International Journal of Grid and Distributed Computing
Vol. 9, No. 12 (2016), pp.333-340
http://dx.doi.org/10.14257/ijgdc.2016.9.12.29

An Improved Artificial Bee Colony Algorithm for Order Acceptance and Scheduling in Flow Shop with Hybrid Blocking

Chunhua Jin

Postgraduate, University of Science and Technology Beijing
Beijing Information & Science Technology University
E-mail bimjch@126.com

Abstract

For order acceptance and scheduling with hybrid blocking in flow shop, the improved artificial bee colony (IABC) with NEH_BV heuristic algorithm, local search and artificial bee colony (ABC) is proposed for resolving the problem. In this algorithm, we first use NEH_BV heuristic algorithm to get some initial solutions, then updating the above initial solutions with improved artificial bee colony. In updating process, employed bees, onlooker bees and scout bees are inserted heuristic will increase exploitation and exploration of the algorithm. At last, we will get the near optimal solution by IABC algorithm. The effectiveness and efficiencies of the IABC algorithm has been proved by the case's analysis and been compared with artificial bee colony and harmony search (HS).

***Keywords**: Order Acceptance; Scheduling; Hybrid Blocking; IABC; Harmony Search*

1. Introduction

The limited storage ability of factory in process of production leads to blocking problem of production that brings the difficulty for scheduling and reduces production efficiency. The blocking problem is not infinite buffer capacity between two adjacent processing node. If the order that is completed in a processing node and next node is not released processing capacity will stay the node refuses the request of next order to be processed. Commonly, blocking problem includes RSb(Release when Starting Blocking) and RCb (Release when Completing Blocking). RSb is a processing node is blocked until the order into next node is available. RCb is only available on a processing node until the order is completed on the next processing node.

In general, blocking of flow shop scheduling is different from flowshop scheduling, most of the blocking problems is NP hard[1]. About typical constraints(RSb) study, Caraffa proves that the flow shop blocking problem in processing environment is more than two machines is strongly NP hard problem and designs genetic algorithm to solve large scale flow shop blocking problem[2]. Grabowski uses tabu search algorithm to solve the permutation flow shop problem with blocking constraints, add local search and the dynamic tabu list to emergency search to improve the efficiency of algorithm[3]. Wang design the hybrid discrete differential evolution algorithm to solve flow shop scheduling problem with blocking constraints and compare to other algorithms to get better results[4].

The strategy of order acceptance and order completion time set have important influences on company performance, Arredondo and Martinez show order acceptance policy to maximize company income per unit cost while the capacity of company production is greater than customers' demand[5]. Assuming order rejected can affect future income, Lewis extends the above model to multi period production[6]. Xiao et al research order acceptance based on permutation flow shop problem and design meta heuristic and simulated annealing algorithm to solve the problem to obtain approximate optimal solution[7].

ISSN: 2005-4262 IJGDC

Blocking will affect the completion time of order with multi-node production environment, how to select order and arrange production scheduling will affect profits of company. The order acceptance is constructed as a nonlinear programming model based on flow shop with hybrid blocking and solved by heuristic and intelligent algorithm. Research on the problem of order acceptance and scheduling is helpful to perfect production theory and provide the theoretical basis for company's actual production.

2. Problem Description

Order acceptance and scheduling with multi-node production environment is expressed as: it has n processing nodes and one order can only be completed on one node at a time; all orders arrive at the same time, each node can only process an order at a time; all orders have the same processing sequence and each order must be processed on all nodes; determining the optimal processing sequence to reduce completion time of order. The objective function is to maximize total returns of all orders, i.e. total revenue minus order delivery delay penalty cost, idle time penalty cost and blocking time penalty cost. Generally, it has infinite storage or RSb blocking with multi-node production, but the research on RCb blocking problem is less.

Many complex processes of multi-node production lines can been lead to blocking because of it is influenced of the process, technology, types of equipment and different production conditions. Different blockings will cause costs of idle time and blocking time, and can influence completion time and returns of order. Therefore, research on order acceptance and scheduling of different blockings in multi-node production environment has practical significance.

3. Model Formulation

3.1 Order acceptance and scheduling of hybrid blockings

The sequence of order is processed can pass m nodes in order acceptance and scheduling problem in multi-node production with hybrid blocking. Compared to the general order acceptance and scheduling in multi-node production, the final completion time varies with blockings. The processing route of order is determined to calculate returns of order, if the net returns is negative so the order is refused to accept.

Hypothesis of order acceptance and scheduling in multi-node production with hybrid blocking: (1) orders are mutually independent and orders are processed at time zero; (2) transfer time, preparation time, installation time includes in the processing time between adjacent nodes; (3) order processing time is known and fixed, and return of order is known; (4) when the order is processing at one node, it is not allowed to interrupt; (5) there are a variety of blocking and unlimited buffers between any two nodes; (6) nodes are available and are not maintenance and breakdown.

3.2 Mathematical model

(1) notation and variables.

$i \in I = \{1, 2, \ldots, n\}$ -Number of order; $j \in J = \{1, 2, \ldots, m\}$ -number of node; $k \in K = \{1, 2, \ldots, k\}$ -number of position of order in processing sequence Q; $P_{i,j}$: the processing time of i order in j node, $i = 1, 2, \ldots, n, j = 1, 2, \ldots m$; $B_{b,j}$: binary variable, equate1when blocking between node j and node $j+1$, else 0; α_i: unit penalty coefficient of order tardiness; β: unit penalty coefficient of idle time and blocking time of order; D_i: delivery time of order, $d_i \geq 0$, $i = 1, 2, \ldots, n$; R_i: incomes of completed order i, $R_i \geq 0$; $S_{k,j}$: start time of position k of order in node j; $C_{k,j}$: completion time of position k of order in node j;

$Q_{i,k}$: binary variable; X_i : binary variable.

(2) model.

According to the above descriptions and symbols, and definition of variable, the model is formulated as follows:

$$\max f = \sum_{k=1}^{n}\sum_{i=1}^{n} Q_{i,k} X_i \left\{ \begin{array}{l} R_i - \alpha_i \max\left\{C_{k,m} - D_i, 0\right\} \\ -\beta(C_{k,m} - \sum_{j=1}^{m} P_{i,j}) \end{array} \right\} \tag{1}$$

Subject to

$$S_{k,j} \geq S_{k,j-1} + \sum_{i=1}^{n} P_{i,j-1} Q_{i,k} \tag{2}$$

$$,k \in \{1,2,\ldots,n\}\ j \in \{1,2,\ldots,m\}$$

$$S_{k,j} \geq C_{k-1,j} B_{1,j} + S_{k-1,j+1} B_{2,j} \tag{3}$$

$$+C_{k-1,j+1} B_{3,j}, k \in \{2,\ldots,n\}\ j \in \{1,2,\ldots,m-1\}$$

$$S_{k,m} \geq C_{k-1,m}, k \in \{2,\ldots,n\} \tag{4}$$

$$C_{k,j} \geq S_{k,j} + \sum_{i=1}^{n} P_{i,j} Q_{i,k} \tag{5}$$

$$,k \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,m\}$$

$$C_{k,j} \leq S_{k,j+1}, k \in \{1,2,\ldots,n\} \tag{6}$$

$$, j \in \{1,2,\ldots,m-1\}$$

$$\sum_{i=1}^{n} Q_{i,k}, k \in \{1,2,\ldots,n\} \tag{7}$$

$$\sum_{k=1}^{n} Q_{i,k}, i \in \{1,2,\ldots,n\} \tag{8}$$

$$Q_{i,k} \in \{0,1\}, i \in \{1,2,\ldots,n\}, k \in \{1,2,\ldots,n\} \tag{9}$$

$$X_i \in \{0,1\}, i \in \{1,2,\ldots,n\} \tag{10}$$

$$S_{k,j} \geq 0, k \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,m\} \tag{11}$$

$$C_{k,j} \geq 0, k \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,m\} \tag{12}$$

Eq. (1) is objective function that is used to maximize return of order; Eq. (2) is expressed that start time of order is more than or equal sum of start time and completion time of order in last node while order is processing in adjacent nodes; Eq.(3) specifies different blockings exists in adjacent nodes, if there is RSb between node j and $j+1$, so Eq.(3) presents $S_{k,j} \geq S_{k-1,j+1}$; Eq.(4) is a special of Eq.(3) and ensures start time of order in the last node is more than completion time; In Eq.(5), completion time is greater than or equal sum of start time and processing time; Eq.(6) guarantees that the completion time of order is less than start time of next order when order is processed in adjacent nodes; Eq.(7) presents each position of processing sequence has an order; Eq.(8) ensures an order can only be put a position in processing sequence; Eq.(9) and (10) are defines

$Q_{i,k}$ and X_i as binary variable; Eq.(11) and (12) are the non-negativity constraint, which ensures that the above variables are nonnegative.

4. Improved Artificial Bee Colony Algorithm

The artificial bee colony algorithm (ABC) is an optimization algorithm based on the intelligent foraging behavior of honey bee swarm, proposed by Karaboga[17]. In the ABC model, the colony consists of three groups of bees: employed bees, onlookers and scouts. It is assumed that there is only one artificial employed bee for each food source.

4.1 *Initialization*

Each order has amounts of profits and completion time in multi-node production. If NEH algorithm is chose to solve this problem, so some orders that gain large profits can be excluded to lose profits but order can be effectively completed. Therefore, Total processing and profits are simultaneously considered for order acceptance and scheduling, the processing priority of order is determined by NEH_BV algorithm.

4.2 Employed Bee

The new food source of employed bee is generated based on formula (1) in artificial bee algorithm. The new food source is compared to current food source and the better food source is selected.

$$v_{i,j} = x_{i,j} + \lambda_{i,j}(x_{i,j} - x_{k,j}) \qquad (1)$$

Therein, $i \in \{1,2,\ldots,SN\}, k \in \{1,2,\ldots,SN\}, j \in \{1,2,\ldots,D\}$, and randomly generated.

4.3 Onlooker Bee

When search is completed by all employed bees, they will share the information of food source and quantity of honey through dancing and observation, i.e. order sequence and fitness is passed to onlooker bee to determine. Onlooker bee selects processing queue based on fitness. The processing queue that is generated by roulette algorithm in stage of onlooker has greater fitness to be selected.

4.4 Scout Bee

While all employed bees and onlookers finish searching, the standard is set by the algorithm will determine whether fitness of some food sources is not improved to abandon. A counter is called "limit" is set to judge food source in a iteration. When the "limit" is reach, the food source is abandoned to let scout bee to find new food source. Scout bee randomly search new food source based on original food sources, if new food source is found to abandon original food source.

4.5 Local Search

At the stage of onlooker, p is set at [0,1] to judge if local search is selected to insert neighborhood. Random r is randomly generated at [0,1]. If r is less than p, local search is selected , else next step to calculation.

5. Simulation

5.1 Experimental Design

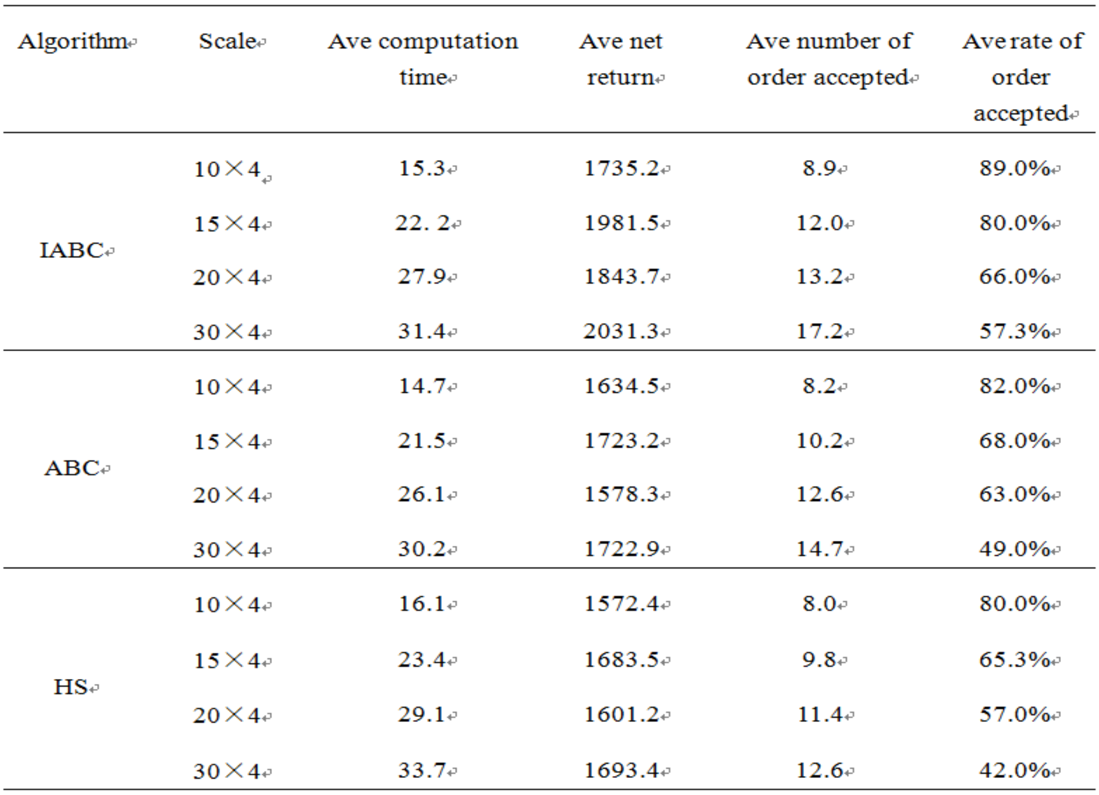
Simulations were with C++ performed on an Pentium4 2.8 GHz processor with 2.24 GB of RAM running on Windows 7. The parameters for ABC, IABC and HS algorithms are kept fixed. For artificial bee colony algorithm, initial population size is 50, employed bee number is 25, onlooker bee number is 25, scout bee number is 1, limit value is 25×D(processing sequence vector dimension), Maxiter is 100. For improved artificial bee colony algorithm, initial population size is 50, employed bee number is 25, onlooker bee number is 25, scout bee number is 1, limit value is 25×D(processing sequence vector dimension), P =0.1, w=0.5, Maxiter is 100. For harmony search algorithm, initial population size is 50, HMCR=0.99, PAR=1.0, PAR_{min}=-1.0, PAR_{max}=1.0, bw=0.01, Maxiter is 100.

Experimental data is set as follows: the number of orders n =10, 15, 20, 30, return of order evenly distribute within [200,2000] and is generated randomly, processing time of order in each node evenly distribute within [1,5] and is generated randomly, delivery time of order evenly distribute within [10,30] and is generated randomly, unit penalty coefficient of storage time and blocking time of node is 1, unit penalty coefficient of delayed completion time is 3. Assuming there are four stages for multi-node production environment.

5.2 Analysis of Experimental Results

The average value of index is calculated by 20 examples of each group. The results is shown in table 1:

Table 1. Experiment Results of Order Acceptance and Scheduling with Hybrid Blocking in Multi-node Production



Algorithm	Scale	Ave computation time	Ave net return	Ave number of order accepted	Ave rate of order accepted
IABC	10×4	15.3	1735.2	8.9	89.0%
	15×4	22. 2	1981.5	12.0	80.0%
	20×4	27.9	1843.7	13.2	66.0%
	30×4	31.4	2031.3	17.2	57.3%
ABC	10×4	14.7	1634.5	8.2	82.0%
	15×4	21.5	1723.2	10.2	68.0%
	20×4	26.1	1578.3	12.6	63.0%
	30×4	30.2	1722.9	14.7	49.0%
HS	10×4	16.1	1572.4	8.0	80.0%
	15×4	23.4	1683.5	9.8	65.3%
	20×4	29.1	1601.2	11.4	57.0%
	30×4	33.7	1693.4	12.6	42.0%

The conclusions of the above experiment data is obtained as follows:

(1) As shown from table 2, the feasible solution can got by IABC, ABC and HS algorithm at short time and the calculation time increases by amounts of order are increased.

(2) The number of order are 8.9, 12.0, 13.2, 17.2 and the average net return of order is 1735.2, 1981.5, 1843.7, 2031.3 in IABC algorithm. The average rate of order increases by amounts of order are increased, the average net return of order increases a value and then decreases. Therefore, if the production capacity is limited and there are bottlenecks about ability of order accepted, more number of order accepted is not more the net return of order.

(3) Compared to data of IABC, ABC and HS algorithm, the result can be obtained that average time of IABC is better than HS, but is little better than ABC. Average return rate of order and number of order accepted is the best by IABC, ABC followed and is worst by HS.

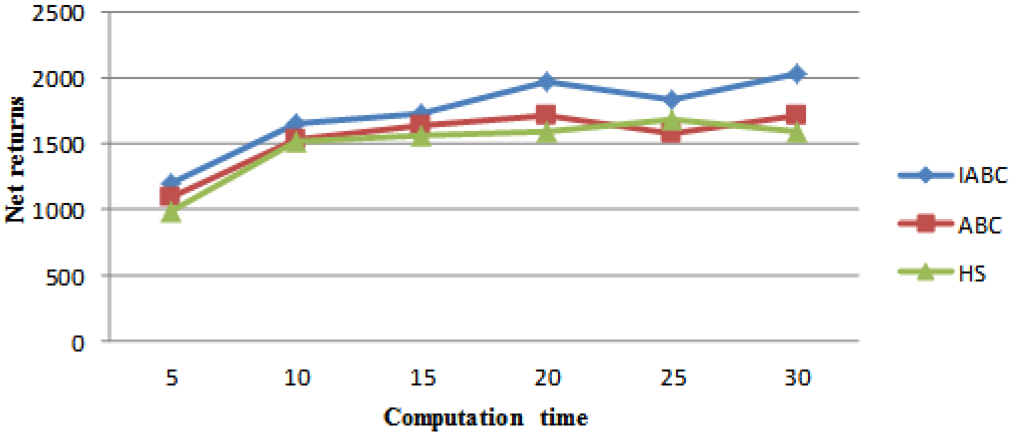


Figure 1. The Convergence Trend of IABC, ABC and HS

The convergence trend of IABC, ABC and HS, number of order is 30, the number of iterations is 100. X-coordinate represents net return of order, Y- coordinate represents computation time. Figure 4 illustrates convergence rate of IABC is better than ABC and HS, ABC and HS is similar. IABC is best to solve the problem efficiently, and ABC is better than HS.

6. Conclusion

Research on order acceptance and scheduling in multi-node production wit hybrid blocking, and the problem is solved by improved artificial bee colony algorithm. Compared to artificial bee colony and harmony search, improved artificial bee colony algorithm is best to solve. Three algorithms within the effective time are able to get the approximate optimal solution, and the total return of orders accepted is obtained. Future research can be carried out in the following directions: (1) Artificial bee colony algorithm is improved for order acceptance and scheduling in multi-node with hybrid blocking; (2) Order acceptance and scheduling in multi-node and multi-route production is considered; (3) Order acceptance and scheduling with batching in multi-node production is considered.

References

[1] N. G. Hall and C. Sriskandarajah, “A survey of machine scheduling with Blocking and no-wait in process”, Operations Research, vol. 44, no. 3, **(1996)**, pp. 510-525.

[2] N. Caraffa, S. Ianes and T. P. Bagchi, “Minimizing makespan in a blocking flowshop using genetic algorithms”, International Journal Production Economy, vol. 70, no. 2, **(2001)**, pp. 101-115.

[3] J. Grabowski and J. Pempera, “The permutation flow shop problem with blocking”, A tabu search approach, OMEGA, vol. 35, no. 3, **(2007)**, pp. 302-311.

[4] L. Wang, Q. K. Pan and P. N. Suganthan, “A novel hybrid discrete differential evolution algorithm for blocking flow shop scheduling problems”, Computers & Operations Research, vol. 37, no. 3, **(2010)**, pp. 509-520.

[5] F. Arredondo and E. Martinez, “Learning and adaptation of a policy for dynamic order acceptance in make-to-order manufacturing”, Computers & Industrial Engineering, vol. 58, no. 1, **(2010)**, pp. 70-83.

[6] H. F. Lewis and S. A. Slotnick, “Multi-period job selection: planning workloads to maximize profit”, Computers & Operations Research, vol. 29, no. 8, **(2002)**, pp. 1081-1098.

[7] Y. Y. Xiao, R. Q. Zhang and Q. H. Zhao, « Permutation flow shop scheduling with order acceptance and weighted tardiness”, Applied Mathematics and Computation, vol. 218, no. 15, **(2012)**, pp. 7911-7926.

[8] S. Senju and Y. Toyoda, “An approach to linear programming with 0–1 variables”, Management Science, vol. 15, no. 4, **(1968)**, pp. 196-207.

[9] S. K. Jain, K. L. Stott and E. G. Vasold, “Order book balancing using a combination of linear programming and heuristic techniques”, Interfaces, vol. 9, no. 1, **(1978)**, pp. 55-67.

[10] B. Pourbabi, “A short term production planning and scheduling model”, Engineering Costs and Production Economics, vol. 18, **(1989)**, pp. 159-167.

[11] B. Pourbabi, “Optimal selection of orders in a just-in-time manufacturing environment: A loading model for a computer integrated manufacturing system”, International Journal of Computer Integrated Manufacturing, vol. 5, no. 1, **(1992)**, pp. 3-44.

[12] R. Kolisch, “Integrated production planning, order acceptance, and due date setting for make-to-order manufacturing”, Operations Research Proceedings, **(1998)**.

M. Snoek, “Neuro-genetic order acceptance in a job shop setting”, Proceedings of the Seventh International Conference on Neural Information Processing, **(2000)**.

[13] A. Nandi and P. Rogers, “Using simulation to make order acceptance/rejection decisions”, Simulation, vol. 80, no. 3, **(2004)**, pp. 131-142.

[14] P. Rogers and A. Nandi, “Judicious order acceptance and order release in make-toorder manufacturing systems”, Production Planning and Control, vol. 18, no. 7, **(2007)**, pp. 610-625.

[15] M. R. A. Moreira and R. A. F. S. Alves, “A new input–output control order release mechanism: How workload control improves manufacturing operations in a jobshop”, European Simulation and Modelling Conference, **(2005)**.

[16] M. R. A. Moreira and R. A. F. S. Alves, “Does order negotiation improve the job-shop workload control?”, The European Operations Management Association Conference- Moving Up the Value Chain, **(2006)**.

[17] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm”, Journal of Global Optimization, vol. 39, **(2007)**, pp. 459-471.

Author

Chunhua Jin

Male, 42 years old, Postgraduate, University of Science and Technology Beijing Professor in the school of Economic and Management, Beijing Information Science & Technology University, No.12 at the east road of Qinghe Xiaoying, Haidian District, Beijing, 100192.