Int j simul model 18 (2019) 3, 521-530
ISSN 1726-4529 Original scientific paper

AN IMPROVED WHALE OPTIMIZATION ALGORITHM FOR JOB-SHOP SCHEDULING BASED ON QUANTUM COMPUTING

Zhu, J.[*]; Shao, Z. H.[**,#] & Chen, C.[***]
[*] Fuzhou University of International Studies and Trade, Fuzhou 350108, China
[**] China Digital Fujian IoT Laboratory of Intelligent Production (Minjiang University), Fuzhou 350108, China
[***] Xiamen Hai Hui Yun Tong Software CO., Ltd., Xiamen 361000, China
E-Mail: 603795008@qq.com, 172097792@qq.com, chencheng@changoip.com
([#] Corresponding author)

Abstract
The traditional swarm intelligence algorithms are inefficient and difficult to converge to the optimal solution of the job-shop scheduling problem (JSP). In this paper, an improved whale optimization algorithm (IWOA) is proposed based on quantum computing to solve the discrete JSP. The algorithm was subjected to the analysis on computing complexity, the demonstration of global convergence, and simulation verification on a benchmark example of the JSP. Through the simulation, our algorithm achieved better minimum value, mean value and optimization success rate than traditional swarm intelligence algorithms. The results prove the convergence accuracy and global search ability of the IWOA.
(Received, processed and accepted by the Chinese Representative Office.)

Key Words: Job-Shop Scheduling Problem (JSP), Swarm Intelligence, Quantum Computing, Whale Optimization Algorithm (WOA), Global Convergence

1. INTRODUCTION

Job-shop scheduling problem (JSP) [1-3] is the most classical scheduling problem for discrete manufacturing systems. As one of the most difficult combinatorial optimization problems, the JSP has been proved to be non-deterministic polynomial-time (NP) hard. No algorithm can converge to the optimal solution of the problem in polynomial time.

The JSP attempts to optimize different performance indices through rational scheduling of jobs and processes under varied constraints. The problems can be classified by the research perspective: static JSP [4] and dynamic JSP [5] by processing features, open-shop scheduling problem (OSSP) [6] and flow-shop scheduling problem (FSP) [7] by the composition of job and job-shop, single-machine scheduling problem [8] and parallel machine scheduling problem [9] by scheduling parallelism. The algorithms to solve these scheduling problems have become a research hotspot in the field of the JSP.

Swarm intelligence [10, 11], a new evolutionary computing technology, maintains a special relationship with evolutionary strategy and genetic algorithm (GA). Two important algorithms have been inspired by swarm intelligence theory: ant colony optimization (ACO) [12, 13] and particle swarm optimization (PSO) [14, 15]. Drawing on the foraging behaviour of ants, the ACO has been successfully applied to many discrete optimization problems. The PSO, mimicking simple social systems, was originally used to simulate the food search process of birds, but was later found to be a good optimization tool.

This paper proposes an improved meta-heuristics swarm intelligence algorithm based on quantum computing to solve discrete JSP. The algorithm was subjected to the analysis on computing complexity, the demonstration of global convergence, and simulation verification.

The results prove the superiority of our algorithm over other swarm intelligence algorithms in solving the JSP.

2. LITERATURE REVIEW

Being a branch of computational intelligence, swarm intelligence regards the foraging behaviour and communication mechanism of organisms as the optimization process to adapt to the environment. Below is a brief introduction to typical swarm intelligence algorithms.

The GA [16] is a stochastic adaptive search algorithm based on Darwin's evolutionary theory and Mendelian genetics. The main operations of the algorithm include the genetic selection, crossover and mutation in the evolutionary process of similar species. The ACO [17] imitates how foraging ants search for the shortest path to the food source: the ants communicate with each other with pheromones, and complete the search based on positive feedbacks. The PSO [18] is a simple, fast-converging algorithm based on the predatory behaviour of bird swarm, but it is easy to fall into the local optimum trap and unstable in multi-modal optimization. The artificial fish-swarm algorithm (AFSA) [19] emulates the behaviours of fish swarm, namely, random behaviour, preying, swarming and following, and is controlled by the mobile strategy. The artificial bee colony (ABC) algorithm [20], inspired by bees' honey-collection behaviour, is suitable for multi-modal and high-dimensional optimization, requiring a small population and a few parameters. The hybrid frog leaping algorithm (HFLA) [21] fuses frog foraging features into the PSO and memetic algorithm (MA). The HFLA needs a large population, and must reorder the individuals in each generation. Thus, the computing complexity of the algorithm increases with the number of iterations, making it ineffective in high-dimensional multi-modal optimization. The firefly algorithm (FA) [22] was developed in the light of firefly aggregation and luminescence. The FA bears high resemblances with the AFSA, and faces problems like long distance computing and single-step movement mode.

The whale optimization algorithm (WOA) [23] arises from the hunting behaviour of humpback whale. However, the WOA is easy to fall into the local optimum trap, as well as slow and inaccurate in convergence. Thus, the algorithm has been improved repeatedly to enhance the convergence speed and accuracy. For example, Aljarah et al. [24] presented an adaptive WOA based on the original WOA. Zhong and Long [25] proposed an efficient WOA with randomly adjustable control parameters. Liu and Li [26] developed a sine-cosine chaotic two-chord WOA. These improved WOAs have been applied to continuous optimization problems, rather than discrete JSP.

The ABC algorithm boasts fast speed and high efficiency in solving the JSP. Li and Ma [27] created a Pareto-based hybrid ABC algorithm for the JSP. Thammano and Phu-ang [28] designed a hybrid ABC algorithm with local search to solve the JSP. Banharnsakun et al. [29] put forward an ABC algorithm based on the best-so-far principle for the JSP. Wang et al. [30] generated the initial solution using several strategies and used Pareto to store and record the non-dominant solution.

3. PRELIMINARIES

3.1 Mathematical description of the JSP

The JSP is generally described as the optimization of certain performance indices through rational arrangement of the processing sequence of n jobs on m machines, under the given processing technique, machine sequence and processing time of each job. The JSP must satisfy the following hypotheses:

Hypothesis 1: The machine never breaks down, and the job is processed from zero time.

Hypothesis 2: Each machine can only process one job at a time, and each job cannot be processed on more than one machine at the same time.

Hypothesis 3: The operations of a job must be completed within the processing time. Once started, the operation sequence cannot be interrupted. Pre-emptive processing is not allowed between the operations.

Hypothesis 4: Each job has a fixed operation sequence, i.e., each machine is responsible for a specific operation of the job, and each operation takes a fixed amount of time. There is no constraint on the sequence between different jobs.

Mathematically, the JSP can be described by linear programming model, integer programming model and disjunctive graph model. In this paper, the JSP is described by integer programming model as:

$$\min \quad \max_{\substack{1\le j\le m \\ 1\le i\le n}}\{c_{ij}\} \tag{1}$$

s.t.

$$c_{ij} - t_{ij} + M(1 - a_{ihj}) \geq c_{ih} \tag{2}$$

$$c_{kj} - c_{ij} + M(1 - x_{ikj}) \geq t_{kj} \tag{3}$$

$$c_{ij} \geq 0 \tag{4}$$

$$a_{ihj}, x_{ikj} = 0, 1 \tag{5}$$

$$i, k = 1, 2, \dots, n \tag{6}$$

$$j, h = 1, 2, \dots, m \tag{7}$$

where, t_{kj} and c_{ih} are the processing time and makespan of job i on machine k, respectively; $a_{ihj} = 1$ means machine h precedes machine j in processing job i; $a_{ihj} = 0$ means machine j precedes machine h in processing job i; $x_{ikj} = 1$ means job i is processed on machine j prior to job k; $x_{ikj} = 0$ means job i is processed on machine k prior to job j.

Eq. (1) specifies the objective of the JSP: minimizing the maximum makespan. Eqs. (2) and (3) are the constraints of the problem, namely, the operation sequence and the non-blocking requirement on the operations.

3.2 The WOA

The WOA includes three operators to simulate the search for prey, encircling prey, and bubble-net foraging behaviour of humpback whales. During bubble-net foraging, each whale chooses between spiral updating and shrinking encircling based on the value of random probability. The spiral updating can be modelled as:

$$Y(t+1) = D' \cdot e^{al} \cdot cos(2\pi l) + Y^*(t) \tag{8}$$

$$D' = |Y^*(t) - Y(t)| \tag{9}$$

where, t is the current number of iterations; vector $Y(t)$ and vector $Y(t+1)$ are the current position and the iteratively updated position of an individual whale, respectively; vector $Y^*(t)$ is best foraging position among the whales; a is a constant about the shape of logarithmic helix; l is a random number within [-1, 1]. Together, a and l control the spiral updating of the whales. If $l = -1$, then the whales are the closest to the prey; If $l = 1$, then the whales are the farthest from the prey.

Let p be the probability of an individual whale to choose spiral updating. Then, the bubble-net foraging behaviour can be described as:

$$Y(t+1) = \begin{cases} X^*(t) - A \cdot D & p < p^* \\ D' \cdot e^{al} \cdot \cos(2\pi l) + Y^*(t) & p \geq p^* \end{cases} \tag{10}$$

If $p < p^*$, the individual whale updates its position and moves to a better position, to achieve shrinking encircling of the prey.

The shrinking encircling can be modelled as:

$$D = |C \cdot Y^*(t) - Y(t)| \tag{11}$$

$$Y(t+1) = Y^*(t) - A \cdot D \tag{12}$$

$$A = 2\alpha \cdot r_1 - \alpha \tag{13}$$

$$C = 2r_2 \tag{14}$$

$$\alpha = 2 - \frac{2t}{T_{max}} \tag{15}$$

where, T_{max} is the maximum number of iterations; α decreases linearly from 2 to 0 with the increase in the number of iterations; r_1 and r_2 are random vectors within [0, 1]; coefficient vectors C and A control the moving pattern of individual whale. If $|A|$<1, then individual whale $Y(t)$ is close to the current optimal position $Y^*(t)$ of the population, and will move to that position; otherwise, the individual whale will start to search for prey.

The search for prey can be modelled as:

$$Y(t+1) = Y_{rand} - A \cdot D_{rand} \tag{16}$$

$$D_{rand} = |C \cdot Y_{rand} - Y(t)| \tag{17}$$

where, Y_{rand} is another random individual whale in the population. Eqs. (16) and (17) show that, if $|A|$ is not less than 1, the individual whale will move randomly towards another whale.

4. IMPROVED WOA (IWOA) BASED ON QUANTUM COMPUTING

4.1 Quantum computing and quantum optimization

In quantum computing, the information is stored in basic units called quantum bits. Compared with classical bit, a quantum bit can be stored continuously in a random manner, allowing the two polarized states to superpose in any form. The quantum bits can be divided into single quantum bit, double quantum bit and multi-quantum bit. The superposition state of a single quantum bit can be expressed as:

$$|\varphi| > \alpha \succ + \beta \prec \tag{18}$$

$$|\alpha|^2 + |\beta|^2 = 1 \tag{19}$$

where, $\succ$ and $\prec$ are the Dirac notations of the two basic states of microparticles in quantum computing, respectively; α and β are plurals representing the probability amplitude of quantum states.

The quantum methods that logically transform quantum states are called quantum logic gates, which are the basis of quantum computing. The main quantum gates include Hadamard gate, Pauli-X gate, Pauli-Y gate, Pauli-Z gate, phase gate, $\pi/8$ gate and quantum rotation gate (QRG).

The QRG can be described as:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{20}$$

The updating process of its quantum state can be depicted as:

$$\begin{bmatrix} \alpha' \\ \beta' \end{bmatrix} = R(\theta) \begin{bmatrix} \alpha \\ \beta \end{bmatrix} \tag{21}$$

where, θ is the rotation angle.

4.2 The IWOA

The basic WOA is poor in convergence accuracy and easy to fall into the local optimum trap. To overcome the defects, this algorithm was improved by the theory of quantum computing and quantum optimization. Specifically, the QRG operation was introduced and the position updating mechanism was modified, aiming to enhance population diversity and convergence accuracy. Let $Y(t)$ and $QY(t)$ be the position of an individual whale in the search space and that after QRG operation, respectively. Since $\beta = \sqrt{1-\alpha^2}$ (Eq. (19)), the updating process of QRG can be simplified as:

$$\alpha' = \left|\alpha \cdot \cos(\theta) - \sqrt{1-\alpha^2} \cdot \sin\theta\right| \tag{22}$$

The operation mode of QRG is called single-chain coded QRG operation. Under this operation, the position of the individual whale can be defined as:

$$QY(t) = \left|Y(t) \cdot \cos(\theta) - \sqrt{1-Y(t)^2} \cdot \sin\theta\right| \tag{23}$$

After a whale has been updated by a QRG, the updated solution is compared with the pre-update solution, and the better one should be selected.

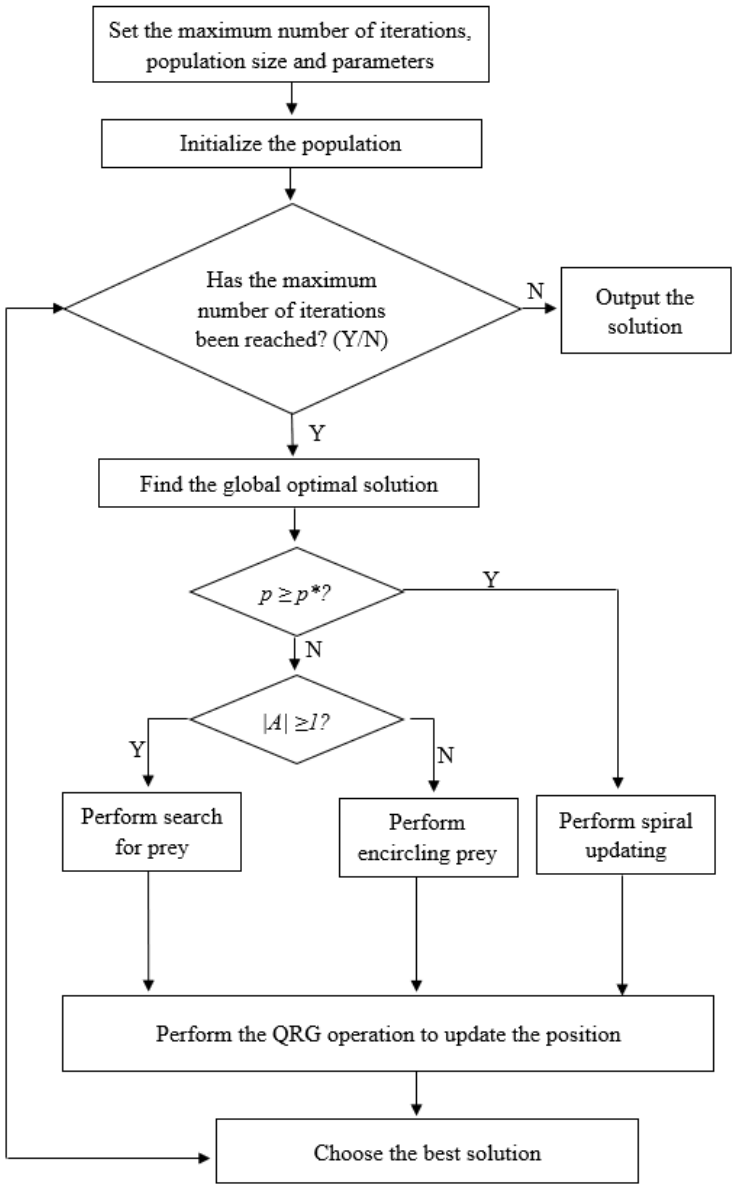

Figure 1: The workflow of the IWOA based on quantum computing.

As shown in Fig. 1, the IWOA based on quantum computing can be implemented in the following steps:

Step 1: Set the maximum number of iterations, population size and parameters.

Step 2: Randomly initialize the population.

Step 3: Calculate the fitness of individual whales, and find the current global optimal solution.

Step 4: Update the position of individual whales in each dimension based on the values of p and $|A|$.

Step 5: Perform the QRG operation for variables in each dimension of whale individuals.

Step 6: Evaluate the updating effects of Steps 4 and 5, and retain the better solution.

Step 7: Repeat Steps 3 to 6 until reaching the maximum number of iterations.

Step 8: Output the solution.

4.3 Demonstration of algorithm convergence

According to the criterion and theorem on global convergence, an algorithm capable of converging to the global optimal solution must satisfy the following two assumptions:

Assumption 1. If $f(D(z,\xi)) \le f(z)$ and $\xi \in S$, then $f(D(z,\xi)) \le f(\xi)$, where f is the objective function of the minimization problem, D is an operator or function that guarantees a better and newest solution, ξ is the vector generated by the algorithm in the sample space, and z is a point in subset *S* of the solution space that minimizes the function value or produces an acceptable lower bound of the function value on *S*. This assumption ensures that the optimization algorithm operates correctly, such that its solution sequence converges to the lower bound of function f on *S*.

Assumption 2. For any Borel subset A in *S*, if its probability measure $V[A] > 0$, then $\prod_{k=0}^{\infty}(1-\mu_k[A]) = 0$, where $\mu_k[A]$ is a probability measure of A generated by distribution μ_k. This assumption ensures that, if the measure is greater than 0, the algorithm will not miss any Borel subset *A* in solution space *S* after infinite iterations.

The global convergence of the IWOA can be proved based on the two assumptions.

Lemma 1. The IWOA satisfies Assumption 1.

Function *D* of the IWOA can be defined as:

$$D\left(G_t, Y_{i,t}\right) = \begin{cases} G_t & f\left(g\left(Y_{i,t}\right)\right) \ge f(G_t) \\ g\left(Y_{i,t}\right) & f\left(g\left(Y_{i,t}\right)\right) < f(G_t) \end{cases} \tag{24}$$

where, D is a function that computes and selects of global optimal positions for the whale population; g is a continuous function corresponding to the QRG operation function; $g\left(Y_{i,t}\right)$ is the position of individual whale i after the t^{th} iteration; G_t is the current global optimal position.

The definition of our algorithm shows that the fitness corresponding to G_t is monotonic and nonincreasing, and gradually converges to the lower bound of the solution space. Thus, Lemma 1 is proved.

Lemma 2. The IWOA satisfies Hypothesis 2.

$$g\left(Y_{i,j,t}\right) = \left|Y_{i,j,t} \times \cos\theta - \sqrt{1 - Y_{i,j,t}} \times \sin\theta\right| \tag{25}$$

where, $Y_{i,j,t}$ is the position of the *j*-dimensional variable of the i^{th} whale individual after the t^{th} iteration; $g\left(Y_{i,j,t}\right)$ and $Y_{i,j,t}$ fall within [0, 1].

The following equation can be derived from Eq. (25):

$$\theta = \cos^{-1}\left(\pm g\left(Y_{i,j,t}\right)\right) - \sigma \tag{26}$$

where, σ falls within [0, $\pi/2$]; $Y_{i,j,t}$ is a point in the j-dimension; $g(Y_{i,j,t})$ is any point different from $Y_{i,j,t}$ in the j^{th} dimension. Eq. (26) is valid when σ falls between $-\pi/2$ and $3\pi/2$). The probability for $Y_{i,j,t}$ to reach $g(Y_{i,j,t})$ is greater than zero, if the quantum rotation angle falls within the range of θ.

Let $\mu_{i,t}$ be the uniform distribution of n-dimensional variables corresponding to individual whales. For any Borel subset A in S, if $V[A] > 0$, then

$$0 < \mu_{i,t}[A] < 1 \tag{27}$$

Then, the probability measure of A generated by μ_t can be expressed as:

$$\mu_t[A] = 1 - \prod_{i=1}^{S}(1 - \mu_{i,t}[A]) \tag{28}$$

Eqs. (27) and (28) show that

$$\prod_{t=0}^{\infty}(1 - \mu_t[A]) = 0 \tag{29}$$

Thus, Lemma 2 is proved.

To sum up, the IWOA is a global convergence algorithm that satisfies both assumptions.

5. SIMULATION AND RESULTS ANALYSIS

To verify its performance, our algorithm was applied to solve the benchmark example of the JSP, and compared with other swarm intelligence algorithms like the grey wolf optimizer (GWO) and the cuckoo search (CS).

The parameters were set as follows: the number of jobs, n; the number of machines, m; the maximum number of iterations, 200. The results of the three algorithms on the benchmark example are compared in Table I below. Each sample consists of three rows, namely, the maximum makespan, the maximum machine load and the total machine load. Obviously, the IWOA outperformed the contrastive algorithms in all samples.

Table I: Comparison between the results of the three algorithms on the benchmark example.

Problem scale	**IWOA**		**GWO**		**CS**	
	Minimum value	Mean value	Minimum value	Mean value	Minimum value	Mean value
4×5	11	12.7	13	14.2	12	14.8
	10	13.6	12	16.1	12	16.3
	32	33.7	35	37.2	36	38.7
8×8	12	15.8	13	17.3	13	17.6
	14	16.2	15	18.2	15	18.4
	77	79.4	80	82.7	79	83.7
10×7	13	16.1	13	17.5	14	17.9
	14	16.8	15	18.9	16	19.2
	83	85.7	84	87.6	83	88.1
10×10	14	16.7	14	17.7	14	18.6
	16	17.1	17	17.9	16	18.2
	85	89.2	88	90.2	87	91.5
15×10	16	17.2	17	18.3	18	19.1
	18	19.8	18	20.3	19	21.2
	90	93.8	95	97.5	94	97.9
15×15	17	17.6	18	18.9	18	19.4
	20	22.1	21	23.6	21	23.9
	92	95.7	95	99.7	95	100.2

Fig. 2 is the Gantt diagram of the optimal solution obtained by the IWOA for sample 10×7.

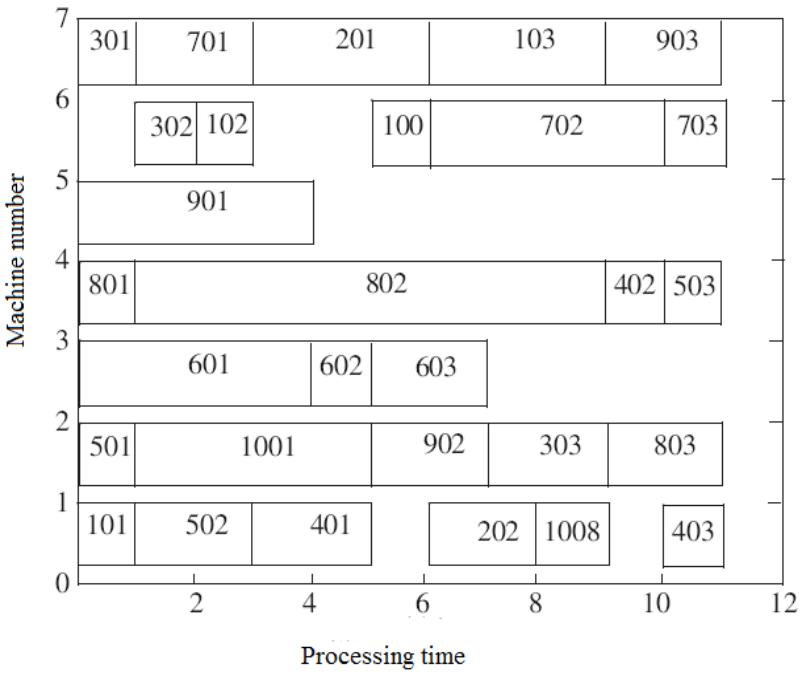

Figure 2: The optimal solution obtained by the IWOA for sample 10×7.

In Fig. 2, each square represents an operation. For each square, the vertical coordinate is the serial number of the operation machine, and the left and right boundaries are the start time and end time of the operation, respectively. As shown in Fig. 2, the scheduling results basically converged to the best-known solution.

For sample 10×10, all three algorithms reached the optimal solution.

Fig. 3 is the Gantt diagram of the optimal solution obtained by the IWOA for sample 15×10, the most complex case in the benchmark example.

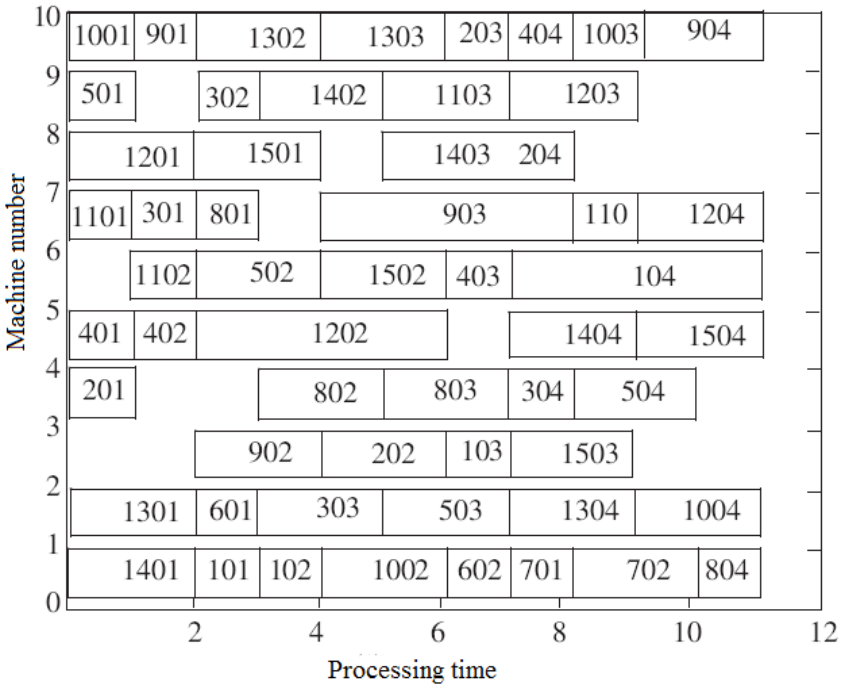

Figure 3: The optimal solution obtained by the IWOA for sample 15×10.

It can be seen that our algorithm still converged to the optimal solution, despite the high complexity of the sample.

6. CONCLUSIONS

The WOA is a new swarm intelligence algorithm. But the algorithm has many defects in solving discrete optimization problems like the JSP. To solve the defects, this paper puts forward the IWOA based on quantum computing. The algorithm was subjected to the analysis on computing complexity, the demonstration of global convergence, and simulation verification. The results prove our algorithm outperformed the other swarm intelligence algorithms in the number of iterations, convergence accuracy and global search ability in solving the JSP. The future research will further improve the application performance of the IWOA.

ACKNOWLEDGEMENT

This paper is supported by Natural Science Foundation of Fujian Province under grants (Grant numbers 2016H6019, 2016J01267), in part by Scientific and Technological Projects of Fuzhou City (Grant number 2019-G-49), in part by scientific research project of Xiamen City (Grant number 3502Z20179033), in part by the Scientific Research Items of MJU (Grant number MJY18003).

REFERENCES

[1] Kong, L.; Li, T.; Wang, K.; Zhu, H.; Takano, M.; Yu, B. (2015). An improved shuffled frog-leaping algorithm for flexible job shop scheduling problem, *Algorithms*, Vol. 8, No. 1, 19-31, doi:10.3390/a8010019

[2] Lei, D.; Zheng, Y.; Guo, X. (2017). A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption, *International Journal of Production Research*, Vol. 55, No. 11, 3126-3140, doi:10.1080/00207543.2016.1262082

[3] Li, J.; Pan, Q.; Xie, S. (2012). An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems, *Applied Mathematics and Computation*, Vol. 218, No. 18, 9353-9371, doi:10.1016/j.amc.2012.03.018

[4] Qiu, X.; Lau, H. Y. K. (2014). An AIS-based hybrid algorithm for static job shop scheduling problem, *Journal of Intelligent Manufacturing*, Vol. 25, No. 3, 489-503, doi:10.1007/s10845-012-0701-2

[5] Gromicho, J. A. S.; van Hoorn, J. J.; Saldanha-da-Gama, F.; Timmer, G. T. (2012). Solving the job-shop scheduling problem optimally by dynamic programming, *Computers & Operations Research*, Vol. 39, No. 12, 2968-2977, doi:10.1016/j.cor.2012.02.024

[6] Panahi, H.; Tavakkoli-Moghaddam, R. (2011). Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization, *Expert Systems with Applications*, Vol. 38, No. 3, 2817-2822, doi:10.1016/j.eswa.2010.08.073

[7] Pan, Q.-K.; Tasgetiren, M. F.; Suganthan, P. M.; Chua, T. J. (2011). A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, *Information Sciences*, Vol. 181, No. 12, 2455-2468, doi:10.1016/j.ins.2009.12.025

[8] Chand, S.; Schneeberger, H. (1986). A note on the single-machine scheduling problem with minimum weighted completion time and maximum allowable tardiness, *Naval Research Logistics Quarterly*, Vol. 33, No. 3, 551-557, doi:10.1002/nav.3800330319

[9] Vallada, E.; Rubén, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times, *European Journal of Operational Research*, Vol. 211, No. 3, 612-622, doi:10.1016/j.ejor.2011.01.011

[10] Sun, G.; Bin, S. (2018). A new opinion leader detecting algorithm in multi-relationship online social networks, *Multimedia Tools and Applications*, Vol. 77, No. 4, 4295-4307, doi:10.1007/s11042-017-4766-y

[11] Pandi, V. R.; Panigrahi, B. K. (2011). Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm, *Expert Systems with Applications*, Vol. 39, No. 7, 8509-8514, doi:10.1016/j.eswa.2011.01.050

[12] Yang, Z.; Yu, B.; Cheng, C. (2006). A parallel ant colony algorithm for bus network optimization, *Computer-Aided Civil & Infrastructure Engineering*, Vol. 22, No. 1, 44-55, doi:10.1111/j.1467-8667.2006.00469.x
[13] Quan, F. (2018). Design of robot ant colony algorithm to reduce transport risks of dangerous chemicals, *Revue d'Intelligence Artificielle*, Vol. 32, No. S1, 57-66, doi:10.3166/ria.32.s1.57-66
[14] Sun, G.; Bin, S. (2017). Router-level internet topology evolution model based on multi-subnet composited complex network model, *Journal of Internet Technology*, Vol. 18, No. 6, 1275-1283, doi:10.6138/JIT.2017.18.6.20140617
[15] Huang, C.; Zhou, X.; Hou, D. (2018). Online no-wait scheduling of leather workshop supply chain based on particle swarm optimization, *Journal Européen des Systèmes Automatisés*, Vol. 51, No. 1-3, 153-167, doi:10.3166/jesa.51.153-167
[16] Maulik, U.; Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique, *Pattern Recognition*, Vol. 33, No. 9, 1455-1465, doi:10.1016/S0031-3203(99)00137-5
[17] Deng, L. L. (2013). Based on the ant colony algorithm TSP optimization simulation, *Applied Mechanics and Materials*, Vol. 275-277, 2501-2505, doi:10.4028/www.scientific.net/AMM.275-277.2501
[18] Arora, S.; Singh, S. (2013). The firefly optimization algorithm: convergence analysis and parameter selection, *International Journal of Computer Applications*, Vol. 69, No. 3, 48-52, doi:10.5120/11826-7528
[19] Gan, X.-S.; Tang, X.-Q.; Gao, H.-L. (2015). Research on nonlinear approximation model of radial basis function neural network trained using artificial fish swarm algorithm with adaptive adjustment, *Applied Mechanics and Materials*, Vol. 713-715, 1855-1858, doi:10.4028/www.scientific.net/AMM.713-715.1855
[20] Zhu, G.; Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation*, Vol. 217, No. 7, 3166-3173, doi:10.1016/j.amc.2010.08.049
[21] Xu, Y.; Wang, L.; Wang, S.; Liu, M. (2013). An effective shuffled frog-leaping algorithm for solving the hybrid flow-shop scheduling problem with identical parallel machines, *Engineering Optimization*, Vol. 45, No. 12, 1409-1430, doi:10.1080/0305215X.2012.737784
[22] Kora, P.; Krishna, K. S. R. (2016). Hybrid firefly and particle swarm optimization algorithm for the detection of bundle branch block, *International Journal of the Cardiovascular Academy*, Vol. 2, No. 1, 44-48, doi:10.1016/j.ijcac.2015.12.001
[23] Mirjalili, S.; Lewis, A. (2016). The whale optimization algorithm, *Advances in Engineering Software*, Vol. 95, 51-67, doi:10.1016/j.advengsoft.2016.01.008
[24] Aljarah, I.; Faris, H.; Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm, *Soft Computing*, Vol. 22, No. 1, 1-15, doi:10.1007/s00500-016-2442-1
[25] Zhong, M. H.; Long, W. (2017). Whale optimization algorithm based on stochastic adjustment control parameter, *Science Technology and Engineering*, Vol. 17, No. 12, 68-73
[26] Liu, Z.; Li, S. (2018). Whale optimization algorithm based on chaotic sine cosine operator, *Computer Engineering and Applications*, Vol. 54, No. 7, 159-163, doi:10.3778/j.issn.1002-8331.1610-0395
[27] Li, X.; Ma, S. (2017). Multiobjective discrete artificial bee colony algorithm for multiobjective permutation flow shop scheduling problem with sequence dependent setup times, *IEEE Transactions on Engineering Management*, Vol. 64, No. 2, 149-165, doi:10.1109/TEM.2016.2645790
[28] Thammano, A.; Phu-ang, A. (2013). A hybrid artificial bee colony algorithm with local search for flexible job-shop scheduling problem, *Procedia Computer Science*, Vol. 20, 96-101, doi:10.1016/j.procs.2013.09.245
[29] Banharnsakun, A.; Achalakul, T.; Sirinaovakul, B. (2011). The best-so-far selection in artificial bee colony algorithm, *Applied Soft Computing*, Vol. 11, No. 2, 2888-2901, doi:10.1016/j.asoc.2010.11.025
[30] Wang, L.; Zhou, G.; Xu, Y.; Liu, M. (2012). An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling, *International Journal of Advanced Manufacturing Technology*, Vol. 60, No. 9-12, 1111-1123, doi:10.1007/s00170-011-3665-z