

SEQUENCE-DEPENDENT SETUP TIMES IN A TWO-MACHINE MAKESPAN JOB- SHOP SCHEDULING PROBLEM

Yuri N. Sotskov ^{*}, Natalja G. Egorova ^{*} and Frank Werner ^{**}

^{} United Institute of Informatics Problems, National
Academy of Sciences of Belarus, Surganova Str. 6, Minsk,
Belarus, e-mail: sotskov@newman.bas-net.by*

*^{**} Faculty of Mathematics, Otto-von-Guericke University,
Postfach 4120, D-39016, Magdeburg, Germany, e-mail:
frank.werner@mathematik.uni-magdeburg.de*

1. PROBLEM DESCRIPTION

$$J = \{1, 2, \dots, n\}, \quad M = \{1, 2\}$$

each machine $m \in M$ processes any job $j \in J$ at most once

$$J = J_{12} \cup J_1 \cup J_2 \cup J_{21}$$

$$n_k = |J_k|, k \in \{1, 2, 12, 21\}$$

O_{jm} - operation of job $j \in J$ on machine $m \in M$

p_{jm} - processing time of operation O_{jm}

If job $i \in J$ is directly followed by job $k \in J$ on machine $m \in M$, then the setup time is equal to a non-negative real number s_{ik}^m (s_{0k}^m, s_{i0}^m)

$$S^1 = \|s_{ij}^1\|, \quad r_1 \times r_1, \quad r_1 = n - n_2 + 1$$

$$S^2 = \|s_{ij}^2\|, \quad r_2 \times r_2, \quad r_2 = n - n_1 + 1$$

$$\pi' = (i'_1, i'_2, \dots, i'_{r_1}) \quad i'_k \in J_{12} \cup J_1 \cup J_{21}$$

$$\pi'' = (i''_1, i''_2, \dots, i''_{r_2}) \quad i''_k \in J_{12} \cup J_2 \cup J_{21}$$

$$C_{\max}(\pi', \pi'') = \max \left\{ C_{i'_{r_1}}(\pi', \pi'') + s_{i'_{r_1}0}^1, C_{i''_{r_2}}(\pi', \pi'') + s_{i''_{r_2}0}^2 \right\} \quad (1)$$

$$J_2 \mid s_{jk} \mid C_{\max}$$

2. MODIFICATION OF SETUP, REMOVAL AND PROCESSING TIMES

For each $i \in J_1 \cup J_{12}$:

$$s^1(\rightarrow i) = \min\{s_{ki}^1 \mid i \neq k \in \{0\} \cup J \setminus J_2\} \quad (2)$$

$$p'_{i1} = s^1(\rightarrow i) + p_{i1} \quad (3)$$

$$s_{ki}^{(1)} = s_{ki}^1 - s^1(\rightarrow i), \quad i \neq k \in \{0\} \cup J \setminus J_2 \quad (4)$$

$$s_{ki}^{(1)} \geq 0 \quad \text{for each } i \in J_1 \cup J_{12}$$

For each $i \in J_2 \cup J_{21}$:

$$p'_{i2} = s^2(\rightarrow i) + p_{i2} \quad (5)$$

$$s_{ki}^{(2)} = s_{ki}^2 - s^2(\rightarrow i), i \neq k \in \{0\} \cup J \setminus J_1 \quad (6)$$

$$s^2(\rightarrow i) = \min\{s_{ki}^2 \mid i \neq k \in \{0\} \cup J \setminus J_1\} \quad (7)$$

For each $j \in J_1 \cup J_{21}$:

$$s^1(j \rightarrow) = \min\{s_{jk}^1 \mid j \neq k \in \{0\} \cup J \setminus J_2\} \quad (8)$$

$$p'_{j1} = s^1(j \rightarrow) + p_{ij} \quad (9)$$

$$s_{jk}^{(1)} = s_{jk}^1 - s^1(j \rightarrow), \quad j \neq k \in J \setminus J_2 \quad (10)$$

$$s_{j0}^{(1)} = s_{j0}^1 - s^1(j \rightarrow) \quad (11)$$

For each $i \in J_2 \cup J_{12}$:

$$p'_{i2} = s^2(i \rightarrow) + p_{i2} \quad (12)$$

$$s_{ik}^{(2)} = s_{ik}^2 - s^2(i \rightarrow), i \neq k \in J \setminus J_2 \quad (13)$$

$$s_{j0}^{(2)} = s_{j0}^2 - s^2(j \rightarrow) \quad (14)$$

$$s^2(i \rightarrow) = \min\{s_{ik}^2 \mid i \neq k \in \{0\} \cup J \setminus J_2\} \quad (15)$$

$$p_{0m} = s^m(0) \quad (16)$$

$$s_{0j}^{(m)} = s_{0j}^m - s^m(0), j \in J \setminus J_{3-m} \quad (17)$$

$$s^m(0) = \min\{s_{0j}^m \mid j \in J \setminus J_{3-m}\} \quad (18)$$

$$p_{n+1,m} = s^m(n+1) \quad (19)$$

$$s_{j0}^{(m)} = s_{j0}^m - s^m(n+1), j \in J \setminus J_{3-m} \quad (20)$$

$$s^m(n+1) = \min\{s_{j0}^m \mid j \in J \setminus J_{3-m}\} \quad (21)$$

Theorem 1. An instance of problem $J2 | s_{jk} | C_{\max}$ is equivalent to the modified instance that differs from the original one only by setup, removal, and processing times of jobs $J \cup \{0, n+1\}$ modified due to formulas (2) - (21).

3. SUFFICIENT OPTIMALITY CONDITION OF JACKSON'S PERMUTATIONS

Theorem 2. Jackson's pair (π', π'') of job permutations constructed for the instance of problem $J2 \parallel C_{\max}$ with job processing times obtained by means of formulas (2), (3), (5), (7), (8), (9), (12), (15), (16), (18), (19), (21) remains optimal for the corresponding instance of problem $J2 | s_{jk} | C_{\max}$, if the main machine $m \in M$ for the semiactive schedule defined by (π', π'') has no idle times and has only zero modified setup and removal times.

4. HEURISTIC ALGORITHM H

1. Construct a modified instance that is equivalent (due to Theorem 1) to the original instance of problem $J2 | s_{jk} | C_{\max}$.
2. Find Jackson's pair (π', π'') of job permutations constructed for problem $J2 || C_{\max}$ corresponding to the modified instance of problem $J2 | s_{jk} | C_{\max}$.
3. Test the sufficient condition for optimality of (π', π'') for the modified instance of problem $J2 | s_{jk} | C_{\max}$ given by Theorem 2 (and others).

4. If at least one of the sufficient conditions holds, the original instance of problem $J2 | s_{jk} | C_{\max}$ is solved by the pair (π', π'') of job permutations. Otherwise, the semiactive schedules constructed for the corresponding instance of problem $J2 || C_{\max}$ (those constructed for the corresponding instance of problem $J2 | s_{jk} | C_{\max}$) polynomially provide lower bounds (upper bounds, respectively) for the objective function (1).

WORST CASE ANALYSIS OF ALGORITHM *H*

$$n_{\min} = \min\{\min\{|J \setminus J_1|, |J \setminus J_2|\}, \\ \min\{|J_{12}| + 1, |J_{21}| + 1\}\}$$

$$n_{\max} = \max\{\max\{|J \setminus J_1|, |J \setminus J_2|\}, \\ \max\{|J_{12}| + 1, |J_{21}| + 1\}\}$$

$$s_{\min} = \min\{s_{ij}^m \mid m \in M, i \in J, i \neq j \in J\}$$

$$s_{\max} = \max\{s_{ij}^m \mid m \in M, i \in J, i \neq j \in J\}$$

a) $s_{ij}^m \leq p_{jm}, i \in J, i \neq j \in J.$ (22)

$$C_{\max}(\pi', \pi'') \leq 2C_{\max}^* - n_{\min} s_{\min}$$

b) $p_{jm} \leq s_{ij}^m \leq 2p_{jm}, i \in J, i \neq j \in J.$ (23)

$$C_{\max}(\pi', \pi'') \leq 3/2 C_{\max}^*$$

c) General case:

$$C_{\max}(\pi', \pi'') \leq C_{\max}^* + n_{\max}(s_{\max} - s_{\min})$$

5. BRANCH AND BOUND ALGORITHM AND COMPUTATIONAL RESULTS

- Implicit enumeration of feasible semiactive schedules
- Algorithm is based on the lower and upper bounds obtained by Algorithm H
- Stopping criterion: CPU limit of 900 s (Pentium PC; 2800 MHz) and at most 15,000,000 vertices

Computational results for problems with p_{ij} from $[10, 100]$ and s_{ij} from $[0, 10]$

# jobs	#unsolved problems	Average CPU time	Average # vertices	Maximal CPU time
100	0	20.7	448,735	105
160	0	97.6	1,039,838	244
220	0	267.8	1,065,730	453
240	0	388.9	929,035	541
260	3	533.3	894,748	538
280	1	798.9	1,337,519	872

Computational results for problems with p_{ij} from $[1, 100]$ and s_{ij} from $[0, 10]$

# jobs	# unsolved problems	Average CPU time	Average # vertices	Maximal CPU time
100	1	91.3	1,759,915	516
160	1	96.9	1,317,195	399
220	2	317.3	1,291,140	569
240	2	413.3	1,442,193	604
260	3	542.0	837,037	668
280	4	818.8	1,644,841	882

Computational results for problems with 100 jobs

$n_{12} + n_1 + n_2 + n_{21}$	# uns. problems	Average CPU time	Average # vertices	Maximal CPU time
45+5+5+45	1	22.7	203,082	58
40+10+10+40	2	220.6	1,931,838	499
30+20+20+30	0	152.7	1,719,753	712
20+30+30+20	1	45.7	799,549	174
10+40+40+10	2	112.2	2,003,443	509
5+45+45+5	0	22.1	745,053	78