STABILITY OF AN OPTIMAL SCHEDULE FOR A JOB-SHOP PROBLEM WITH TWO JOBS

Yu. N. Sotskov, N. Yu. Sotskova

Abstract: The usual assumption that the processing times of the operations are known in advance is the strictest one in scheduling theory. This assumption essentially restricts practical aspects of deterministic scheduling theory since it is not valid for the most processes arising in practice. The paper is devoted to a stability analysis of an optimal schedule, which may help to extend the significance of scheduling theory for decision-making in the real-world applications. The term stability is generally used for the phase of an algorithm, at which an optimal solution of a problem has already been found, and additional calculations are performed in order to study how solution optimality depends on variation of the numerical input data.

Keywords: Job shop problem, geometric approach, stability analysis

Introduction

The problem under consideration is to minimize the value of the given objective function of completion times of n jobs J = {1, 2, …, n} processed on m machines M = {1, 2, …, m}. First, we assume that processing time $t_{j,k}$ of job $j \in J$ on machine $k \in M$ (i.e., processing time of operation $O_{j,k}$) is known before scheduling. Operation preemptions are not allowed. This problem is denoted as $J||\Phi$ where Φ defines objective function.

Let $C_{i,k}$ denote the completion time of the job in position i on machine $k \in M$. We assume that objective function $\Phi(C_{1,m}, C_{2,m}, \ldots, C_{n,m})$ is non-decreasing function of job completion times. Such a criterion is called regular.

For the job-shop problem $J|n=2|C_{max}$ with two jobs and makespan objective function $C_{max} = \max\{C_{1,m}, C_{2,m}, \ldots, C_{n,m}\}$, the geometric algorithm was proposed by Akers and Friedman [1] and developed by Brucker [2], Szwarc [7], Hardgrave and Nemhauser [4]. Sotskov [5] generalized the geometric algorithm for the problem $J|n=2|\Phi$ with any given regular criterion. Sotskov [6] proven that both problems

$$J|n=3|C_{max} \text{ and } J|n=3|\sum C_{i,m}$$

are binary NP-hard. Hereafter, the criterion $\sum C_{i,m}$ means minimization of total completion time

$$\sum_{i=1}^{n} C_{i,m}.$$

Geometric algorithm

For simplicity, we describe geometric model for the case of a flow-shop problem $F|n=2|\Phi$, i.e., when all n jobs have the same technological route through m machines, namely, (1, 2, …, m).

Let $TM_{j,k}$ denote the sum of the processing times of job $j \in J = \{1, 2\}$ on a subset of k machines $\{1, 2, ..., k\} \subseteq M$:

$$TM_{j,k} = \sum_{i=1}^{k} t_{j,i}, \ 1 \le k \le m.$$

It is assumed that $TM_{1,0} = TM_{2,0} = 0$. We introduce a coordinate system xy on the plane, and draw the rectangle H with corners (0, 0), $(TM_{1,m}, 0)$, $(0, TM_{2,m})$ and $(TM_{1,m}, TM_{2,m})$. In the rectangle H, we draw m rectangles H_k, $k \in \{1, 2, \ldots, m\}$, with corners $(TM_{1,k-1}, TM_{2,k-1})$, $(TM_{1,k}, TM_{2,k-1})$, $(TM_{1,k-1}, TM_{2,k})$, $(TM_{1,k}, TM_{2,k})$. We denote south-west corner $(TM_{1,k-1}, TM_{2,k-1})$ of the rectangle H_k as SW_k, north-west corner $(TM_{1,k-1}, TM_{2,k})$ as NW_k, south-east corner $(TM_{1,k}, TM_{2,k-1})$ as SE_k, and north-east corner $(TM_{1,k}, TM_{2,k})$ as NE_k. Obviously, point (0, 0) is SW_1 and point $(TM_{1,m}, TM_{2,m})$ is NE_m.

We use Chebyshev's metric, i.e., the length d[(x, y), (x', y')] of a segment [(x, y), (x', y')] connecting points (x, y) and (x', y') in the rectangle H is calculated as follows:

$$d[(x, y), (x', y')] = \max\{|x - x'|, |y - y'|\}.$$

The length $d[(x_1, y_1), (x_2, y_2), \ldots, (x_r, y_r)]$ of a continuous polygonal line $[(x_1, y_1), (x_2, y_2), \ldots, (x_r, y_r)]$ is equal to the sum of the lengths of its segments.

Since $\Phi(C_{1,m}, C_{2,m})$ is a non-decreasing function, the search for the optimal schedule can be restricted to set S of schedules in which at any time of the interval $[0, \max\{C_{1,m}, C_{2,m}\}]$ at least one job is processed. A schedule from set S can be suitably represented within the rectangle H on the plane xy as a trajectory (continuous polygonal line) $\tau = [SW_1, (x_1, y_1), (x_2, y_2), \ldots, (x_r, y_r), NE_m]$ where either $x_r = TM_{1,m}$ or $y_r = TM_{2,m}$.

Let a point (x, y) belong to the trajectory τ and let d be the length of the part of trajectory τ from the point SW_1 to the point (x, y). The coordinate x (coordinate y) of point (x, y) defines the state of processing job 1 (job 2) as follows.

If $SW_u \le x \le SE_u$ and $SW_v \le y \le NW_v$, $u \in M$, $v \in M$, then job 1 (job 2) is completed on the machines 1, 2,…, u-1 (on the machines 1, 2,…, v-1) at time d. Moreover at time d, job 1 (job 2) has been processed on machine u (machine v) during $x - SW_u$ (during $y - SW_v$) time units.

Since a machine cannot process more than one job at a time and operation preemptions are not allowed, each straight segment [(x, y), (x', y')] of a trajectory τ may be either

- horizontal (when only job 1 is processed) or
- vertical (when only job 2 is processed) or
- diagonal with slope of 45^0 (when both jobs are processed simultaneously).

It is clear that a horizontal segment (vertical segment) can only pass along south boundary (west boundary) of the rectangle H_k, $k \in M$, or along north (east) boundary of the rectangle H. The diagonal segment of trajectory τ can only pass either outside rectangle H_k or through point NW_k or point SE_k.

Sotskov [5] proven that problem $J|n=2|\Phi$ of finding the optimal schedule or, in other words, of finding the optimal trajectory, can be reduced to the shortest path problem in the digraph (V, A) constructed by the following Algorithm 1. Again for simplicity, we describe this algorithm for the case of a flow-shop problem $F|n=2|\Phi$, when all n jobs have the same technological route through m machines.

Vertex set V of the digraph (V, A) is a subset of set

$$V^0 = \{SW_1, NE_m\} \cup \{NW_k, SE_k : k \in M\} \cup \{(x_k, TM_{2,m}), (TM_{1,m}, y_k) : k \in M\}.$$

Algorithm 1

1. Set $V = \{SW_1, SE_1, NW_1, NE_m\}$ and $A = \{(SW_1, SE_1), (SW_1, NW_1)\}$.
2. Take vertex $(x, y) \in V \setminus \{NE_m\}$ with zero outdegree. If $(x, y) = SE_k$, go to step 3. If $(x, y) = NW_k$, go to step 4. If set $V \setminus \{NE_m\}$ has no vertex with zero outdegree.
   STOP
3. Draw a diagonal line with slop 45^0 starting from vertex SE_k until either east boundary $[(TM_{1,m}, 0), NE_m]$ of the rectangle H is reached in some vertex $(TM_{1,m}, y_k)$ or open south boundary (SW_h, SE_h) of the rectangle H_h, $k+1 \le h \le m$, is reached. In the former case, set $V: =V \cup \{(TM_{1,m}, y_k)\}$ and $A: =A \cup \{(SE_k, (TM_{1,m}, y_k)), ((TM_{1,m}, y_k), NE_m)\}$. In the latter case, set $V: =V \cup \{SE_h, NW_h\}$ and $A : =A \cup \{(SE_k, SE_h), (SE_k, NW_h)\}$.
   Go to step 2.
4. Draw a diagonal line with slope 45^0 starting from vertex NW_k until either north boundary $[(0, TM_{2,m}), NE_m]$ of the rectangle H is reached in some vertex $(x_k, TM_{2,m})$ or open west boundary (SW_h, NW_h) of the rectangle H_h, $k+1 \le h \le m$, is reached. In the former case, set $V: = V \cup \{(x_k, TM_{2,m})\}$ and $A: = A \cup \{(NW_k, (x_k, TM_{2,m})), ((x_k, TM_{2,m}), NE_m)\}$. In the latter case, set $V: = V \cup \{SE_h, NW_h\}$ and $A: = A \cup \{(NW_k, SE_h), (NW_k, NW_h)\}$.
   Go to step 2.

In order to find the optimal path (i.e., optimal schedule) for the problem $J|n=2|\Phi$ we can use the following Algorithm 2, where the length of arc $((x, y), (x', y')) \in A$ is assumed to be equal to the length of the polygonal line constructed by Algorithm 1 with origin in the point (x, y) and with end in the point (x', y').

Algorithm 2

1. Construct the digraph (V, A) using Algorithm 1 and find all border vertices in the digraph (V, A), i.e., the vertices (x, y) either of the form $(x_k, TM_{2,m})$ or of the form $(TM_{1,m}, y_k)$.
2. Construct the set of trajectories corresponding to the shortest paths in the digraph (V, A) from the vertex SW_1 to each of the border vertices.

3. Find an optimal trajectory (optimal path in (V, A)) in the set constructed at step 2 that represents a schedule with minimal value of the objective function Φ.
   STOP

It was proven that Algorithm 2 takes O(m log m) time for problem F|n=2|Φ and its generalization for problem J|n=2|Φ takes O(m^2 log m) time (see Sotskov [5, 6]).

Example

Next, we demonstrate the geometric algorithm for the problem Fk|n=2|Φ using five machines and processing times given in Table 1. We call this example as Example 1.

Machine m	= 1	= 2	= 3	= 4	= 5
$t_{1,m}$	[illegible]	0	[illegible]	[illegible]	2
$t_{2,m}$	6	[illegible]	[illegible]	[illegible]	[illegible]

Table 1. Processing times of two jobs

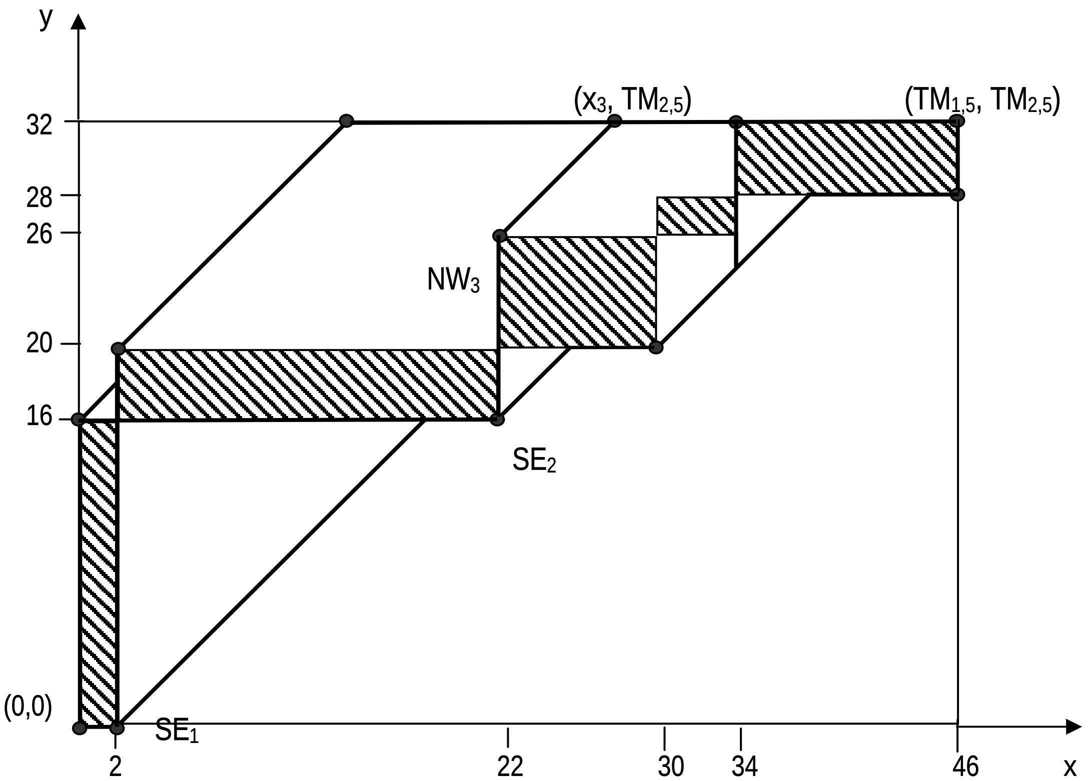

Figure 1. Trajectories representing active schedules for Example 1

For Example 1, we see that the shortest path ((0, 0), SE_1, SE_2, NW_3, (x_3, $TM_{2,5}$)) from vertex (0, 0) to the border vertex (x_3, $TM_{2,5}$) in the digraph (V, A) constructed by Algorithm 1 specifies trajectory [(0, 0), (2, 0), (18, 16), (22, 16), (22, 26), (28, 32)] in the rectangle R. Using Algorithm 2 we indicate that the schedule represented by this trajectory is optimal for the problem F5|n=2|$\Sigma C_{i,5}$ with total completion time criterion. See Fig. 1, where the rectangles R_m for Example 1 are shaded.

Stability analysis

In what follows, we consider stability of an optimal schedule with respect to possible variations of the given vector $t = (t_{1,1}, t_{1,2}, \dots, t_{1,m}, t_{2,1}, t_{2,2}, \dots, t_{2,m})$ of operation processing times.

Let (V_t, A_t) denote the digraph (V, A) constructed by Algorithm 1 for the problem $F|n=2|\Phi$ with vector t of operation processing times. Let $\mathbf{P}_t$ be set of all shortest paths from vertex SW_1 to the border vertices in the digraph (V_t, A_t). As follows from Algorithm 1, the same path may belong to sets $\mathbf{P}_t$ constructed for different vectors t of operation processing times (since for any vector t we have $V_t \subseteq V^0$). Notation $s_u(t)$ is used for a schedule defined by path $\tau_u \in \mathbf{P}_t$. The objective function value calculated for schedule $s_u(t)$ is denoted as $\Phi(s_u(t))$.

A schedule is called *active* if none of the operations can start earlier than in this schedule, provided that the remaining operations could start no later. It is known (see Giffler and Thompson [3]) that a set of active schedules is dominant (i.e., it contains at least one optimal schedule) for any regular criterion. The following claim may be proven by induction with respect to number of machines m.

Theorem 1: *If $\mathbf{P}_t$ is set of all shortest paths from vertex SW_1 to the border vertices in the digraph (V_t, A_t), then set $\mathbf{P}_t$ defines all active schedules for the problem $F|n=2|\Phi$ with operation processing times defined by vector t.*

Let R^{2m} be space of non-negative 2m-dimensional real vectors $t = (t_{1,1}, t_{1,2}, \dots, t_{1,m}, t_{2,1}, t_{2,2}, \dots, t_{2,m})$ with Chebyshev's metric

$$d(t, t^0) = \max\{|t_{i,j} - t^0_{i,j}| : i \in \{1, 2\}, j \in \{1, 2, \dots, m\}\}$$

where $t^0 = (t^0_{1,1}, t^0_{1,2}, \dots, t^0_{1,m}, t^0_{2,1}, t^0_{2,2}, \dots, t^0_{2,m}) \in R^{2m}$. Let path $\tau_u \in \mathbf{P}_t$ be optimal for the problem $F|n=2|\Phi$ with operation processing times defined by vector t. If for any small positive real number $\varepsilon > 0$ there exists vector $t^0 \in R^{2m}$ such that $d(t, t^0) = \varepsilon$ and path τ_u is not optimal for the problem $F|n=2|\Phi$ with operation processing times defined by vector t^0, then optimality of path τ_u is *not stable*. Otherwise, optimality of path τ_u is *stable*.

Let $\delta(\tau_u)$ denote the set of all operations $O_{j,k}$, $j \in \{1, 2\}$, which are processed by machine $k \in M$ in such a way that at the same time job $i = 3 - j$ waits since operation $O_{i,k}$ (which is ready to be processed) needs the same machine k. Obviously, if $O_{1,k} \in \delta(\tau_u)$ (respectively, $O_{2,k} \in \delta(\tau_u)$), then trajectory defined by path τ_u includes a horizontal segment $[(x, y), SE_k]$ (vertical segment $[(x, y), NW_k]$).

Theorem 2: *Let path $\tau_u \in \mathbf{P}_t$ be optimal for the problem $F|n=2|\Phi$ where Φ is continuous increasing function of job completion times. Optimality of path τ_u is stable if and only if set $\mathbf{P}_t$ does not contain another optimal path for the problem $F|n=2|\Phi$ with operation processing times defined by vector t.*

Proof: *Sufficiency.* Since set of active schedules is dominant for any regular criterion, it is sufficient to compare schedule $s_u(t)$ with other active schedules. So due to Theorem 1, we have to compare path τ_u with other paths $\tau_v \in \mathbf{P}_t$, $\tau_v \neq \tau_u$. Since path τ_u is unique optimal path, we get inequality $\Phi(s_v(t)) - \Phi(s_u(t)) > 0$. Since Φ is increasing function, in order to overcome the difference $\Phi(s_v(t)) - \Phi(s_u(t))$ for the new vector t^0 of operation processing times, we have to increase the processing times for operations from the set $\delta(\tau_u)$ or (and) to decrease the processing times for operations from the set $\delta(\tau_v)$. Since Φ is continuous function, we can reach equality $\Phi(s_v(t^0)) - \Phi(s_u(t^0)) = 0$ only if $d(t, t^0) > 0$. Thus, optimality of path τ_u is stable.

Necessity. Let equality $\Phi(s_w(t)) = \Phi(s_u(t))$ hold. Since optimal paths τ_w and τ_u are different, either set $\delta(\tau_w) \setminus \delta(\tau_u)$ or set $\delta(\tau_u) \setminus \delta(\tau_w)$ is not empty. In the former case (we call it as case (a)), there exists at least one operation $O_{j,k} \in \delta(\tau_w) \setminus \delta(\tau_u)$ such that trajectory defined by path τ_w includes some segment of a boundary of rectangle H_k while trajectory defined by path τ_u does not include a segment of this boundary. In the latter case (we call it as case (b)), there exists at least one operation $O_{i,r} \in \delta(\tau_u) \setminus \delta(\tau_w)$ such that trajectory defined by path τ_u includes some segment of a boundary of rectangle H_r while trajectory defined by path τ_w does not include a segment of this boundary. Note that Φ is increasing function of job completion times.

Therefore, if in the case (a) we subtract any small positive value $\varepsilon > 0$ from the value $t_{j,k}$ with remaining the same all other components of the vector t, then we get such a vector t^0 of operation processing times that inequality $\Phi(\tau_w(t^0)) < \Phi(\tau_v(t^0))$ holds. On the other hand, if in the case (b) we add any small positive value $\varepsilon > 0$ to the value $t_{i,r}$ with remaining the same all other components of the vector t, then we get such a vector t^* of operation processing times that inequality $\Phi(\tau_w(t^*)) < \Phi(\tau_v(t^*))$ holds. Since value ε can be as small as desired, we conclude that optimality of path τ_u is not stable in both cases (a) and (b). ■

Returning to the Example 1, we see that the shortest path ((0, 0), SE_1, SE_2, NW_3, (x_3, $TM_{2,5}$)) from vertex (0, 0) to the border vertex (x_3, $TM_{2,5}$) in the digraph (V, A) is stable since set $\mathbf{P}_t$ does not contain another optimal path for the problem $F|n=2|\Phi$ with operation processing times defined by vector t given in Table 1.

Conclusion

Both Theorems 1 and 2 will be correct if flow shop problem $F|n=2|\Phi$ will be replaced by job shop problem $J|n=2|\Phi$.

To test whether optimality of the path $\tau_u \in \mathbf{P}_t$ is stable takes $O(m \log m)$ time for problem $F|n=2|\Phi$ and $O(m^2 \log m)$ time for problem $J|n=2|\Phi$. Indeed, we can use Algorithm 2 for the vector t of the operation processing times and construct optimal paths with different border vertices. Number of the optimal paths which have to be tested due to Theorem 2 is restricted by the number of border vertices asymptotically restricted by $O(m)$ for problem $F|n=2|\Phi$ and by $O(m^2)$ for problem $J|n=2|\Phi$.

It is easy to convince that for the above sufficiency proof of Theorem 2 we can replace increasing function Φ by non-decreasing function Φ. It should be noted that the most objective functions considered in classical scheduling theory are continuous non-decreasing functions of job completion times, e.g.,

- makespan C_{max},
- total completion time $\sum_{i=1}^{n} C_{i,m}$,
- maximal lateness $L_{max} = \max\{C_{i,m} - D_i : i \in J\}$ and
- total tardiness $\sum_{i=1}^{n} T_{i,m} = \sum_{i=1}^{n} \max\{0, C_{i,m} - D_i : i \in J\}$, where D_i denotes the given due date for a job i.

However, function $\Phi = \sum_{i=1}^{n} \mathrm{sign}(\max\{0, C_{i,m} - D_i\})$ equaled to the number of late jobs is not continuous, and so sufficiency of Theorem 2 may be violated in the break points of such a function Φ.

The research of the first author was supported by INTAS (Project 00-217).

Bibliography

1. S.B. Akers and J. Friedman, A non-numerical approach to production scheduling problems, Operations Research 3, 1955, 429 - 442.
2. P. Brucker, An efficient algorithm for the job-shop problem with two jobs, Computing 40, 1988, 353 - 359.
3. B. Giffler and G.L. Thompson, Algorithms for solving production-scheduling problems, Operations Research 8, N 4, 1960, 487 - 503.
4. W.W. Hardgrave and G. Nemhauser, A geometric model and graphical algorithm for a sequencing problem, Operations Research 11, N 6, 1963, 889 - 900.
5. Yu.N. Sotskov, Optimal scheduling two jobs with regular criterion, In: Design Processes Automating, Institute of Engineering Cybernetics, Minsk, 1985, 86 - 95 (in Russian).
6. Yu.N. Sotskov, The complexity of shop-scheduling problems with two or three jobs, European Journal of Operational Research 53, 1991, 326 - 336.
7. W. Szwarc, Solution of the Akers-Friedman scheduling problem, Operations Research 8, 1960, 782 - 788.

Author information

Yuri N. Sotskov - United Institute of Informatics Problems of National Academy of Sciences of Belarus, Surganov Str. 6, 220012 Minsk, Belarus; e-mail: sotskov@newman.bas-net.by

Nadezhda Yu. Sotskova - Hochschule Magdeburg-Stendal, Fachbereich Wasserwirtschaft, PSF 3680, D-39011 Magdeburg, Germany; OR Soft Jänicke GmbH, Geusaer Str. 104, FH, D-06217 Merseburg, Germany; e-mail: nadezhda.sotskova@orsoft.de