Bull. Nov. Comp. Center, Comp. Science, 10 (1999), 31–36

The Akers problem for variable operation processing times

O.A. Lyakhov

The well-known "job-shop" problem in the wider formulation (many machines of the same sort and many workers per one operation) is discussed in the paper. The graphic approach to the problem with two jobs and variable processing times of operations is proposed. It is shown that the more general problem can be formulated as creating a trajectory, consisting of the minimum number of one-unit-of-time segments.

1. Definition of the problem

A well-known so-called "job-shop" problem is to minimize a completion time of m jobs which pass through n machines. All the jobs orders of operations are different. All the processing times are known. Each machine processes only one operation every moment. The next operation of job must not start before the completion of previous operations.

The graphical approach to this problem was proposed in [1] for $m = 2$ which was developed by many scientists (see [2, p. 152]). The wider formulation of this scheduling problem, many machines of the same sort and many jobs of the same sort[1], is considered in this paper.

Designations:

$(L_1^i, L_2^i, \ldots, L_{r_1}^i)$ – the order of operations for job i, $i = \overline{1,2}$;

L_j^i – number of a resource (a machine) which is needed for operation j of job i;

V_j^i – the amount of resource L_j^i for operation j of job i;

r_i – the number of operations in job i;

R_k – the available amount of k – resource at each moment of time, $k = \overline{1,K}$;

F_j^i – the maximum possible amount of k – resource which may be used by operation j of job i in each time interval, $j = \overline{1,r_1}$, $i = 1,2$;

x_{jt}^i – the amount of resource j which is needed for operation j of job i in time interval t (intensity), $i = \overline{1,2}$, $j = \overline{1,r_1}$, $t = 1,2,\ldots$;

t_{Hj}^1 – the moment of starting operation j of job i, $i = \overline{1,2}$, $j = \overline{1,r_i}$;

t_0^i – the moment of finishing operation j of job i, $i = \overline{1,2}$, $j = \overline{1,r_i}$.

---

[1] For example, production lot.

The problem may be formulated as follows:

$$t^1_{H1} = t^2_{H1} = 0; \qquad (1)$$

$$t^i_{Hj} \geq t^i_{0j-1}, \qquad i=1,\ j=\overline{2,r_1} \qquad i=2,\ j=\overline{2,r_2}; \qquad (2)$$

$$\sum_{t=t^i_{Hj}}^{t^i_{0j}} x^i_{jt} = V^i_j, \qquad i=1,\ j=\overline{1,r_1} \qquad i=2,\ j=\overline{1,r_2}; \qquad (3)$$

$$0 \leq x^i_{jt} \leq F^i_j, \qquad i=1,\ j=\overline{1,r_1} \qquad i=2,\ j=\overline{1,r_2}; \qquad (4)$$

$$\sum_{\substack{i=1,2,\\ L^1_j=L^2_j=k}} x^i_{jt} \leq R_k, \qquad t=\overline{1,T}, \quad k=\overline{1,K}; \qquad (5)$$

$$T = \max\{t^1_{0r_1},\ t^2_{0r_2}\} \to \min\,. \qquad (6)$$

It is necessary to create the minimizing completion time schedule (6) (to find integer t^i_{Hj}, t^1_{0j}, t^2_{Hj}, t^2_{0j}, x^1_{jt}, x^2_{jt}), to process all the operations in the given orders (2) and in necessary volumes (3) with limited intensities (4), and to fulfil resources restrictions (5).

Every moment each operation needs the integer amount of resource. Intensities of operations have high and low levels and may be changed at integer moments. Zero intensity is corresponds to operation interruption. In comparison with the job shop problem the processing times are not known in advanced and are not fixed. They are connected with resources allocation in operations and among them. In practice, the volumes of operations and intensities are measured in natural or monitory units, but usually in timing units.

For example, an operation is to make a lot of the same two products with the use of three units of machines. The volume of operation is four hours. The high level of intensity is two hours per hour (the third machine is not needed). Some variants of resource allocation in the operation are given in Table 1.

Table 1

Variant	Intensities	Time, hours	
		intervals	processing
1	2	[0, 2]	2
2	2 1	[0, 1] [1, 3]	3
3	2 0 2	[0, 1] [1, 2] [2, 3]	3

2. Graphic form of the problem

Let us designate

$$x_u = \sum_{j=1}^{u} V^1_j, \quad u=\overline{1,r_1}; \qquad y_v = \sum_{j=1}^{v} V^2_j, \quad v=\overline{1,r_2}.$$

In the diagram (Figure 1), let us mark the points x_u ($u = \overline{1, r_1}$) on the axis X and the points y_v ($v = \overline{1, r_2}$) – on the axis Y. The schedule is an uninterrupted broken line between the coordinates (0,0) and (x_{r_1}, y_{r_2}) (trajectory).

A segment of trajectory is the part of operation which is processed with the constant intensity during one unit of time. The length of trajectory is the sum of all the segments. Each point on trajectory corresponds to the volume of all the completed operations: the coordinate along the axis X – the first job, along the axis Y – the second job.

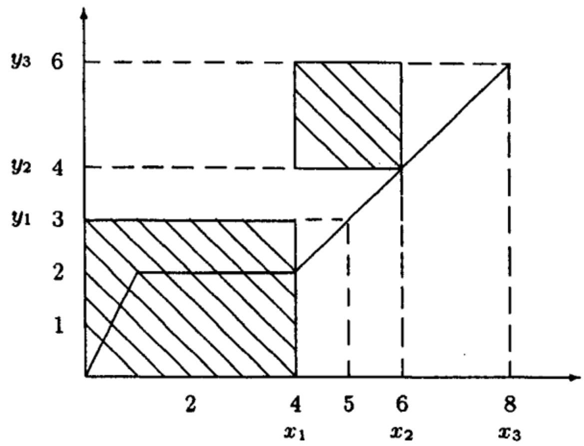

Figure 1

Like in [2, p. 112], let us call the rectangle P_{uv} with the apexes (x_{u-1}, y_{v-1}), (x_u, y_{v-1}), (x_{u-1}, y_v), (x_u, y_v) "special" if $L_u^1 = L_v^2$ and $F_u^1 + F_v^2 > R_{L_u^1}$ ($1 \leq u \leq r_1$, $1 \leq v \leq r_2$), i.e., if the operations u and v use the same resource and compete for it (in Figure 1, two special rectangles are marked for the example, given in Table 2).

Table 2

Operation	Job 1			Job 2		
	V	F	R	V	F	R
1	4	3	1	3	2	1
2	2	1	2	1	1	3
3	2	2	3	2	2	2

Here V is a volume, F is a maximal intensity, and R is a resource

Note: $R_1 = 3$, $R_2 = 2$, $R_3 = 3$.

The parameters of the trajectory shown in Figure 1 are given in Table 3.

Each segment of trajectory starts and finishes at an integer point. The intensity in a segment is defined as a distance from the beginning of the

Table 3

Coordinates of segments (x, y)	Time	Job 1			Job 2		
		Intensity	Resource	Operation	Intensity	Resource	Operation
(1, 2)	1	1	1	1	2	1	1
(4, 2)	2	3	1	1	0	1	1
(5, 3)	3	1	2	2	1	1	1
(6, 4)	4	1	2	2	1	3	2
(8, 6)	5	2	3	3	2	2	3

coordinates up to the ending of the coordinates, for job one – along the axis X, for job two – along the axis Y.

3. Trajectories in special rectangle

Let us designate by F_1, F_2 the maximum possible intensities of operations in the special rectangle $(0,0)$, $(0,Y)$, $(X,0)$, (X,Y),[2] R is an available amount of resource which can be used by operations at each moment of time. The problem is to find the minimum trajectory between the points $(0,0)$ and (X,Y) with restrictions: $F_1 \leq R$ and $F_2 \leq R$ (else F_1 or (and) F_2 may be determined equal to R); $F_1 + F_2 > R$ (else, the rectangle is not special).

Lemma 1 [3]. *If intensities and processing times are not to be necessarily integer, the minimum complete time of both operations (in a special rectangle) is:*

$$Q = \max\left\{\frac{X+Y}{R}, \frac{X}{F_1}, \frac{Y}{F_2}\right\}.$$

Let us designate by $T = \lceil Q \rceil$ a minimal integer such that $T \geq Q$.

Lemma 2 [3]. *If R, F_1, F_2, X, Y are integer, then there exists a trajectory between $(0,0)$ and (X,Y), consisting of T one-unit-of-time segments, so that ρ_i, μ_i, x_i, y_i are integer and the following constraints are valid:*

$$0 \leq \rho_i = x_i - x_{i-1} \leq F_1, \qquad i = \overline{1,T}, \tag{7}$$

$$0 \leq \mu_i = y_i - y_{i-1} \leq F_2, \qquad i = \overline{1,T}, \tag{8}$$

$$\rho_i + \mu_i \leq R, \qquad i = \overline{1,T}, \tag{9}$$

$$\sum_{i=1}^{T} \rho_i = X, \qquad \sum_{i=1}^{T} \mu_i = Y. \tag{10}$$

Here x_i and y_i are the coordinates of one-unit-of-time segments, ρ_i and μ_i are the intensities of operations in the segment i.

Remark. The minimum length of a trajectory between (x_1, y_1) and (x_2, y_2) (where (x_1, y_1), (x_2, y_2) belong to the same special rectangle and $x_1 \leq x_2$, $y_1 \leq y_2$) is $T = \lceil Q \rceil$ where

$$Q = \max\left\{\frac{x_2 - x_1 + y_2 - y_1}{R}, \frac{x_2 - x_1}{F_1}, \frac{y_2 - y_1}{F_2}\right\}.$$

A simple algorithm of finding a trajectory, consisting of T one-unit-of-time segments, was given in [3].

[2]Further we omit the indices of operations and resources.

Example. Let $F_1 = 2$, $F_2 = 3$, $R = 3$, $X = 4, Y = 3$. The minimal trajectories with their timing values (T) from $(0,0)$ to $(\alpha, 3)$, $\alpha = 1, \ldots, 4$, and to $(4, \beta)$, $\beta = 1, 2, 3$ in a special rectangle with apexes $(0,0)$, $(0,3)$, $(4,0)$, $(4,3)$ are given in Figure 2.

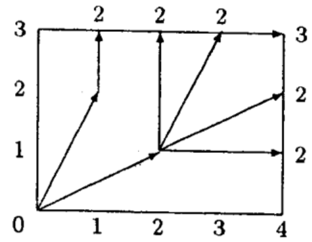

Figure 2

4. An algorithm for finding the schedule minimizing complete time of two jobs

The algorithm creates a graph, consisting of the paths $(0,0), \ldots, (x_{r_1}, y_{r_2})$, one of them being optimal.

Designations: J is a set of points (α, β) (nodes of a graph) which belong to the rectangle $(0,0)$, $(0, y_{r_2})$, $(x_{r_1}, 0)$, (x_{r_1}, y_{r_2}), so that α, β are integers and $0 \le \alpha \le x_{r_1}$, $0 \le \beta \le y_{r_2}$.

Each record of J consists of (α, β), $T(\alpha, \beta)$, $(\bar{\alpha}, \bar{\beta})$, where $T(\alpha, \beta)$ is the length of the trajectory $((0,0), \ldots, (\alpha, \beta))$; $(\bar{\alpha}, \bar{\beta})$ is a predecessor to (α, β), Γ is a set of nodes of a graph, consisting of an optimal path from $(0,0)$ to (x_{r_1}, y_{r_2}). Each record of Γ consists of $(\alpha, \beta), (\bar{\alpha}, \bar{\beta})$.

The scheme of the algorithm:

Step 1. Inclusion of the node $(0,0)$ in J, $T(0,0) = 0$, $\Gamma = \emptyset$.

Step 2. If J contains only one node (x_{r_1}, y_{r_2}), then stop.

Step 3. Selection from J of any node (α, β) (except (x_{r_1}, y_{r_2})).

Step 4. If (α, β) belongs to the special rectangle (x_u, y_v), (x_{u+1}, y_v), (x_u, y_{v+1}), (x_{u+1}, y_{v+1}) so that $(x_u \le \alpha < x_{u+1}$ and $\beta = y_v)$, or $(\alpha = x_u$ and $y_v \le \beta < y_{v+1})$, then go to Step 5, else go to Step 9.

Step 5. Calculating the lengths of trajectories in a special rectangle. The first node of the trajectory belongs to one of two sides: (x_u, y_v), (x_{u+1}, y_v) or (x_u, y_v), (x_u, y_{v+1}).

Using the formulas for the lengths of paths in special rectangles the lengths of the trajectories $(\alpha, \beta), \ldots, (d, t)$ are calculated, where (d, t) are the nodes on opposite sides of the rectangle, i.e., (α, y_{v+1}), $(\alpha+1, y_{v+1}), \ldots, (x_{u+1}, y_{v+1})$ is the first side, (x_{u+1}, β), $(x_{u+1}, \beta+1)$, $\ldots, (x_{u+1}, y_{v+1} - 1)$ is the second side.

The lengths of the trajectories $(0,0)$, ..., (α,β), (d,t) are the sums of $T(\alpha,\beta)$ and the lengths of paths from (α,β) to (d,t) in the special rectangle. The nodes between (α,β) and (d,t) are not kept.

Step 6. Saving in J the nodes (d,t) which were found at Steps 5, 9 (only one node is found at Step 9).

Step 7. Cutting off ineffective points from J. The point (x_{u+1},t) is ineffective if $T(x_{u+1},t) \geq T(x_{u+1},i)$, $i = \overline{t+1, y_{r_2}}$. By analogy, the point (t, y_{v+1}) is ineffective if $T(t,y_{v+1}) \geq T(i,y_{v+1})$, $i = \overline{t+1, x_{r_1}}$.

Step 8. Carrying the record (α,β), $(\bar{\alpha},\bar{\beta})$ from J to Γ and excluding the record (α,β), $T(\alpha,\beta)$, $(\bar{\alpha},\bar{\beta})$ from J. Go to Step 2.

Step 9. Creating a trajectory in non-special rectangles. A trajectory from (α,β) to (d,t) is created, according to the requirements of maximum intensities of operations. The point (d,t) is on a side of a special rectangle or on the boundary of the range of admissible solutions if a path $(\alpha,\beta),\ldots,(d,t)$ does not cross any non-special rectangle. Go to Step 6.

A result of the algorithm is a set of the points Γ which are on sides of special rectangles and the boundary of the range of admissible solutions. The optimal trajectory can be created from (x_{r_1}, y_{r_2}) to (0,0) by using addresses of the predecessors $(\bar{\alpha},\bar{\beta})$ which exist in records of Γ.

5. Conclusion

This algorithm will not have applications in practical management, because the number of jobs is not equal to two in real life problems, and increasing the number of jobs quickly exhausts the possibilities of computers.

The main aim of the article is to show that the more general problem (many same machines of the same sort, or many workers per one operation, etc.) can be formulated as creating a trajectory, consisting of the minimum number of one-unit-of-time segments.

References

[1] Akers S.B. A graphic approach to production scheduling problems // Operation Research. – 1956. – Vol. 4, No. 2.

[2] Tanaev V.S., Sostkov Yu.N., Strusevich V.A. Scheduling Theory. Multistage Systems. – Moscow: Nauka, 1989 (in Russian).

[3] Lyakhov O.A. Aker's problem for variable intensities of operations // Trudy VTc SO RAN. Sistemnoe modelirovanie. – Novosibirsk, 1997. – Issue 4 (22). – P. 48–58 (in Russian).