International Journal of u- and e- Service, Science and Technology
Vol.6, No.5 (2013), pp.1-16
http://dx.doi.org/10.14257/ijunesst.2013.6.5.01

Concurrent Open-shop Scheduling Accurate Algorithm Research

YanMin Ma and Di Jin

School of computer and information engineering
Harbin University of Commerce
Harbin, China
mym654321@163.com

Abstract

According to constraint conditions of concurrent open-shop and its description in mathematical language, the integer programming model of this scheduling problem, applied to the IBM ILOG CPLEX OPTIMIZATION mathematical optimization software combined with the five groups of numerical examples verify the correctness of the established model, is set up. Integer programming model can not only use mathematical language to describe the scheduling problem, and if we make a traversal of the whole solution space in established model, then we can get the exact solution of the problem.

Keywords: *Open-shop; workpiece; concurrent open shop scheduling*

1. Introduction

Concurrent Open-Shop Scheduling Problem is an important research branch of Open-Shop Scheduling. It has wide applications in many aspects, such as modern transport and logistics industry, modern service industry, large-scale systematic maintenance industry, clothing industry, health care and so on [1]. Scheduling Problem Compared with the traditional open workshop, they have similar features. Their process is not bound by the orders, the difference between them is concurrent open-shop can make multiple machines work on a workpiece simultaneously, which has more feasible solution space therefore most of the concurrent open-shop scheduling problem is NP-complete problem that makes us cannot get accurate optimal solution in time. In recent years, artificial intelligence [2], computational intelligence, real-time intelligence and other research results have been applied to the solving process of Concurrent Open-Shop Scheduling Problem, which achieved remarkable results. Therefore, analysis of the Concurrent Open-Shop Scheduling Problem not only for plays an important role in promoting scheduling theory, but also has practical significance.

TEOFILO and SARTAJ [3] are considered the first two to put forward the definition of the open-shop scheduling and the open-shop scheduling application examples, and give the open workshop the detailed description, and introduce the interruptible and non-interruptible open-shop scheduling problem which proves that the two machine open-shop scheduling problem is polynomial complexity problem. Later scholars have found that most of the open-shop scheduling problem is NP problem; it can't give exact solution of the problem. Sergey V. Evastianov and Gerhard J. Oeginger used a polynomial time approximation algorithm to the solving of open-shop scheduling problem with the minimum manufacturing period.

Then many scholars proposed concurrent workshop model of the open-shop scheduling

Project Supported: social science research project of Heilongjiang province（12D056）; natural science foundation of Heilongjiang province(F201243); 2012 higher education teaching reform project of Harbin University of Commerce (201242)

ISSN: 2005-4246 IJUNESST

based on the open-shop scheduling. Concurrent open-shop scheduling model in 1979 was the earliest, [4] Baker put forward m completely machine parallel machine environment of workshop, and if the m value is an arbitrary value, will this problem markers for PD, m parallel machines in this paper refers to m machine can processing that can be together do concurrent processing. Recently, Roemer [5] for concurrent open-shop scheduling problem has carried on the classification of the different target summary, the complexity of the problem are given for different target.

In this paper, the concurrent open-shop problem is under description, and the integer programming model is established, in order to verify whether the class NP problem model is correct, two modeling methods are used; we will build model practical example application software for simulation. At the same time, the calculation used to record time, when the calculation time is too long, we will use other methods to solve this problem.

2. The Precise Solution to the Scheduling Problem

The enumeration method is the most intuitive accurate solution method, the applied enumeration method can be implement traversal for all solutions of the problems, finally it can get the optimal solution of the problem, but this kind of solving method for NP problems could not finish in limited time, so the enumeration method shall not be used in solving NP problem. Shop scheduling problem solving method generally uses mathematical programming method and combination of the optimized method, their core is to establish mathematical model of the problem, and find out the optimal solution of the model.

Mathematical programming method should make n related and quantifiable decisions into the Decision Variables(x1, x2,..., xn), the reasonable measure of the results is presented as a mathematical function of these Decision Variables, the function is called the objective function, restrictions of a scheduling model will be expressed by inequalities among Decision Variables, then it becomes the mathematical programming model, solving the scheduling problem can get the optimal solution. But this method will lead to a very large solution space, if you use computer calculation; calculated time is often unacceptable too, so this method can be used when solving small problems. When the problem is NP problem, the established model is just a representation of the scheduling problem, because it is difficult to give the optimal solution of the problem in limited time.

Using branch demarcation method and cutting plain method is an effective method to solve the established mathematical programming model, using these two methods can lead the solving of mathematical programming model solving towards a same goal, and it will reduce the unnecessary solving process, but in solving large-scale problems, it will also cost a long time.

3. Mathematical Model of Concurrent Open-Shop Scheduling Problem

Before the integer programming model of Open Concurrent workshop is set up, first of all, workshop model should be described in details. Machine and workpiece of Concurrent Open-Shop scheduling Problem with should satisfy the following constraints:

1) Workpiece in the processing process have no process constraints, it means workpieces can be processed on each machine-in accordance with any process.

2) Machines can't overlap from each other; each machine should process only one workpiece at most at any time.

3) Workpiece can overlap from each other; each workpiece is allowed to be processed on multiple machines at the same time.

4) Each workpiece should be on each machine processing time, processing time can be the various workpiece for each machine-is different; the same workpiece processing time on different machines may also be different.

Figure 3.1 (a) for job-shop schematic, workpieces are strictly accordance with the established process operation, the following processing closed until the processing before it finished, which the common one is flow shop; Figure 3.1 (b) for the traditional open workshop (dashed lines means that the workpiece can only choose one of the machine to process at the same time); Figure 3.1 (c) is a concurrent open-shop (the solid line means the workpiece can choose multiple machines at the same time). This paper wills introduce the following constraints on the basis of normal concurrent open-shop scheduling problem:

(1) Each piece has its own ready time, namely it's the time when workpiece can start processing, before the workpiece's ready time, don't allow the processing the workpiece.

(2) There is interference in the machine, the machine can process more than one workpiece at the same time, but due to machine's limitation with each other, some individual machines cannot process a workpiece at the same time, the interference between the machines can't process concurrently.

(3) The workpiece process is not allowed to interrupt, and once the processing started, the workpiece can't be removed utile the end of machine's processing.

(4) Ignore the workpiece's time of handling, removing, and clamping in the machine.

This section will set up mathematical model to concurrent open-shop according to the constraint condition above, and the exact solutions of scheduling problem can be obtained.

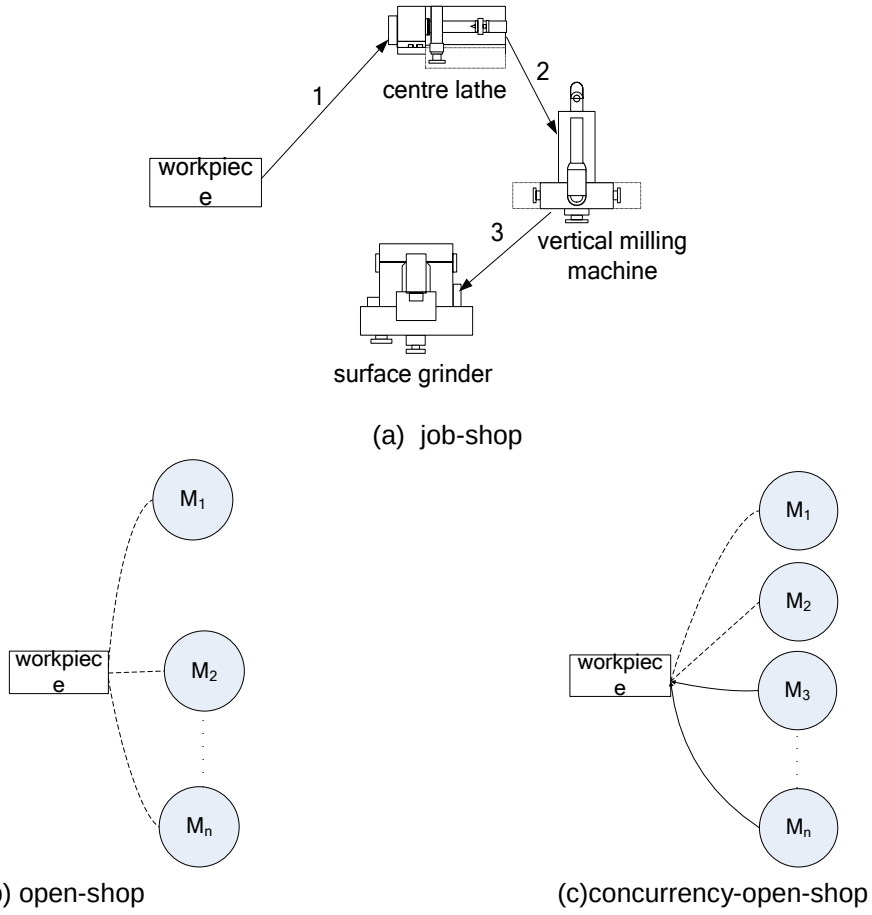

Figure 3.1. Shop-Models

3.1. Modeling Method Introduction

In this paper, the concurrent open shop scheduling problem uses integer programming model to solve, integer programming method in operational research is a common method to solve the problem of resource allocation. [7], by application of equality and inequality relation of resource constraints about decision variables, the decision variables can be workpiece ready time, finish time and start processing time, the objective function can be expressed as the effect of reasonable measurement which is expressed as a mathematical function of the decision variables, the range of relationship between decision variables and objective function establishes the integer programming model of the problem. The establishment of the inequality relationship requires to cover demand-constraints completely cover, which is help to obtain accurate solutions, but in the model, the establishment of constraint cannot have overlapping part, overlapping will lead to subsequent solution become harder, which makes the solving efficiency reduced.

Before establishing mathematical model, we introduce symbols which are commonly used in integer programming model of scheduling problem, and explain. Concurrent open-shop scheduling problem, in this paper, can be used sign language to describe as follows: there is m dedicated machines M, respectively, and m1,m2,…,m_m, this m machine used for processing n workpiece-J, expressed as j1,j2,…,j_n. Every workpiece has to process on all the machines, so, each of which has m processes, but there is no processing order constraints, workpiece-j_j on the mi processing expressed as o_{ij} for i= 1, 2,... , n, j = 1, 2,...m. Each workpiece-j_j has its own ready time-r_j. Each workpiece's processing time in a machine is p_{ij}, which means that the workpiece-j needs time- p_{ij} on machine-i, S_{ij}represented as the start processing time of workpiece-j on the machine-i, C_{ij}represented as completion time of workpiece-j in the machine-i.

3.2. Integer Programming model based on Workpiece's Completion Time

Workpiece completion time variable integer programming model is based on the workpiece completed time nodes, then we establish relationship between relevant workpiece completion time and processing time, the relationship between the different constraint conditions should be established by the unequal relationship between the workpiece completing time node and the beginning time of processing, such as workpiece-j should be processed after the workpiece-k, establishing the constraint relations with completion time variables can be expressed as:

$$c_{ik} \geq \max(s_{ik}, c_{ij}) + p_{ik} ,$$

Inequalities are given work-piece k's completion time to greater than or equal to j's completion time which is ahead of k, and the work-piece ready time in the larger value of k, and sum of the processing time of k.

There is no accurate solution method for solving multi-objective problems now, thus for validating model, we can only model and solve to the target of the minimized manufacturing duration. The integer programming model according to the completion time variables is set up as follows, notes for IP3-1:

$$\text{minimize } \max c_{ij} \quad \forall j \in (1,2,...,n), \forall i \in (1,2,...,m) \tag{3-1}$$

$$c_{ij} \geq c_{i'j} + p_{i'j} \text{或} c_{i'j} \geq c_{ij} + p_{ij} \quad \forall j \in (1...n), \forall i \in (1...m), \ i, i' \in (1...m) \text{且} i \neq i' \tag{3-2}$$

$$c_{ij} \geq c_{ik} + p_{ij} \text{或} c_{ik} \geq c_{ij} + p_{ik} \quad \forall j,k \in (1...n) \text{且} j \neq k, \forall i \in (1...m) \tag{3-3}$$

$$s_{ij} \geq r_j \qquad \forall j \in (1...n), \forall i \in (1...m) \tag{3-4}$$

$$c_{ij} = s_{ij} + p_{ij} \qquad \forall j \in (1...n), \forall i \in (1...m) \tag{3-5}$$

$$c_{ij} \text{ integer} \qquad \forall j \in (1...n), \forall i \in (1...m) \tag{3-6}$$

$$s_{ij} \text{ integer} \qquad \forall j \in (1...n), \forall i \in (1...m) \tag{3-7}$$

The following details are explanations to the integer programming model, using completion time variable, of concurrent open-shop:

Formula (3-1) is the solving goal of the integer programming model, it tends to minimize the maximum completion time, and minimize the manufacturing period. Constraints (3-2) said machine-i' can't process the same workpiece with the machine-i, it also means that machine-i' and machine-i are the intrusive machine. Inequality (3-3) said the completion time of workpiece-j on the machine-i is greater than the sum of completion time of workpieces before workpiece-j on machine-i and the completion time of workpiece-j, it also means that the completion time of workpiece-k after workpiece-j is greater than the sum of completion time of workpiece-j and the completion time of workpiece-k, inequality (3-3) tends to meet the constrains of concurrent open-shop scheduling problem: a machine should only process one workpiece ta most at the same time; inequality (3-4) said the processing time of workpieces in the beginning should be greater than the ready time. Constraints (3-5) said the completion time of workpieces should be equal to the sum of the beginning time of processing and the completion time of workpiece, at the same time it also ensures the processing is non-interruptible, once the processing starts, we shouldn't remove workpieces until the processing completes. Formula (3-6) and (3-7) requires workpiece the beginning time of processing and completion time should be integer.

3.3. The Integer Programming Model based on Workpiece's Processing Order

Sequential variables namely accords to the workpiece's processing sequence, 0-1 integer programming is applied to describe the workpiece's processing sequence. Each workpiece constraints' existence relationship represents with 0 and 1 switch function. There are constraints, if open the switch function, the function value is 1, otherwise, the constraint does not exist, the switch is closed, and the function value is 0.

Introducing the switching function-δ_{ijk}, when the δ_{ijk} = 1, it represents that work-j processes on the machine-i, ahead of workpiece-k; when δ_{ijk} = 0, it represents that workpiece-k processes on the machine-i, ahead of workpiece-j, this variable is called sequential variables, through this variables which can reflect the workpiece's processing sequence. The use of this variable modeling problem can be established in based on the idea of workpiece in the machine processing order.

Integer programming model of aiming at minimize the manufacturing period is as follows, notes for IP3-2:

$$\text{minimize max } c_{ij} \qquad \forall j \in (1...n), \forall i \in (1...m) \tag{3-8}$$

$$c_{ij} \geq \left(c_{ik} + p_{ik}\right)\delta_{ijk} \qquad \forall j,k \in (1...n) \text{且} j \neq k, \forall i \in (1...m) \tag{3-9}$$

$$\delta_{ijk} + \delta_{ikj} = 1 \qquad \forall j,k \in (1...n) \text{且} j \neq k, \forall i \in (1...m) \tag{3-10}$$

$$c_{ij} \geq c_{i'j} + p_{i'j} \text{或} c_{i'j} \geq c_{ij} + p_{ij} \quad \forall j \in (1...n), \forall i \in (1...m), \quad i' \in (1...m) \text{且} i \neq i' \quad (3\text{-}11)$$

$$s_{ij} \geq r_j \quad \forall j \in (1...n), \forall i \in (1...m) \quad (3\text{-}12)$$

$$c_{ij} = s_{ij} + p_{ij} \quad \forall j \in (1...n), \forall i \in (1...m) \quad (3\text{-}13)$$

$$c_{ij} \text{ integer} \quad \forall j \in (1...n), \forall i \in (1...m) \quad (3\text{-}14)$$

$$s_{ij} \text{ integer} \quad \forall j \in (1...n), \forall i \in (1...m) \quad (3\text{-}15)$$

Above all, the following details are explanations to the integer programming model of concurrent open-shop:

Formula (3-8) is the solving object of this integer programming model, also, is the minimization of maximum-completion-time, equally, is the minimization of manufacturing-period. Formula (3-9) represents that the beginning processing time of workpiece-j on the machine-i is greater than or equal to the finishing time of any workpieces (like, workpiece-k) that is ahead of workpiece-j, in the machine-i and the sum of the processing completion time of workpiece-j. Formula (3-10) represents two single-natural-orders limitation on machine-i, namely, two workpieces on a machine with only one processing order. Formula (3-11) represents that machine-i' can't process the same workpiece with other machines at the same time, the machines- i' is interference. Formula (3-12) represents that workpiece's processing beginning time is greater than the workpiece's ready time. (3-13) said workpiece's completion time is equal to the processing beginning time plus the total processing time. Formula (3-14), (3-15) requires that processing beginning time and completion time of workpiece are integer.

4. Integer Planning Model Validation

This model validation method based on computer simulation experiment, and applied to mathematical programming software. A given number of examples to integer programming model are the validation method which uses application of mathematical programming software for example and makes analysis to the results. Then, we could make sure whether the results of numerical example meet the constraint conditions of model, and validate the correctness of mathematical model.

4.1. The Simulation for Integer Programming Model

The search policy of CPLEX software's model solution is points branch defined law, the method can reduced search space, relatively, compared to other heuristic algorithm searching for the optimistic solution. This section will use five-group cases to simulate the integer planning model IP3-1 and IP3-2 of concurrent open-workshop scheduling , data of cases takes randomly, the matrix scales of workpiece processing time window, respectively, are 4*3, 5*4, 6*4, 6*6, 7*5. According to the constraints of model, we write CPLEX program, and give the examples of processing-time's matrix, respectively, into a simulation, simulation results data unit is minutes.

Example one:

A given instance of 4 pieces and 3 machines is in example one, and given processing time matrix shows in Table 4.1, additional constraints that machine 3 and machine 2 cannot be concurrent processing, table data unit is in minutes.

Table 4.1. Example-processing-time's Matrix

	machine1	machine 2	machine 3	ready time
workpiece 1	3	7	5	1
workpiece 2	1	4	7	2
workpiece 3	8	2	6	1
workpiece 4	4	9	4	0

Integer programming model for application IP3-1 establishes simulation results of manufacturing up to 23 minutes, start processing time of workpiece in the machine-is shown in Table 4.2, and completion time of workpiece in the machine-is shown in Table 4.3.

Table 4.2. Workpiece Process beginning Time

	machine 1	machine 2	machine 3
workpiece 1	1	16	4
workpiece 2	8	3	15
workpiece 3	15	1	9
workpiece 4	9	7	0

Table 4.3. Workpiece Completion Time

	machine1	machine2	machine3
workpiece 1	4	23	9
workpiece 2	9	7	22
workpiece 3	23	3	15
workpiece 4	13	16	4

Integer programming model for application IP3-2 establishes simulation results of manufacturing up to 23 minutes; start processing time of workpiece in the machine-is shown in Table 4.4, and completion time of workpiece in the machine-is shown in Table 4.5.

Table 4.4. Workpiece began Processing Time

	machine1	machine2	machine3
workpiece 1	4	23	9
workpiece 2	9	7	22
workpiece 3	23	3	15
workpiece 4	13	16	4

Table 4.5. The Workpiece Completion Time

	machine1	machine2	machine3
workpiece 1	1	16	4
workpiece 2	8	3	15
workpiece 3	15	1	9
workpiece 4	9	7	0

Example two:

A given instance of 5 pieces and 4 machines is in example two, and given processing time matrix shows in Table 4.6, additional constraints that machine 3 and machine 2 cannot be concurrent processing, table data unit is in minutes.

Table 4.6. Example-processing-time's Matrix

	machine1	machine2	machine3	machine4	ready time
workpiece 1	3	7	5	4	1
workpiece 2	1	4	7	2	2
workpiece 3	8	2	6	5	4
workpiece 4	2	6	9	2	3
workpiece 5	4	6	3	9	2

Integer programming model for application IP3-1 establishes simulation results of manufacturing up to 31 minutes, start processing time of workpiece in the machine-is shown in Table 4.7, and completion time of workpiece in the machine-is shown in Table 4.8.

Table 4.7. Workpiece Process Beginning Time

	machine1	machine2	machine3	machine4
workpiece 1	23	24	1	23
workpiece 2	4	3	15	6
workpiece 3	10	7	25	8
workpiece 4	20	18	6	27
workpiece 5	5	11	22	13

Table 4.8. Workpiece Completion Time

	machine1	machine2	machine3	machine4
workpiece 1	26	31	6	27
workpiece 2	5	7	22	8
workpiece 3	18	9	31	13
workpiece 4	22	24	15	29
workpiece 5	9	17	25	22

Integer programming model for application IP3-2 establishes simulation results of manufacturing up to 31 minutes; start processing time of workpiece in the machine-is shown in Table 4.9, and completion time of workpiece in the machine-is shown in Table 4.10.

Table 4.9. Workpiece Began Processing Time

	machine1	machine2	machine3	machine4
workpiece 1	23	24	1	23
workpiece 2	4	3	15	6
workpiece 3	10	7	25	8
workpiece 4	20	18	6	27
workpiece 5	5	11	22	13

Table 4.10. The Workpiece Completion Time

	machine1	machine2	machine3	machine4
workpiece 1	26	31	6	27
workpiece 2	5	7	22	8
workpiece 3	18	9	31	13
workpiece 4	22	24	15	29
workpiece 5	9	17	25	22

Example three:

A given instance of 6 pieces and 4 machines is in example three, and given processing time matrix shows in Table 4.11, additional constraints that machine 3 and machine 2 cannot be concurrent processing, table data unit is in minutes.

Table 4.11. Example-Processing-Time's Matrix

	machine1	machine2	machine3	machine4	ready time
workpiece 1	3	7	5	2	1
workpiece 2	1	4	7	3	7
workpiece 3	8	2	6	4	4
workpiece 4	2	6	9	5	3
workpiece 5	1	2	4	6	5
workpiece 6	2	7	9	3	4

Integer programming model for application IP3-1 establishes simulation results of manufacturing up to 41 minutes, start processing time of workpiece in the machine-is shown in Table 4.12, and completion time of workpiece in the machine-is shown in Table 4.13.

Table 4.12. Workpiece Process Beginning Time

	machine1	machine2	machine3	machine4
workpiece 1	18	13	1	26
workpiece 2	14	26	30	14
workpiece 3	23	21	15	6
workpiece 4	15	32	6	17
workpiece 5	35	30	37	31
workpiece 6	38	4	21	10

Table 4.13. Workpiece Completion Time

	machine1	machine2	machine3	machine4
workpiece 1	21	20	6	28
workpiece 2	15	30	37	17
workpiece 3	31	23	21	10
workpiece 4	17	38	15	22
workpiece 5	36	32	41	37
workpiece 6	40	11	30	13

Integer programming model for application IP3-2 establishes simulation results of manufacturing up to 41 minutes; start processing time of workpiece in the machine-is shown in Table 4.14, and completion time of workpiece in the machine-is shown in Table 4.15.

Table 4.14. Workpiece Began Processing Time

	machine1	machine2	machine3	machine4
workpiece 1	18	13	1	26
workpiece 2	14	26	30	14
workpiece 3	23	21	15	6
workpiece 4	15	32	6	17
workpiece 5	35	30	37	31
workpiece 6	38	4	21	10

Table 4.15. The Workpiece Completion Time

	machine1	machine2	machine3	machine4
workpiece1	21	20	6	28
workpiece2	15	30	37	17
workpiece3	31	23	21	10
workpiece4	17	38	15	22
workpiece5	36	32	41	37
workpiece6	40	11	30	13

Example four:

A given instance of 6 pieces and 4 machines is in example four, and given processing time matrix shows in Table 4.16, additional constraints that machine 3 and machine 2 cannot be concurrent processing, table data unit is in minutes.

Table 4.16. Example-processing-time's Matrix

	machine1	machine2	machine3	machine4	machine5	machine	ready time
workpiece1	6	7	5	3	9	4	1
workpiece2	2	8	0	8	1	3	2
workpiece3	7	3	6	5	4	5	0
workpiece4	2	5	9	1	8	7	3
workpiece5	5	2	9	4	7	9	2
workpiece6	4	2	8	3	1	7	1

Integer programming model for application IP3-1 establishes simulation results of manufacturing up to 37 minutes, start processing time of workpiece in the machine-is

shown in Table 4.17, and completion time of workpiece in the machine-is shown in Table 4.18.

Table 4.17. Workpiece Process Beginning Time

	machine1	machine2	machine3	machine4	machine5	machine6
workpiece1	21	29	6	20	12	13
workpiece2	12	13	6	26	2	33
workpiece3	29	9	0	14	22	8
workpiece4	4	21	11	25	28	26
workpiece5	7	5	28	6	3	17
workpiece6	14	2	20	1	1	1

Table 4.18. Workpiece Completion Time

	machine1	machine2	machine3	machine4	machine5	machine6
workpiece1	27	36	11	23	21	17
workpiece2	14	21	6	34	3	36
workpiece3	36	12	6	19	26	13
workpiece4	6	26	20	26	36	33
workpiece5	12	7	37	10	10	26
workpiece6	18	4	28	4	2	8

Integer programming model for application IP3-2 establishes simulation results of manufacturing up to 37 minutes; start processing time of workpiece in the machine-is shown in Table 4.19, and completion time of workpiece in the machine-is shown in Table 4.20.

Table 4.19. Workpiece Began Processing Time

	machine1	machine2	machine3	machine4	machine5	machine6
workpiece1	20	11	6	18	12	1
workpiece2	18	3	37	2	11	19
workpiece3	7	31	0	13	28	31
workpiece4	15	22	28	10	3	5
workpiece5	2	27	11	21	21	22
workpiece6	29	1	20	29	1	12

Table 4.20. The Workpiece Completion Time

	machine1	machine2	machine3	machine4	machine5	machine6
workpiece1	26	18	11	21	21	5
workpiece2	20	11	37	10	12	22
workpiece3	14	34	6	18	32	36
workpiece4	17	27	37	11	11	12
workpiece5	7	29	20	25	28	31
workpiece6	33	3	28	32	2	19

Example five:

A given instance of 7 pieces and 5 machines is in example five, and given processing time matrix shows in Table 4.21 additional constraints that machine 3 and machine 2 cannot be concurrent processing, table data unit is in minutes.

Table 4.21. Example-Processing-Time's Matrix

	machine1	machine2	machine3	machine4	machine5	ready time
workpiece1	2	7	6	4	6	1
workpiece2	1	4	7	2	3	2
workpiece3	8	2	6	3	1	4
workpiece4	4	9	4	6	7	1
workpiece5	3	2	8	2	2	3
workpiece6	5	6	4	1	4	4
workpiece7	3	9	4	6	4	7

Integer programming model for application IP3-1 establishes simulation results of manufacturing up to 40 minutes, start processing time of workpiece in the machine-is shown in Table 4.22, and completion time of workpiece in the machine-is shown in Table 4.23.

Table 4.22. Workpiece Process Beginning Time

	machine1	machine2	machine3	machine4	machine5
workpiece1	1	1	27	35	20
workpiece2	3	19	33	23	11
workpiece3	4	17	21	27	14
workpiece4	34	25	1	16	28
workpiece5	30	23	9	4	4
workpiece6	16	34	5	22	15
workpiece7	27	8	17	10	7

Table 4.23. Workpiece Completion Time

	machine1	machine2	machine3	machine4	machine5
workpiece1	3	8	33	39	26
workpiece2	4	23	40	25	14
workpiece3	12	19	27	30	15
workpiece4	38	34	5	22	35
workpiece5	33	25	17	6	6
workpiece6	21	40	9	23	19
workpiece7	30	17	21	16	11

Integer programming model for application IP3-2 establishes simulation results of manufacturing up to 40 minutes; start processing time of workpiece in the machine-is shown in Table 4.24, and completion time of workpiece in the machine-is shown in Table 4.25.

Table 4.24. Workpiece Began Processing Time

	machine1	machine2	machine3	machine4	machine5
workpiece1	1	1	27	35	20
workpiece2	3	19	33	23	11
workpiece3	4	17	21	27	14
workpiece4	34	25	1	16	28
workpiece5	30	23	9	4	4
workpiece6	16	34	5	22	15
workpiece7	27	8	17	10	7

Table 4.25. The Workpiece Completion Time

	machine1	machine2	machine3	machine4	machine5
workpiece1	3	8	33	39	26
workpiece2	4	23	40	25	14
workpiece3	12	19	27	30	15
workpiece4	38	34	5	22	35
workpiece5	33	25	17	6	6
workpiece6	21	40	9	23	19
workpiece7	30	17	21	16	11

4.3. Analysis of Simulation Data

From the results of the simulation, solutions of simulation results meet needs of constraints of open-shop scheduling, matrix processing time requirements are met by the workpiece, workpiece began processing time is greater than or equal workpiece's ready time, the machine can process the same workpiece concurrently, interference machines are not able to process in parallel, each machine can process one workpiece parallel without interruption. The simulation result of model based on completion time variable and processing sequence variable is consistent, which has proven that the correctness of them, and means that all of them can be applied to setup model to solve the open-shop scheduling problem.

But the two methods, when solving the same example, computing performance is different, following is a comparison of the performance of models, from calculations time, number of constraints, to solving speed based on same variables.

From the data completed by CPLEX, the time variable modeling to solve five groups example takes respectively 0.02 s, 1.95 s, 7.05 s, 48.7 s, 11678.1 s, processing order variable modeling to solve five groups example takes respectively 0.05 s, 5.91 s, 23.07s, 105.6s, 49865.6s, five groups example produces variable number-24,40,48,60,70-respectively, and constraints of IP3-1 model are 156, 325, 462, 574, 784, respectively, the constraints of IP3-2 model are 288, 485, 702, 1028, 1204 , respectively, the calculation speed of IP3-1 model are 17948.4, 22114.8, 22737.4, 16575, 17723, the calculation speed of IP3-2 model are 5982.9, 7313.1, 6948.8, 5176.6, 4150.6, the unit is: the solution/s. Contrast diagram of variables establishment and computer calculation time is shown in Figure 4.1, contrast diagram of the variables establishment and the constraints the model produces is shown in Figure 4.2, contrast diagram of variables establishment and solving speed is shown in Figure 4.3

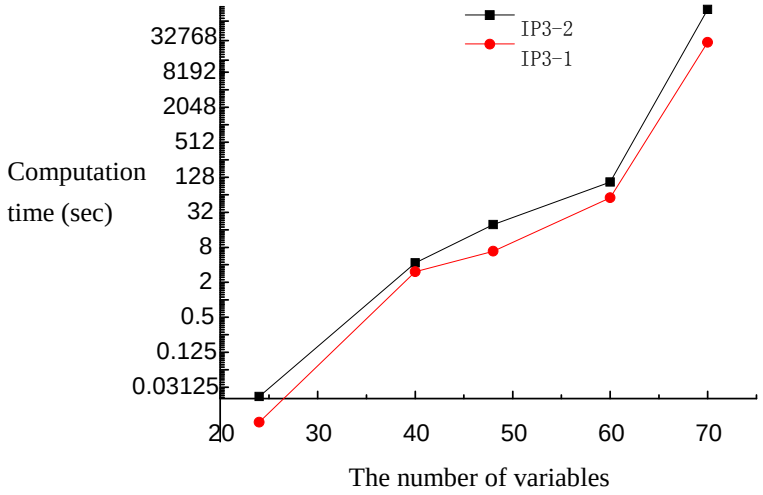

Figure 4.1. Time Comparison of Two Modeling Methods of Calculation

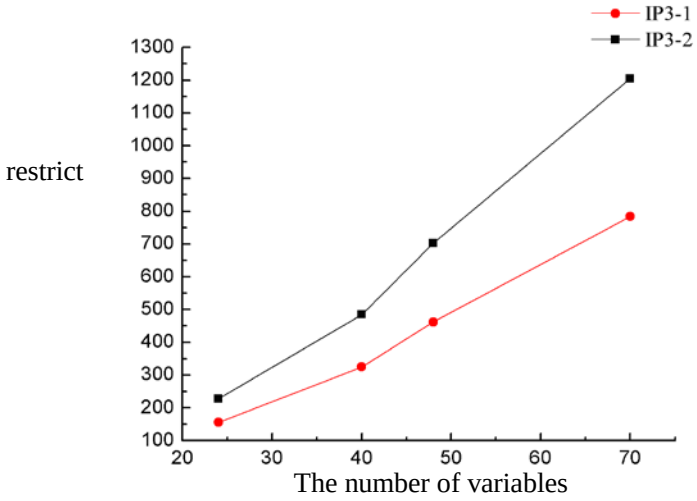

Figure 4.2. Constraints Comparisons of Two Kinds of Modeling

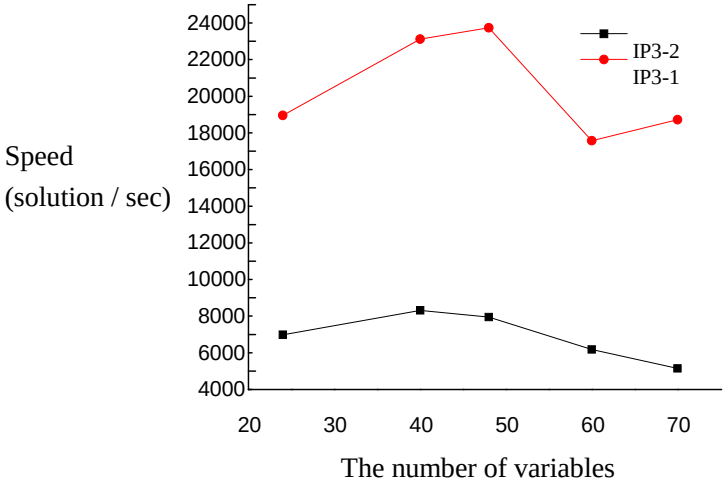

Figure 4.3. Speed Comparisons of Two Kinds of Modeling Method

From chart of contrast we can see, the performance of model IP3-1 using completed time variable is better than model IP3-2 using processing variable. Compared with model IP3-1, the increasing rate of model IP3-2' time consuming grows more obviously, while problem scale increases, under general situation, the constraints created by model IP3-1 is less than model IP3-2's, and on calculation speed, IP3-1's calculation speed is significantly higher model IP3-2. From these comparative analyses of open-shop scheduling, the integer programming model using workpiece-completion-time variable is preferable to integer programming model using processing sequence variable.

5. Summary

This paper on the integer programming model is to solve open-shop scheduling. Model uses two kinds of modeling methods, respectively, according to the time variable and the variable according to the processing order, also, is checked by the IBM CPLEX software for simulation which uses the five groups of numerical example and simulation results are given, which, respectively, shows that the modeling is correct. From the contrast of process parameters of the simulation, we can see that the computational time complexity of modeling on completion time variable is less than using modeling on processing order variable. But from the overall results, when the problem size increases gradually, using precise solution will not be able to figure out the solution of problems in a relatively short time. So, to the large-scale concurrent open-shop scheduling problem, the precise solution is not available.

References

[1] J. Di, M. Yanmin, F. ZhiPeng and F. Xiaolin, "A RFID anti-collision algorithm based on multithread regressive-style binary system", 2012 International Conference on Measurement, Information and Control, **(2012),** pp. 365-369.
[2] Y. M. Ma and D. Jin, "Anti-collision Algorithm based on Multi-threaded RFID Lock Position", International Journal of Hybrid Information Technology, vol. 6, no. 3, **(2013)**, pp. 95-104.
[3] M. Modarres and M. Ghandehari, "Applying circular coloringto open shop scheduling", Scientia Iranica, vol. 15, no. 5, **(2008)**, pp. 652-660.
[4] S. Ghosh Dastidar and R. Nagi, "Batch splitting in an assembly scheduling environment", Production Economics, vol. 105, no. 2, **(2007)**, pp. 372-384.
[5] R. L. Graham, E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling:A Survey. Annals of Discrete Mathematics", vol. 5, **(1979)**, pp. 287-326.
[6] C. S. Sung and H. Ah Kim, "A two-stage multiple-machine assembly scheduling problem for minimizing sum of completion times", Production Economics, vol. 113, **(2008)**, pp. 1038-1048.
[7] T. Gonzale and S. Sahni, "Open shop scheduling to minimize finish time", Journal of the Assooauon for Computing Machiner, vol. 23, no. 4, **(1976)**, pp. 665-679.

