Fuqing Zhao, Jianxin Tang

Lanzhou University of Technology

A memetic algorithm combined particle swarm optimization with simulated annealing and its application on multiprocessor scheduling problem

***Abstract**. A memetic algorithm, which combines globe search with local search strategies, is presented to deal with the multiprocessor scheduling problem(MSP). During the processes, an improved particle swarm optimization is employed to execute the globe search optimization, and the simulated annealing is adopted to improve the quality of the selected candidates based on a certain strategy. Simulations show that the proposed method performs well on the globe exploration. Experimental results based on MSP show that the algorithm achieved an efficient makespan.*

***Streszczenie.** Przedmiotem artykułu jest algorytm memetyczny do optymalizacji szeregowania zadań w systemie wieloprocesorowym (ang. Multiprocessor Scheduling Problem), łączący w sobie strategie wyszukiwania globalnego i lokalnego. W celu optymalizacji wyszukiwania globalnego zastosowano ulepszoną metodę optymalizacji PSO (ang. Particle Swarm Optimization) oraz algorytm symulowanego wyżarzania (ang. Simulated Annealing) w celu poprawy jakości wybranych elementów. (**Algorytm memetyczny w optymalizacji szeregowania zadań w systemie wieloprocesorowym – optymalizacja PSO i algorytm symulowanego wyżarzania**)*

Keywords: memetic algorithm; particle swarm optimization; simulated annealing; multiprocessor scheduling problem
Słowa kluczowe: Algorytm memetyczny, PSA, symulowane wyżarzanie (SA), optymalizacja szeregowania zadań w systemie wieloprocesorowym

Introduction

Multiprocessor scheduling problem has always been an NP-hard optimization problem. With the development of computer science rapidly and the increasing requirements from modern society, the multiprocessor system has been well applied to computer systems widely. The typical applications involved in the fields of learning control of mobile robots [1], weather forecasting and real time dynamic system simulations [2], etc. Theory proved that multi-processor task scheduling problem is a NP-complete problem [3], and there has not an effective algorithm to deal with this problem in polynomial time at present. The difficulty of MSP depends on the following indicators, such as the topology structure of directed acyclic graph (DAG), which shows the tasks with precedence relations, the topology structure of multi-processor system, the number of processors, uniformity level of process time of tasks and the performance of the schedule system. The way of assigning tasks with precedence relations to the parallel processors so that the system performs well is the key problem to be considered. The scheduling strategy determines the system whether performs satisfactory responsiveness and output-input ratio.

Multiprocessor task scheduling system is a technology much admired in the fields of science computation, manufacture controlling and aerospace industry. With the promotion, many professional experts and researchers have presented vary algorithms for the MSP. Inspired by a simple coevolutionary algorithm based on Bak-Sneppen model, Piotr Switalski and Franciszek Seredynski proposed a generalized external optimization algorithm to cope with MSP problem. And the scheduling experiment proved that this method achieved better results than the genetic algorithm. Mohammad Reza Bonyadi and Mohsen Ebrahimi Moghaddam presented a bipartite genetic algorithm for multi-processor task scheduling problem. The proposed method is a bipartite algorithm in a way that each part is based on different genetic schemes, such as genome presentation and genetic operators. Typical practical engineering applications showed that the PSO performed well in complex optimization problems, and improved PSO algorithms have been proposed and applied to scheduling problems. Based on improved PSO with dynamic varying inertia weight, P. Visalakshi and S. N. Sivanandam [6] solved the dynamic multiprocessor task scheduling problem with the restraint of load balancing. Compared with the basic PSO algorithm, this method performed well in the evolution rate. To deal with the scheduling problem with the restraint of minimizing makespan, zhang chang sheng, etc. [7] proposed a hybrid self adaptive algorithm imported genetic operators to PSO algorithm effectively to solve the flow shop scheduling problem. The hybrid algorithm performed well stability and satisfactory solutions.

However, the evolutionary algorithms simulating nature phenomenon based on swarm search have typical problems of slow convergence and the tendency to local optimal. To deal with these problems, improved evolutionary algorithms have been presented. In the last decades, memetic algorithm [8] has been a hot issue in the evolutionary field. It is a hybrid evolutionary algorithm combined local search strategy and global search. Memes are the units of cultural evolution in contrast to the genes that are the units of biological evolution. In MAs, individuals may undergo memetic evolution or local improvement before transmitting their genes to their offspring. Yew Soon Ong and Andy J. Keane [9] proposed a memetic algorithm based on adaptive local search strategy. The spirit of Lamarckian learning strategy is chosen as local method to locally improve the next chromosome. F. Choong presented two hybrid heuristic algorithms that combine PSO with SA and tabu search, respectively. And the hybrid algorithms were applied on the hybrid flow shop scheduling problem. Ruey Maw Chen, etc. proposed a hybrid algorithm combined PSO and SA, and applied the algorithm on Grid Computing Scheduling Problem, Simulation results show that the grid scheduling problem can be solved efficiently by the proposed method [11]. Typical applications showed that memetic algorithm converges to the optimal solution with a fast convergence rate in combinational, nonstationary and multiobjective optimization problems. Inspirited by ideology of MAs, this paper presents a memetic algorithm combined improved PSO and SA, and apply the MA on the MSP. Simulation results show that this method performed well in searching the globe optimization with a fast speed rate, correspondingly.

Multiprocessor scheduling problem

This paper considers the multiprocessor task scheduling problem with the following scenario. The system consists of M heterogenous processors with different memory and processing resources, which implies that the tasks executed on different processors encounters different execution and communication cost. The tasks in such a problem is usually considered as a directed acyclic graph (DAG) which provides precedence, dependency, and priority tasks together with communication cost among tasks and their precedence. The DAG can be showed by $G=\{V,E\}, V=\{1,2,...,n\}(|V|=n)$ presents the set of all the tasks, and E is the set of the directed edges, which presents the restrictions of tasks in the DAG. That is, $\forall(i,j)\in E$, task T_i will be processed before task T_j. A MSP example with 9 tasks that is shown by DAG is given in Fig.1.

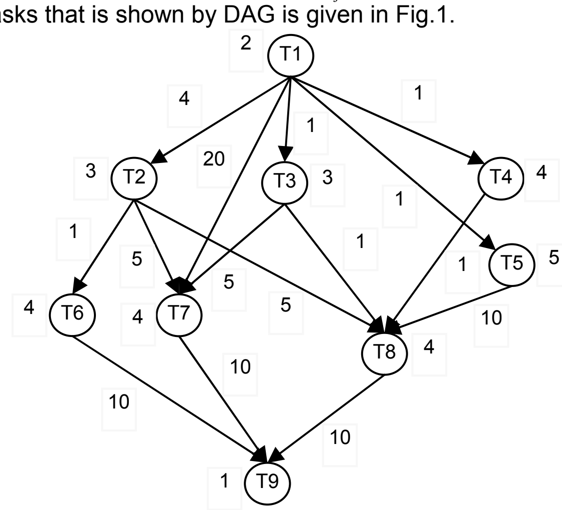

Fig.1. A MSP with 9 tasks shown by DAG

A simplified MSP can be formulated as that the system assigns the n tasks with precedence relations to the m processors using a definite regulation. The objective is to minimize the total execution and communication cost encountered by the task assignment subject to the resource constraints. The function model is given as following.

$$\min\{\max\{x_j y_{ij}\}\} \quad (1)$$

$$x_k - p_k \geq x_j,\ T_j \prec T_k \quad (2)$$

$$t_{\max} - \sum_{j=1}^{n} p_j y_{ij} \geq 0,\ i=1,2,...,m \quad (3)$$

$$\sum_{i=1}^{m} y_{ij} = 1,\ j=1,2,...,n \quad (4)$$

$$y_{ij} = 0 \text{ 或 } 1,\ i=1,2,...,m,\ j=1,2,...,n \quad (5)$$

Where, y_{ij} is set to 1 if task T_j is assigned to processor P_i. Otherwise, y_{ij} is set to 0. x_j denotes the complete time and P_j denotes the processing time of task T_j respectively. t_i demotes the communication cost if task T_j is assigned to processor P_j. In addition, $t_{max}=max\{t_i\}$. $\prec$ denotes the precedence relation, $T_j \prec T_k$ shows that task T_k is assigned to processor ahead of task T_j.

The proposed memetic algorithm

The memetic algorithms are hybrid algorithms combined evolution algorithm operators and local optimizer strategies. Memes, as defined by Dawkins, are the units of cultural evolution in contrast to the genes that are the units of biological evolution. MA is a powerful tool that can be applied in a wide range of optimization problems. The main research contents concentrate on proposing novel co-evolutionary framework, analyzing the theory and mechanism, finding effective local optimizer and implying MA on multi-objective optimization problems.

In MAs, individuals may undergo memetic evolution (or local improvement) before transmitting their genes to their offspring. In the processes, the EA operator executes the globe search, and the LS operator is used to improve the current personal solutions. Typical applications show that PSO testified as a successfully optimization method in a wide variety of research areas, such as artificial neutral networks, optimal design of generator, pattern recognition and image processing. SA is a stochastic optimization that based on Monte Carlo iteration. Improved SA can converges to globe optimization with the probability of 1.

The particle swarm optimization (PSO) algorithm, which is based on the ideology of simulating the bird flocking food and fish schooling behavior, is proposed by Kennedy and Eberhart firstly in 1995 [12]. Like other evolutionary algorithms, PSO is also based on concepts of population and evolution. In the case of PSO, the system initializes a swarm sized of N particles with random positions and velocities in a D dimensional search space initially. In the evolutionary processes, all particles dynamically update the velocities according to the cognitions of personal component and the social component to "search" in the solutions space, and convergence to the globe optimization solution finally. Y.H. Shi [13] introduced inertia weight to eliminate the need for velocity clamping. The following formulas (6) and (7) depict the model of standard PSO.

$$v_{id}^{t+1} = \omega v_{id}^{t} + c_1 rand_1 (p_{id}^{t} - x_{id}^{t}) + c_2 rand_2 (p_{gd}^{t} - x_{id}^{t}) \quad (6)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \quad (7)$$

For $i=1,2,...,N$, $d=1,2,...,D$, where $c_1$and $c_2$are acceleration coefficients, generally $c_1=c_2=2$, $r_1$and $r_2$are random positive numbers, drawn from uniform distribution $U(0,1)$. v_{id}^{t} denotes the velocity in the dth dimension of particle i at time step t, and x_{id}^{t} denotes the position in the dth dimension of particle i at time step t. p_{id}^{t} is the personal best solution of particle i at time step t, p_{gd}^{t} is the best position found by the neighborhood of particle i at time step t.

PSO has attracted the interest of researchers and practitioners around the globe on researching its mechanism and applications for the reasons that its framework is simple and it involves less evolutionary operators, such as selection, crossover or mutation vectors, and it does not require adjusting many parameters. However, like other heuristic algorithm based on biology swarm research, PSO also has the problems of the tendency to trap to local optimization easily and slow convergence. To deal with the problems, an improved PSO algorithm named DDPSO with a linear decreasing disturbance index was proposed in [14]. In the improved PSO, the velocity model can be represented as following:

$$v_{id}^{t+1} = \omega v_{id}^{t} + c_1 rand_1 (p_{id}^{t} - x_{id}^{t}) + c_2 rand_2 (p_{gd}^{t} - x_{id}^{t}) + r_3 l \quad (8)$$

where $l=d_1(t\Delta x-d_2)$, where d_1 and d_2 are small constants settled by dynamically according to optimization problems. Δx is a step size at time step t. r_3 is a random positive number drown from uniform distribution [0, 1].

SA was introduced by Metropolis in 1953. The ideology roots in the annealing processes of solidity. Initial temperature and condition are set firstly, then, the algorithm searches the optimization in the candidates' space with the decreasing temperature. The algorithm terminates if the optimization is fund or the maximum iterations is met. SA is an efficient method to solve complex optimization problems, and it has been widely implied on a range of optimizations. The original SA algorithm framework could be performed as follows.

Initial a solution S*, compute* E
Set the initial $T,k,\ r$
While $E<>0$
S'*= Generate the new solution*
Compute energy E' *and* $\Delta E=E'-E$
if $\Delta E<0$ *then accept* $S=S',E=E'$
else Compute the $\delta = e^{\frac{-\Delta E}{kT}}$
Accept the new solution when random number < δ
decrease the temperature $T=T*r$

Where E is defined as:

$$\{\max(C_{exe}(k)+C_{com}(k))\}+Penalty(k),\ k\in\{1,2,...,N\}$$

In the evolution processes of MA, the optimization dependents completely on the neighborhood function and the initial candidates. For the neighborhood function directs the system how to produce a new population from the old group, therefore, the construction mechanism of neighborhood affects the quality of the optimal solution and the convergence rate of the algorithm. Experimental simulations showed that the way by auditioning disturbance based on the conception of distance to construct neighborhood function is a method used widely by researchers.

However, there are always unavoidable individuals with bad fitness values in the globe searching processes of PSO, and the bad individuals will influence the optimizations. Local search algorithms can improve the bad individuals in a local candidate space, and maintain the improved individuals in an acceptable candidate space, and this is the ideology of MAs. Therefore, MAs can be seen as special searching methods which are done in a local optimization space. Inspired by this ideology, this paper presents a memetic algorithm based on combining PSO with simulated annealing algorithm.

The technological process is given in Fig.2.

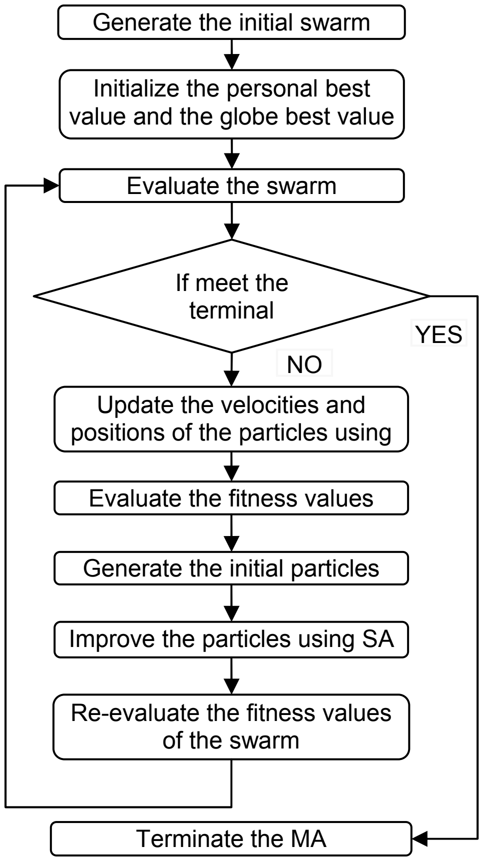

Fig.2. The flow chart of the proposed MA

Typical simulation results and analysis

To evaluate the performance of the MA presented in this paper, four typical benchmark optimization functions are simulated.

Schaffer f7: $f(x,y)=\sum_{i=1}^{n}(x_i^2+y_i^2)^{0.25}\times[\sin(50\times(x_i^2+y_i^2)^{0.1})+1.0]$

Foxhole: $f(x)=418.9829n+\sum_{i=1}^{n}X_i\sin(\sqrt{|x_i|})$

Schaffer f6: $f(x,y)=0.5+\dfrac{(\sin\sqrt{x_i^2+y_i^2})^2-0.5}{(1+0.01(x_i^2+y_i^2)^2)^2}$

Ackley: $f(x,y)=20+e^{-20}e^{-\frac{1}{5}\sqrt{\frac{1}{n}(x^2+y^2)}}-e^{\frac{1}{n}(\cos(2\pi x)+\cos(2\pi y))}$

For the globe searching processes, the system initializes a population sized of 20 in the search space of (-30, 30). The other parameters are set according to that of DDPSO. Fig.3-Fig.6 show the simulation of the four benchmark functions. Fig.7-Fig.10 show the evolutionary processes and the optimal values for the given benchmark functions respectively applied by the MA and a bipartite genetic algorithm (BGA). The compared evolutionary curve show that MA converges to better optimization values than the BGA.

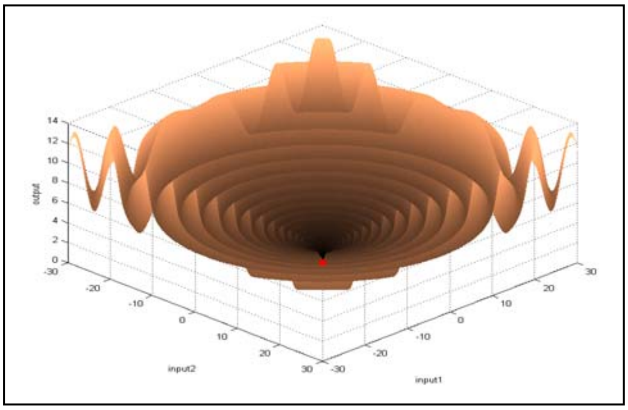

Fig.3. The simulation of Schaffer f7

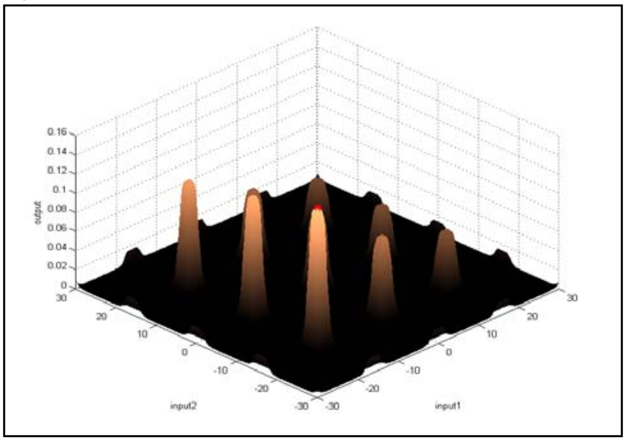

Fig.4. The simulation of Foxhole

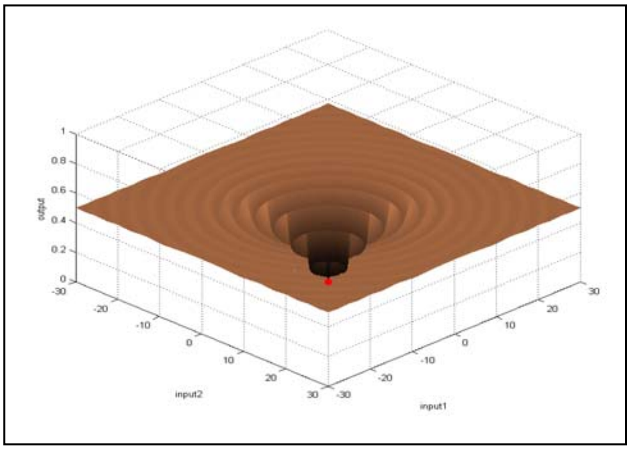

Fig.5. The simulation of Schaffer f6

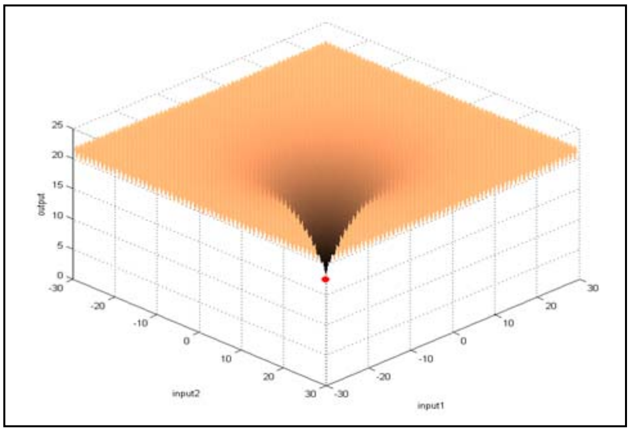

Fig.6. The simulation of Ackley

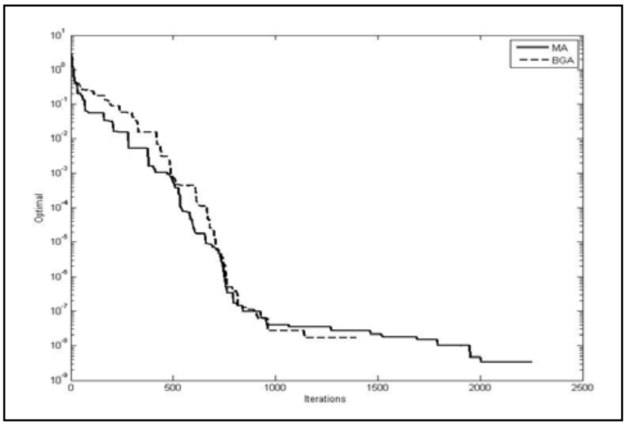

Fig.7. Evolutionary processes of Schaffer f7

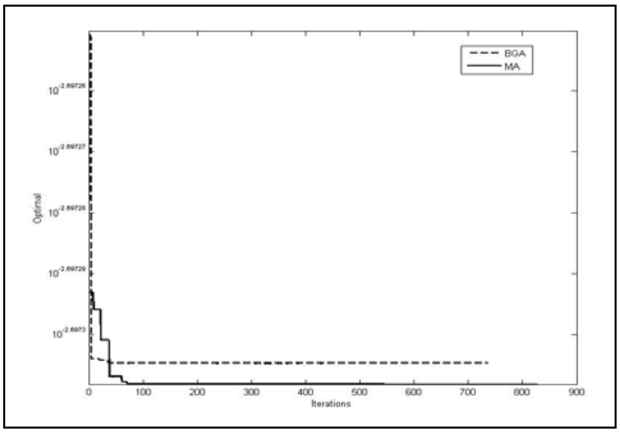

Fig.8. Evolutionary processes of Foxhole

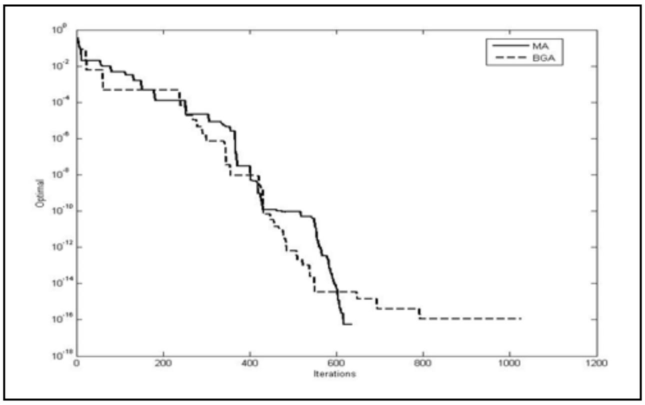

Fig.9. Evolutionary processes of Schaffer f6

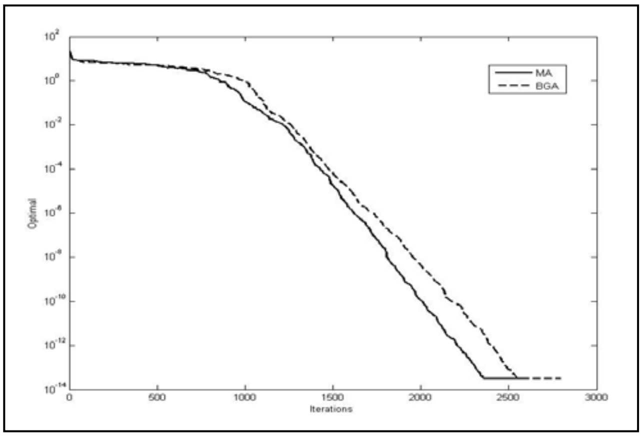

Fig.10. Evolutionary processes of Ackley

In consideration of the effective MA in testing typical Benchmark functions, this paper applied the MA on MSP. PSO has always been applied to single objective and continuous optimization problems, however, the makespan and the cost functions in the MSP are non-increasing discrete multi-objective optimization problems. Therefore, certain strategy must be adopted to map the tasks in the DAG into the parameters of PSO. In this proposed MA, available encoding mode in reference [16] is employed to encode the positions of the particles. During the evolutionary iterations, DDPSO executes the globe search firstly, then the initial bad individuals are constructed, and SA is adopted to execute the local search for the initial individuals to improve the quality of personal solutions. The MA termites when the maximum iteration is met or the acceptable optimization is converged. In the DAG showed in Fig.1, the encoding length of $|V|$ is 9. That is each task node maps a corresponding encoding bit of the particle.

The 9 tasks given in the DAG in Fig.1 are assigned to 2 processors realized by the proposed MA. Fig.11 shows the Gantt chart for this pair array and the compared results applied by MA and some typical proposed algorithms are given in Table 1. Fig.12 shows the iterate curves of the MA solving MSP compared with BGA [17], and the cost results of applying MA and a bipartite genetic algorithm are given in Table 2. Considering the communication cost, the system assigns task *T1* to processor *P2*. For task *T2* executes earlier than task *T3*, the system assigns task *T2* to processor *T2* and task *T3* to processor *P2* after task *T1* is completed. When the 9 tasks are completed, The MA algorithm terminates.

In Fig.12, Cost1 denotes the cost by the MA presented in this paper, and Cost2 denotes the cost by the BGA. Table 2 shows that the initial communication cost by MA is 932.

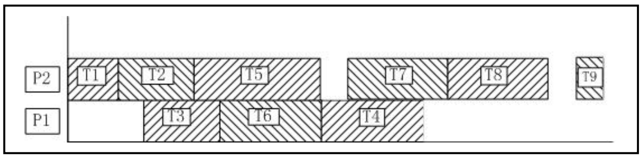

Fig.11. Assignment results of the 9 tasks

Table 1. Compared results by MA and some typical heuristic algorithms on the given problem

Algorithms	MCP	DSC	MD	DCP	BGA	MA
No. of proc.	3	4	2	2	2	2
Best solution	29	27	32	32	21	21

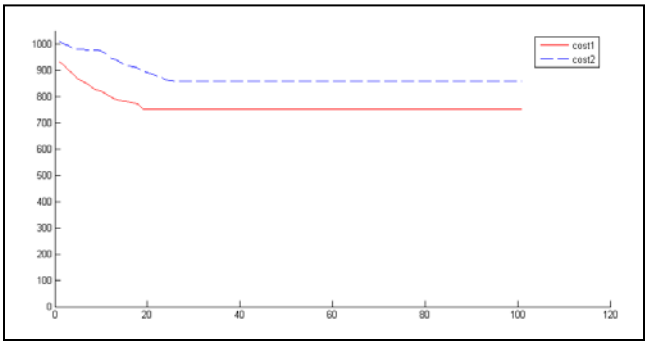

Fig.12. The minimizing cost results by MA compared with BGA

Table 2. The cost results of applying MA and BGA

Algorithm	Completion time	Initial cost	Final cost	Iterations
MA	21	932	751	21
BGA	21	1011	857	28

With the iterations increasing, the swarm converges in a fast rate and stagnates at the 19 iteration, and the communication cost is 751. The initial communication cost by BGA is 1011, and the algorithm stagnated at 28th iteration, and the final cost is 857. The results imply that the MA algorithm converges to globe optimization in a rapid rate profiting from the initializing strategy and the local search ability of SA.

Conclusions

Multiprocessor task scheduling problem has been a complex problem, and there are multi different system targets according to various requirements. Effective algorithms, which affect the assigning mechanism and the scheduling prosperity of the multiprocessor system, have been introduced by researchers to solve the MSP at present. To deal with the problem of minimizing the makespan when the system reasonable assign the tasks represented by a directed acyclic task graph, a novel memetic algorithm combined particle swarm optimization with simulated annealing was presented in this paper. The SA used as local search strategy improved the fitness values of the swarm, which avoid the swarm to strap into local optimization. Simulated results show that the MA performances well in minimizing the makespan problem of MSP.

Acknowledgement

This work was financially supported by the National Natural Science Foundation of China under Grant No.61064011. And it was also supported by China Postdoctoral Science Foundation, Science Foundation for The Excellent Youth Scholars of Lanzhou University of Technology, and Educational Commission of Gansu Province of China under Grant No.20100470088, 1014ZCX017 and 1014ZTC090, respectively.

REFERENCES

[2] Johnson B., Pike G.E., Preparation of Papers for Transactions, *IEEE Trans. Magn*., 50 (2002), No. 5, 133-137

[1] Antonini, P. Ippoliti, G. Longhi, Learning control of mobile robots using a multiprocessor system, *Control Engineering Practice*, 11(2006), No. 14, 1279-1295

[2] Nakata, Toshiyuki Tanabe, Norio Onozuka, etal. MULTIPROCESSOR SYSTEM FOR MODULAR CIRCUIT SIMULATION. *IEEE International Conference on Computer-Aided Design: ICCAD-87 - Digest of Technical Papers*. (1987), Santa Clara, CA, USA: IEEE

[3] Garey M. R., D.S. Johnson, Computers and Intractability, New York: W. H. Freeman and Company, 1979

[4] Piotr Switalski, Franciszek Seredynski. Generalized Extremal Optimization for Solving Multiprocessor Task Scheduling Problem, *Lecture Notes in Computer Science*, (2008), No. 5361, 161-169

[5] Mohammad Reza Bonyadi, Mohsen Ebrahimi Moghaddam. A Bipartite Genetic Algorithm for Multi-processor Task Scheduling. *Int J Parallel Prog*, (2009), No. 37, 462-487

[6] P Visalakshi, S N Sivanandam. Dynamic Task Scheduling with Load Balancing using Hybrid Particle Swarm Optimization, *Int. J. Open Problems Compt. Math.*, 3(2009), No. 2, 475-488

[7] ZHANG Chang-Sheng, SUN Ji-Gui, OUYANG Dan-Tong, ZHANG Yong-Gang, A Self-Adaptive Hybrid Particle Swarm Optimization Algorithm for Flow Shop Scheduling Problem, *CHINESE JOURNAL OFCOMPUTERS*, 11(2009), No. 32, 2137-2146

[8] Pablo Moscato, Carlos Cotta, A gentle introduction to memetic algorithms. (1989), 1-60

[9] Yew Soon Ong, Andy J. Keane, Meta-Lamarckian Learning in Memetic Algorithms, *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, 2(2004), No. 8, 99-110

[10] F. Choong, S. Phon-Amnuaisuk, M. Y. Alias, Metaheuristic methods in hybrid flow shop scheduling problem, *Expert Systems with Applications*, (2011), No. 38, 10787-10793

[11] Ruey-Maw Chen, Der-Fang Shiau, Shih-Tang Lo, Combined Discrete Particle Swarm Optimization and Simulated Annealing for Grid Computing Scheduling Problem, (2009), 242-251

[12] James Kennedy, Russell Eberhart, Particle Swarm Optimization, *IEEE International Conference on Neural Networks*, (1995), No. 4, 1942-1948

[13] Shi Y. H., Eberhart R. C., A modified particle swarm optimizer, *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, (1998), 69-73

[14] Zhao Fuqing, T.Jianxin, Wang Jizhe, Wei Chunmiao, An Improved PSO Algorithm with Decline Disturbance Index, *Journal of Computers*, (2011), No. 6, 691-697

[15] M. Nawaz, E. Enscore, I. Ham, A heuristic algorithm for the machine,n-job flow-shop sequencing problem, *Omega*, (1983), No. 11, 1191-1195

[16] ZHANG Xiao-dong, LI Xiao-ping, WANG Qian, YUAN Ying-chun, Hybrid particle swarm optimization algorithm for cost minimization in service-workflows with due dates, *Journal on Communications*, 8(2008), No. 29, 87-99

[17] S. N. Sivanandam, P. Visalakshi, A. Bhuvaneswari, Multiprocessor Scheduling Using Hybrid Particle Swarm Optimization with Dynamically Varying Inertia, *International Journal of Computer Science & Applications*, 3(2007), No. 4, 95-106

***Authors**: Dr.Fuqing Zhao, School of Computer and Communication,Lanzhou University of Technology. E-mail: Fzhao2000@hotmail.com; dr Jianxin Tang, School of Computer and Communication,Lanzhou University of Technology, E-mail:* jxtang2011@hotmail.com.

The correspondence address is:
e-mail: fzhao2000@hotmail.com, jxtang2011@hotmail.com