Mathl Comput. Modelling, Vol. 13, No. 3, pp. 81–83, 1990
Printed in Great Britain.

0895-7177/90 $3.00 + 0.00

A SIMPLER AND FASTER ALGORITHM FOR OPTIMAL TOTAL-WORK-CONTENT–POWER DUE DATE DETERMINATION

S. L. VAN DE VELDE
Centre for Mathematics and Computer Science, P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

Abstract—Due date determination problems comprise scheduling problems in which not only the scheduling of n jobs is involved, but the assignment of due dates to jobs as well. This paper considers the case where a schedule is given and there is a single decision variable involved: the due date multiplier. The multiplier is used to set the due dates in order to minimize a composite objective function. Recently, an $O(n^2)$ algorithm was presented, the validity of which was proved on basis of the dual of the linear programming formulation of this problem. We give a simpler and faster algorithm based upon strictly primal arguments, requiring only $O(n \log n)$ time. In addition, we point out that these arguments can be employed for alternative proofs in case of a common due date.

1. PROBLEM FORMULATION

Due date determination problems comprise scheduling problems in which not only the scheduling of jobs is involved, but the assignment of due dates to jobs as well. Usually, due dates need to be set by use of some *due date assignment rule*, and the objective function expresses the desire to let the completion times of the jobs coincide with their due dates.

This paper addresses the problem of determining due dates by use of the so-called "total-work-content–power" rule for a *given* schedule; such a problem typically occurs when the overall problem is difficult to solve and optimization takes place by implicit enumeration of all possible schedules. Recently, Cheng [1] presented an analytical method to solve this problem. Let us describe the features of the problem in detail. Throughout the paper, we adopt Cheng's notation, except that we assume that the jobs are numbered according to increasing completion times.

There is one machine available, on which a set N of n independent jobs have to be scheduled. The machine can handle only one job at a time and machine idleness is not permitted. Job i $(i = 1, \ldots, n)$ requires a positive uninterrupted processing time t_i. A *schedule* defines the order in which the jobs pass through this machine. Given a schedule, the completion time for job i $(i = 1, \ldots, n)$, denoted by C_i, is simply determined by $C_i = \Sigma_{j \leqslant i} t_j$. Job i is called *tardy* if its completion time exceeds its due date d_i. Similarly, a job is *early* if it is completed before its due date. The total-work-content–power rule assigns for each job i $(i = 1, \ldots, n)$ a due date d_i according to

$$d_i = k t_i^m,$$

where m is a given exponent and k is the *due date multiplier* (Cheng uses the term "due date multiple factor").

The optimality criterion is a linear combination of two components that are both defined in terms of the due date multiplier. This first part of the cost function is proportional with this multiplier, the second part reflects the linear penalties inflicted if jobs deviate from their due dates. The objective function can be written as

$$f(k) = \sum_{i \in N} (\alpha k + |C_i - d_i|), \tag{1}$$

where α is the cost per unit value of k. The optimal due date determination problem for a *given* schedule can then be formulated as

$$\min \{f(k) | k \geqslant 0\}. \tag{2}$$

Note that the due date multiplier is the *only* decision variable involved.

Cheng reformulates problem (2) in terms of a linear programming problem. The dual of this linear programming problem and the strong duality theorem of linear programming are required to prove the following theorem. This theorem is needed to determine the optimal due date multiplier.

Theorem 1 [1]. Define for each job i $(i = 1, \ldots, n)$ the sets $s(i)$ and $S(i)$ as

$$s(i) = \{j: C_i t_j^m < C_j t_i^m, j \neq i, j \in N\}$$

and

$$S(i) = \{j: C_i t_j^m \geqslant C_j t_i^m, j \neq i, j \in N\},$$

respectively. The optimal due date multiplier is $k^* = C_r / t_r^m$, where $r \in N$ is such that

$$t_r^m - n\alpha \geqslant \sum_{j \in S(r)} t_j^m - \sum_{j \in s(r)} t_j^m \geqslant -t_r^m - n\alpha.$$

Note that a procedure based upon this theorem requires $O(n^2)$ time.

In the next section, we present a simpler and faster algorithm for solving problem (2) to optimality, for which only *primal* arguments are needed. Its running time is shown to be $O(n \log n)$. Since the algorithm we propose is based upon generic arguments, some extensions will be given in Section 3.

2. A SIMPLER AND FASTER ALGORITHM

The crucial argument for the new algorithm is the fact that the cost functions

$$g_i: k \to |C_i - k t_i^m| \qquad (i = 1, \ldots, n)$$

are *convex* functions in $\mathbb{R}^+ \cup \{0\}$. As the sum of a finite number of convex functions is again a convex function, we assert that $f(k)$ *is a convex piecewise linear function in* k. Since $f(k)$ is non-differentiable, the above notions are extremely important. They tell us that every *locally* optimal point is *globally* optimal, and that an optimal point is attained in a point of non-differentiability, or *breakpoint*, of the cost function $f(k)$. Proofs of these assertions are straightforward [e.g. 2] and therefore omitted.

Taking a closer look at the function $f(k)$, we detect that the points of non-differentiability are, besides the value $k = 0$, those values of k for which $C_i = k t_i^m$, for some i $(i = 1, \ldots, n)$. Let us assume, without loss of generality, that this gives us precisely $n + 1$ points of non-differentiability. The next step is to sort these $n + 1$ values in increasing order, and number them accordingly. Let $k_{[i]}$ denote the $(i + 1)$th smallest breakpoint, while k_i is the breakpoint that corresponds with job i. The last step is to evaluate sequentially the objective values $f(k_{[i]})$ for $i = 0, \ldots, n$, until we find a local optimum and, because of the convexity of $f(k)$, thereby global optimum. We deduce from equation (1) that, for any k, the cost function is

$$f(k) = \left(\sum_{i \in T_k} C_i - \sum_{i \in E_k} C_i \right) + k \left(n\alpha - \sum_{i \in T_k} t_i^m + \sum_{i \in E_k} t_i^m \right), \tag{3}$$

where, for that particular value of k, E_k denotes the set of early and just-in-time jobs, and T_k the set of late jobs, respectively. Note that $N = E_k \cup T_k$ for each $k \geqslant 0$, and that these sets remain intact between two subsequent points of non-differentiability. Hence, $f(k)$ can be written as

$$f(k) = \left(\sum_{i \in N} C_i - 2 \sum_{i \in E_{k_{[j]}}} C_i \right) + k \left(n\alpha - \sum_{i \in N} t_i^m + 2 \sum_{i \in E_{k_{[j]}}} t_i^m \right), \quad k \in [k_{[j]}, k_{[j+1]}], \quad j = 0, \ldots, n. \tag{4}$$

The evaluation of $f(k)$ at all the $n \times 1$ breakpoints takes no more than $O(n)$ time altogether. As $f(k)$ consists of $n + 1$ components, the calculation of $f(k_{[0]})$ takes linear time. However, the objective value at each of the subsequent breakpoints can be evaluated in *constant* time. At each breakpoint, exactly one specified job becomes just-in-time, namely the one associated with this breakpoint. As can be seen in equation (4), only a fixed number of operations is needed to define

and evaluate the linear function on the next interval. The overall running time of the algorithm is therefore $O(n \log n)$, since it is dominated by the time complexity of the sorting routine.

By now, we have proven the following theorem:

Theorem 2. Let k^* be the smallest breakpoint of $f(k)$ for which

$$n\alpha - \sum_{i \in N} t_i^m + 2 \sum_{i \in E_{k^*}} t_i^m \geqslant 0.$$

Then k^* is the optimal due date multiplier. It can be determined in $O(n \log n)$ time.

To illustrate this result, we consider the example from Ref. [1]. The processing times and other relevant data for the jobs can be found in Table 1.

Table 1

i	1	2	3	4	5
t_i	1	2	4	8	10
t_i^2	1	4	16	64	100
C_i	1	3	7	15	25
k_i	1	$\frac{3}{4}$	$\frac{7}{16}$	$\frac{15}{64}$	$\frac{1}{4}$

On $[0, \frac{15}{64}]$ and $[\frac{15}{64}, \frac{1}{4}]$, the objective functions are $f(k) = 51 - 184k$ and $f(k) = 21 - 56k$, respectively. For $k \in [\frac{1}{4}, \frac{7}{16}]$, we have $f(k) = 29 + 144k$. Hence, by use of Theorem 2, the optimal due date multiplier is $k^* = \frac{1}{4}$, and the corresponding objective value is $f(\frac{1}{4}) = 7$.

3. EXTENSIONS

The presented algorithm can be extended in two directions. As long as we are provided with a set of completion times, the above procedure remains valid. This means that machine idleness, job splitting, multiple machines, different machine speeds and capacities can be included as well, since they do not affect the solution procedure.

Furthermore, the general idea of the algorithm is applicable to any piecewise linear convex (or concave) function in a single variable for which the points of non-differentiability can be readily inspected [cf. 3, 4]. Therefore, it extends to problems in which the jobs have a *common* due date, that is, $d_i = k$ for each i. For instance, Cheng [5] considered a common due date problem in which the objective function to be minimized is

$$f(k) = \sum_{i \in N} w_i |C_i - k|, \tag{5}$$

where w_i denotes the weighting factor of job i, and where again the job completion times are given. In contrast to Cheng, the reader might employ purely primal arguments to prove the following theorem:

Theorem 3. The optimal common due date coincides with the completion time of the first job in the schedule p, say, job r, for which

$$2 \sum_{i=1}^{r} w_i - \sum_{i=1}^{n} w_i \geqslant 0.$$

REFERENCES

1. T. C. E. Cheng, Optimal total-work-content–power due-date determination and sequencing. *Computers Math. Applic.* **14,** 579–582 (1987).
2. C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, N.J. (1982).
3. M. H. Karwan and B. Ram, A Lagrangian dual-based solution method for a special linear programming problem. *Computers Ops Res.* **14,** 67–73 (1987).
4. S. L. van de Velde, *Minimizing Total Completion Time in the Two-machine Flow Shop by Lagrangian Relaxation.* Report OS-R8808, Centre for Mathematics and Computer Science, Amsterdam (1988).
5. T. C. E. Cheng, On optimal common due date determination. *IMA Jl math. Mgmt* **1,** 39–43 (1986).