Appl. Math. and Comp. Sci., 1994, vol.4, No.4, 635–641

GROUP SEQUENCING SUBJECT TO PRECEDENCE CONSTRAINTS†

MIKHAIL Y. KOVALYOV*, ALEXANDER V. TUZIKOV*

The problem of sequencing jobs for processing on a single machine to minimize maximum penalty is studied. It is assumed that the jobs are classified into several families on the basis of group technology. Precedence constraints are specified on the set of jobs for each family and on the set of families. The Lawler's polynomial time algorithm is generalized to solve this problem and some problems with two ordered criteria.

1. Introduction

We consider the problem of sequencing n jobs for processing on a single machine without idle times. Each job is available for processing at time zero and it is assigned to one of F families on the basis of group technology (Ham *et al.*, 1985; Mitrofanov, 1966; Potts and Van Wassenhove, 1991). Jobs of a specific family have to be processed continously. An arbitrary (binary, transitive, irreflexive) relation $\rightarrow$ is determined on the set of jobs for each family. If jobs i and j belong to the same family and $i \rightarrow j$, then the processing of job j cannot be started before job i has finished processing. The analogous relation for which we use the same notation is determined on the set of families. If I and J are some families and $I \rightarrow J$, then processing jobs of family J cannot be started before all jobs of family I have finished processing. For each family J, a machine set-up time $s_J \geq 0$ is required immediately before the jobs of this family are processed. A processing time $p_j > 0$, a due date $d_j > 0$ and a weight $w_j > 0$ are specified for each job j. Given a job sequence, the completion time C_j for each job j is easily determined. The objective is to find a sequence which satisfies the group technology and precedence constraints and minimizes a maximum penalty function

$$f_{max}(C_1, \ldots, C_n) = \max\Big\{f_j(C_j) | j = 1, \ldots, n\Big\}$$

where $f_j(t)$, $j = 1, \ldots, n$, are some non-decreasing real-valued functions.

Adopting the notation for scheduling problems of Lawler *et al.* (1989), we denote our problem by $1/GT, prec/f_{max}$, where acronyms GT and $prec$ indicate that the group technology and precedence constraints, respectively, have been specified on the set of jobs. For this problem without the group technology constraint, i.e. when there is only one family, an $O(n^2)$ algorithm is presented by (Lawler, 1973). For

† This work was supported by INTAS grant No.93-257

* Institute of Engineering Cybernetics, Belarus Academy of Sciences, Surganova 6, 220012 Minsk, Belarus

the problem with the group technology constraint and no precedence constraints to minimize maximum lateness (in this case $f_j(t) = t - d_j$, $j = 1, \dots, n$), a simple $O(n \log n)$ algorithm is described by (Potts and Van Wassenhove, 1991).

In this paper, the problem with the group technology and precedence constraints is considered. The paper consists of four sections. In the next section, Lawler's $O(n^2)$ algorithm for $1/prec/f_{max}$ is generalized to solve $1/GT,\ prec/f_{max}$ in $O(Fn^2)$ time. Problems, for which this time complexity may be decreased, are indicated. In the following section, we show how to adopt our algorithm to solve some single machine group scheduling problems with two ordered criteria. The last section summarizes results of the paper.

2. Generalization of the Lawler Algorithm

It is convenient to introduce some terminology. Given a set Q, an element $x^0 \in Q$ is called a *minimal element* of the set Q with respect to the relation $\rightarrow$ if there is no element $x \in Q$ such that $x^0 \rightarrow x$. The set of all minimal elements of the set Q with respect to $\rightarrow$ is denoted by Q^-.

We first consider the problem in which there is only one family and the corresponding set-up time is zero. In this case, the following $O(n^2)$ algorithm presented by Lawler (1973) constructs an optimal job sequence.

Algorithm 1 *(for one family).*

INITIALIZATION: Set $J_n = \{1, \dots, n\}$, $P_n = \sum_{j=1}^{n} p_j$.

GENERAL STEP Construct J_n^-. Find $i_n \in J_n^-$ such that $f_{i_n}(P_n) = \min\{f_l(P_n) | l \in J_n^-\}$ settling ties arbitrarily. If $n = 1$, then STOP: an optimal job sequence $(i_1, \dots, i_n)$ is constructed. Otherwise, set $J_{n-1} = J_n - \{i_n\}, P_{n-1} = P_n - p_{i_n}$ and repeat General Step for $n := n - 1$.

We now assume that there is more than one family. Algorithm 1 is then generalized in the following way.

Algorithm 2 *(for F families).*

INITIALIZATION Set $H_F = \{1, \dots, F\}$, $R_F = \sum_{j=1}^{n} p_j + \sum_{I=1}^{F} s_I$.

GENERAL STEP Construct H_F^-. For each family $L \in H_F^-$, apply Algorithm 1 in which $J_n = \{L\}$ and $P_n = R_F$. If $\pi_L = (i_1, \dots, i_k)$ is the sequence given by this algorithm, then calculate the value of this sequence

$$\Phi_L(R_F) = \max\left\{ f_{i_l}(C_{i_l}) | l = 1, \dots, k \right\}$$

where C_{i_l} is the completion time of job i_l in the sequence π_L subject to $C_{i_k} = R_F$. Find family I_F such that $\Phi_{I_F}(R_F) = \min\{\Phi_L(R_F) | L \in H_F^-\}$ settling ties arbitrarily. If $F = 1$, then STOP: an optimal sequence of families $(\pi_{I_1}, \dots, \pi_{I_F})$ is constructed. Otherwise, set $H_{F-1} := H_F - \{I_F\}$, $R_{F-1} := R_F - (s_{I_F} + \sum_{j \in I_F} p_j)$ and repeat General Step for $F = F - 1$.

To demonstrate the application of Algorithm 2, consider the problem with six jobs having processing times $p_1 = 2, p_2 = 3, p_3 = 1, p_4 = 3, p_5 = 4, p_6 = 2$ and due dates $d_1 = 8, d_2 = 5, d_3 = 10, d_4 = 4, d_5 = 7, d_6 = 3$. The jobs are divided into three families, say A, B and C, so that $A = \{1,2,3,4\}, B = \{5\}$ and $C = \{6\}$. Set-up times are $s_A = 2, s_B = 1, s_C = 3$. The precedence constraints are given so that $B \to C$ for the families and $1 \to 2, 4 \to 3$ for the jobs. The objective is to minimize the maximum lateness, i.e. penalty functions are $f_j(C_j) = C_j - d_j$ for $j = 1, \dots, 6$. Algorithm 2 solves this problem as follows.

1. Set $F = 3$, $H_3 = \{A, B, C\}$. Calculate $R_3 = 2+3+1+3+4+2+2+1+3 = 21$. Construct $H_3^- = \{A, C\}$.
2. (Application of Algorithm 1 for family A)
   - 2.1. Set $n = 4, J_4 = \{A\} = \{1,2,3,4\}$, $P_4 = R_3 = 21$. Construct $J_4^- = \{2,3\}$. Calculate $f_2(P_4) = 21 - d_2 = 16$, $f_3(P_4) = 21 - d_3 = 11$. Since $\min\{f_2(P_4), f_3(P_4)\} = f_3(P_4)$, then set $i_4 = 3$.
   - 2.2. Set $n = 3$, $J_3 = \{1,2,4\}$, $P_3 = P_4 - p_3 = 20$. Construct $J_3^- = \{2,4\}$. Calculate $f_2(P_3) = 20 - d_2 = 15$, $f_4(P_3) = 20 - d_4 = 16$. Set $i_3 = 2$.
   - 2.3. Set $n = 2$, $J_2 = \{1,4\}$, $P_2 = P_3 - p_2 = 17$. Construct $J_2^- = \{1,4\}$. Calculate $f_1(P_2) = 17 - d_1 = 9$, $f_4(P_2) = 17 - d_4 = 13$. Set $i_2 = 1$.
   - 2.4. Set $n = 1$, $J_1 = \{4\}$, $P_1 = P_2 - p_4 = 14$. Construct $J_1^- = \{4\}$. Calculate $f_4(P_1) = 14 - d_4 = 10$. Set $i_1 = 4$. Thus, $\Phi_A(R_3) = \max\{f_4(P_1), f_1(P_2), f_2(P_3), f_3(P_4)\} = 15$.
3. (Application of Algorithm 1 for family C)
   - 3.1. Set $n = 1$, $J_1 = \{C\} = \{6\}$, $P_1 = R_3 = 21$. Construct $J_1^- = \{6\}$. Calculate $f_6(P_1) = 21 - d_6 = 18$. Thus, $\Phi_C(R_3) = f_6(P_1) = 18$.
4. Since $\min\{\Phi_A(R_3), \Phi_C(R_3)\} = \Phi_A(R_3)$, we set $I_3 = A$ and $\pi_{I_3} = \pi_A = (i_1, i_2, i_3, i_4) = (4,1,2,3)$.
5. Set $F = 2$, $H_2 = \{B, C\}$. Calculate $R_2 = R_3 - (2+9) = 10$. Construct $H_2^- = \{C\}$.
6. (Application of Algorithm 1 for family C)
   - 6.1. Set $n = 1$, $J_1 = \{C\} = \{6\}$, $P_1 = R_2 = 10$. Construct $J_1^- = \{6\}$. Calculate $f_6(P_1) = 10 - d_6 = 7$. Thus, $\Phi_C(R_2) = f_6(P_1) = 7$, $I_2 = C$ and $\pi_{I_2} = \pi_C = (6)$.
7. Set $F = 1$, $H_1 = \{B\}$. Calculate $R_1 = R_2 - (3+2) = 5$. Construct $H_1^- = \{B\}$.
8. (Application of Algorithm 1 for family B)
   - 8.1. Set $n = 1$, $J_1 = \{B\} = \{5\}$, $P_1 = R_1 = 5$. Construct $J_1^- = \{5\}$. Calculate $f_5(P_1) = 5 - d_5 = -2$. Thus, $\Phi_B(R_1) = f_5(P_1) = -2$, $I_1 = B$ and $\pi_{I_1} = \pi_B = (5)$.
9. Finally, an optimal job sequence is $(\pi_B, \pi_C, \pi_A) = (5,6,(4,1,2,3))$. The corresponding optimal objective value is $\max\{\Phi_B(R_1), \Phi_C(R_2), \Phi_A(R_3)\} = \max\{-2, 7, 15\} = 15$.

The following theorem holds.

Theorem 1. *Algorithm 2 solves problem* $1/GT,\ prec/f_{max}$.

Proof. To prove this theorem, it is easy to adopt the proof of the correctness of Lawler's algorithm (Lawler 1973). The idea of our proof is as follows. Let H be the set of families that have no successors with respect to the relation $\rightarrow$ and let R be the sum of all the processing times and all the set-up times. Consider the problem of scheduling jobs of some family $L \in H$ to minimize f_{max} subject to the relation $\rightarrow$ and the condition that all jobs are available for processing at time $R - (s_L + \sum_{i \in L} p_i)$. Let π_L be an optimal sequence of jobs for this problem and let $\Phi_L(R)$ be the corresponding value of the criteria f_{max}. If $\Phi_K(R) = \min_{L \in H}\{\Phi_L(R)\}$, then there exists an optimal solution to the problem $1/GT,\ prec/f_{max}$ in which jobs of family K are sequenced last in π_K order. Since K is the group which is chosen by Algorithm 2, repeating this argument shows that Algorithm 2 solves the problem $1/GT,\ prec/f_{max}$. ■

Since Algorithm 1 applied for family I has $O(|I|^2)$ running time, the time complexity of Algorithm 2 does not exceed $O(F\sum_{I=1}^{F}|I|^2)$, or equivalently, $O(Fn^2)$ if computations of penalty functions are not taken into consideration. This time complexity can be decreased in some cases as indicated in (Kovalyov and Tuzikov, 1991). For example, $O(n^2)$ and $O(n\log n)$ algorithms can be developed for solving $1/GT,\ prec/f_{max}$ and $1/GT/f_{max}$, respectively, if

$$f_j(C_j) \in \{L_j = C_j - d_j, T_j = \max\{0, L_j\}, U_j = sign(T_j)\}, j = 1, \ldots, n$$

If $f_j(C_j) = w_jT_j$ for $j = 1, \ldots, n$, then $O(Fn\log^2 n)$ algorithm can be developed for $1/GT/f_{max}$ using results of (Hochbaum and Shamir, 1989). If jobs have different release dates r_j and equal due dates $d_j = d$, $j = 1, \ldots, n$, then the problems $1/r_j, GT,\ prec/f_{max}$ and $1/r_j, GT/f_{max}$, where $f_j(C_j) = f(C_j - d)$, $j = 1, \ldots, n$, can be solved in $O(n^2)$ and $O(n\log n)$ times, respectively. Furthermore, Algorithm 2 can naturally be generalized to the more complicated situation where some families are divided into subfamilies and so on. In this case, to calculate the values of f_{max} for families, it is necessary to calculate the values of f_{max} for subfamilies. This recursive procedure is fulfilled until we reach jobs.

3. Bicriterion Problems

Algorithm 2 can also be applied to solve some single machine bicriterion problems with the group technology constraint. In this section, we give some examples.

The first problem, denoted by $1/GT,\ prec/(g_{max}, f_{max})$, is to minimize f_{max} on the set of all optimal sequences for $1/GT,\ prec/g_{max}$ where g_{max} is another maximum penalty function. In order to solve this problem, we begin by applying Algorithm 2 and finding the objective function optimal value g^* for $1/GT,\ prec/g_{max}$. Then we solve $1/GT,\ prec/f_{max}$ using Algorithm 2 modified in the following way. In General Step of Algorithm 1, which is the part of Algorithm 2, we redefine $f_l(P_n) = \infty$ for $l \in J_n^-$ if $g_l(P_n) > g^*$. It is not difficult to prove that this modification of Algorithm 2 solves $1/GT,\ prec/(g_{max}, f_{max})$ in $O(Fn^2)$ time.

In the second problem, $1/GT/(\sum w_jC_j, f_{max})$, it is assumed that there are no precedence constraints and f_{max} is minimized on the set of all optimal sequences for

$1/GT/\sum w_jC_j$. To solve this problem, we first determine a relation $\rightarrow$ on the set of jobs for each family $J, J = 1, \ldots, F$, so that for any pair i, j of jobs from the same family, the following property holds

$$i \rightarrow j \quad \text{if and only if} \quad p_i/w_i < p_j/w_j$$

Then we determine relation $\rightarrow$ on the set of families so that for any pair I, J of families, such that

$$I \rightarrow J \quad \text{if and only if} \quad \Big(s_I + \sum_{j\in I} p_j\Big) / \sum_{j \in I} w_j < \Big(s_J + \sum_{j \in J} p_j\Big) / \sum_{j\in J} w_j$$

The following theorem holds.

Theorem 2. *The sequence of jobs is optimal for* $1/GT/\sum w_jC_j$ *if and only if it is feasible with respect to relation* $\rightarrow$.

Proof. This theorem is easily proved if we consider families as *composite jobs* with processing times $P_I = s_I + \sum_{j\in I} p_j$ and weights $W_I = \sum_{j\in I} w_j$. Then, following (Smith, 1956), we show that there is no interchange of the jobs or composite jobs which improves the objective value if the job sequence is feasible with respect to relation $\rightarrow$. Conversely, if a sequence is not feasible with respect to $\rightarrow$, then there is an interchange of the jobs or composite jobs which improves the objective value. ∎

Thus, $1/GT/(\sum w_jC_j, f_{max})$ is equivalent to $1/GT$, $prec/f_{max}$, where precedence constraints are given by the relation $\rightarrow$. Therefore, we can apply Algorithm 2 to solve it.

We note that an analog of Theorem 2 can be proved for a more general function than the weighted sum of completion times. Let us assume that for the function $f(\pi)$ there exists a function $r(\pi)$ such that for any job sequences $\pi' = (u, a, b, v)$ and $\pi'' = (u, b, a, v)$ the following statement holds

$$r(a) < r(b) \quad \text{implies} \quad f(\pi') < f(\pi'')$$

Here u, a, b, v are disjoint subsequences (strings) of jobs. In this case, following the definition of Lawler (1983), we say that the function $f(\pi)$ possesses a *strong string interchange property*. As well as the weighted sum of completion times, the function

$$f(\pi) = \sum_{j\in\{\pi\}} \Big(w_j \exp(\gamma C_j) + b_j\Big), \qquad \gamma \neq 0$$

possesses this property. In this case, we have

$$r(\pi) = \Big(f(\pi) - \sum_{j\in\{\pi\}} b_j\Big) / \Big(1 - \exp(\gamma \sum_{j\in\{\pi\}} p_j)\Big)$$

These and other examples of functions possessing the strong string interchange property can be found in (Shafransky and Tuzikov, 1991; Tanaev *et al.*, 1984).

If function f has the strong string interchange property, then we define relation $\rightarrow$ on the set of jobs for each family so that, for any pair i, j of jobs from the same family, $i \rightarrow j$ if and only if $r(i) < r(j)$. We define relation $\rightarrow$ on the set of families so that, for any pair I, J of families, $I \rightarrow J$ if and only if $r(\pi_I) < r(\pi_J)$. Here π_I and π_J are any sequences of jobs of families I and J, respectively, feasible with respect to relation $\rightarrow$. Then the following theorem holds.

Theorem 3. *If the function f possesses the strong string interchange property, then the sequence of jobs is optimal for $1/GT/f$ if and only if it is feasible with respect to relation $\rightarrow$.*

Proof. An interchange argument presented by (Smith, 1956) can easily be adopted to prove this theorem. ■

Theorem 3 is the basis to apply Algorithm 2 for solving the problem $1/GT/(f, f_{max})$ where the function f has the strong string interchange property. This theorem shows, in fact, how to describe the set of all optimal sequences of jobs using a binary relation on the set of jobs. Therefore, it is closely related to the results of Monma and Sidney (1987) and Tuzikov (1985), where the problem of describing the set of all optimal sequences is studied for the case of precedence constraints on the set of jobs.

4. Conclusion

The main result of this paper is the generalization of the Lawler algorithm (Lawler, 1973) to the case of the group technology constraint. This generalized algorithm is applied to solve a number of single machine group scheduling problems with various maximum penalty functions and some hierarchical bicriterion problems where the second criterion is a maximum penalty. We believe that the application of the presented algorithm is not limited by the problems considered and there exist other real-life situations including the group technology constraint where it can be used.

References

Ham I., Hitomi K. and Yoshida Y. (1985): *Group Technology.* —Dordrecht: Kluwer-Nijhoff.

Hochbaum D.S. and Shamir R. (1989): *An $O(n \log^2 n)$ algorithm for the maximum weighted tardiness problem.* — Information Processing Letters, v.31, pp.215–219.

Kovalyov M.Y. and Tuzikov A.V. (1991): *Minimizing maximum penalty on a single machine under group technology and precedence constraints.* — Preprint No.35, Institute of Engineering Cybernetics, Academy of Sciences of Belarus, Minsk.

Lawler E.L. (1973): *Optimal sequencing a single machine subject to precedence constraints.* — Management Science, v.19, pp.544–546.

Lawler E.L. (1983): *Recent results in the theory of machine scheduling,* In: Mathematical Programming: the State of the Art, (Bachem A., Grotschel M., Korte B. Eds.). — Berlin: Springer, pp.202–234.

Lawler E.L., Lenstra J.K., Rinnooy Kan A.H.G. and Shmoys D.B. (1989): *Sequencing and scheduling: algorithms and complexity.* — Report 8934/*A*, Econometric Institute, Erasmus University, Rotterdam.

Mitrofanov S.P. (1966): *Scientific Principles of Group Technology.* — Yorkshire: National Lending Library.

Monma C.L. and Sidney J.B. (1987): *Optimal sequencing via modular decomposition: characterization of sequencing functions.* — Mathematics of Operations Research, v.12, pp.22–31.

Potts C.N. and Van Wassenhove L.N. (1991): *Integrating scheduling with batching and lotsizing: a review of algorithms and complexity.* — J. Operational Research Society, v.43, pp.395–406.

Shafransky Y.M. and Tuzikov A.V. (1991): *Sequencing with ordered criteria.* — Preprint No.36, Institute of Engineering Cybernetics, Academy of Sciences of Belarus, Minsk.

Smith W.E. (1956): *Various optimizers for single-stage production.* — Naval Research Logistics Quarterly, v.3, pp.59–66.

Tanaev V.S., Gordon V.S. and Schafransky Y.M. (1984): *Scheduling Theory. Single-Stage Systems.* — Moscow: Nauka (in Russian).

Tuzikov A.V. (1985): *Methods of solving some classes of multicriterion scheduling problems.* — Cand. physic.-mathem. sciences thesis: Institute of Engineering Cybernetics, Minsk (in Russian).

Received: November 26, 1993
Revised: April 28, 1994