TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.5, May 2014, pp. 4024 ~ 4029
DOI: http://dx.doi.org/10.11591/telkomnika.v12i5.5093

The Hybrid Flow Shop Scheduling with Special Time Constraints

Xiaobing Cheng*[1], Mingping Xia[2], Xiuying Wang[3] Yongjun Xiao[4]
[1,2,3]College of Automation, Beijing Union University, Beijing, P.R.China, 100101
[4]China Electronic Information Industry Development Research Institute, Beijing, 100036
*Corresponding author, e-mail: xiaobingcheng@sina.com[1], zdhtmingping@buu.edu.cn[2], zdhtxiuying@buu.edu.cn[3], yxiao1979@126.com[4]

Abstract

A constraint satisfaction optimization model is established for just-in-time HFS scheduling problem with limited waiting time. Considering the problem's complicated characteristic of having binary variables, this paper decomposes it into a Multiple Capacitated Flow Shop Scheduling (MCFS) problem and a machine allocation problem. During the process of solving the MCFS problem, a local search is embedded into the procedure of constraint satisfaction optimization so as to improve the convergence of the algorithm. Data experiments show that both the model and algorithm are feasible and effective.

Keywords: *hybrid flow shop (HFS), just in time, limited waiting time, constraint satisfaction optimization*

1. Introduction

Just-in-time HFS scheduling problem with limited waiting time exists widely in the metallurgical, petrochemical and pharmaceutical process industry production management practice. For example, in the steel-making and continuous casting process of the iron and steel enterprises the casting machine has the strict requirements on the steel composition and temperature, and therefore it is not allowed that the molten steel in high temperature has a long waiting time in the process, in order to avoid the decreased temperature affecting the quality of the molten steel. In addition, continuous cast constraints of continuous casting machine has punctual requirements to arrival of the furnace, namely the refined molten steel must on time arrive continuous casting machine when continuous casting machine complete a casting furnace, in order to avoid casting break and loss of the heat waiting.

In theory, HFS scheduling problem is NP-problem, so just-in-time HFS scheduling problem with limited waiting time is also NP-problem. Some related issues were discussed in literature: literature [1-3] explored different methods for solving general HFS scheduling, including operations research methods based on accurate mathematical model and artificial intelligence methods to obtain a satisfactory solution for the purpose; literature [4] overviewed the computational complexity of the no waiting and blocking scheduling problem; literature [5] explored the performance of the no-wait flow shop scheduling algorithm; literature [6] put forward the minimum deviation algorithm for two-stage no-wait HFS scheduling problem; literature [7] established mathematical model to minimize waiting time and earliness/tardiness punishment for the optimization goal for the steel-making-continuous casting production scheduling problem, and put forward practical algorithm based on genetic algorithm and linear programming.

In this paper, using constraint satisfaction optimization method [8] solves just-in-time HFS scheduling problem with limited waiting time. First, we build a constraints of the problem meet the optimization model; then the original problem is decomposed into multi-capacity flow shop scheduling constraints to meet the optimization problem and machine assignment constraint satisfies problem; then we put forward a method based on neighborhood search constraints to meet optimization algorithm; at last we use data experiments to verify the effectiveness of the model and algorithm.

Received November 2, 2013; Revised December 26, 2013; Accepted January 16, 2014

2. Problem Description and Model

Just-in-time HFS scheduling problem with limited waiting time refers to the N work performing process along the same direction through s stages, and each stage has $l_j \geq 1$ (j=1, 2, ... , s) with the same speed parallel machine, wherein at least one phase exists the parallel machine. It is known that the released time of each job i is rd_i and the delivery time is dd_i, the processing time of the j-th stage process (operation) o_{ij} is p_{ij}, as well as to allow the maximum time w_j in the adjacent phase waiting, and requirements determine starting time and processing machinery of the work at various stages under meeting the assumptions and constraints, making the sum of job completion time earliness/tardiness penalties minimize. If the start time of the operation and processing machine mapping is variable V, the optional starting time of operation and processing machine mapping is the range of variable D, the constraints mapping is the constraints set C, and the goal of problem is expressed as a function F, so the scheduling problem can be transformed into the constraint satisfaction optimization problem (Constraint Satisfaction Optimization Problem, CSOP) Θ=(V, D, C, F).

To establish constraint satisfaction optimization model for convenience, we introduce symbols and variables: i is the work number, $i \in I=\{1, 2, ..., n\}$; j is the procedure number, $j \in J=\{1, 2, ..., s\}$; o_{ij} is the j-th process of job i, also known as operating o_{ij}; p_{ij} is the processing time of operation o_{ij}; c_{ij} is completion time of the manipulate o_{ij}; rd_i is the release period of the job i; dd_i is the delivery time of job i; w_j is maximum wait time of work between the j and j +1 phase; $C1_i$ is per unit time penalty of completed ahead of schedule of the job i; $C2_i$ is the tardiness completion per unit time punishment of work i; s_{ij} is start time of the operation o_{ij}; m_{ij} is the processing machine of the operation o_{ij}.

CSOP model of constraint satisfaction optimization model is:

$$[M1]\min\left(\sum_{i\in I}\max\left\{c1_i(dd_i - c_{is})\right\} + \sum_{i\in I}\max\left\{c2_i(c_{is} - dd_i),0\right\}\right) \tag{1}$$

S.T. $$c_{ij} = s_{ij} + p_{ij}, i \in I, j \in J \tag{2}$$

$$s_{ij+1} \geq c_{ij}, i \in I, j \in \{1,2,...,s-1\} \tag{3}$$

$$(m_{i_1 j} \neq m_{i_2 j}) \vee (s_{i_1 j} \geq c_{i_2 j} \vee s_{i_2 j} \geq c_{i_1 i}), i_1, i_2 \in I, i_1 \neq i_2, j \in J \tag{4}$$

$$s_{ij+1} - c_{ij} \leq w_j, i \in I, j \in \{1,2,...,s-1\} \tag{5}$$

$$s_{ij} \geq rd_i, i \in I, j \in J \tag{6}$$

$$m_{ij} \in R_j = \{r_{j1}, r_{j2}, ..., r_{jl_j} \mid j \in J\}, i \in I \tag{7}$$

Among them, the objective function formula (1) shows the earliness/tardiness penalty sum of minimizing work; the formula (2) shows once you start processing the work, it is not allowed to be interrupted; the formula (3) is the timing constraints of the operation, which means when the last stage ends , we can start the processing of the next stage; the formula (4) is the disjunction constraint of the operation, which means it is impossible that the two work performs processing in any one machine at the same time; the formula (5) shows the work cannot exceed the upper limit in the waiting time of the adjacent stage; the formula (6), (7) denote the start time variables of the operation and the variables range of processing machine.

3. Method for Solving

Because the parallel exist in the HFS environment, the CSOP model of scheduling problem exists with operating o_{ij} determining the binary variables: starting time variable s_{ij} and processing machine variable m_{ij}. Aiming at this characteristic, the HFS Scheduling problem is decomposed into two sub-problems to solve: (1) ignore the assigned process of the machine of the work in each phase, and just-in-time HFS scheduling problem with limited waiting time is simplified to the corresponding more ability flow shop Scheduling (Multiple Capacitated Flow Shop Scheduling, MCFS) problem, and use constraint satisfaction optimization algorithm to obtain MCFS scheme; (2) by solving machine assignment problem to determine processing machine of the operation, to ensure operations of assigned to the same machine satisfy the corresponding disjunctive constraints.

3.1. MCFS Problem

Learn from the literature [9] we get the definition of the multi ability to Job Shop, MCFS problem can be considered to be a special kind of flow shop scheduling problem. It is different from the assumption of any machine at most processing a work at the same time in the general flow shop scheduling problem, and the machines in the MCFS problem within the capacity of the machine (resource) process several works at the same time.

In order to accurately establish just-in-time MCFS problem model with limited waiting time, we introduce the discrete time t and the Boolean variable B_{ijt} :

$$B_{ijt} = \begin{cases} 1, s_{ij} \le t < c_{ij}, \forall t \in \{s_{ij} \mid \forall i \in I, \forall j \in J \mid\} \\ 0 \end{cases}$$

CSOP model of MCFS problem is:

[M2] (1)
S.T. (2), (3), (5), (6)

$$\sum_{i \in I} B_{ijt} \le |R_j|, \forall j \in J, \forall t \in \{s_{ij} \mid \forall i \in I, \forall j \in J\} \quad (8)$$

In the above model, the left item of the formula (8) means resource load of stage j at a time t phase, and the right term represents the resource capability of stage j. The style is a machine resource capacity constraints, which means that at any time in stage j processing work number does not exceed its available resources $|R_j|$.

The paper refers to remaining resource capacity analysis method that the literature [9] uses, and the combined model is for the characteristics of the target to minimize the earliness/tardiness punishment, first the structure feasible initial scheduling on the time by the relaxation resource capacity constraint; Then use the constraint consistency technique to repair the resource conflicts, and get completely feasible scheduling; in order to get a better solution in a short period of time, embed neighborhood search in restarting the search mechanism of constraint satisfaction optimization to make up for its lack in convergence.

3.1.1. Generate Feasible Solution

(1) Initialization scheduling

The initialization scheduling without considering the resource capacity constraints (8), according to delivery time of the work and the timing constraints of the operation, calculate the optimal start time of each operation:

$$s_{ij} = dd_i - \sum_{k=j}^{s} p_{ik}, \forall i \in I, j \in J \quad (9)$$

We call that the time is the relaxation (resource capacity constraints) optimal start time, and this time scheduling is said feasibly initial scheduling on the time. If the scheduling meets all of resource capacity constraints, it is a no earliness / tardiness optimal final scheduling; otherwise, according to the resource capacity constraints of problem, based on satisfying the timing constraints, adjust the starting time to the operation.

(2) Constraint consistency

Constraint consistency implementation includes constraint checking and constraint propagation and it is the core technology of the CSP solving process. Constraint consistency implementation aims at discovering and eliminates conflicts due to improper variable assignment in MCFS problem solving process: (1) the timing conflict due to the operation starting time violating timing constraints; (2) resource load exceeds the resource conflict that the resource ability causes. Timing conflict is examined by the constraint Equation (3) and directly according to the Equation (3) trim the starting time of the operation and be cancelled. The resource conflict is checked by the remaining resource ability function (10):

$$RC\,(j,t) = |R_j| - \sum_{i \in I} B_{ijt}, \forall j \in J, \forall t \in \{s_{ij} \mid \forall i \in I, \forall j \in J\} \tag{10}$$

If the RC (j, t) <0, it means that the resource load exceeds the resource capacity at the stage j time t, it is need to trim the starting time of the operation to reduce the resource load. The basic idea to trim operation\tarting time is that we add timing constraints between the operations at occupied t time resource, and select the operation by the forward-looking and backtracking and trim the start time. Based on the constraint consistency technical resource collision repair process is as follows:

Step 1: Add operation of t(t=S$_{ij}$) time to the conflict operation set $\cos_{jt}$:

$\cos_{jt} = \{o_{pj} \mid s_{pj} \le t < c_{pj}, t = s_{ij} \mid\}$, suppose : $\cos^0_{jt} := \cos_{jt};$

Step 2: If $\cos_{jt} = \phi,$, turn to step 5; otherwise, according to the following formula select the earliest starting time in conflict operation set $\cos_{jt}$: $o_{p*j} = \{o_{pj} \mid s_{p*j} = \min\{s_{pj}\}, o_{pj} \in \cos_{jt}\}$, if it exists several o_{p*j}, select any one, $\cos_{jt} := \cos_{jt} - o_{p*j}$;

Step 3: Trimming the earliest starting time of operation o_{p*j}, $s_{p*j} = s_{ij} - p_{p*j}$;

Step 4: Constraint propagation, to all 1<k<j, if $s_{p*k} \ge c_{k-1}$ (or $s_{p*k} \ge rd$ (k=1)), trimming the earliest starting time $s_{p*k} = s_{p*k+1} - p_{p*k} - w_{j-1}$ of operation o_{p*k} , turn to step 8; otherwise, back to step 3 and turn to step 2;

Step 5: according to the formula we select the minimum operation that the completion time and time t has in conflict operation set $\cos_{jt}$: $o_{p'j} = \{o_{pj} \mid s_{p'j} = \min\{c_{pj} - t\}, o_{pj} \in \cos^0_{jt}, t = s_{ij}\}$, if there exist several $o_{p'j}$, we can select any one;

Step 6: trimming the starting time $s_{ij} = c_{p'j}$ of operation o_{ij} ;

Step 7: constraint propagation, trimming the starting time to all $j < k \le s$ $s_{ik} = s_{ik-1} + p_{ik-1}$;

Step 8: output current scheduling and end the process.

3.1.2. Neighborhood Search

Restarting search mechanism was used to optimize the objective function to MCFS scheduling problem using optimization constraint satisfaction, and the results show that the optimization process is fast, easy to implement, but the convergence is less than ideal. So, we embed Local Search (LS) replacing the neighborhood search base on forward in the constraint satisfaction optimization process, and the pseudo-code is:

```
Procedure LS (sch, $sch^*$ =sch){
generate a neighbor of $o_{is}$: $N(o_{is})$;
while $N(o_{is}) \ne \phi$ do {
        select $\forall o_{ks} \in N(o_{is})$, $N(o_{is}) := N(o_{is}) - o_{ks}$;
        exchanging positions of $o_{is}$ and $o_{ks}$;
        generate a neighbor $sch^k$ through constraint propagation;
        If a feasible solution $sch^k$ exists & F($sch^k$)<F($sch^*$) then{
        $sch^* := sch^k$;
      }
  }
  Return: $= sch^*$;
  }
```

3.2. Machine Assignment Problem

Machine assignment shows that the operation in the MCFS problem scheduling results is assigned to the parallel machine in the corresponding HFS. The problem is the constraint satisfaction problem, consisted by the operation of processing machine variable m_{ij} and constraint (4) and (7).

In solving process of the machine assignment problem, the choice of variables and values use the first come first service rules (FCFS) and the first idle machine priority rule (FAM) [10], and it can quickly get a solution of a constraint satisfaction problem.

4. Simulation Experiment

Using DELPHI as the programming language to realize the algorithm in the paper, the experimental environment is desktop of Pentium4/CPU 3.00GHz/RAM 1GB. The experimental data set the work number n={10,20,50}, the number of stages s={2,3,5}, the machine number of each stage $|R_j|$=3, the work processing time p_{ij} follows the uniform distribution integers of U[10,30], the released period rd_i=0, the delivery time dd_i= $dd_i = \sum_{j=1}^{s} p_{ij} + U[0,1] \times \sum_{i=1}^{n}\sum_{j=1}^{s} p_{ij} / \sum_{j=1}^{s} |R_j|$, rounded, the work waiting time limit w_j={5,10} between adjacent phases, and the work in advance, tardiness penalties coefficient $C1_i$, $C2_i$ are taken 5 and 10, 20 groups of question instances are randomly generated for each problem structure.

Table 1. Data Experimental Results

Work×stage	Waiting time ω_j	Termination cycle number		CPU/s		Improvement rate
		CSO	LSBCSO	CSO	LSBCSO	η/(%)
10×2	5	80.38	53.85	3.37	0.81	2.97
	10	106.45	56.01	2.46	0.85	4.53
10×3	5	139.93	80.51	6.89	1.82	4.25
	10	171.05	88.31	5.95	2.00	6.63
10×5	5	248.41	98.53	14.33	3.71	5.37
	10	208.32	105.67	12.02	3.98	7.23
20×2	5	402.23	228.70	18.57	6.90	6.14
	10	416.91	213.50	25.29	6.44	8.08
20×3	5	778.85	339.15	53.93	15.34	7.76
	10	733.71	335.08	64.41	16.47	9.70
20×5	5	833.75	377.80	111.33	28.48	11.48
	10	688.97	354.02	79.51	31.14	15.56
50×2	5	958.15	265.51	110.57	20.02	13.66
	10	882.68	247.76	101.86	18.68	22.05
50×3	5	1217.24	421.95	244.72	47.72	21.04
	10	1264.28	385.66	218.85	43.62	31.14
50×5	5	1029.16	538.43	378.19	101.49	25.61
	10	1169.88	500.89	337.51	94.41	32.99

The experimental results are shown in Table 1, the terminated cycle number is the objective function value improvement rate not more than 0.5% of the cycle number within continuous 20 cycles; CPU time is the program consumed time when it reaches the termination cycle number; improvement rate $\eta=(F_{LSBCSOFCSO})-F_{CSO})/F_{CSO}$, F_{LSBCSO} and F_{CSO} respectively refer to the optimal solution of constraint satisfaction algorithm L_{SBCSO} embedded neighborhood search constraint and the optimal solution of constraint satisfaction algorithm CSO without the neighborhood search.

Table 1 lists the results of two different algorithms which are CSO and LSBCSO, and we can see:

(1) Two algorithms obtain a satisfactory scheduling result within an acceptable time range.

(2) From the algorithm efficiency, we can see that to structure of the same problem, the computational efficiency of LSBCSO is better than that of the CSO, which shows that the embedded neighborhood search can greatly improve the efficiency of constraint satisfaction optimization algorithm. The bigger scale of the problem the more obvious efficiency will be.

(3) From the experimental effect, neighborhood search for the target value improvement rate of constraint satisfaction optimization algorithm CSO and the problem scale and the waiting limit time are related. The reason is that the greater problem size is the greater feasible solution space and the more obvious improvement effect of the solution will be, the greater waiting time constraints limit is, the more relaxed problem constraint and the greater feasible solution space will be, so the more obvious improvement effect of solution will be.

5. Conclusion

Just-in-time HFS scheduling problem with limited waiting time in the industrial production system exists many examples. HFS scheduling problem with limited waiting time on minimizing the earliness/tardiness penalty for the optimization objective were studied. Constraint satisfaction technology was used to establish the optimization model; based on the complexity characteristics of the model with two variables, we decompose the problem into several ability flow shop scheduling problem and the machine assignment problem. We focus on the former and present constraint satisfaction optimization algorithm based on neighborhood search. The experimental results show that, neighborhood search in optimization restarting the search mechanism of the embedded constraint satisfaction can improve the convergence of constraint satisfaction optimization methods; the algorithm put forward in the paper is feasible and effective.

References

[1] Brah SA, Loo LL. Heuristics for scheduling in a flow shop with multiple processors. *European Journal of Operational Research*. 1999; 113(1): 113-122.

[2] Moursli O, Pochet Y. A branch-and-bound algorithm for the hybrid flow shop. *International Journal of Production Economics*. 2000; 64(1/3): 113-125.

[3] Wardono B, Fathi Y. A tabu search algorithm for multi-stage parallel machine problem with limited buffer capacities. *European Journal of Operational Research.* 2004; 155(2): 380-401.

[4] Halln G, Sriskandarajah C. A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*. 1996; 44(3): 510-525.

[5] Sriskandarajah C. The performance of scheduling algorithms for nowait flow shops with parallel machines. *European Journal of Operational Research*. 1993; 70(3): 365-378.

[6] Xie Jinxing, Xing Wenxun, Liu Zhixin, et al. Minimum deviation algorithm for two-stage no-wait flowshops with parallel machines. *Computers & Mathematics with Applications*. 2004; 47: 1857-1863.

[7] Li Tieke, Zhou Jian, Sun Lin. Steelmaking continuous casting and rolling and cold-mounted hot-rolled side-by-side environment-Continuous Casting Production Scheduling Model and Algorithm. *System engineering theory and Practice*. 2006; 6: 117-123.

[8] Dorndorf U, Pesch E, Phan-Huy T. Constraint propagation techniques for the disjunctive scheduling problem. *Artificial Intelligence*. 2000; 122: 189-240.

[9] Nuijten W, Aarts E. A computational study of constraint satisfaction for multiple capacitated job shop scheduling. *European Journal of Operational Research*. 1996; 90(2): 269-284.

[10] Valerie BG. Hybrid flow shop scheduling with precedence constraints and time lags to minimize maximum lateness. *International Journal of Production Economics*. 2000; 64: 101-111.