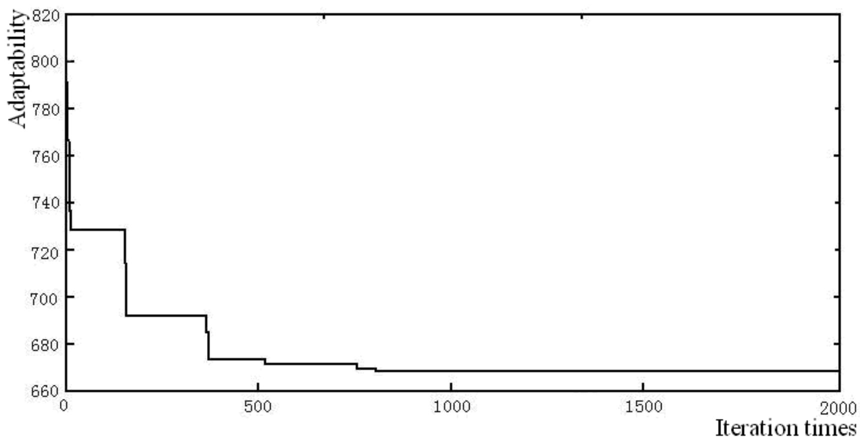

Fig. 5. Convergence curves of LA01 problem by hybrid PSO algorithm.

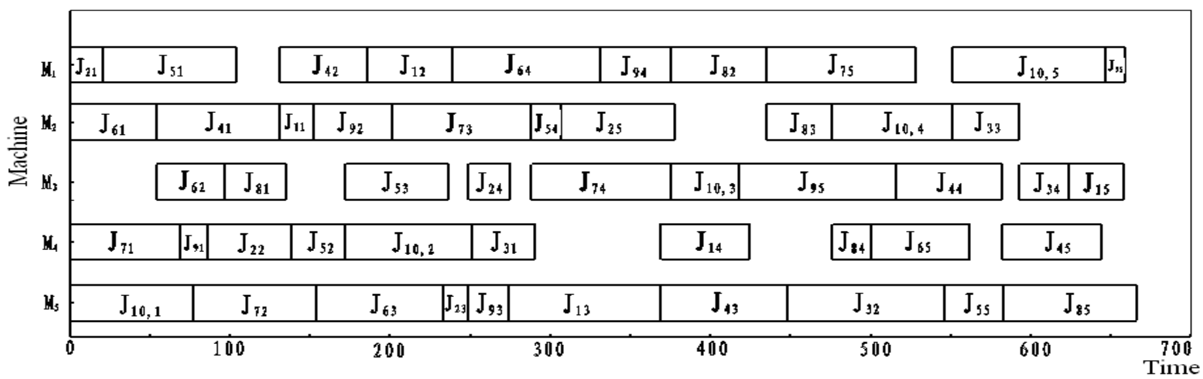

Fig. 6. Optimum Gantt chart of LA01 problem by hybrid PSO algorithm.

For the LA01 problem, twenty times calculations were continuously implemented on the population incorporating twenty particles. Meanwhile, eighteen times calculations could converge at the optimal solution 666, one times calculation converged at the least solution 681, the average convergence generation quantity was 750 generations, the convergence percentage was 18/20, and average time of converging at the optimal solution was 0953s.

4. Conclusions

Aiming at the concrete problem that standard PSO algorithm difficultly solved the complex task schedule in the workshop for the optimal solution, we introduced the Metropolis sampling criteria in simulation annealing algorithm into algorithm, proposed three kinds of hybrid PSO algorithms integrating simulation annealing, and applied the algorithm to solving part of FT problems and LA problems. Hybrid PSO algorithm convergence was validated through analysis of calculation results and comparison with solving results of other algorithms.

References

[1]. Yubo Tian, Renjie Zhu, Zhengqiang Li, Research on Updating Particles of Particle Swarm Optimization Method, *Journal of Jiangsu University of Science and Technology*, Vol. 22, Issue 5, 2008, pp. 67-72.

[2]. Ling Wang, Bo Liu, Particle Swarm Optimization and Scheduling, *Tsinghua University Press*, 2008.

[3]. Xuedong Zhang, Chuanxin Zhao, Yimu Ji. A Hybrid Particle Swarm Optimization and Applications In the Job Shop Problem, *Computer Technology and Development,* Vol. 16, Issue 9, 2006, pp. 109-111.

[4]. Sha D Y, Hsu C Y. A Hybrid Particle Swarm Optimization for Job Shop Scheduling Problem, *Comput. Ind. Eng,* Vol. 51, 2006, pp. 791-808.

2012 Copyright ©, International Frequency Sensor Association (IFSA). All rights reserved.
(http://www.sensorsportal.com)