J. Operations Research Soc. of Japan
Vol. 17, No. 1, March 1974

A GRAPHICAL BRANCH-AND-BOUND ALGORITHM FOR THE JOB-SHOP SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT SET-UP TIMES

THEODOR SIEGEL

Technische Universität Berlin

(Received December 11, 1972)

Abstract

The job-shop scheduling problem is considered for the case of sequence-dependent set-up times. The solution of Nabeshima [1] to an example problem is improved, and it is pointed to the cause of the nearoptimality of Nabeshima's solution. Then a new approach is made to the same example, by applying a graphical branch-and-bound algorithm. By means of an example with three jobs it is shown that this algorithm can be applied to an arbitrary number of jobs.

1. Introduction

In general, the job-shop scheduling problem [1, p. 73] is researched without consideration of sequence-dependent set-up times. However, authors as Charlton/Death [2], Müller-Merbach [3], Mensch [4], and Nabeshima [1] point to the possibility that set-up times are dependent on the job sequence, and to the necessity of their simultaneous optimization.

In a recent publication, Nabeshima [1, pp. 87 ff.] makes the first published algorithmical approach to such a sequencing problem. Nabeshima's procedure is an efficient algebraical branch-and-bound algorithm which takes advantage of the disjunctive graph formulations of Balas [5] and Charlton/Death [2].

The intention of this contribution is to present a different solution to the same problem. The algorithm is shown by means of the example of Nabeshima and then extended to $n>2$. At first some remarks on Nabeshima's algorithm may be allowed.

2. Remarks on Nabeshima's Approach

The following matrix is corresponding to the data of the example of Nabeshima:

i	t_{iA1}	t_{iA2}	t_{iB1}	S_i
1	3	5 or 11*	1	A, B, A
2	4	—	6	A, B

$$* \ t_{1A2}=\begin{cases}11 \text{ if } F_A=\cdots, (2A1), (1A2), \cdots \\ 5 \text{ otherwise}\end{cases}$$

$$\Delta t_{1A2}= 6$$

(ijk) is the kth operation ($k\in\{1; 2\}$) of job $i\in\{1; 2\}$ on machine $j\in\{A, B\}$; t_{ijk} is its operation time including its own set-up time (in contrast with Nabeshima's d_{ij} which include set-up time for next operation. The choice of t_{ijk} and their definition are preferred because an operation is in a closer relation to its own set-up time. The d_{ij}'s are changed correspondingly). T_{ijk} is the completion time of operation (ijk). S_i determines the technological order of job i. F_j means the solution order on machine j. ($F_A=\cdots$, (2A1), (1A2), $\cdots$ signifies that (2A1) is directly followed by (1A2).)

The objective is to minimize the makespan:

$$T=\max_{(ijk)} T_{ijk}=\min!$$

The solution of Nabeshima is represented by this Gantt chart (cf. [1], Fig. 17, p. 91):

Now it is easy to see that operation (1A2) can be moved back to begin in $t=8$ without offending against any restriction. So the

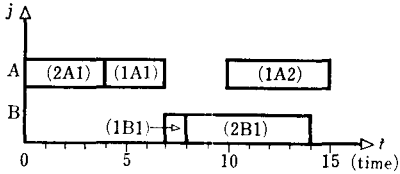

Fig. 1.

optimal solution has

$$T = \max_{(ijk)} T_{ijk} = T_{2B1} = 14$$

The cause of the error is in the regard of the arc (4, 3) in the solution graph of Nabeshima [1, p. 90]. This arc must only be considered in the case $F_A = \cdots$, (2A1), (1A2), $\cdots$ (or, in terms of Nabeshima, operation No. 3 follows d i r e c t l y to operation No. 4). In the case of Nabeshima's solution there is F_A=(2A1), (1A1), (1A2), and no economical reason is to be seen that operation (1A2) should have a set-up time of 6 time units in case of a predecessor operation $(ijk) \neq (2A1)$.

Nevertheless, the approach of Nabeshima is a very interesting one. It should be modified in that way that, *e.g.*, if arc (4, 1) is taken to a solution arc (4, 3) is to be taken out of consideration.

3. The Graphical Branch-and-Bound Algorithm for Two Jobs

The graphical algorithm which is now proposed with regard to the general scheduling problem with sequence-dependent set-up times is the extension of the Akers solution [6] by a branch-and-bound procedure and a particularity for dependence of sequence.

The graphical formulation of the example of Nabeshima can be shown in the following "operations area" (Fig. 2):

The shortest way from point ($t_1=0$; $t_2=0$) to point ($t_1 = \sum_{jk} t_{1jk} = 9$; $t_2 = \sum_{jk} t_{2jk} = 10$) (from now the t_{ijk} are taken with their minimal values) is to be found regarding that conflict areas (shaded) must not be crossed and that only horizontal, vertical, and 45-degree segments, each corresponding to one time unit, are allowed. t_i means the sum

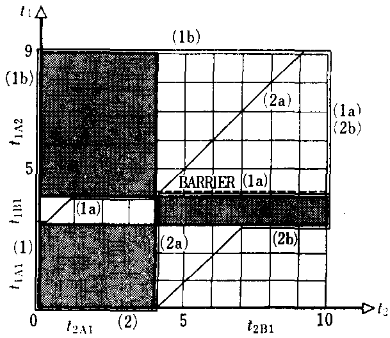

Fig. 2.

of time units having been operated on job i. The fact that t_{1A2} is 6 more in case $F_A=\cdots$, (2A1), (1A2), $\cdots$ as otherwise is respected by a "barrier" of $\Delta t_{1A2}=6$ time units' length. The barrier must not be crossed (for an exception see below). The barrier is marked as a conditional extension of t_{1A2} (cf. d_{43} of Nabeshima), so that is has the same effect on t_1 as constructing a new operations area with $t_{1A2}=11$.

The application of branch-and-bound is as follows: Already in point ($t_1=0$; $t_2=0$), a conflict area requires a branching decision:

Way (1) with $F_A=$(1A1), (2A1), $\cdots$ or

Way (2) with $F_A=$(2A1), (1A1), $\cdots$

Lower bounds for makespan are calculated as follows, not paying regard to other conflict areas in this moment (in this simple case, it would be possible not to neglect them):

Way: Makespan (lower bound):

(1) $T_{(1)}\geq t_{1A1}+\max(t_{2A1}+t_{2B1};\ t_{1B1}+t_{1A2})=13$

(2) $T_{(2)}\geq t_{2A1}+\max(t_{2B1};\ t_{1A1}+t_{1B1}+t_{1A2})=13$

As both lower bounds are equal, Way (1) may be taken. In point ($t_1=4$; $t_2=1$) it meets another conflict area. Going round as

Way (1a) with $F_A=\cdots$, (2A1), (1A2)

it must respect the barrier. In case of

Way (1b) with $F_A=\cdots$, (1A2), (2A1) there is no barrier. The lower bounds are:

Way: Makespan (lower bound):

(1a) $T_{(1a)} \geq t_{1A1} + \max(t_{1B1};\ t_{2A1}) + \Delta t_{1A2} + t_{1A2} = 18$

(1b) $T_{(1b)} \geq t_{1A1} + t_{1B1} + t_{1A2} + t_{2A1} + t_{2B1} = 19$

Way (1a) does not meet any more conflict areas, so its lower bound marks a solution makespan. As the lower bound of Way (2) is 13, Way (2) may be better as Way (1a). Following Way (2), in point ($t_1=3$; $t_2=7$) there is a conflict. Branching in

Way (2a) with $F_B=$(1B1), (2B1) or

Way (2b) with $F_B=$(2B1), (1B1)

the decision of continuing is based on the following calculations:

Way: Makespan (lower bound):

(2a) $T_{(2a)} \geq t_{2A1} + t_{1A1} + t_{1B1} + \max(t_{1A2};\ t_{2B1}) = 14$

(2b) $T_{(2b)} \geq t_{2A1} + \max(t_{1A1};\ t_{2B1}) + t_{1B1} + t_{1A2} = 16$

(Way (2a) must not respect the barrier as this way has $F_A=$(2A1), (1A1), ...)

The solution of Way (2a) is optimal because there is no more conflict area, and all other lower bounds (here: that of Way (2b)) are >14. $T=14$ is corresponding to the solution by the modified Nabeshima algorithm.

4. More than Two Jobs

It is an important question whether this graphical branch-and-bound algorithm can be extended to $n>2$ (more than two jobs). Though Akers [6] does not believe in this possibility and others deny this, the graphical algorithm is, at least in theory, not limited to a certain value of n. The procedure [7] is a "decomposition" of the n-job problem to $\binom{n}{2}$ 2-job problems where every action in the representation of one 2-job problem (like Fig. 2) has its reflection in all other representations in which one of these jobs is participated.

This is the only graphical algorithm for $n>2$ known to the author. The graphical approaches by Hardgrave/Nemhauser [8] and Mensch

[9, pp. 89 ff.] are limited to $n=2$ (or extremely 3 when suffering difficulties in representation).

The procedure is sketched in the case of sequence-dependent set-up times as follows. To simplify the exposition, only one machine ($j=A$) demands sequence-dependent set-up times. The data of the problem considered can be found in this matrix:

i	t_{iA} in case of predecessor operation				t_{iB}	t_{iC}	t_{iD}	S_i
	—	t_{1A}	t_{2A}	t_{3A}				
1	3	—	7	3	4	2	3	D, A, C, B
2	5	6	—	6	4	2	4	B, A, C, D
3	2	3	4	—	4	3	4	C, D, B, A

(Operation time here is t_{ij}- job i on machine j- as every job is worked only once on each machine.)

The $\binom{n}{2}=\binom{3}{2}=3$ operations areas are constructed in Fig. 3, containing the optimal way:

Every solution way tries to have as many diagonal steps as possible in all operations areas. Four diagonal steps can be made until in point ($t_1=4$; $t_2=4$) there is a conflict between jobs 1 and 2 at machine A. The calculations are:

Way: Makespan (lower bound):

(1) $T_{(1)}\geq u^c+t_{1D}+t_{1A}-t_1^c+\max(\Delta t_{2A(1A)}+t_{2A}+t_{2C}+t_{2D};\ t_{1C}+t_{1B})=18$

(2) $T_{(2)}\geq u^c+t_{2B}+t_{2A}-t_2^c+\max(\Delta t_{1A(2A)}+t_{1A}+t_{1C}+t_{1B};\ t_{2C}+t_{2D})=22$

u^c is the number of steps until the actual conflict ($u^c=4$). t_i^c is the position of job i in the operations areas at the time of this conflict ($t_1^c=4$; $t_2^c=4$). Both ways have to respect barriers of additional set-up times Δt_{iA} for the second operation on machine A. $\Delta t_{iA(pA)}$ means the difference between t_{iA} in case of predecessor operation (pA), and t_{iA} in case of no predecessor operation. (In the equation for lower bound, t_{iA} is taken with its minimal value.) Corresponding to the graphical representation, the lower bounds can be calculated in a different but

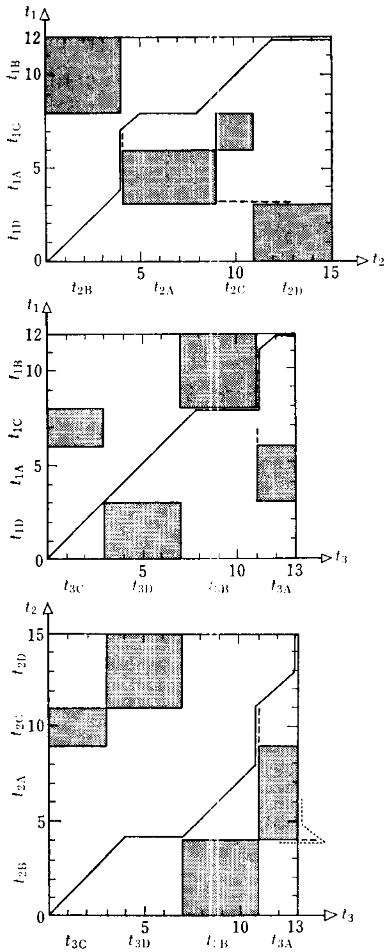

Fig. 3.

equivalent way, *e.g.*:

$$T_{(2)} \geq u^c + t_{2B} + t_{2A} + \Delta t_{1A(2A)} - t_2^c$$
$$+ \max\left(t_{1A} + t_{1C} + t_{1B};\ t_{2C} + t_{2D} - \Delta t_{1A(2A)}\right) = 22$$

The solution is continued by Way (1). That means job $i=2$ has to wait in the operations areas for 3 time units (2 in reality, and 1 to pay regard to the extension of operation time t_{2A}). This causes three non-diagonal steps in every operations area containing $i=2$. The next step is diagonal in all operations areas, leading to $t_1=8$; $t_2=5$; $t_3=8$. In point ($t_1=8$; $t_3=8$) a conflict is met at machine B ($u^c=t_1^c=t_3^c=8$).

Way: Makespan (lower bound):

(1a) $T_{(1a)}\geq u^c+\sum_{j\in J} t_{1j}-t_1^c+t_{3B}+t_{3A}=18$

(1b) $T_{(1b)}\geq u^c+t_{3C}+t_{3D}+t_{3B}-t_3^c+\max(t_{1B};\ t_{3A})=15$

($j\in J$ means all $j\in\{A, B, C, D\}$.) The continuation of the way is preferred by (1b) because $T_{(1b)}$ is expected to be lower than $T_{(1a)}$. $T_{(1b)}\geq 15$ is to be read $T_{(1b)}\geq 18$ as $T_{(1)}\geq 18$. Three steps only for $i=2$; 3 are done until the end of the conflict. Then there is a new conflict: $u^c=11$; $t_2^c=8$; $t_3^c=11$. It is solved by:

Way: Makespan (lower bound):

(1ba) $T_{(1ba)}\geq u^c+t_{2B}+t_{2A}-t_2^c+\max(\Delta t_{3A(2A)}+t_{3A};\ t_{2C}+t_{2D})=18$

(1bb) $T_{(1bb)}\geq u^c+\sum_{j\in J} t_{3j}-t_3^c+\Delta t_{2A(3A)}+t_{2A}+t_{2C}+t_{2D}=25$

The solution way is continued by way (1ba), which effects 3 time units waiting for $i=3$ (1 of them in reality). No further conflict occurs, so way (1ba) ends with $T_{(1ba)}=18$. This way is optimal since no unchecked branching can have a $T<18$. The solution is represented by this Gantt chart (Fig. 4):

Some remarks are added with regard to ways different from this

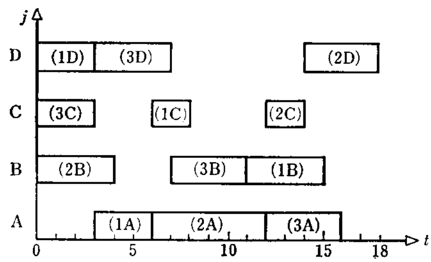

Fig. 4.

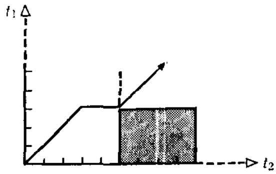

Fig. 5.

solution way. Way (1bb), if relevant, could be demonstrated in the operations area of $i=2$; 3 as partially shown by the dotted line.

In general, set-up time can begin before the job arrives at a machine—if the machine is not occupied. Therefore a barrier must not be effective in such cases. These can be found out in the operations areas.

For a way like the given one (partially) in Fig. 5, the barrier does not exist as setting-up can be executed in the two time units before arriving at the barrier.

It is to be expected that the algorithm of Nabeshima can be modified to take such possibilities into consideration. But the disjunctive graphs of Balas and Nabeshima are very difficult to be surveyed for increasing numbers of jobs and machines. The graphical branch-and-bound algorithm shows the relations clearly by constructing the operations areas, and it can be taken advantage of the graphical representation of operation times. However, computing the optimal solution is possibly not economical in problems of realistic size. So a heuristic procedure, which can be based on the graphical algorithm or an algebraical equivalent, is proposed.

References

[1] Nabeshima, I., "General Scheduling Algorithms with Applications to Parallel Scheduling and Multiprogramming Scheduling," *J. Operations Res. Soc. of Japan*, **14** (1971), 72-99.

[2] Charlton, J.M., and C.C. Death, "A Generalized Machine-scheduling Algorithm," *Operational Res. Quart.*, **21** (1970), 127-134.

[3] Müller-Merbach, H., *Operations Research*, 2, Auflage, München, 1971.

[4] Mensch, G., "Das Trilemma der Ablaufplanung," *Zeitschrift für Betriebswirtschaft*, **42** (1972), 77-88.

[5] Balas, E., "Machine Sequencing via Disjunctive Graphs: An Implicit Enumeration Algorithm," *Operations Res.*, **17** (1969), 941-957.

[6] Akers, S.B., Jr., "A Graphical Approach to Production Scheduling Problems," *Operations Res.*, **4** (1956), 244-245.

[7] Siegel, T., *Das Reihenfolgeproblem der Maschinenbelegungsplanung und ein graphischer Branch-and-Bound-Algorithmus zu seiner Lösung*, Technische Universität Berlin, Dissertation 1973; *Optimale Maschinenbelegungsplanung*, Berlin, 1974.

[8] Hardgrave, W.W., and G.L. Nemhauser, "A Geometric Model and a Graphical Algorithm for a Sequencing Problem," *Operations Res.*, **11** (1963), 889-900; **12** (1964), 364.

[9] Mensch, G., *Ablaufplanung*, Köln/Opladen, 1968.