JOB SHOP SCHEDULING OPTIMIZATION AND SIMULATION BASED ON HYBRID OF NEIGHBORHOOD SEARCH AND GENETIC ALGORITHM

Ye WANG
School of Computer Science and Technology, Huaiyin Normal University, Jiangsu 223300, China.
E-mail: wang_yweio@163.com

***ABSTRACT**: Reasonable optimization scheduling of flexible work shop will greatly reduce process costs and improve production efficiency. This paper creatively builds a neighborhood search and genetic hybrid algorithm that allows for idle time, and applies it to the shop floor optimization scheduling. The proposed algorithm is tested for feasibility by a typical example and compared with other algorithms. Here, we investigate the semi-active, active and fully active decoding modes in the production scheduling process, and build an idle time neighborhood structure on critical path of production scheduling. It is also analyzed that there should be constraints for ensuring scheduling algorithm has a feasible solution in the transfer process. A search method for idle processes in different cases on the critical path is given herein. The integration of neighborhood search method with the traditional genetic algorithm can effectively improve the solution precision and efficiency of the algorithm. The results from simulation test show that relative deviation of the proposed algorithm is 0.22, far lower than that of other traditional algorithms, which further demonstrates the superiority of overall computation performance of the proposed algorithm. With the conclusions derived herein, it is certain to provide the clue to optimizing large-scale production scheduling.*

***KEYWORDS**: production shop scheduling, neighborhood search, genetic algorithm, simulation, decoding mode.*

1 INTRODUCTION

Job-shop Scheduling Problem (JSP) is a simplified model for practical production scheduling of manufacturer. It is attributed to reasonable arrangement of product transportation and machining sequence, normal service schedule for process equipment, and the reduction of additional production energy consumption in the job floor that the production cost can be significantly cut down, and the production efficiency can be improved. It is more significant for the production scheduling of flexible job shops (Vinod & Sridharan, 2008; Dong et al., 2012; Torabi et al., 2005; Barbosapóvoa, 2005; Liu et al., 2008).

The JSP is mainly solved by two algorithms, i.e. accurate and approximation algorithms. Accurate algorithm mostly applies the operations research or mathematical programming to solve well-established production scheduling model (Ramezanian et al., 2013; Mohammadi & Jafari, 2011). For example, Luh et al. improved the traditional job shop scheduling based on the Lagrangian relaxation method (Luh et al., 2000); Fernandes et al. adopted the branch and bound method to limit the search space of non-optimal solutions, thereby improving the evolutionary efficiency of algorithm in the search process of optimal solution (Fernandes, 2007); Aiex et al. used the parallel acceleration processors to optimize traditional branch and bound method and apply the improved method to the job shop scheduling (Aiex et al., 2003). The accurate algorithm can obtain an optimal solution when the production scheduling scale is small, but there is a huge computing burden. The exact algorithm is no longer applicable as the problem scale is enlarged (Wu et al., 2003). The approximation algorithm has been widely used for solving large-scale JSPs due to its shorter computation time and high quality of approximate solutions (Zhang et al., 2018). Approximation algorithms mainly include the heuristic scheduling, artificial intelligence scheduling, and computational intelligent scheduling methods (Zoghby et al., 2005; Goryachev et al., 2012; Shi et al., 2012; Liu et al., 2008).

It has been discovered from available literature that two or more optimization algorithms are hybridized to supply their respective gaps. In this way, the search capacity and convergence performance of individual algorithm can be improved (Tonelli et al., 2013; Clausen & Ju, 2006). For example, Zuo et al. merged the artificial immune algorithm with the emergency search algorithm, and used the fused algorithm to solve the flexible job shop scheduling (Zuo & Tan, 2012); Gao et al. integrated local search algorithm with cultural algorithm and applied it to the solution of JSP model (Gao et al., 2011); Rego et al. took the bottleneck movement strategy to improve the neighborhood search algorithm (César & Duarte, 2009). Ren et al. used the local search algorithm to improve the traditional genetic algorithm (Ren & Wang, 2012). The above hybrid algorithms have yielded certain fruits for the optimal scheduling of the flexible work shop.

Based on existing study, this paper creatively builds a neighborhood search and genetic hybrid algorithm that allows for idle time, and applies it to the production job shop optimization scheduling. It is tested for feasibility by a typical example and compared with other algorithms. With conclusions derived herein, it is useful for providing the clues to optimizing large-scale production floor scheduling.

2 PROBLEM DESCRIPTION

The traditional workshop production scheduling can be described as such a line that *n* products should be machined on some machines according to certain process path, and the product is not allowed to access the next machine for machining until the machining process has been finished on the previous machine, and the production process can't stop midway once it starts.

The objective function is to minimize the production machining time.

$$C_{max} = \min\left(\max_{i=1}^{n}(C_i)\right) \quad (1)$$

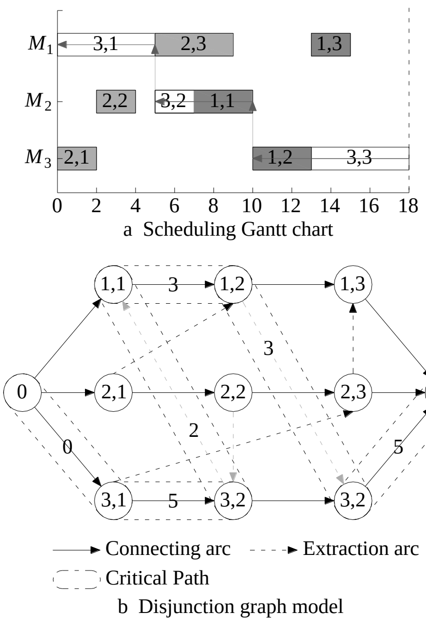


Fig. 1 Gantt and disjunction maps of job shop scheduling

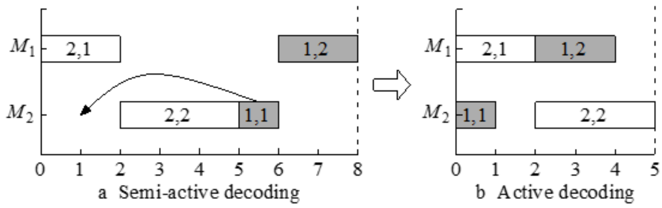
The job shop production scheduling is described by taking three products J_1 - J_3 and three machines M_1 - M_3 as examples. The Gantt and the disjunction diagrams of the production workshop scheduling under the above conditions are shown in Fig. 1.

In Fig. 1(a), the machining path in the direction of the arrow is the critical path of the production floor, and the path in the disjunction map is the longest from 0 to #, that is, the path as indicated by the arrow in the dotted line. The Gantt and disjunction maps can be used to determine the optimal machining sequence of products and the feasible solution of the scheduling.

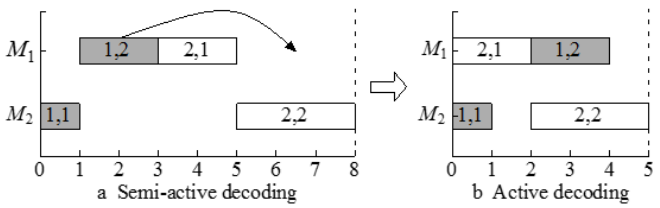
3 NEIGHBORHOOD SEARCH ALGORITHM ALLOWING FOR MACHINING IDLENESS TIME

3.1 Neighborhood structure

In the production scheduling scheme design, the product machining process code is first generated, when is then decoded to obtain the Gantt chart for scheduling. As shown in Fig. 2(a) and Fig. 2(b), the process in which the typical semi-active decode is converted into active decode which is then converted into full-active decode in the production scheduling decoding mode are given there.



(a) Semi-active decode is converted into active decode



(b) Active decode is converted into full-active decode

Fig. 2 Conversion of production processing scheduling decoding modes

The initial machining time of the process *n* of semi-active decoding is the maximum of finishing time of all products in the previous process; the active decoding is to optimize based on semi-active decoding, and all machining processes are shifted to the left. The full active decoding is further to move all machining processes to the right based on active decoding. In addition to semi-active decoding, the other two decoding modes can be allowed to move the processes only at the idle time period of machining.

Some processes can be deferred in production process to reduce shorter slots of idle time, thus diminishing the holistic machining time of production scheduling. Further illustrate by Fig. 3, as the idle time slot analysis of the Fig. 1 Gantt chart for scheduling, wherein Fig. 3(a) illustrates that on the critical path (indicated by the arrow in the figure) of the initial planning, the latter process starts immediately after the previous process is finished, and there is no idle time slot between adjacent processes. In the non-critical path, there are two smaller idle time slots on the M2 machine, within which any process can not be completed. The processes (1, 1) are moved to the positions 1 and 2 in Fig. 3(a), respectively, so that the Gantt charts of Fig. 3(b) and Fig. 3(c) are formed. In this way, the initial machining of processes (2,2) and (3, 2) are delayed, but the wholistic machining durations of the production workshop are reduced to 12 and 15, respectively, so that the completion time of the product is significantly lessened.

3.2 Shifting conditions for processes on critical path

The critical path for product machining is the major influence factor of the maximum completion time of workshop production. Based on the neighborhood search algorithm proposed herein, it is required to ensure that there is feasible solution after the process movement on the critical path.

In order to ensure that feasible solution always exists in production scheduling, this paper defines the following

Theorem 1: Assume the adjacent two processes on the one machine are *u* and *v*, respectively (*u* is implemented first), and the feasible solution is *s*; when *u* is the last process of product machining, or conversely, and

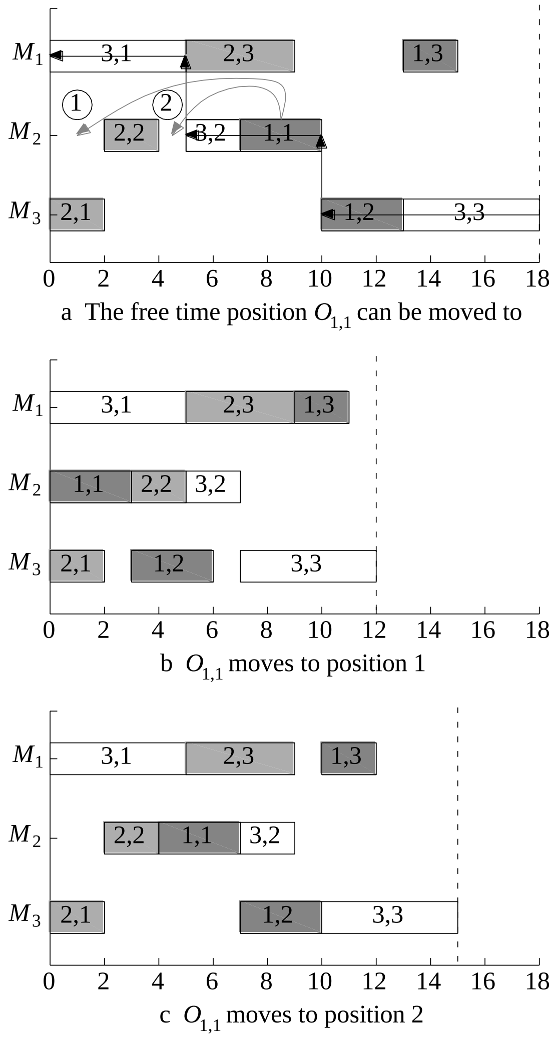
$$L(v,\#) \geq L(JS[u],\#) \quad (2)$$

It is meant that the feasible solution of the workshop production scheduling still exists when the process *v* is preceded the process *u*. In the formula, *L* is the length of the critical path, and JS[u] is the process *u* after the movement.

Theorem 2: Assume the adjacent two processes on the one machine is *u* and *v*, respectively (process u is implemented first), and the feasible solution is *s*; when *v* is the first process of product machining, or conversely, and

$$L(0,u) + p_u \geq L(0,JP[v]) + p_{JP[v]} \quad (3)$$

It is meant that the feasible solution of the workshop production scheduling still exists when the process *v* is preceded the process *u*.



a The free time position $O_{1,1}$ can be moved to

b $O_{1,1}$ moves to position 1

c $O_{1,1}$ moves to position 2

Fig. 3 Analysis of idle time slots of machines

The above two theorems will be interpreted with reference to Fig. 4. After the proves *u* is moved, the positions of all processes in the production are rearranged and reset, as shown in Fig. 4(b). When *u* is not the last process of product machining, then

$$L(v,\#) = makespan - s^{L}(v) \quad (4)$$

$$L(JS[u],\#) = makespan - s^{L}(JS[u]) \quad (5)$$

makespan represents the maximum completion time of workshop production. From the above relationship, the following conclusions can be further derived.

$$L(v,\#) \geq L(JS[u],\#) \Leftrightarrow s^{L}(v) \leq s^{L}(JS[u]) \quad (6)$$

When *v* is the first process of product machining, then

$$L(0,u) + p_u = c^{E}(u) \quad (7)$$

$$L(0,JP[v]) + p_{JP[v]} = c^{E}(0,JP[v]) \quad (8)$$

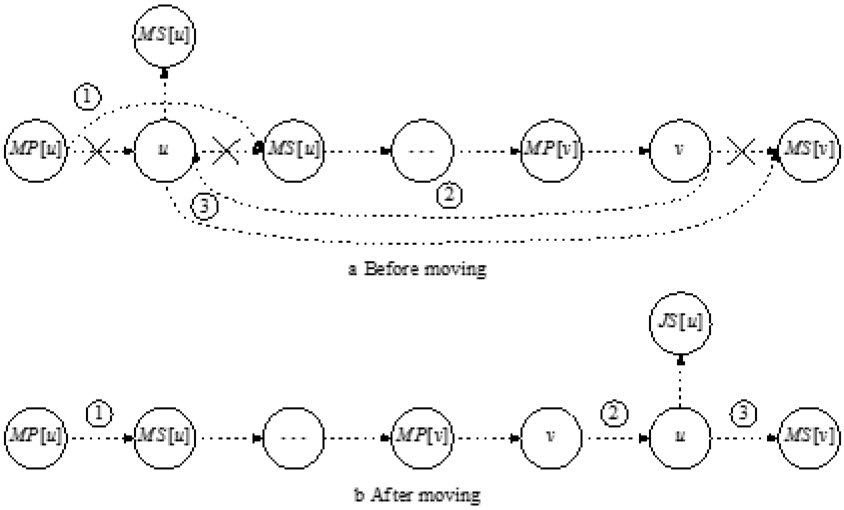


Fig. 4 Impact of the movement of different machining processes on feasible solution of scheduling

The following conclusion can be available according to Formulas (3), (7) and (8).

$$L(0,u)+p_u \ge L(0,JP[v])+p_{JP[v]} \Leftrightarrow c^E(u) \ge c^E(0,JP[v]) \quad (9)$$

3.3 Analysis of idle time slots between different machining processes

Further, the idle time search process between adjacent processes is described. To make a distinction from the analysis of the previous section, it is assumed that the adjacent processes are x and y, and the maximum idle time between the two is IT (x, y), its maximum is

$$IT(x,y)=s^L(y)-c^E(x) \quad (10)$$

That is, the initial time of the previous process x is the minimum, and the completion time of the next process y is the maximum. As shown in Fig. 5, the idle time lookup process between adjacent processes is analyzed in the form of a Gantt chart.

In Fig. 5(a), each process in the initial planning scheme has its minimum machining and completion time slots. All processes in the figure are moved to the positions where they are started and finished at the latest time within the planned range, as shown in Fig. 5(b). In Fig. 5(a), the maximum completion time of the processes (3, 1) and (2, 3) on the machine M_1 is 7, the value IT before the process (2, 2) and free time after the process (1, 3) are all 3. Free time can be available by the idle time search process proposed herein can obtain statistically, as the foundation for subsequent optimization.

In Fig. 5, the proposed neighborhood structure that allows for the idle time of the processes is shown in Fig. 6. Let w be one process on the critical path of production scheduling process of the workshop, then the maximum idle time of w is the difference between the earliest completion time of the previous process and the latest completion time of the next process.

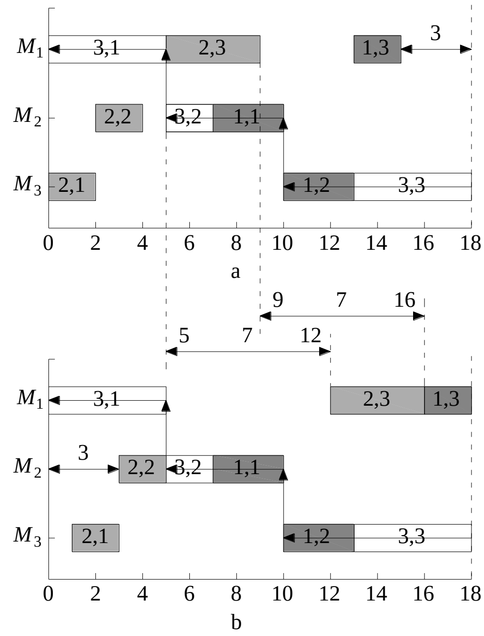


Fig. 5 Idle job search between adjacent processes

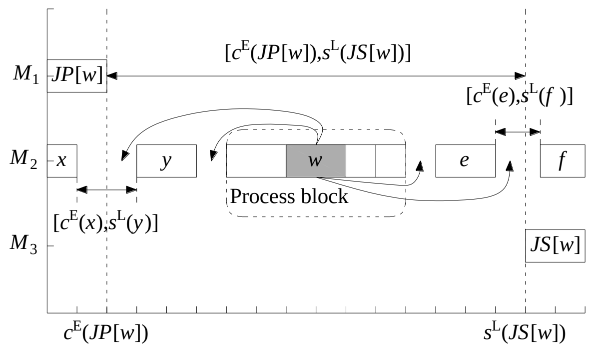


Fig. 6 Neighborhood structure that allows for process idle time

The process conversion does not obtain the feasible solution, and may also lead to the existence of a non-feasible solution to production scheduling.

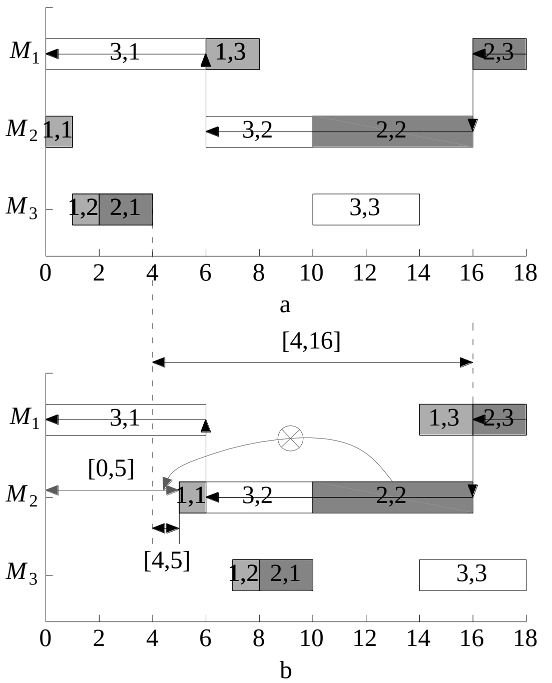


Fig. 7 Infeasible solution process formed after process transfer

Fig. 7 shows the infeasible solution flow that forms the production schedule after process transfer. Processes (2, 2) are on the critical path of production scheduling. In the machining time slot [4, 16], there is an idle time slot [4, 5] in front of the processes (1, 1). When the processes (2, 2) are preceded the processes (1, 1), the deadlock phenomenon occurs in 4 processes on the machines M_2 and M_3, which makes the whole production schedule infeasible.

To eliminate the above phenomenon, based on the idea of neighborhood search algorithm, this paper makes a mobile feasibility analysis of the process on the critical path during the production scheduling.

Since there is no idle time between adjacent machining processes in the one block, only the idle processes should be searched at the initial and completion time in the whole process.

For the first process w in the block, any adjacent processes x and y processed before w satisfy the following formula:

$$\left[c^E(x), s^L(y)\right] \cap \left[c^E(JP(w)), s^L(JS(w))\right] = 0 \quad (11)$$

Then there is no idle time before the first process *w*, and the idle time search is limited to the backward operation. Similarly, when the machining process *w* in the block is the last one, the subsequent adjacent processes *e* and *f* satisfy

$$\left[c^E(e), s^L(f)\right] \cap \left[c^E(JP(w)), s^L(JS(w))\right] = 0 \quad (12)$$

That is, there is no idle time after *w*, and the idle time search should be performed with the migration operation.

3.4 Analysis of idle interval between different processes

Based on the above analysis, the job shop optimization scheduling process based on the hybrid of neighborhood search and genetic algorithm that allows for idle time is shown in Fig. 8.

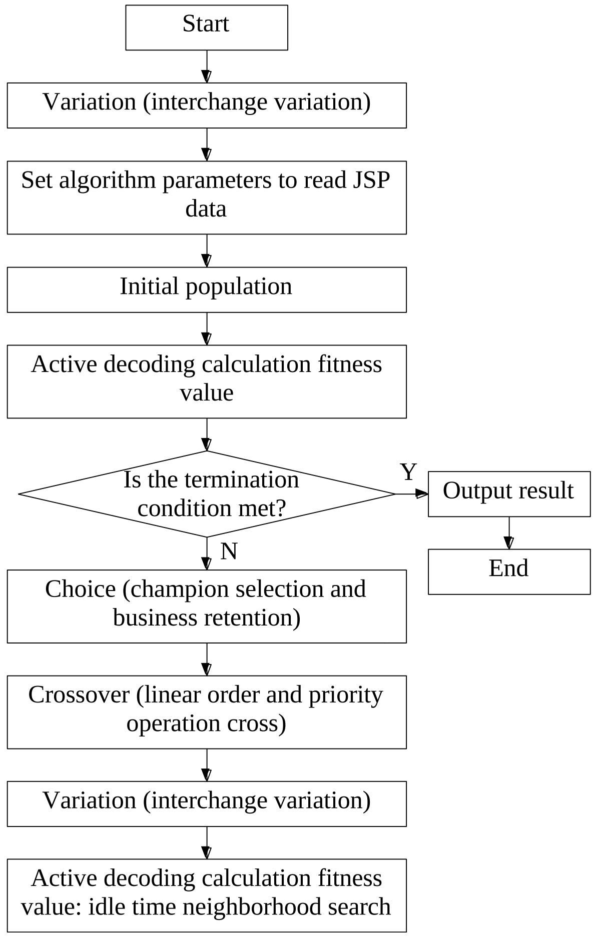


Fig. 8 Infeasible solution process formed after process transfer

4 SIMULATION CASE AND ANALYSIS

Test the availability of the algorithm proposed herein, there are 20 cases used to test the algorithm. The problem size of the cases 1 - 5 is 15×10; the problem size of the cases 6 - 10 is 20×10; the problem size in the cases 11 - 15 is 30×10, and the problem size of cases 15 - 20 is 15×15.

The crossover and mutation probabilities are 0.8 and 0.2, respectively, and the maximum number of iterations is calculated as 100 generations.

When the algorithm proposed in this paper is compared with the traditional production scheduling method AIS-TS hybrid algorithm and PSO-AIS hybrid algorithm. As shown in Fig. 9, these three algorithms obtain their respective optimal maximum completion durations C_{max} and theoretical optimal completion durations C_t.

With the relative deviation R, the deviation in the optimal completion time C_{max} and the theoretical optimal time C_t between different algorithms

$$Re = (C_{max} - C_t)/C_t \times 100\% \quad (13)$$

The average relative deviation ave(R) from 20 cases is available. The average relative deviation of the algorithm is ave(R)=0.22. AIS-TS and PSO-AIS hybrid algorithms are 0.30 and 0.33, respectively. The overall performance of the proposed algorithm is significantly better than other algorithms.

In Fig. 10 (a, b, c) is Gantt chart of production scheduling based on calculation cases 5, 7 and 16.

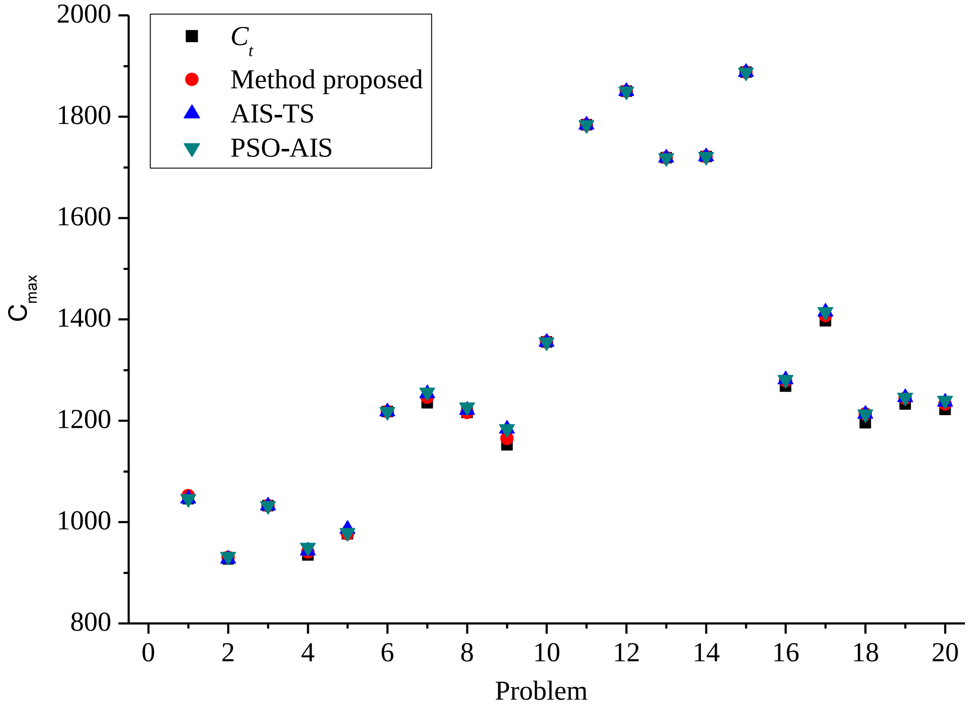
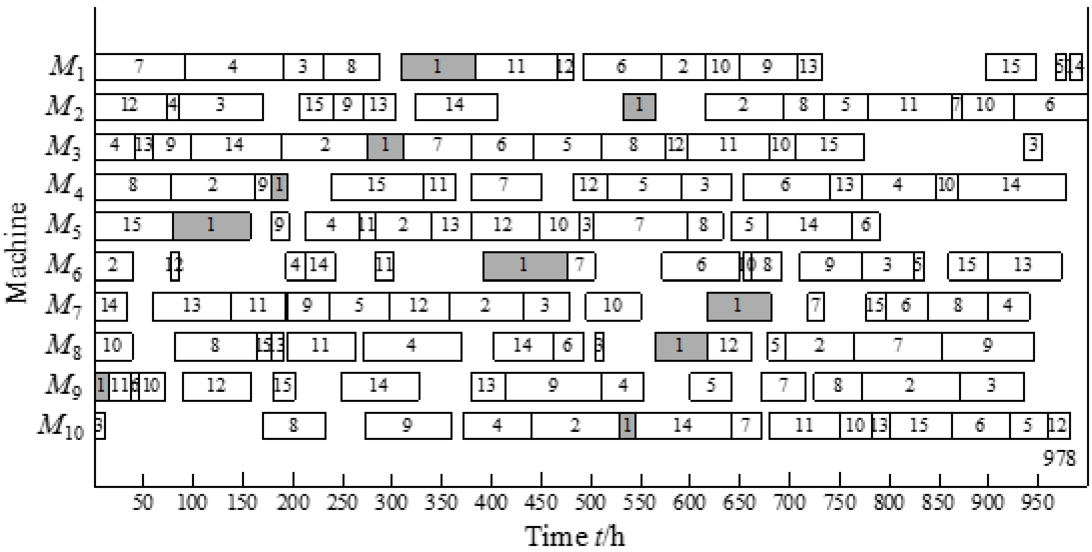
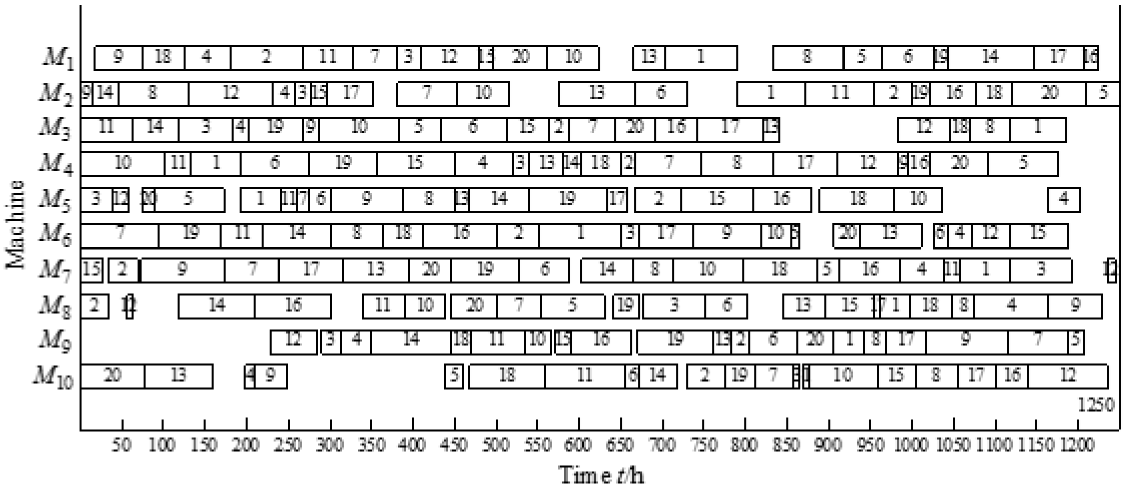


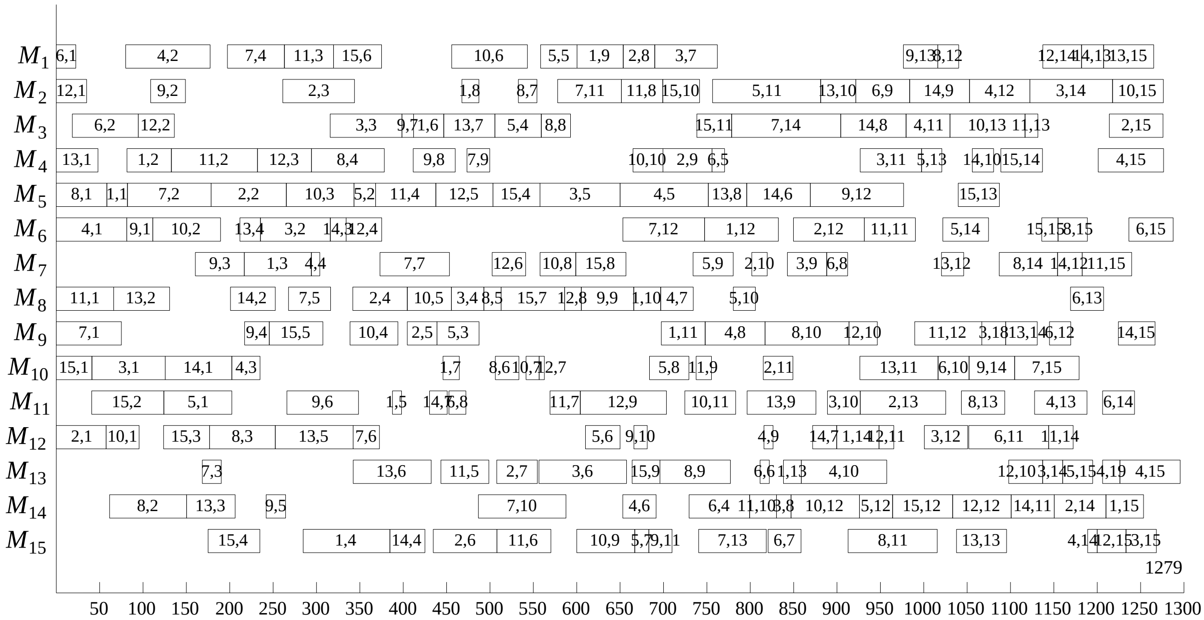
Fig. 9 Optimal completion time statistics of production scheduling with different algorithms



(a) Calculation case



(b) Calculation case 7



(c) Calculation case 16

Fig. 10 Gantt chart of production scheduling with three algorithms

5 CONCLUSION

This paper creatively builds a neighborhood search and genetic hybrid algorithms that allows for idle time, and applies it to the production job shop optimization scheduling. It is tested by the typical example and comparing with other algorithms for the feasibility. Here come several conclusions:

(1) The semi-active, active and all-active decoding modes in the production scheduling process are studied, and the idle time neighborhood structure on the critical path of production scheduling is then constructed. It is also analyzed there should be constraints for ensuring the feasible solution of the scheduling algorithm in the transfer process. In addition, the search method is given for idle processes in different cases on the critical path. The neighborhood search algorithm is integrated into the traditional genetic algorithm, which effectively improves the solution precision and efficiency of the algorithm.

(2) The results from simulation test reveal that the relative deviation of the proposed algorithm is 0.22, far lower than other traditional algorithms, which demonstrates the superiority of the overall computation performance of the proposed algorithm.

6 REFERENCES

►Aiex, R. M., Binato, S., Resende, M. G. C. (2003). Parallel grasp with path-relinking for job shop scheduling. Parallel Computing, 29(4), 393-430.

►Barbosapóvoa, A. P. (2005). Optimal scheduling for flexible job shop operation. International Journal of Production Research, 43(11), 2323-2353.

►Clausen, J., Ju, S. (2006). A hybrid algorithm for solving the economic lot and delivery scheduling problem in the common cycle case. European Journal of Operational Research, 175(2), 1141-1150.

►Dong, Q. Y., Lu, J. S., Gui, Y. K. (2012). Integrated optimization of production planning and scheduling in mixed model assembly line. Procedia Engineering, 29(4), 3340-3347.

►Fernandes, S. (2007). A GRASP and branch-and-bound metaheuristic for the job-shop scheduling. European Conference on Evolutionary Computation in Combinatorial Optimization, 4446, 60-71.

►Gao, L., Zhang, G., Zhang, L., Li, X. (2011). An efficient memetic algorithm for solving the job shop scheduling problem. Computers & Industrial Engineering, 60(4), 699-705.

►Goryachev, A., Kozhevnikov, S., Kolbova, E., Kuznetsov, O., Simonova, E., Skobelev, P., Alexander, T., Shepilov, Y. (2012). “Smart factory”: intelligent system for workshop resource allocation, scheduling, optimization and controlling in real time. Advanced Materials Research, 630, 508-513.

►Liu, X. X., Xie, L. Y., Tao, Z., Hao, C. Z. (2008). Flexible job shop scheduling for decreasing production costs. Journal of Northeastern University, 29(4), 561-564.

►Liu, Y. X., Xiong, J., Sun, B. M. (2008). Research on dynamic scheduling of job-shop production with the ant colony optimal algorithm. Applied Mechanics and Materials, 10-12, 109-113.

►Luh, P. B., Zhao, X., Wang, Y., Thakur, L. S. (2000). Lagrangian Relaxation Neural Networks for Job Shop Scheduling. IEEE International Conference on Robotics & Automation, 16(1), 78-88.

►Mohammadi, M., Jafari, N. (2011). A new mathematical model for integrating lot sizing, loading, and scheduling decisions in flexible flow shops. International Journal of Advanced Manufacturing Technology, 55(5-8), 709-721.

►Ramezanian, R., Saidimehrabad, M., Teimoury, E. (2013). A mathematical model for integrating lot-sizing and scheduling problem in capacitated flow shop environments. International Journal of Advanced Manufacturing Technology, 66(1-4), 347-361.

►Rego, C., Duarte, R. (2009). A filter-and-fan approach to the job shop scheduling problem. European Journal of Operational Research, 194(3), 650-662.

►Ren, Q. D. E. J., Wang, Y. (2012). A new hybrid genetic algorithm for job shop scheduling problem. Computers & Operations Research, 39(10), 2291-2299.

►Shi, C. T., Yang, G. J., Li, T. K. (2012). An evolutionary neighborhood search algorithm for flexible job-shop scheduling problem in steel tube production. Advanced Materials Research, 566, 620-627.

►Tonelli, F., Paolucci, M., Anghinolfi, D., Taticchi, P. (2013). Production planning of mixed-model assembly lines: a heuristic mixed integer programming based approach. Production Planning & Control, 24(1), 110-127.

►Torabi, S. A., Karimi, B., Ghomi, S. M. T. F. (2005). The common cycle economic lot scheduling in flexible job shops: the finite horizon case. International Journal of Production Economics, 97(1), 52-65.

►Vinod, V., Sridharan, R. (2008). Scheduling a dynamic job shop production system with sequence-dependent setups: an experimental study. Robotics & Computer Integrated Manufacturing, 24(3), 435-449.

►Wu, L. Y., Hu, Y. D., Xu, D. M., Hua, B. (2003). Solving batch production scheduling using genetic algorithm. Computer Aided Chemical Engineering, 15, 648-653.

►Zhang, X., Xu, D., Du, D., Wu, C. (2018). Approximation algorithms for precedence-constrained identical machine scheduling with rejection. Journal of Combinatorial Optimization, 35(1), 318-330.

►Zoghby, J., Barnes, J. W., Hasenbein, J. J. (2005). Modeling the reentrant job shop scheduling problem with setups for metaheuristic searches. European Journal of Operational Research, 167(2), 336-348.

►Zuo, X., Tan, W. (2012). Two heads are better than one: an ais- and ts-based hybrid strategy for job shop scheduling problems. International Journal of Advanced Manufacturing Technology, 63(1-4), 155-168.