Control and Cybernetics

vol. **36** (2007) No. 2

On a bicriteria optimal production plan

by

Dorota Kuchta

Institute of Industrial Engineering and Management
Wroclaw University of Technology
ul. Smoluchowskiego 25, 50-370 Wroclaw, Poland
e-mail: Dorota.Kuchta@pwr.wroc.pl

Abstract: The classical mathematical programming problem used for the determination of a production plan maximising total income or profit is complemented with a second objective, concerning the makespan of the products being manufactured on two machines. As a result, a bicriterial integer linear programming problem is obtained, which can be solved by means of classical methods. A computational example is presented and discussed.

Keywords: production schedule, makespan, flow shop, open shop.

1. Introduction

In the paper we consider a classical mathematical programming problem used for the determination of a production plan maximising total income or profit, complemented with a second objective, concerning the makespan of the products being manufactured on two machines. This second objective is usually not considered directly as an objective or even in the constraints. Even if the limit of machine-hours is taken into account in the constraints, the idle time on the machines is usually not counted. And the makespan may be of high importance to the total profit as well as to the workers' satisfaction with working conditions.

Thus, we propose to combine in one bicriterial problem the objective of income (profit) maximisation and that of makespan minimisation. The compromise solutions will be determined by solving mixed 0-1 single-objective mathematical programming problem or a single objective mathematical programming problem with continuous variables together with several open shop problems. Two scheduling cases will be considered: the two machines flow shop and open shop problem.

2. Two machines flow shop problem

Before we go over to the two machines flow shop problem, we will consider two more general problems: the m machines job shop problem, which is a more general problem comprising the flow shop problem, and the m machines flow shop problem.

Let us consider a problem in which we have m different machines (M_k (k=1,..., m)) and n jobs J_j ($j = 1, ..., n$). Each job has to be processed on a sequence $\left(i_j^1, i_j^2, ..., i_j^{l(j)}\right)$ $\left(i_j^i \in \{1, ...m\}\right)$ of machines and the processing time of job J_j on machine M_k (it may be equal to zero for those machines on which the given job does not have to be processed) is equal to p_j^k. In each moment on each machine at most one job can be processed. We assume that we do not change the processing order of jobs on different machines. We can ask the question what is the optimal order in which the jobs should be processed so that the makespan M, representing the earliest moment when all machines are free, is minimal.

If we denote by $s_j(i_j^r)$ $(j = 1, ...n; r = 1, ...l(j))$ the start of the processing of job J_j on machine i_j^r, define 0-1 variables y_{ij} equal to 0 if job J_i precedes job J_j and to 1 otherwise, and choose C to be a sufficiently big constant, the problem can be solved by means of the following mixed linear problem (Dauer, Lin, 1990):

Objective function: $M \to \min$

Constraints ensuring that each job is processed on the correct sequence of machines:

$$s_j(i_j^r) \geqslant s_j(i_j^{r-1}) + p_j^{i_j^{r-1}} \quad (j = 1, \ldots, n;\ r = 2, \ldots, l(j)) \tag{1}$$

Constraints ensuring the needed meaning of the objective function:

$$s_j(i_j^{l(j)}) + p_j^{i_j^{l(j)}} \leqslant M \quad (j = 1, \ldots, n) \tag{2}$$

Constraints ensuring that in each moment on each machine there is at the most one job being processed:

$$\begin{aligned} & s_{j_1}(i_{j_1}^{r^*}) + p_{j_1}^{i_{j_1}^{r^*}} - s_{j_2}(i_{j_2}^{s^*}) \geqslant C y_{j_1 j_2} \\ & \text{for all } j_1^*, r^*, j_2^*, s^* \text{ such that } i_{j_1}^{r^*} = i_{j_2}^{s^*} = k,\ k = 1, \ldots, m \\ & y_{j_1 j_2} + y_{j_2 j_1} = 1 \text{ for all } j_1, j_2 = 1, \ldots, n \end{aligned} \tag{3}$$

The above problem is very hard from the computational point of view. In the paper we will concentrate on a special case: the flow shop problem.

In the flow shop problem the order of processing is fixed and the same for each job: each job has to go through M_1 first, then M_2 etc and finally through

M_m. Thus $(i_j^1, i_j^2, \ldots, i_j^{l(j)}) = (1, 2, \ldots, m)$ for each $j = 1, \ldots, n$. Here the above mixed linear problem takes on the following form:

$$M \to \min$$

$$s_j(k) \geqslant s_j(k-1) + p_j^{k-1} \quad (j = 1, \ldots, n;\ k = 2, \ldots, m) \tag{4}$$

$$s_j(m) + p_j^m \leqslant M \quad (j = 1, \ldots, n) \tag{5}$$

$$s_{j_1}(k) + p_{j_1}^k - s_{j_2}(k) \geqslant C y_{j_1 j_2}, \quad y_{j_1 j_2} + y_{j_2 j_1} = 1 \quad \text{for } k = 1, \ldots, m,\ j_1, j_2 = 1, \ldots, n. \tag{6}$$

If we consider the case with only two machines, the problems becomes easy to solve. Johnson (1954) presented an $\mathrm{O}(n \log n)$ algorithm for solving the problem. An optimal solution is found in the following way:

1) all the jobs $J_j (j = 1, \ldots, n)$ should be divided into two sets: T_1 is composed of those jobs for which the condition $p_j^1 \leqslant p_j^2$ is fulfilled, T_2 is formed by the remaining jobs;
2) jobs from T_1 are scheduled first, in the order of increasing p_j^1
3) then jobs from T_2 are scheduled, in the order of decreasing p_j^2.

It follows that in the linear programming formulation for the two machines case the number of variables (6) could be reduced. The model would be composed of (3)(4)(5) and

$$s_{j_1}(k) + p_{j_1}^k - s_{j_2}(k) \geqslant 0 \quad \text{for } j_1, j_2 \text{ such that } J_{j_1} \in T_1 \text{ and } J_{j_2} \in T_2 \tag{7}$$

$$s_{j_1}(k) + p_{j_1}^k - s_{j_2}(k) \geqslant C y_{j_1 j_2}, \quad y_{j_1 j_2} + y_{j_2 j_1} = 1 \quad \text{for } j_1, j_2 \text{ such that } J_{j_1}, J_{j_2} \in T_1$$

$$s_{j_1}(k) + p_{j_1}^k - s_{j_2}(k) \geqslant C y_{j_1 j_2}, \quad y_{j_1 j_2} + y_{j_2 j_1} = 1 \quad \text{for } j_1, j_2 \text{ such that } J_{j_1}, J_{j_2} \in T_2.$$

Now we will consider another two machines problem.

3. Two machines open shop problem

In the two machines open shop problem each job has to be processed on both machines, but the order can be any: M_1 first, then M_2 or the other way round. Thus, this is not a job shop, where the order of "visiting" the machines is fixed for each job. In Gonzales and Sahni (1976) an $\mathrm{O}(n)$ algorithm has been developed for solving the problem (again, with makespan minimization). We will not quote this algorithm here, but we will restrict ourselves to quoting the following theorem:

THEOREM 1 *The optimal makespan is equal to the maximum of*

$$\sum_{j=1}^{n} p_j^1,\ \sum_{j=1}^{n} p_j^2,\ \max_{j=1,\ldots,n} (p_j^1 + p_j^2).$$

4. Single-criterion production plan problem

Let us consider the following optimal production plan problem: suppose that a company manufactures n products, one unit of the j-th product generating the income c_j. Let $X = (x_1, \ldots, x_n)$ denote the vector of decision variables representing the amounts of each product to be manufactured in a given period, and $\mathbf{AX} \leqslant \mathbf{B}$ represent the constraints of the production plan. Then, of course, the optimal production plan for the objective "total income" can be found by means of the following programming problem and the classical methods of its solution:

$$\begin{aligned} &\sum_{j=1}^{n} c_j x_j \to \max \\ &\mathbf{AX} \leqslant \mathbf{B}, \quad \mathbf{X} \geqslant \mathbf{0}. \end{aligned} \tag{8}$$

In the next section we will consider an additional criterion for the optimal production plan problem.

5. Bi-criteria production plan problem

Let us suppose that the products considered in (8) should be processed on two machines (a generalisation to m machines for any m will be straightforward, it is just that it will be harder from the computational point of view). The total available processing time on the machines may or may not be a part of the constraints in (8), but the minimisation of the makespan may be an additional objective. In fact, the longer the makespan, the higher the cost of executing the obtained production plan, the higher the non-availability time of the machines and of the persons operating them for other tasks etc. For this reason we propose to complete (8) with additional criterion and consider the following bicriterial problem:

$$\begin{aligned} &\sum_{j=1}^{n} c_j x_j \to \max \\ &M \to min \\ &\text{constraints ensuring the proper meaning of } M \\ &\mathbf{AX} \leqslant \mathbf{B}, \quad \mathbf{X} \geqslant \mathbf{0}. \end{aligned} \tag{9}$$

We will consider the two machines flow shop and open shop problems. In the flow shop case each product j has to be processed through M_1 first, then through M_2, in the open shop case the order of processing trough both machines can be either. In both cases let us denote as t_j^1 (t_j^2) the time each unit of product j will need to be processed on machine M_1 (M_2), thus the processing times will be respectively $p_j^1 = t_j^1 x_j$ ($p_j^2 = t_j^2 x_j$) and will depend on the solution selected. We assume that processing of all the units of one product on one machine cannot be interrupted.

Thus we propose to solve the following bicriterial problem:

$$
\begin{aligned}
&\sum_{j=1}^{n} c_j x_j \to \max\\
&M \to \min\\
&\text{constraints ensuring the right meaning of } M\\
&\mathbf{AX} \leqslant \mathbf{B}, \quad \mathbf{X} \geqslant \mathbf{0}.
\end{aligned} \tag{10}
$$

For the flow shop case we make (10) concrete by using (4),(5),(7), as T_1 and T_2, which depend only on the relations between t_j^1 and t_j^2 for each $j = 1, \ldots, n$, will be known without knowing the actual processing times, which depend, in turn, on vector $\mathbf{X}$. Thus, we will solve the following problem:

$$
\begin{aligned}
&\sum_{j=1}^{n} c_j x_j \to \max\\
&M \to \min\\
&s_j(k) \geqslant s_j(k-1) + t_j^{k-1} x_j \quad (j = 1, \ldots, n;\ k = 2, \ldots, m)\\
&s_j(m) + t_j^m x_j \leqslant M \ (j = 1, \ldots, n)\\
&s_{j_1}(k) + t_{j_1}^k x_{j_1} - s_{j_2}(k) \geqslant 0 \text{ for } j_1, j_2 \text{ such that } J_{j_1} \in T_1 \text{ and } J_{j_2} \in T_2\\
&s_{j_1}(k) + t_{j_1}^k x_{j_1} - s_{j_2}(k) \geqslant C y_{j_1 j_2},\ y_{j_1 j_2} + y_{j_2 j_1} = 1\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{for } j_1, j_2 \text{ such that } J_{j_1}, J_{j_2} \in T_1\\
&s_{j_1}(k) + t_{j_1}^k x_{j_1} - s_{j_2}(k) \geqslant C y_{j_1 j_2}, \quad y_{j_1 j_2} + y_{j_2 j_1} = 1\\
&\qquad\qquad\qquad\qquad\qquad\qquad \text{for } j_1, j_2 \text{ such that } J_{j_1}, J_{j_2} \in T_2\\
&\mathbf{AX} \leqslant \mathbf{B}, \quad \mathbf{X} \geqslant \mathbf{0}.
\end{aligned} \tag{11}
$$

We can use any of the multiple objective linear programming methods and algorithms to solve the above problem, according to the needs of the decision maker – some algorithms determine a collection of solutions (e.g. the non-dominated ones) from which the decision maker has to choose one, some of them determine just one solution thanks to a prior modification of the problem according to the preferences of the decision maker (e.g. the lexigraphic approach). It is beyond the scope of the paper to discuss the individual methods, details can be found e.g. in Steuer (1986). We think that the most useful approach would be either to use the goal programming (e.g. Schniederjans, 1995), which is a method widely accepted in practical applications because of its simplicity and intuitiveness, or to determine the whole set of non-dominated solutions, which give the decision maker the possibility to select one from among a broad spectrum of solutions, changing with respect to the preferences given to individual objectives. An algorithm for determining such a set can be found e.g. in Gal (1977) and Dauer, Liu (1990). This algorithm allows to determine the set of non-dominated solutions in case of any number of objective functions. If there are just two objective functions, like in our case (with the objective functions and constraints being linear, which assures that the non-dominated set has "good" properties, see, e.g., Galas, Nykowski, Żółkiewski, 1987), it is also possible to find the set of all non-dominated solutions by means of solving a single-objective linear programming problem with a parameter in the objective function, where the parameter runs the interval [0,1]. This is the method, called weighted objective function approach (see, e.g., Steuer, 1986), that we use in the computational example. This method has the drawback that if the parameter takes on the value 0 and 1, which corresponds to not taking into account one of the objective functions, and if there are alternative solutions, the ones obtained may not be non-dominated. Thus, at the parameter values 0 and 1 it should be checked whether there are alternative solutions and if there are, all the basic ones should be checked (e.g. by the classical simplex algorithm) and the one which is non-dominated should be selected (there is always such one). This problem is discussed more deeply in Galas, Nykowski, Żółkiewski (1987) and Wierzbicki (1986). The set of non-dominated solutions is characterized in Gallager, Saleh (1993) and the problem of its stability in Gal and Wolf (1986).

For the open shop case we make (2) concrete in a two-stage algorithm. First of all, we determine the compromise values of both objective functions together with the production plans, without determining the corresponding schedules yet. Thanks to Theorem 1, this is possible by solving the following bicriterial problem:

$$
\begin{aligned}
&\sum_{j=1}^{n} c_j x_j \to \max \\
&M \to \min \\
&M \geqslant \sum_{j=1}^{n} t_j^k x_j \quad (k = 1, 2) \\
&M \geqslant (t_j^1 + t_j^2) x_j \quad (j = 1, \ldots, n) \\
&\mathbf{AX} \leqslant \mathbf{B}, \quad \mathbf{X} \geqslant \mathbf{0} \\
&\lambda \in [0, 1]\,.
\end{aligned}
\tag{12}
$$

In the second stage, knowing for the solution of (12) (again, obtained by means of any multicriteria linear programming algorithm) the corresponding production plan, we will be able to find, using the algorithm from Gonzales and Sahni (1976), the corresponding schedule.

6. Computational example

A company manufactures 4 products. The income generated by one unit of each of the products is as follows: $c_1 = 4$; $c_2 = 3$; $c_3 = 3$; $c_4 = 12$. Constraints $\mathbf{AX} \leqslant \mathbf{B}$ from (8) have the following form:

$$\begin{aligned}
&x_1 + 2x_2 + 15x_3 + 6x_4 \leqslant 90000\\
&2x_1 + 2x_2 + 15x_3 + 4x_4 \leqslant 120000.
\end{aligned}$$

The processing times of one unit of each product on each machine are as follows:

$$t_1^1 = 3;\ t_2^1 = 1;\ t_3^1 = 3;\ t_4^1 = 2;\ t_1^2 = 2;\ t_2^2 = 4;\ t_3^2 = 1;\ t_4^2 = 5.$$

Let us consider the flow shop case. Set T_1 will be composed of jobs representing the manufacturing of the 2nd and the 4th product, jobs representing the manufacturing of the 1st and 3rd product will form T_2. Thus, using the weighted function approach as a method of turning the bicriterial problem into a single objective one, we will get the following concretisation of (11):

$$\begin{aligned}
&\lambda(4x_1 + 3x_2 + 3x_3 + 12x_4) - (1-\lambda)M \to \max\\
&x_1 + 2x_2 + 15x_3 + 6x_4 \leqslant 90000\\
&2x_1 + 2x_2 + 15x_3 + 4x_4 \leqslant 120000\\
&s_1(1) + 3x_1 \leqslant s_1(2)\\
&s_2(1) + x_2 \leqslant s_2(2)\\
&s_3(1) + 3x_3 \leqslant s_3(2)\\
&s_4(1) + 2x_4 \leqslant s_4(2)\\
&s_1(2) + 2x_1 \leqslant M\\
&s_2(2) + 4x_2 \leqslant M\\
&s_3(2) + x_3 \leqslant M\\
&s_4(2) + 5x_4 \leqslant M\\
&s_2(1) + x_2 - s_1(1) \leqslant 0\\
&s_2(1) + x_2 - s_3(1) \leqslant 0\\
&s_4(1) + 2x_4 - s_1(1) \leqslant 0\\
&s_4(1) + 2x_4 - s_3(1) \leqslant 0\\
&s_2(2) + 4x_2 - s_1(2) \leqslant 0\\
&s_2(2) + 4x_2 - s_3(2) \leqslant 0\\
&s_4(2) + 5x_4 - s_1(2) \leqslant 0\\
&s_4(2) + 5x_4 - s_3(2) \leqslant 0
\end{aligned}$$

$$
\begin{aligned}
&s_2(1)+x_2-s_4(1)\leqslant Cy_{24}; \quad s_4(1)+2x_4-s_2(1)\leqslant Cy_{42}\\
&s_2(2)+4x_2-s_4(2)\leqslant Cy_{24}; \quad s_4(2)+5x_4-s_2(2)\leqslant Cy_{42}\\
&y_{24}+y_{42}=1\\
&s_1(1)+3x_1-s_3(1)\leqslant Cy_{13}; \quad s_3(1)+3x_3-s_1(1)\leqslant Cy_{31}\\
&s_1(2)+2x_1-s_3(2)\leqslant Cy_{13}; \quad s_3(2)+x_3-s_1(2)\leqslant Cy_{31}\\
&y_{13}+y_{31}=1\\
&y_{24},y_{42},y_{13},y_{31}=0,1, \text{other variables non-negative,}\\
&\lambda\in[0,1]\,.
\end{aligned}
$$

In case of $\lambda = 0$ and $\lambda = 1$ there are no alternative solutions, thus by solving the above problem we are sure to get all the basic non-dominated solutions. Here they are, presented only in the criteria space:

Table 1. Basic non-dominated solutions for the example

Solution	**Income** $\sum_{j=1}^{n} c_j x_j$	**Makespan** M
A1	0	0
A2	42 434	83 486
A3	209 423	112 500
A4	219 130	121 304
A5	270 000	240 000

The whole non-dominated set is composed of all the line segments combining the points listed above.

Solutions A1 and A5 correspond to the single-objective problems: the first one ignores the objective of income, the second one ignores the objective of makespan. As we can see, if we ignore the objective of makespan, we miss solutions like A3 and A4, where the income is not much smaller than the ideal one (income for A4 is by less than 20% smaller than that for A5, for A3 the corresponding difference is less than 25%) , but the makespan is considerably shorter (by almost 50% in A4 and by more than 50% in A3). The thus freed amount of time might be used to generate income significantly exceeding what we lose with respect to A5.

It might be useful to consider the following trade-off:

$$R(j)=\frac{\mathit{Income}(A_j)-\mathit{Income}(A_{j-1})}{\mathit{Makespan}(A_j)-\mathit{Makespan}(A_{j-1})} \quad (j=2,\ldots,5)).$$

It shows how much income we gain if we gradually pass from solution A_{j-1} to solution A_j by allowing one more unit of time for the makespan. This value should be compared with the additional cost linked to each additional unit of the makespan. The inverse of $R(j)$ shows how much time we gain if we gradually pass from solution A_j to solution A_{j-1} by consenting to lose one unit of income. This value should be compared to additional income we can get from the freed amount of time. Here are both trade-offs calculated for the example:

Table 2. Ratios characterising moves from one basic non-dominated solution to another one in the example

j	$R(j)$	$1/R(j)$
2	1.97	0.51
3	0.17	5.75
4	0.91	1.10
5	2.33	0.43

For example, if we are at point A4 with makespan equal to 121 304 time units and allow one time unit more for the makespan, we can increase our income by 2.33 units. The same decision taken in A3 would bring only 0.91 additional income unit. If each additional unit of makespan causes the additional cost of 1, the move from A4 towards A5 would be desirable, but the move from A3 towards A4 would not. Similarly, if we are at A5 (the point corresponding to the ideal income) and consent to give up one unit of income, we would gain only 0.43 units of time, the same move from A4 towards A3 would free for us more than 5 units of time. We would have to see how much additional income we can generate in the freed amount of time.

In Kaliszewski (2000) a method is proposed which allows to set bounds on the trade-offs in such a way that no solution with trade-offs above these bounds is generated by the algorithm.

7. Conclusions

We have proposed a way of introducing the makespan, as a second criteria, into the problem of determining an optimal production schedule. In this way, while looking for the optimal production plan, we take into account also idle states of machines, which obviously influence the cost and the satisfaction degree, but are not taken into account in classical formulation. The cases discussed comprise only the simplest scheduling cases and further research is needed to generalize the approach to other situations that occur in reality.

References

Dauer, J.P. and Liu, Y.-H. (1990) Solving multiple objective linear programs in objective space. *European Journal of Operational Research* **46**, 350–357.

Foo, S.Y., Yoshiyasu, T. and Szu, H. (1994) Job-shop scheduling based on modified tank-hopfield linear programming networks. *Engineering Applications of Artificial Intelligence* **7** (3), 321–327.

Gal, T. (1977) A general method for determining the set of all efficient solutions to a linear vectormaximum problem. *European Journal of Operational Research* **1**, 307–322.

Gal, T. and Wolf, K. (1986) Stability in vector maximization - a survey. *European Journal of Operational Research* **25**, 169-182.

Galas, Z., Nykowski, I. and Żółkiewski Z. (1987) *Programowanie wielokryterialne*. PWE, Warszawa.

Gallager, J.G. and Saleh, O.A. (1993) A representation of an efficiency equivalent polyhedron for the objective set of a multiple objective linear program. *European Journal of Operational Research* **80**, 214–212.

Gonzales, T. and Sahni, S. (1976) Open shop scheduling to minimize finish time. *J. Assoc. Comput. Mach.* **25**, 92–101.

Johson, S.M. (1954) Optimal two- and three-stage production schedules with setup times included. *Naval Res. Logist. Quart.* **1**, 61–86.

Kaliszewki, I. (2000) Using trade-off information in decision-making algorithms. *Computer & Operations Research* **27**, 161–182.

Schniederjans, M. (1995) *Goal Programming, Methodology and Applications.* Springer Verlag.

Steuer, R. (1986) *Multiple Criteria Optimization.* John Wiley and Sons, New York.

Wierzbicki, A. (1986) On the completness and constructiveness of parametric characterization to vector optimization problems. *OR Spectrum* **8**, 73–87.