# Parallel Branch and Bound Algorithm for Product Testing Job Scheduling Problems using MapReduce

# Yu Yu

Abstract—This paper analysis the similarities and differences between test scheduling and production scheduling. A job parallelization scheduling model based on characteristics of test scheduling is proposed. Further, the branch and bound search algorithm of job shop scheduling problem is studied. The MR-WFBB algorithm based on cloud computing MapReduce computing model is proposed. This algorithm is a novel job shop scheduling parallelization breadth-first branch and bound algorithm. Based on the actual test scheduling, this paper proposes the constraints of the job parallelization scheduling mode, solves the job parallelization scheduling problem under certain constraints and gives the Gantt chart and the assignment table corresponding to the optimal solution. The optimal solution can provide calibration and comparison for various artificial intelligence scheduling algorithms.

Index Terms—MapReduce, Job shop, parallel algorithm.

#### I. INTRODUCTION

Product test job scheduling aims at improving test efficiency through scheduling both test devices and test samples. Most of recent researches focus on scheduling in the field of manufacturing. The typical shop scheduling problem includes a set of work pieces to be processed, each of which includes a set of processes and each process needs to use production resources such as machine tools and process according to a certain process route. Different machine tools can be processed differently. The purpose of scheduling is to allocate the machine tool and other resources reasonably, so that some indicators can be optimized while satisfying certain constraints. Most of the shop scheduling problems are NP-hard, that is, there is no polynomial time complexity algorithm, which makes the scheduling problem a hot research topic. This paper first introduces the related concepts of shop scheduling, then analyzes the difference between test scheduling and shop scheduling. The job parallelization scheduling model suitable for test scheduling is first proposed and the mathematical definition of the model is given. Aiming at the optimal solution of the job parallelization scheduling mode, the MR-WFBB algorithm is proposed. The MR-WFBB algorithm is a parallel deterministic job shop scheduling algorithm based on the cloud computing MapReduce calculation model. The experimental part verifies the correctness of the model and algorithm using the published benchmark example.

#### II. CLASSIFICATION OF SHOP SCHEDULING PROBLEMS

The shop scheduling problem comes from the actual production scheduling. According to different processing steps and processing machines, it can be divided into three categories: job shop scheduling, open shop scheduling and flexible job shop scheduling.

## A. Job Shop Scheduling

In the deterministic job shop scheduling problem, the machining process of each work piece is determined, and the processing time and processing machine of each process are also predetermined [1].

#### B. Open Shop Scheduling

In the open shop scheduling problem, the order of processing between the processes of each work piece is arbitrary. The machining of the work piece can start from any process and end at any one step. There is no specific technical route constraints for the machining of work pieces and there is no sequence constraint between the processes [2].

#### C. Flexible Job Shop Scheduling

In the flexible job shop scheduling problem, the processing machine of each process is uncertain, and each process of each work piece can be processed multiple selectable processing machines. The processing times of the same work piece on different processing machines are different [3].

# III. EVALUATION CRITERIA FOR WORKSHOP SCHEDULING PROBLEMS

The evaluation criteria are the key for evaluating the pros and cons of the scheduling scheme. Common scheduling indicators are: 1) cost indicators. 2) Scheduling performance indicators. 3) User demand indicators. This paper mainly uses Makespan [4] as the scheduling performance indicator, which refers to the minimum value of the maximum completion time of all jobs.

## IV. RELATED WORKS

## A. Heuristic-Based Algorithm

The bottleneck transfer algorithm [5] is a solution space search algorithm based on heuristic rules. The bottleneck transfer algorithm converts the scheduling problem of

Manuscript received May 13, 2019; revised January 3, 2020.

YuYu is with Institute of Ocean Instruments and Metrology, Qilu University of Technology (Shandong Academy of Sciences), Qingdao, China (e-mail: rainertop@126.com).

multiple machines and multiple work pieces into scheduling problems of multiple single machines, find the bottleneck machine and schedules the bottleneck machine. The basic principle of heuristic rules is to reduce the impact of bottlenecks on the evaluation criteria of scheduling problems. Each round of execution of the bottleneck transfer algorithm selects a bottleneck machine and the algorithm ends after executing the round of the number of machines included, so the bottleneck transfer algorithm is a fast scheduling algorithm.

# B. Neighborhood Search Algorithm

- Taboo search: Taboo search [6]-[9] is widely used in the field of combinatorial optimization such as scheduling problems. The basic idea of taboo search is to complete the search for feasible scheduling schemes in the search space through a series of moves.
- 2) Simulated annealing algorithm: Simulated annealing algorithm [10], [11] is based on the simulation of the physical phenomenon of metal annealing cooling process. The current state of the thermodynamic system corresponds to the current scheduling scheme. The energy equation of the thermodynamic system corresponds to the objective function of the scheduling and the ground state corresponds to the global optimal scheme. In addition to the global energy of the system J, there is also a global temperature T, which is gradually reduced as the iterative process of simulated annealing. The simulated annealing algorithm generates a random scheduling scheme by sampling the probability distribution of the system, as shown in Eq. (1).

$$P_j \propto \exp(-T(\Delta J_{best} - \Delta J_i)/K) \tag{1}$$

Among them,  $P_j$  represents the probability of moving to *j* from all neighbors.  $\Delta J_{best}$  represents the optimal selection of the increment corresponding to the objective function,  $\Delta J_i$  representing the corresponding increment of *j*. The coefficient *K* is the normalization factor. Due to the existence of probability  $P_j$ , the simulated annealing algorithm can jump out of the local optimal solution.

- 3) Genetic algorithm: The genetic algorithm [12], [13] is an optimal solution search method for simulating the biological evolution process of Darwin's biological evolution theory and the genetic mechanism. In essence, the genetic algorithm encodes the parallel search process of the concept space, in which each process attempts to search the concept space by the coarse-grained hill-climbing method. The steps of genetic algorithms consist of four parts. 1) Fixed-length string encoding for the solution to the problem. 2) A suitable moderation function to divide the level of each solution. 3) Solution population initialization method. 4) Crossover, mutation and other domain-related genetic operators that are applied to the parent population to alter the gene composition of the offspring.
- 4) Branch and bound method: Compared with other algorithms, the branch and bound method [14], [15] has

the greatest advantage that it is an algorithm for determining the optimal solution. The branch and bound method can be divided into the depth-first branch and bound method and the breadth-first branch and bound method. For the job shop scheduling problem with the lower limit, the depth-first branch and bound method can get the solution of the problem earlier, thus having high probability of completing more pruning through the lower bound of the node for search space reducing.

#### V. JOB PARALLELIZATION SCHEDULING MODEL

In actual product testing, testers usually use multiple samples of the same model for testing to improve product testing efficiency. The biggest difference between product test scheduling and manufacturing scheduling is here. Product test scheduling can increase the efficiency of product testing by increasing the number of test samples for job parallelization. In the field of manufacturing, the efficiency is often improved by increasing the number of machines. In the study of job parallelization scheduling mode, the focus of researches is on job parallelization and the solution of job shop scheduling problem based on it.

# A. Example Study

The classic example of job shop scheduling in [16] is used to illustrate the job parallel scheduling model. As shown in Table I, in which *i* in the operation  $P_{ij}$  represents the machine number corresponding to the operation. *j* indicates the job number corresponding to the operation.

TABLE I: SCHEDULIN	G EXAMPLI
--------------------	-----------

Job	Machine Sequence	Processing Time
1	$1 \rightarrow 2 \rightarrow 3$	$p_{11} = 10, p_{21} = 8, p_{31} = 4$
2	$2 \rightarrow 1 \rightarrow 4 \rightarrow 3$	$p_{22} = 8, p_{12} = 3, p_{42} = 5, p_{32} = 6$
3	$1 \rightarrow 2 \rightarrow 4$	$p_{13} = 4, p_{23} = 7, p_{43} = 3$

Fig. 1 shows an optimal scheduling scheme for the job shop scheduling example shown in Table I. The maximum completion time is 28.



Assume that job 2 can be parallelized with a degree of parallelism of 2. The problem can be translated into a job shop scheduling problem as shown in Table II. The machine order 2→1→4→3 of the job 2 can be divided into {2→1} | {4→3} according to the business rule. The symbol | indicates that there is no order between {2→1} and {4→3}.



TADLE II. IOD DAD ALLEY ZATION

TABLE II. JOB TAKALLELIZATION					
Job	Machine Sequence	Processing Time			
1	$1 \rightarrow 2 \rightarrow 3$	$p_{11} = 10, p_{21} = 8, p_{31} = 4$			
2	$2 \rightarrow 1$	$p_{22} = 8, p_{12} = 3$			
2'	$4 \rightarrow 3$	$p_{42}' = 5, p_{32}' = 6$			
3	$1 \rightarrow 2 \rightarrow 4$	$p_{13} = 4, p_{23} = 7, p_{43} = 3$			

Fig. 2 demonstrates an optimal scheduling scheme for the parallelized scheduling of the job 2 as shown in Table II. The maximum completion time is 27.

2) Assume that jobs 2 and 3 can be parallelized with a degree of parallelism of 2. The problem can be converted into a job scheduling problem as shown in Table III.

TABLE III: PARALLELIZATION OF JOB 2 AND 3

Job	Machine Sequence	Processing Time
1	$1 \rightarrow 2 \rightarrow 3$	$p_{11} = 10, p_{21} = 8, p_{31} = 4$
2	$2 \rightarrow 1$	$p_{22} = 8, p_{12} = 3$
2'	$4 \rightarrow 3$	$p_{42}' = 5, p_{32}' = 6$
3	$1 \rightarrow 2 \rightarrow 4$	$p_{13} = 4, p_{23} = 7$
3'	4	$p'_{43} = 3$

The machine sequence  $1 \rightarrow 2 \rightarrow 4$  can be decomposed into  $\{1 \rightarrow 2\} | \{4\}$  according to business rules. Fig. 3 demonstrates an optimal scheduling scheme for the parallelized scheduling of job 2 and 3 as shown in III. The maximum completion time is 25. Comparing Fig. 1, Fig. 2 and Fig. 3, job parallelization effectively shortens the maximum completion time of job shop scheduling, thus improving the efficiency of testing.



Fig. 3. An optimal scheduling scheme for parallelization of job 2 and 3.

#### B. Mathematical Description

The test scheduling problem studied in this paper can be described as follows: Each test order contains N jobs, corresponding to the N test sub-items in the test order. M test station corresponds to M device, and each test

station can execute a test sub-item. The order of the test sub-items in the test order represents the corresponding process flow of the job shop scheduling problem.

Definition: job parallelization means that J job can be decomposed into k sub-jobs that are independent of the execution order, expressed as  $J \Rightarrow J'_1|J'_2|...J'_k$ , where the symbol | indicates that there is no order between the jobs. In the job parallelization scheduling model, the machines involved in the process flow before and after the job parallelization are unchanged, and only the number of samples is increased.

$$|J| = J_1' |+|J_2' |+...|J_k'|$$
(2)

where, |J| represents the number of machines involved in the process flow J that can parallelized, k < |J|.

In the test scheduling problem, job parallelization is also constrained by sample manufacturing capabilities C, and the number of samples cannot be unlimited. That is, for p parallelizable scheduling job, select q job that can be parallelized  $J_1, J_2, \ldots, J_q, q < p$  and satisfy:

$$k_1 + k_2 + \dots k_q \le C \tag{3}$$

The sample manufacturing capability is also manifested by a constraint on the number of sample types T, that is only T types of samples that can be manufactured.

$$q \le T \le p \tag{4}$$

The goal of job parallelization test scheduling is to minimize the maximum completion time if certain constraints are met.

Further a discrete programming model for job shop scheduling problems is presented.  $C_{\max}$  indicates the maximum completion time for job shop scheduling problems. Variable  $y_{ij}$  indicates the start time of the operation, N indicates a collection of all operations. A stands for all process order constraints  $(i, j) \rightarrow (k, j)$ , symbol  $\rightarrow$  indicates in the process flow of job j, machine i needs to be before machine k.

$$y_{kj} - y_{ij} \ge p_{ij}, (i, j) \to (k, j) \in A$$
(5)

$$C_{\max} - \mathbf{y}_{ii} \ge p_{ii}, (i, j) \in N \tag{6}$$

$$y_{ij} - y_{il} \ge p_{il} \text{ or } y_{il} - y_{ij} \ge p_{ij}$$
 (7)

$$y_{ii} \ge 0, (i, j) \in N \tag{8}$$

where formula 5 indicates that the operation (k, j) cannot be started before the operation (i, j) is completed. Formula 6 indicates the maximum completion time  $C_{\max}$  is equal to the completion time of the last operation. Formula 7 indicates there is a certain order relationship between different job jand l. Formula 8 indicates that start time of all operations is greater than or equal to 0.

#### VI. MR-WFBB ALGORITHM

According to the mathematical description of the job parallelization scheduling model, the job parallelization scheduling problem is equivalent to solving several job shop scheduling problems with selected job shop scheduling problems with selected shop scheduling scheme that minimizes the maximum completion time  $C_{\rm max}$ . According to the description of the constraint conditions of the job parallelization scheduling model, the scheduling scheme minimizes the maximum completion time  $C_{max}$  and minimizes the number of samples q and the sample type T, which is the optimal operation parallel scheduling. However, job shop scheduling itself has proven to be NP-Hard, making it difficult to solve job parallel scheduling. Based on the MapReduce solution framework for general combinatorial optimization problems, this paper proposes a branch and bound algorithm MR-WFBB based on MapReduce programming model, which realizes the parallel solution of job parallelization scheduling.

## A. Combined Optimization Problem Solving Framework

The MapReduce programming model is very suitable for solving combinatorial optimization problems (See Algorithm 1). For general combinatorial optimization problems, the key steps for solving the MapReduce programming model are described as follows: The advantage of the MapReduce solution model for combinatorial optimization problem is that for a specific problem, there is no need to have a deep understanding of the problem, and only the feasibility algorithm of the solution and the calculation method of the objective function can be solved. However, for the combinatorial optimization problem with large solution space, the MapReduce solution model of the general combinatorial optimization problem is not applicable. Taking the problem  $FT6 \times 6$  as an example, the size of the solution space has reached to  $(6!)^6$ .

*Algorithm 1 MapReduce* solution *framework for general Combinatorial* optimization *problems Input: Combined* optimization *problem;* 

**Output:** Optimal solution;

1: Generating solution space.

2: Determine the feasibility of the current solution in the Map function. If not feasible, discard directly.

3: For the feasible solution  $S_i$ , calculate the value of the

corresponding *objective function*  $T_i$  *in the Reduce* function,

save the key value pair  $(S_i, T_i)$ .

4: Sort all  $(S_i, T_i)$  by  $T_i, T_j = \max(T_1, \dots, T_n)$ , so  $S_j$  is the optimal solution.

### B. Branch and Bound Algorithm for Job Shop Scheduling

The branch and bound algorithm of job shop scheduling is represented by a discrete graph model based on job ship scheduling problem. Fig. 4 shows the discrete graph corresponding to the example shown in Table I. Where the solid line between the nodes represents the routing constraints between nodes represents jobs that are executed on the same machine. The feasible solution of the job shop scheduling corresponds to selecting one of a pair of oppositely directed dashed lines in the figure, and satisfying the condition that the graph is an acyclic graph.



Fig. 4. Discrete graph representation of job shop scheduling.

The longest distance from the source node to the sink node is the maximum completion time  $C_{\rm max}$  of the job shop scheduling problem represented by the discrete graph. The solution to the job shop scheduling is to find the optimal scheduling scheme to minimize  $C_{\rm max}$ .

The branch and bound algorithm is an algorithm for solving the optimal solution of job shop scheduling problem. In the actual solution process, the branch and bound algorithm of job shop scheduling is very effective for solving the optimal solution. The reason is that the branch and bound algorithm is only searching in the feasible solutions space, the key steps of the algorithm are described as show in Algorithm 4-2, and the related definitions of the branch and bound algorithm for job shop scheduling are described in [17].

Algorithm 2 Branch and bound method for job shop scheduling (depth first);

*Input:* Operations (including job information, machine information, and operating time information) and its routing constraints;

**Output:** The optimal solution of job shop scheduling problem and its corresponding scheduling scheme; 1: Collection  $\Omega$  contains the first action of each job. For each element (i, j) in the collection  $\Omega$ , let  $r_{ij} = 0$ .

2: Determine if the collection  $\Omega$  is empty. If the set  $\Omega$  is empty, let  $\theta = \theta \bigcup S_c$  and algorithm return. Where,  $S_c$  is a known solution, the set  $\theta$  is a set of known solutions. Solve the current local scheduling problem,  $t(\Omega) = \min\{r_{ij} + p_{ij}\}$ ,  $(i, j) \in \Omega$ .  $i^*$  represents the machine number corresponding to the operation that obtained the minimum value, where  $p_{ij}$  is the processing time of the operation and  $r_{ij}$  is the release time of the operation (Release time).

3: Let the set  $\Omega'$  denote the operation  $(i^*, j)$  on the machine  $i^*$  that satisfies the condition  $r_{i^*j} < t(\Omega)$ . For each operation *i* in the set  $\Omega'$ , calculate the lower bound  $LB_{i^*j}$ 

As described by Algorithm 2, the depth-first branch and bound algorithm of job shop scheduling is a recursive algorithm. In addition, in the algorithm solving process, the lower bound  $LB_{i*j}$  corresponding to the operation  $(i^*, j)$  needs to be calculated. The calculation method of the lower

bound  $LB_{i^*j}$  corresponds to the solution of n NP-Hard problems, which is limited by the space, and will not be further elaborated here. For related algorithms and proofs, see the reference [17].

# C. Improvement Measures of MR-WFBB Algorithm

The MR-WFBB algorithm is a parallel branch and bound algorithm for solving job shop scheduling problems. In [18], a parallel branch and bound algorithm based on message passing is proposed, which distributes the active nodes of job shop scheduling to multiple on the processor, the branch path in the search tree is concurrently searched, and the minimum value of the known solution is transmitted between the respective processors, thereby accelerating the search process of the branch path.

In the process of parallel computing, the MR-WFBB algorithm needs to pass part of the solution of the discrete graph as a parameter. In the transfer process, only one of the two dashed edges of the current solution represented by the discrete graph needs to be passed, without transmitting the entire discrete graph, as shown in Equation 9.

$$PG = \{(1,i) \to (1,i'), (2,j'), \dots, (m,k) \to (m,k')\}$$
(9)

The MR-WFBB algorithm consists of two main steps: First, get the suboptimal solution of the shop scheduling problem using the fast solution algorithm. Then, the branch ad bound search is performed on the solution space based on the breadth-first search. The suboptimal solution obtained in first step is used to control the search process by searching the lower bound function value of the node.



Fig. 5. Branch-limit search based on breadth-first.

Fig. 5 further describes the step two of the MR-WFBB algorithm. In the breadth-first search process, the nodes are searched layer by layer according to the level of the nodes in the search tree. It can be seen from the discrete graph representation of the job shop scheduling problem that the solution of the shop scheduling problem corresponds to the leaf node in the breath-first search tree, that is, the lowest layer in the search tree. The breadth-first search process is different from the depth-first search process, and it is not possible to update the known solution of the problem during the search process. However, there is no need to update known solutions, making breadth-first search easier to

parallelize than depth-first search. In step two of the algorithm MR-WFBB, the breadth-first search can only compare the suboptimal solution S' obtained by the fast solution algorithm in step one with the lower bound of the node and pruning the node. In Figure 5, the light-colored shading node indicates that the lower bound value of the node is greater than the sub-optimal solution S', so discard it directly.

Algorithm 3 Job Shop Scheduling MR-WFBB Algorithm Input: Operations (including job information, machine information, and operating time information), routing constraints, and suboptimal solutions S';

**Output:** The optimal solution of job shop scheduling problem and its corresponding scheduling scheme;

*I*:Define sets  $\Omega_0$  and  $O_0$ , where  $\Omega_0$  contains the first operation of each job,  $O_0$  is initialized to empty which represents partial solution of set  $\Omega_0$ .

2: Create  $Job_{wi}$ ,  $Job_{wi}$  reads current  $\Omega_{i-1}$  and partial solution  $O_{i-1}$ , output  $\Omega_i$  (next level of  $\Omega_{i-1}$ ) and partial solution  $O_i$ . Among which, the definition of set  $\Omega_{i-1}$  and  $\Omega_i$  are the same as  $\Omega$  in step 4 of Algorithm (2).

3: Repeat step (2) k times, k is the total number of operations.

4: Create  $Job_{c1}$ , the minimum value  $min(value(Output_{c1}))$  in the output  $Output_{c1}$  of  $Job Job_{c1}$  is the optimal solution for job shop scheduling. The corresponding  $key(Output_{c1})$  is the optimal scheduling scheme.

Algorithm 3 describes the key steps of the overall iterative process of the MR-WFBB algorithm for job shop scheduling. Algorithm 4 describes the key steps of the Mapper algorithm for search tasks in the MR-WFBB algorithm. Algorithm 5 describes the key steps of the Mapper algorithm for solving the shop scheduling problem corresponding to the leaf nodes in the MR-WFBB algorithm.

Algorithm 4 Mapper algorithm for Job<sub>wi</sub>

**Input:** Set  $\Omega$ , the current partial solution O, and the suboptimal solution S', where the definition of set  $\Omega$  is the same as the step (4) of algorithm 2;

**Output:** Set  $\tilde{\Omega} = {\tilde{\Omega}_1, ..., \tilde{\Omega}_k}$  and its corresponding partial

solution  $\tilde{O} = \overline{O}_1, \dots, \overline{O}_k$ .

1: Solve the current local scheduling problem,  $t(\Omega) = \min\{r_{ij} + p_{ij}\}, (i, j) \in \Omega$ , the definition of  $i^*$ ,  $p_{ij}$ ,  $r_{ij}$ 

are the same as step (2) in algorithm 2.

2: Let the set  $\Omega'$  denote the operation  $(i^*, j)$  on the machine  $i^*$  that satisfies the condition  $r_{i^*j} < t(\Omega)$ . For each operation in the set  $\Omega'$ , calculate the lower bound  $LB_{i^*j}$  corresponding to the operation  $(i^*, j)$ . If  $LB_{i^*j} > S'$ , then directly discard operation  $(i^*, j)$ . Otherwise, consider this operation as an extended partial scheduling problem for the next operation of machine  $i^*$ .

3: For each extended partial scheduling problem, remove the corresponding operation from the set  $\Omega$  and add the direct subsequent operation from the set  $\Omega$  and add the direct subsequent operations of the operation to the set  $\Omega$ , thereby obtaining the set  $\overline{\Omega}$ . Add set  $\overline{\Omega}$  to the set  $\Omega$  with its corresponding partial solution  $\overline{O}$  added to set  $\widetilde{O}$ .

In Algorithm 3, the number of cycles k is the upper limit of the number of cycles required. Before the algorithm is executed, the value of the number of cycles k cannot be determined in advance. The upper limit calculation step of the number of cycles k is as follows.

1) For *m* machines, the scheduling problem of *n* tasks consists of *l* operations.

$$l = \sum_{i=1}^{n} JM_{i} = \sum_{j=1}^{m} MJ_{j}$$
(10)

where  $JM_i$  represents the number of machines in job *i* with  $JM_i \le m$ .  $MJ_j$  represents the number of tasks performed by machine *j* with  $MJ_j \le n$ .

- 2) Consider a worst case scenario in the branch and bound algorithm searching process of job shop scheduling. The machine sequences corresponding to the minimum value operation in step four of algorithm 4-2 are  $(1...1)_{MJ_1}, (2...2)_{MJ_2}, ..., (m...m)_{MJ_m}$
- 3) From step two and formula 10, the upper limit of the number of cycles  $k = \sum_{j=1}^{m} MJ_j$  can be obtained.

The reducer algorithm of  $Job_{wi}$  does not require special processing and is therefore omitted here.

The reducer algorithm of  $Job_{c1}$  does not require special processing and is therefore omitted here.

Algorithm 3 consists of k + 1 jobs. The role of the k jobs corresponding to  $Job_{wi}$  is to perform breadth-first search on the solution space. In the searching process, the suboptimal solution S' can be compared with the lower bound LB of the current partial solution which is discarded if  $LB \ge S'$ .  $Job_{c1}$  calculates the solution corresponding to the scheduling scheme generated by  $Job_{wk}$  and the minimum value is the optimal solution. Compared with other parallelization algorithms, the MR-WFBB algorithm inherits the simplicity, scalability and fault tolerance of the MapReduce framework.

Algorithm 5 Mapper algorithm for Job<sub>c1</sub>

# Input: Scheduling scheme O;

*Output:* The maximum completion time (Makespan) for the scheduling scheme O.

1:Construct a discrete graph G corresponding to the scheduling scheme O.

2: Mark the Source node in discrete graph G as calculated.

3: In the discrete graph G, find all nodes p whose precursor nodes are marked.

4: Calculate the tag value L(p) of node p. L(p) equals to the maximum value of  $L(\bar{p})$  plus  $T(\bar{p})$ .  $L(\bar{p})$  indicates the tag value of node  $\bar{p}$  which are the direct predecessor node of node p. T(p) indicates the maximum processing time of node  $\bar{p}$ .

5: Mark node p as calculated. If node p is the Sin k node of discrete graph, then exit. Otherwise, turn to Step(3).

6: Return the tag value of node Sin k, which is the maximum completion time Makespan corresponding to scheduling scheme O.

#### VII. EXPERIMENTS AND ANALYSIS

# A. Experimental Data

Experimental data is derived from the  $FT6\times6$  in OR-library.  $FT6\times6$  is first proposed in the literature [19].  $FT6\times6$  is chosen because it is typical and moderately difficult and its optimal solution is known to be 55.

As shown in Table IV, the lines in  $FT6\times 6$  indicate jobs marked 0~5. Odd columns represent machines numbered 0~5. The even column indicates the processing time of the job on the corresponding machine. The order of the odd column labels corresponding to each row of jobs indicates the routing corresponding to the job of the row. For example, the process route corresponding to Job1 is 2,0,1,3,5,4 with processing time are 1,3,6,7,3,6. Based on the  $FT6\times 6$  on the job parallelization scheduling mode is given, as shown in Table IV. For new jobs that are parallelized in the  $FTP6\times 6$  problem, renumber them directly from 11 to distinguish them from the original jobs in  $FT6\times 6$ .

TABLE IV: FT  $6 \times 6$  EXAMPLE

2	1	0	3	1	6	3	7	5	3	4	6
1	8	2	5	4	10	5	10	0	10	3	4
2	5	2	5	4	10	5	10	0	10	3	4
1	5	0	5	2	5	3	3	4	8	5	9
2	9	1	3	4	5	5	4	0	3	3	1
1	3	3	3	5	9	0	10	4	4	2	1

#### B. Analysis of Results

According to the  $FTP6 \times 6$  problem of job parallelization scheduling model, three job parallelization scheduling problems for different constraint combinations are designed. The optimal solution of job parallelization scheduling problem and its corresponding scheduling scheme with dispatch table and Gantt chart are given in three cases. Finally, the optimal solution of the  $FTP6 \times 6$  problem and the dispatch table and Gantt chart of the corresponding scheduling scheme are given.

 Situation 1: Satisfying the condition of formula (4-3) and the maximum completion time for parallelization of all parallelizable jobs is minimal. That is, the constraint of the formula (4-3) is satisfied, the constraints for the formula (4-4) and formula (4-5) for the number of samples and the kind of the sample are ignored, and all parallelizable operations are parallelized.

The MR-WFBB algorithm is used to solve the above problem, and the optimal solution is 43, and the Gantt chart of the corresponding scheduling scheme is shown in Fig. 6. The dispatch table of the corresponding scheduling scheme is shown in Table V. The data item n(st, et) in the dispatch table indicates that the job with the number n has a start time st on the corresponding machine and an end time of et.

2) Situation 2: Under the condition of constraint (4-3) and (4-4), and the maximum completion time of parallelization of all parallelizable job is minimum. That is, considering the constraint condition that satisfies equation (4-3) and equation (4-4) (the number of samples C = 4), the constraints of equation (4-5) for the sample type are ignored, and all parallelizable operations are parallelized. In the case that the constraint C = 4 is satisfied, there are two jobs in the corresponding  $FTP6 \times 6$  that cannot be executed in parallel. The same calculation is performed using the MR-WFBB algorithm, and the optimal solution is also 43, the Gantt chart of the

corresponding scheduling scheme is shown in Fig. 7. The dispatch table of the corresponding scheduling scheme is shown in Table VI.

3) Situation 3: Under the condition of constraint (4-3), (4-4) and (4-5), the maximum completion time for parallelization of all parallelizable operations is minimum. That is, the constraints that satisfying (4-3), (4-4) (C = 4) and (4-5) (T = 2), and all parallelizable operations are parallelized. In the case that the constraints C = 4, T = 2 are satisfied, only two jobs corresponding to  $FTP6 \times 6$  can be executed in parallel. The same calculation is performed using the MR-WFBB algorithm, and the optimal solution is also 49, the Gantt chart of the corresponding scheduling scheme is shown in Fig. 8. The dispatch table of the corresponding scheduling scheme is shown in Table VII.



Fig. 6. Gantt chart for situation 1.







Fig. 8. Gantt chart for situation 3.



Fig. 9. Gantt chart for the optimal solution of  $FT6 \times 6$ 

TABLE V: DISPATCH TABLE FOR SITUATION 1										
Machine1	Machine2	Machine3	Machine4	Machine5	Machine6					
22(0,10)	21(0,3)	11(0,1)	18(0,3)	18(3,11)	20(0,4)					
11(10,13)	13(3,11)	19(1,10)	21(3,6)	22(11,15)	14(4,14)					
14(14,24)	19(11,14)	13(11,16)	12(6,13)	19(15,20)	21(14,23)					
16(24,33)	11(14,20)	22(16,17)	14(24,28)	13(20,30)	12(23,26)					
17(33,38)	17(20,25)	15(17,22)	15(28,32)	12(30,36)	18(26,35)					
20(38,41)	16(33,34)	17(38,43)	20(41,42)	16(36,43)	15(35,43)					
TABLE VI: DISPATCH TABLE FOR SITUATION 2										
Machine1	Machine2	Machine3	Machine4	Machine5	Machine6					
22(0,10)	13(0,8)	1(0,1)	18(0,3)	18(3,11)	14(0,10)					
1(10,13)	17(8,13)	3(1,6)	3(6,10)	22(11,15)	3(10,18)					
14(13,23)	1(13,19)	13(8,13)	1(19,126)	13(15,25)	18(18,27)					
3(23,32)	21(19,22)	19(13,22)	14(26,30)	19(25,30)	1(27,30)					
17(32,37)	19(22,25)	22(22,23)	21(30,33)	1(30,36)	20(30,34)					
20(37,40)	3(32,33)	17(37,42)	20(40,43)	3(36,43)	21(34,43)					
		TABLE VII: DISPATO	TH TABLE FOR SITUATION 3							
Machine1	Machine2	Machine3	Machine4	Machine5	Machine6					
1(1,4)	13(0,8)	1(0,1)	18(0,3)	18(3,11)	14(0,10)					
14(10,20)	1(8,14)	3(1,6)	3(6,10)	13(13,23)	3(10,18)					
17(20,25)	17(14,19)	13(8,13)	1(14,21)	1(24,30)	1(21,24)					
3(25,34)	6(19,22)	5(13,22)	6(22,25)	5(30,35)	6(25,34)					
6(34,44)	5(22,25)	17(25,30)	14(25,29)	3(35,42)	5(35,39)					
5(44,47)	3(34,35)	6(48,49)	5(47,48)	6(44,48)	18(39,48)					
TABLEVIII: DISPATCH TABLE FOR SITUATION 4										
Machine1	Machine2	Machine3	Machine4	Machine5	Machine6					
22(0,10)	13(0,8)	1(0,1)	18(0,3)	18(3,11)	14(0,24)					
1(10,13)	17(8,13)	3(1,6)	3(6,10)	22(11,15)	3(10,18)					
14(13,23)	1(13,19)	13(8,13)	1(19,26)	13(15,25)	18(18,27)					
3(23,32)	21(19,22)	5(13,22)	14(26,30)	5(25,30)	1(27,30)					
17(32,37)	5(22,25)	22(22,23)	21(30,33)	1(30,36)	5(30,34)					
5(37,40)	3(32,33)	17(37,42)	5(40,41)	3(36,43)	21(34,43)					

Comparing Fig. 6 with Fig. 7, the scheduling scheme shown in Fig. 7 is less parallelized than the scheduling scheme shown in Fig. 6. However, the optimal solutions they obtained are the same, both being 43. Therefore, increasing the level of parallelization of the job does not necessarily reduce the minimum value of the maximum completion time.

In the above three examples, the constraint condition is finally transformed into the number of parallelizable jobs in the  $FP6 \times 6$  problem, and further converted into the solution of the job shop scheduling problem corresponding to the permutation combination composed of parallelizable jobs. The problem of permutation and combination of parallelizable jobs can be directly calculated by the combinatorial optimization problem solving framework. Therefore, based on MapReduce task automatic scheduling, the job shop scheduling problem corresponding to each combination scheme is submitted to MapReduce as a task, thus realizing the parallel solution of the combined scheme corresponding to the job shop scheduling problem.

Finally, an optimal solution to the  $FT6\times 6$  problem is given. The solution of the optimal solution can be .converted into the permutation and combination problem of all parallel operations, and the condition that the maximum completion time is minimum, the number of sample *C*, and the sample type *P* are the least. For the  $FT6 \times 6$  problem, the number of parallel jobs can be 1~6, and the parallel operation can be combined into  $C_6^1 \sim C_6^6$ .

The optimal solution is obtained under the condition that only the parallel operations 1, 3, 5 are performed in the corresponding  $FT6\times6$ , and the value is 43. The optimal solution is the same as the optimal solution of the  $FT6\times6$ problem shown in Fig. 6, but the corresponding scheduling scheme uses 3 sample less than Fig. 6 and 1 prototype less than Fig. 7. The corresponding Gantt chart is shown in Fig. 9, and the corresponding dispatch table is shown in Table VIII.

#### VIII. CONCLUSION

Based on the analysis of similarities and differences between test scheduling and production scheduling, this paper proposes a job parallel scheduling model for the characteristics of test scheduling. This paper also studies the branch and bound search algorithm of job shop scheduling problem, and proposes the MR-WFBB algorithm based on MapReduce calculation model. This algorithm is a novel job shop scheduling parallelization breath-first branch and bound algorithm. Starting from the actual test scheduling, this paper proposes the constraints of the job parallelization scheduling mode, solves the job parallelization scheduling problem under certain constraints, and gives the Gantt chart and dispatch table corresponding to the optimal solution. In the future wok, on the one hand, it is proposed to further study the relationship between the parallel path to further study the relationship between the parallel path of the parallelizable operation and the job shop scheduling discrete map, and directly judge whether the job parallelization can reduce the maximum completion time through the critical path. On the other hand, it is proposed to study the fast solution algorithm of test scheduling based on the optimal solution algorithm studied in this paper.

#### CONFLICT OF INTEREST

The authors declare no conflict of interest.

#### AUTHOR CONTRIBUTIONS

YuYu conducted the research, analyzed the data, wrote the paper and had approved the final version.

#### ACKNOWLEDGMENT

This work is supported by the National Key R&D Program of China under the Grant 2017YFC1405600, National Science Foundation for Young Scientists of China under the Grant 41706101, Qingdao source innovation plan (basic research on youth talent application) under the Grant 18-2-2-71-jch.

#### REFERENCES

- G. K. Rand, "Sequencing and scheduling: An introduction to the mathematics of the job-shop," *Journal of the Operational Research Society*, vol. 33, no. 9, pp. 862–862,1982.
- [2] T. F. Gonzalez and S. Sahni, "Open shop scheduling to minimize finish time," *Journal of the ACM*, vol. 23, no. 4, pp. 665–679, 1976.
- [3] P. Brandimarte, "Routing and scheduling in a flexible job shop by tabu search," *Annals of Operations Research*, vol. 41, no. 3, pp. 157–183, Sep 1993.
- [4] C. Rajendran and H. Ziegler, "Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs," *European Journal of Operational Research*, vol. 155, no. 2, pp. 426–438, 2004.
- [5] J. Adams, E. Balas, and D. Zawack, "The shifting bottleneck procedure for job shop scheduling," *Management Science*, vol. 34, no. 3, pp. 391–401, 1988.
- [6] F. Glover, "Tabu search part i," ORSA Journal on Computing, vol. 1, no. 3,1989.
- [7] F. Glover, "Tabu search part ii," *Informs Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1989.
- [8] M. Dellamico and M. Trubian, "Applying tabu search to the job-shop scheduling problem," *Annals of Operations Research*, vol. 41, no. 1, pp. 231–252, 1993.
- [9] J. W. Barnes and J. B. Chambers, "Solving the job shop scheduling problem with tabu search," *Iie Transactions*, vol. 27, no. 2, pp. 257–263, 1995.
- [10] P. P. Van Laarhoven, E. E. Aarts, and J. K. Lenstra, "Job shop scheduling by simulated annealing," *Operations Research*, vol. 40, no. 1, pp. 113–125, 1992.
- [11] M. Kolonko, "Some new results on simulated annealing applied to the job shop scheduling problem," *European Journal of Operational Research*, vol. 113, no. 1, pp. 123–136, 1999.
- [12] W. Ying and L. Bin, "Job-shop scheduling using genetic algorithm," Job-Shop Scheduling Using Genetic Algorithm - Systems, Man and Cybernetics, vol. 3, pp. 1994–1999, 1996.
- [13] T. Yamada and R. Nakano, "A genetic algorithm applicable to large-scale job-shop problems," *Reinhard Manner and Bernard Manderick*, pp. 283–292, 1992.
- [14] P. Brucker, B. Jurisch, and B. Sievers, "A branch and bound algorithm for the job-shop scheduling problem," *Discrete Applied Mathematics*, vol. 49, no. 1, pp. 107–127, 1994.
- [15] J. Carlier and E. Pinson, "An algorithm for solving the job-shop problem," *Management Science*, vol. 35, no. 2, pp. 164–176, 1989.
- [16] C. Y. Lee, "Scheduling: Theory, algorithms, and systems," *lie Transactions*, vol. 28, no. 8, pp. 695–698, 1996.
- [17] B. Giffler and G. L. Thompson, "Algorithms for solving production-scheduling problems," *Operations Research*, vol. 8, no. 4, pp. 487–503, 1960.
- [18] M. Perregaard and J. Clausen, "Parallel branchand-bound methods for thejob-shop scheduling problem," *Annals of Operations Research*, vol. 83, pp. 137–160,
- [19] W. B. Crowston *et al.*, "Probabilistic and parametric learning combinations of local job shop scheduling rules," ONR Research Memorandum, 1963.

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited (<u>CC BY 4.0</u>).



Yu Yu received the Ph.D. degree from Ocean University of China in 2014. From 2014 to 2019, he was an assistant researcher at Institute of Marine Instruments and Metrology, Qilu University of Technology. His research interests lie in the fields of multi-sensor data fusion, marine hydro meteorological observation and artificial intelligence.