OPERATIONS RESEARCH, Vol. 24, No. 4, July-August 1976

Scheduling Parallel Production Lines with Changeover Costs: Practical Application of a Quadratic Assignment/LP Approach

A. M. GEOFFRION and G. W. GRAVES
University of California, Los Angeles, California

(Received original June 27, 1975; final, February 3, 1976)

Production orders for a number of products must be scheduled on a number of similar production lines so as to minimize the sum of product-dependent changeover costs, production costs, and time-constraint penalties. We treat the problem by a quadratic assignment algorithm with a linear programming adjustment, and describe a successful practical application for chemical reactor scheduling.

THE BASIC problem addressed by this paper can be stated as follows (a more elaborate variant will be introduced later). There are several "similar" continuous process facilities or flow-shop production lines operated in parallel. Each can make (process) some subset of products with a known *production rate* of R_{pl} units per day for product p on line l, at a *production cost* of P_{pL} dollars per unit. A *transition cost* of $T_{p,p',l}$ dollars is incurred whenever line l changes over from making product p to product $p.'$ A number of independent *production orders* are given; the k^{th} order specifies a *demand* of D^k units of product p^k to be produced between an *earliest start time* E^k (commonly due to restrictions on raw material availability) and a *latest finish time* L^k (commonly due to customer delivery commitments or a known schedule for further processing). It is desired to find a *production schedule*—which line produces which products and when over a scheduling *horizon* of H days—meeting the given production orders at minimum total cost (transition plus production). Individual production orders may be split into two or more parts to be run noncontiguously on the same line or on different lines. The product that each line is producing immediately prior to the start of the scheduling period is known, so that the proper transition cost (if any) can be charged for the initial product scheduled on each line.

Scheduling/sequencing problems with changeover costs arise frequently in the continuous process industries and in bulk production and packaging. Prabhakar [7] refers, for instance, to chemical reactors and processors that, when switched from one product to another, yield a certain amount of off-grade material before the reaction or process is stabilized at the new operating conditions. The lower market value of the off-grade material, which depends on the products between which a transition is made, is responsible for the changeover cost. Scheduling problems with sequence-dependent changeover costs also arise for multi-use bulk production and packaging lines in the food and consumer goods industries, for continuous treatments in materials fabrication industries, and so on.

Related Literature

A search of the sequencing literature (see, e.g., the surveys of references 2 and 4) revealed a dearth of previous studies closely related to the problem posed above.

Prabhakar [8] developed a mixed integer linear programming model for a similar class of problems that is more general in some respects (e.g., inventory capacities are taken into account) but more restrictive in others (e.g., split runs of a product may not occur on the same line). This model appears satisfactory so long as the number of products that can be made on each line is moderate, for the number of required 0–1 variables equals the sum of the squares of these numbers. Thus five lines, each of which can make any of 15 products, would require $5(15^2)=1125$ binary variables—which would be well beyond the capability of existing mixed integer linear programming codes [5]. (This is the approximate size of the practical problem that led to our interest in this class of problems.)

Deane and White [3] (see also the other references cited therein) developed a specialized branch-and-bound algorithm for a similar class of problems with identical production lines, no time constraints, no job splitting, and with a workload-balancing constraint on the maximum makespan deviation of each line from the average. (In Section 2 we present arguments against the use of such rigid constraints and introduce into our own model a penalty function version of the workload-balancing constraint.) It is difficult to assess the efficiency of Deane and White's algorithm because only preliminary computational experience is presented with fairly small randomly generated problems and no computer times are given.

The only other directly germane reference we are able to cite is Balas, Enos, and Graves' successful treatment of a magazine-bindery scheduling problem [1]. Their quadratic assignment problem approach served, in fact, as the chief inspiration for the approach taken in the present paper.

Outline of the Paper and Summary of Results

Section 1 develops an exact representation of the basic scheduling problem as a quadratic assignment problem, provided two production-line similarity assumptions hold. This representation is of interest because it opens this class of scheduling problems to a new line of attack.

Section 2 points out certain practical difficulties that follow from a literal interpretation of the basic scheduling problem as stated above. In particular, the optimal solution may be unduly sensitive to problem data (e.g., forecast demands) that are normally subject to considerable managerial discretion. We suggest a more flexible statement of the scheduling problem that avoids this objectionable possibility. The new problem formulation is more appealing from a practitioner's viewpoint, diminishes the importance of the production-line similarity assumptions, and permits better control of the size (and hence difficulty) of the associated quadratic assignment problem.

Section 3 describes how the quadratic assignment formulation can be modified in a natural manner, with the help of a linear programming adjustment, to adddress the new problem formulation.

Finally, we describe a practical application to chemical reactor scheduling in which our approach demonstrated a clear superiority over manual solutions produced by experienced schedulers. This application involves scheduling about 25 products monthly on six reactors at Dart Industries, Inc. The computerized scheduling program has been adopted and implemented by the firm as the primary scheduling technique for one of the major product lines.

The computational burden of our approach is sufficiently light, and its flexibility sufficiently great, that it appears promising for various generalizations of the problem class treated here.

1. FORMULATION AS A QUADRATIC ASSIGNMENT PROBLEM

The objective of this section is to give an exact quadratic assignment formulation of the basic problem posed at the outset.

Throughout this section we will assume that the production lines are *similar* to one another in the sense that their production rates and transition costs are proportionally related. More precisely, we impose the following two assumptions.

A1. There is a *standard daily production rate* $\bar{R}_p$ units per day for each product p and a *production proportionality constant* α_l for each line l such that

$$R_{pl}=\alpha_l\bar{R}_p \text{ for all pairs } pl \text{ such that } R_{pl}>0. \tag{1}$$

A2. There is a *standard transition cost* $\bar{T}_{p,p'}$ for changeover from product p to product p', and a *transition cost proportionality constant* β_l for each line, such that

$$T_{p,p',l} = \beta_l \bar{T}_{p,p'} \tag{2}$$

for all triples p,p', l such that line l can produce both products p and p'.

These assumptions hold exactly if all lines are identical. They can be expected to hold to close approximation if all lines are basically of the same technological design except for the scale of implementation or the "speed" of operation. Such was the case in the practical application described in Section 4. Even when the two proportionality assumptions do not hold to very close approximation, it can be argued on statistical grounds that least squares estimates of $\bar{R}_p\ \alpha_l$, $\bar{T}_{p,p'}$, and β_l (aimed at enforcing (1) and (2)) will yield good results for applications where the number of products and the amount produced in a given schedule are large with respect to the number of lines. Moreover, we should point out that the more realistic problem reformulation given at the end of Section 2 greatly mitigates the need for assumptions A1 and A2 to hold exactly.

Quantization

A necessary prerequisite for a quadratic assignment problem formulation is a finite quantization of both time and production. In particular, each production order must be expressed in terms of equal indivisible *lots* and the scheduling period [O,H] on each line must be partitioned into equal indivisible time *slots*. This must be done in such a way that each production lot corresponds to the actual amount of work a line can do in any given time slot.

Such a quantization can be achieved without introducing any error (thanks to the proportionality assumptions) if a suitable basic *time quantum* τ is selected. The choice of τ induces a *standard lot size* of $\bar{R}_p\tau$ units for product p and a *standard slot size* of τ/α_l days for line l. Notice that line l can make $R_{pl}\tau/\alpha_l \overset{(1)}{=} (\alpha_l\bar{R}_p)(\tau/\alpha_l) = \bar{R}_p\tau$ units of product p in one time slot (if it can make any at all), which is exactly one standard lot's worth. The choice of τ further implies that there will be a total of

$$m_l \triangleq H/(\tau/\alpha_l) \tag{3}$$

time slots available on line l, and that the k^{th} production order will be partitioned into

$$n^k \triangleq D^k/(\bar{R}_{p^k}\tau) \tag{4}$$

lots of product p^k that must be scheduled to commence after slot

$$E^k/(\tau/\alpha_l) \tag{5}$$

and finish no later than slot

$$L^k/(\tau/\alpha_l) \tag{6}$$

if line l is used. All of the quantities identified in (3) through (6) will be integers if τ is taken as a common divisor (preferably the greatest common divisor) of the numbers $H\alpha_l, D^k/\bar{R}_{p^k}, E^k\alpha_l, L^k\alpha_l$.

We require that the total number of lots equals the total number of slots,

$$m \triangleq \sum_l m_l = \sum_k n^k. \tag{7}$$

Clearly $\sum_l m_l \geq \sum_k n^k$ must hold or the given problem is not feasible. Equality will be enforced if necessary by adding a *dummy* product with the right number of lots (see Note 1).

Expression as a Quadratic Assignment Problem

Since any single lot of product p can be produced during any single time slot on line l (so long as $R_{pl} \neq 0$), the scheduling problem can be stated succinctly as: find a one-to-one map of the $\sum_l m_l$ time slots onto the $\sum_k n^k$ lots that minimizes the total corresponding costs (which include a prohibitive cost for any disallowed or time-infeasible assignment). More precisely, define ρ to be a one-to-one mapping from the ordered set

$$\{\underbrace{1,\cdots,m_1}_{\text{line 1}};\ \underbrace{m_1+1,\cdots,m_1+m_2}_{\text{line 2}};\ \cdots;\ \cdots m\} \qquad \text{(time slots)} \tag{8}$$

onto the ordered set

$$\{\underbrace{1,\cdots,n^1}_{\text{order 1}};\ \underbrace{n^1+1,\cdots,n^1+n^2}_{\text{order 2}};\ \cdots;\ \cdots m\} \qquad \text{(production lots).} \tag{9}$$

Thus $\rho(i)=h$ means that the i^{th} slot is assigned to the h^{th} lot and the problem can be stated as

$$\sum_{i=1}^{i=m} \sum_{j=1}^{j=m} q_{ij} c_{\rho(i),\rho(j)} + \sum_{i=1}^{i=m} a_{i,\rho(i)} \tag{10}$$

over all one-to-one maps ρ, where the $m \times m$ matrices Q, C and A are defined as explained below. Problem (10) is the desired quadratic assignment problem expressed in the notation of Graves and Whinston [6], the computational approach adopted in our implementation.

The C-matrix re-expresses the standard transition costs of (2) in terms of lots. A typical element $c_{h,h'}$ must equal the standard transition cost associated with line changeover from lot h to lot h'. The lot ordering specified by (9) makes it easy to relate h and h' back to their products, resulting in

$$C \triangleq \begin{array}{c} \\ n_1\{ \\ n_2\{ \\ n_3\{ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} \overbrace{\qquad\quad}^{n_1} & \overbrace{\qquad\quad}^{n_2} & \overbrace{\qquad\quad}^{n_3} & \qquad \end{array} \\ \left[\begin{array}{c:c:c:c} 0 & \bar{T}_{p^1,p^2} & \bar{T}_{p^1,p^3} & \cdots \\ \hdashline \bar{T}_{p^2,p^1} & 0 & \bar{T}_{p^2,p^3} & \cdots \\ \hdashline \bar{T}_{p^3,p^1} & \bar{T}_{p^3,p^2} & 0 & \cdots \\ \hdashline \vdots & \vdots & \vdots & \ddots \end{array}\right] \end{array}$$

Fig. 1. The C-matrix.

a specification for C in partitioned form as given in Fig. 1. (where every entry in a given cell is identical and has the indicated value).

The Q-matrix serves two purposes. The locations of its non-zero entries indicate the direction of time by identifying which time slot immediately follows which time slot on the same line. The magnitude of the non-zero elements conveys the transition cost proportionality constants β_l. The appropriate Q-matrix is 0 except for a band of β's immediately above the principal diagonal; see Fig. 2.

With C and Q defined, it is clear that the first term of the objective function of (10) gives the total transition cost associated with ρ. To see this, notice that the highly special structure of Q implies that the only ij pairs that actually appear in the summation are those for which $j=i+1$ and i is not a final time slot for any of the lines ($i \neq m_1$ or m_1+m_2 or $\cdots$ or m). Denote the set of final time slots by I_f. Then the first term of (10) can be

$$Q \triangleq \begin{array}{c} \\ m_1\{ \\ m_2\{ \\ \\ \end{array} \begin{array}{c} \begin{array}{ccc} \overbrace{\qquad\qquad}^{m_1} & \overbrace{\qquad\qquad}^{m_2} & \qquad \end{array} \\ \left[\begin{array}{c:c:c} \begin{matrix} 0 & \beta_1 & & 0 \\ 0 & 0 & \beta_1 & \\ & \ddots & \ddots & \\ 0 & & 0 & \beta_1 \\ & & 0 & 0 \end{matrix} & 0 & \cdots \\ \hdashline 0 & \begin{matrix} 0 & \beta_2 & & 0 \\ 0 & 0 & \beta_2 & \\ & \ddots & \ddots & \\ 0 & & 0 & \beta_2 \\ & & 0 & 0 \end{matrix} & \cdots \\ \hdashline \vdots & \vdots & \ddots \end{array}\right] \end{array}$$

Fig. 2. The Q-matrix.

written as

$$\sum_{i \notin I_f} q_{i,+1} c_{p(i),p(i+1)}, \tag{11}$$

which consists, for each non-final time slot i, of the β-factor for the corresponding line multiplied by the standard transition cost associated with changeover from the product made during time slot i to the product made during the next time slot on the same line.

The A-matrix has typical entry $a_{i,h}$, which should be thought of as a penalty (or incentive, if negative) associated with assigning slot i to lot h. Its functions are to account for: (i) the fact that certain products cannot be made on certain lines; (ii) the production costs; (iii) the transition cost (if any) incurred at the very beginning of the scheduling period because of the product made at the end of the previous period; (iv) a prohibitive penalty for scheduling a lot of order k prior to its stipulated earliest start time E^k or after its stipulated latest finish time L^k.

One may compose A as a sum of $m \times m$ matrices,

$$A = A^{00} + A^0 + \sum_k A^k, \tag{12}$$

where A^{00} accounts for (i) and (ii), A^0 for (iii), and A^k for (iv) as it pertains to order k.

The coefficient $a_{i,h}^{00}$ will be a very large number if the line corresponding to slot i cannot make the product corresponding to lot h (refer to (8) and (9)). Otherwise, it is equal to $(\bar{R}_{pT})P_{pl}$ for the product p corresponding to lot h and the line l corresponding to lot i.

The entries of A^0 will be 0 except for the rows corresponding to the first time slot for each line (rows 1, m_1+1, m_1+m_2+1, etc.). If product p_1^0 is the initial condition on line 1, then $a_{1,h}^0$ is set equal to β_1 times the standard transition cost from product p_1^0 into the product corresponding to lot h (assuming line 1 can make this latter product). The other first slot rows are specified similarly.

Prohibitive penalties for violating $E^k(L^k)$ are reflected by large positive entries in the columns of A^k corresponding to order k and the rows corresponding to slots that begin before (end after) time $E^k(L^k)$. Of course, if the E^k's and L^k's are such that the problem as originally stated is infeasible, then the quadratic assignment problem solution incurs one or more of these large penalty entries.

2. PRACTICAL CONSIDERATIONS

We have given an exact quadratic assignment problem representation of the scheduling problem posed at the outset. There are, however, at least two practical difficulties yet to be faced.

The Time Quantum τ May Be Too Small

Recall from Section 1 that the time quantum τ must be a common divisor of a certain set of numbers if the quadratic assignment representation is to be exact. Since the map size m varies inversely with τ (refer to (3), (4) and (7)), it is possible that m may be too large to permit optimization of (10) in a reasonable amount of computer time. Even the best available method for obtaining optimal or nearly optimal solutions to quadratic assignment problems, which we believe to be of the type described in reference 6, becomes onerous for m much above the range of 250 or so. This suggests using a larger value of τ than may be justified in a strict sense.

We believe that τ can and should be adjusted if necessary to keep m under control. After all, in most practical applications there is a good deal of managerial discretion as to the choices of the numbers H, D^k, E^k and L^k, which, in turn, determine all of the quantities of which τ is supposed to be a greatest common divisor. This discretion can be used to achieve integral values of m_l and n^k of reasonable size without noticeable distortion of the problem.

The Original Problem Statement May Be Too Inflexible

The second practical difficulty is more subtle: the original problem statement is fine in spirit, but tends to be unrealistically rigid and inflexible. Demand quantities, production order time constraints, and the scheduling period horizon are assumed known as precise numbers. Yet a minute change in one or more of these numbers could conceivably permit a significantly lower cost schedule to become feasible. In other words, the cost of the optimal production schedule could be unduly sensitive to the given data (see Note 2). From a modeling viewpoint, acute sensitivity is unacceptable because the problem data are not known with such great accuracy. Most modelers would be willing to bend the quantities D^k, E^k, L^k and H slightly if this were to result in significantly reduced costs.

A simple example will clarify the nature of the improvements sought by relaxing the rigidity of the problem statement. Consider two hypothetical lines that can both make any of four products with equal efficiency. Product A (C) was being produced on line 1 (2) at the end of the previous scheduling period. Demands are for 15 days' worth each of A and B, 16 days' worth of C, and 14 days' worth of D. The transition costs for each line are all \$1,000 except for $A \rightarrow D$, which is \$200. There are no time constraints and the scheduling period is 30 days. Under these assumptions an obvious optimal schedule is: produce A for 15 days and then B for 15 days on line 1, and C for 16 days and then D for 14 days on line 2. The total transition cost is \$2,000 and both lines run for exactly 30 days. An alternative schedule that runs line 1 for 29 days and line 2 for 31 days and that has transition costs of only \$1,200 is: produce A for 15 days and then D for 14 days on line 1, and C for 16 days and then B for 15 days on line 2. It is likely that this less expensive schedule will be preferred to the first, even though the two lines do not finish their work exactly at the end of the 30th day. The next scheduling period would be one day longer for line 1 and one day shorter for line 2; an alternative might be to adjust the demands for products A and D (B and C) upward (downward) by a total of one day so as to equalize the finish times of both lines.

A variation of the above example in which the $D \rightarrow C$ transition costs \$700 illustrates a related point. The optimal 30-day rigid schedule is to produce A for 15 days, D for 14 days, and finally C for 1 day on line 1; and C for 15 days followed by B for 15 days on line 2. The total transition cost is \$1,900. The second schedule mentioned above (with a cost of \$1,200) is again likely to be preferred to the rigidly optimal one, which now suffers from an "unnecessary" split in the production of product C. Splits can be advantageous in some circumstances but are usually undesirable when one of the split runs is very short.

These two examples show that rigid observance of the original problem statement can lead to missed opportunities for taking advantage of good product sequences and/or can lead to undesirable split production runs. There are two possible remedies. One is to keep the rigid problem statement but to do extensive sensitivity analysis on the "soft" data (i.e., on the D^k's, E^k's, L^k's and H). The other remedy, far superior in our opinion, is to adopt a less rigid problem statement that is inherently less sensitive to problem data.

An Improved Problem Statement

A better problem statement would give lower and upper limits bracketing D^k instead of a single target number, would allow E^k and L^k to be violated for a price (say, for a penalty cost of $\pi_1{}^k$ and $\pi_2{}^k$ dollars per day, respectively), and would allow individual lines to finish their work earlier or later than time H for a penalty cost of $\pi_0{}^l$ dollars per day (either long or short).

An additional benefit of such a reformulation is that the first practical difficulty concerning the size of τ now becomes largely academic. Such flexibility is exactly what we need to keep the map size m within a tractable range. Moreover, the production line similarity assumptions of Section 1 become much less important.

It is evident that a more flexible problem statement of this type removes the scheduling problem from the realm of purely combinatorial formulations.

No longer can it be quantized so as to be representable exactly as a quadratic assignment problem. Yet so long as the sequencing and production costs are strongly dominating in comparison to the penalty costs, an approximate quadratic assignment formulation is the natural choice as the foundation of a practical solution procedure. Such a procedure is given in the next section.

3. A HYBRID QUADRATIC ASSIGNMENT/*LP* APPROACH

The preceding section proposed a more flexible statement of the scheduling problem in order to avoid practical difficulties associated with the original problem statement. If a quadratic assignment algorithm alone were to be applied to the new problem statement, the natural way to do so would be as follows. Lengthen H by 10 or 15 % and add enough demand for a dummy product (see Note 1) to take up the additional production capacity. (This will act as a kind of "slack variable" in each of the line horizon constraints.) Select τ so that integral m_l's and n^k's can be obtained via (3) and (4) using H and D^k's within the allowed range of their target values, and so that m is not too large. Incorporate the time penalty costs (π_1^k, π_2^k, and π_0^l of Section 2) into the A-matrix in the obvious way. For instance, the "prohibitive" penalties of matrix A^k described in Section 1 would be replaced by finite values calculable in a straightforward manner from E^k, L^k, π_1^k, π_2^k and the known duration τ/α_l of time slots on each line. We have found that this quadratic assignment approximation to the revised problem does in fact work quite well.

A still better way to approach the more flexible problem statement, however, is to employ linear programming in tandem with the quadratic assignment algorithm. The basic idea is to use the quadratic assignment algorithm in the manner just described in an effort to determine the best sequencing of products on each of the lines and then to use linear programming to adjust the lengths of the various production runs thereby determined so as to minimize the nonsequencing costs subject to the given minimum-maximum range on each of the D_k's. This is the natural division of labor: a combinatorial algorithm for the combinatorial part of the problem and LP for the remaining "continuous" part of the problem.

To indicate more precisely how the linear program would be set up, it is necessary to extract from the solution of the quadratic assignment problem the pure sequencing information it conveys—production quantities and timing information can be suppressed. The fundamental entity pertaining to sequencing is the production *campaign*, defined as an uninterrupted period of production on a single line earmarked for a single production order. A *campaign sequence* is a time-sequence of campaigns specified as to production line and production order but not specified as to duration. A convenient way to express a campaign sequence is to define a partitioned index set

$$\Gamma \triangleq \{1, \cdots, \gamma_1; \gamma_1+1, \cdots, \gamma_1+\gamma_2; \cdots; \cdots \textstyle\sum_l \gamma_l\} \tag{13}$$

that numbers campaigns in chronological order by line (γ_l is the number of campaigns on line l), and also to define a campaign sequence function $K(\cdot)$ on Γ that specifies the production order index of each campaign. Thus $K(9) = 3$ means that the 9th campaign is part of the 3rd production order. Moreover, if we suppose that $\gamma_1 = 8$, then Γ identifies the 9th campaign as the $(9-8) = 1$st campaign on line 2.

Associated with any one-to-one map ρ from (8) to (9) there will be a unique and easily extracted campaign sequence—call it $K_\rho(\cdot)$ with domain Γ_ρ—identified by the maximal assigned sequences of time-consecutive lots, on the same production line, corresponding to the same production order. The campaign sequence associated with the quadratic assignment problem solution, ρ^0, contains all the information about ρ^0 needed by the linear program.

The function of the linear program can now be stated succinctly: without altering the quadratic assignment campaign sequence (K_{ρ^0} on Γ_{ρ^0}), determine the duration of each campaign so as to minimize the sum of production costs and all time penalties associated with E^k and L^k violations and with departures from H in the finishing time of each production line (see Note 3), subject to the requirement that the total quantity produced by campaigns associated with each production order k must be within the specified limits bracketing D^k.

The natural formulation of this problem has as many variables as there are campaigns, as many two-sided demand constraints as there are production orders, and an objective function that includes piecewise-linear convex penalty terms. Figure 3 illustrates a typical penalty term associated with a campaign corresponding to a production order k with $E^k>0$. Here s stands for the sum of the durations of all campaigns that occur before the subject campaign on the same line (these are readily identified from Γ), and π_1^k is the per diem penalty rate for violation of E^k (as defined at the end of Section 2).

All penalty terms can be re-expressed with the help of a standard trick so that the natural formulation is converted into an ordinary linear program. For example, the penalty term of Figure 3, which can be expressed as $\operatorname{Max}\{0, \pi_1^k E^k - \pi_1^k s\}$, can be replaced in the objective function by an auxiliary variable σ subject to the linear constraints $\sigma \geq 0$ and $\sigma \geq \pi_1^k E^k - \pi_1^k s$.

The approximate quadratic assignment formulation given at the outset of this section may not lead to the best possible campaign sequence in the global sense. The obvious way to deal with this possibility is to iterate the

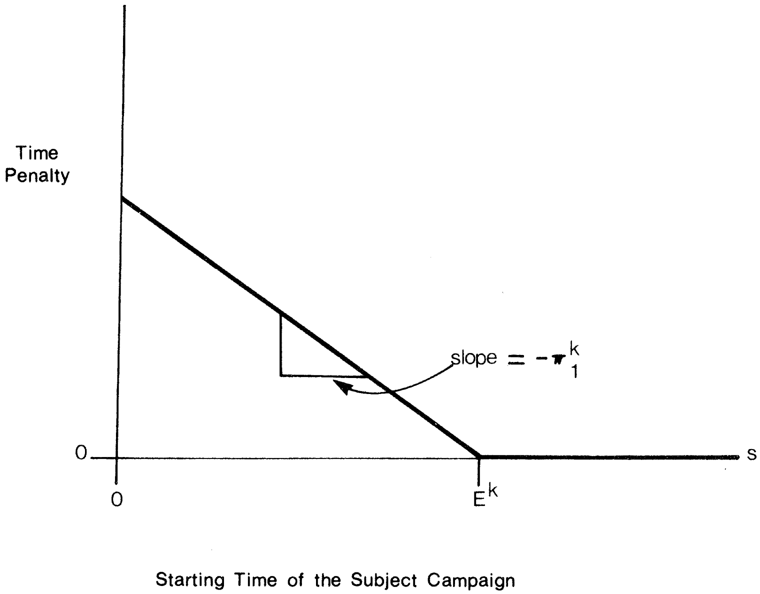

Fig. 3. A typical time penalty.

use of the quadratic assignment algorithm or linear programming or both. At least two kinds of iteration are possible: (a) alter the chosen quadratic assignment approximation and solve both it and its induced linear program again; (b) solve the quadratic assignment problem once for $K_{\rho 0}$, but solve the linear program not only for $K_{\rho 0}$ but also for other plausible campaign sequences that are "close" to $K_{\rho 0}$ in a suitable sense. We have found the second of these approaches to be entirely satisfactory in the practical application now to be described.

4. AN APPLICATION TO CHEMICAL REACTOR SCHEDULING

A practical application has been carried out for a complex of chemical reactors at Dart Industries Inc. As in the example described by Prabhakar [7], the transition costs were the economic loss suffered by the production of off-grade product during changeover. Five reactors of one type produce about twenty different products each month, while one reactor of a second type produces about a half dozen products. Since the products produced by the two types of reactors are disjoint, the scheduling problem decomposes naturally into two separate problems. The five reactors of the first type are similar so that our assumptions A1 and A2 provide a very good approximation. The standard scheduling period is one month. Some demands are a consequence of in-hand customer orders, while others are anticipated on the basis of market forecasts. Earliest start times are occasionally specified to take account of scheduled replenishment of raw material supplies. Latest finish times as a result of customer delivery commitments are more common.

The chemical reactor complex had always been scheduled manually. It was the practice to make a monthly schedule just prior to the beginning of every month, with this schedule being revised as necessary during the month to account for unanticipated reactor downtime or changes in customer delivery requirements and raw material availabilities.

The Computer Implementation

Our algorithmic choice for the quadratic assignment problem is a variant of Graves and Whinston's method [6]. Variances of the completion values of partial assignment maps are not employed, as it has been found that means alone give the best tradeoff between solution quality and computing time.

The quadratic assignment algorithm is interfaced with a linear programming routine in the manner described in Section 3. We employed the second type of *LP* iteration mentioned at the end of that section. More specifically:

Step 1. Set up and solve the quadratic assigment problem described in the beginning of Section 3. Extract the associated campaign sequence S^0. (More precisely, S^0 is the function K_{ρ^0} on Γ_{ρ^0} defined in Section 3.)

Step 2. Set up and solve the *LP* problem induced by S^0 as explained in Section 3. Install S^0 as the *incumbent solution*, S^{INC}, with *total cost* $TC(S^{INC})$ equal to the changeover costs for S^0 plus the minimal time penalty and production costs found by *LP*.

Step 3. Systematically generate "neighbors" of S^{INC}, testing each one to see if it qualifies for further evaluation by having change-over costs strictly less than $TC(S^{INC})$. (Only if this is so does a campaign sequence have a mathematical chance to improve the incumbent.) STOP if S^{INC} has no qualifying neighbor or if all such neighbors have already been passed on to Step 4 for evaluation. Otherwise, go to Step 4 as soon as the next qualifying neighbor S of S^{INC} is found.

Step 4. Evaluate S by solving its induced *LP* problem. If the total cost of S is less than $TC(S^{INC})$, replace S^{INC} by S. In any case, return to Step 3.

A *neighbor* of S^{INC} is defined here as any campaign sequence obtained by applying one of the following operations to S^{INC}:

(a) Move one campaign to another position.

(b) Interchange the positions of two campaigns.

These operations are easily formalized in terms of the notation established for campaign sequences in Section 3. The computer implementation carries out the four-step procedure in its entirety in a single run.

It is clear from the finiteness of the number of campaign sequences that the above procedure must terminate in a finite number of steps with an incumbent that is locally optimal for the improved problem statement of Section 2. By "locally" optimal we mean that no improvement can be obtained except, possibly, by changes to S^{INC} more complex than changing the position of one campaign or interchanging the positions of two campaigns. Our experience has been that scanning higher-order changes is not worth the trouble.

The LP solution is stored along with the incumbent campaign sequence, of course, so that the entire production schedule is at hand upon termination.

Provision is made to allow the user to specify mandatory campaigns of predetermined size for particular products and lines at the beginning of the scheduling period. Among the several uses of this feature is the ability to quickly rerun the scheduling problem in mid-month after some occurrence has caused departure from the intended schedule. Provision is also made to allow the scheduling of intentional down-time or maintenance just as though it were a "product." Neither of these features poses any new difficulties.

Results

A direct comparison between manual schedules made by experienced schedulers and computer-produced schedules was arranged for a three-month test period. This period was preceded by an initial phase during which the computer code was being tuned and the manual schedulers were learning to take advantage of newly developed changeover cost data. The procedure followed was to obtain a problem statement and the corresponding manual schedule, to generate a computer schedule using the given problem statement but without any knowledge of the manual schedule, and then to evaluate the total costs of both schedules using the agreed-upon cost data.

The computer schedule proved superior to the manual schedule for each of the three test months, as indeed it has for all of the less formal comparisons made before and since. The manual schedules were 35 %, 22 %, and 25 % more expensive than the computer schedules. Since changeover costs were amounting to several hundred thousand dollars per year, the savings were judged ample to justify conversion to the new computer scheduling method.

The cost savings demonstrated in these trials, substantial as they are, probably underestimate the true economic benefits of the scheduling program. One reason is that the problem statements and manual schedules were worked out at about the same time by the same people, which suggests that the problem statements transmitted were if anything more restrictive (difficult) than the problems actually addressed by the manual schedulers. Another reason is that the computer program should be better and quicker at revising schedules in mid-month after such disturbances as reactor breakdown, delayed raw material delivery, and urgent customer requests. Mid-month revisions were not part of the tests. This last point suggests one of the more valuable secondary uses of the program: to evaluate (approximately) the implications on cost of accepting versus refusing a customer request to advance the delivery date on his order.

Typical scheduling runs for the 5-reactor complex take the time quantum τ to be in the neighborhood of 3 days, which results in a total quadratic assignment map size m of about 50. The quadratic assignment algorithm takes approximately one minute on the IBM 360/91, and the iterative LP phase typically takes about one fifth that long.

The computer program has been installed on the firm's computer, is in regular use, and is regarded by the firm as a successful application.

NOTES

1. The changeover cost from any real product to the dummy product would be taken as 0, and the cost from dummy to any real product would be taken as prohibitively high so that the dummy product will only be produced after all real production on each line.
2. Sensitivity and continuity analysis of discrete and combinatorial programs is a subject still in its infancy. See, however, Radke [9] and Williams [10].
3. The *finishing time* of each line is the time at which the last real campaign ends and the first campaign (if any) of the dummy product starts.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the close cooperation and able assistance provided by Mr. Anthony G. Posgay, Assistant Director of Operations Research, Corporate Group, and Dr. James B. Wong, Director of Economics and Operations Analysis, Chemical Group, and others at Dart Industries Inc. This work was partially supported by the National Science Foundation and by the Office of Naval Research.

REFERENCES

1. E. Balas, L. T. Enos, and G. W. Graves, "Computerized Scheduling of Printing Operations," presented at the Spring Conference of the Graphic Communications Computer Association, May 23–25, 1973, Toronto.
2. J. E. Day and M. P. Hottenstein, "Review of Sequencing Research," *Naval Res. Log. Quart.* **17,** 11–39 (1970).
3. R. H. Deane and E. R. White, "Balancing Workloads and Minimizing Setup Costs in the Parallel Processing Shop," *Opnl. Res. Quart.* **26,** 45–53 (1975).
4. S. E. Elmaghraby and A. N. Elshafei, "The Scheduling of Jobs on Parallel Processors: A Survey and Annotated Bibliography," OR Report No. 97, Program in Operations Research, North Carolina State University at Raleigh, September 1974.
5. A. M. Geoffrion and R. E. Marsten, "Integer Programming Algorithms: A Framework and State-of-the-Art Survey," *Management Sci.* **18,** 465–491 (May 1972).
6. G. W. Graves and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," *Management Sci.* **17,** 453–471 (1970).
7. T. Prabhakar, "Some Scheduling Applications in the Chemical Industry," in *Symposium on the Theory of Scheduling and Its Applications*, No. 86, Springer-Verlag, 1973.
8. T. Prabhakar, "A Production Scheduling Problem with Sequencing Considerations," *Management Sci.* **21,** 34–42 (1974).
9. M. Radke, "Sensitivity Analysis in Discrete Optimization," Ph.D. Thesis, Graduate School of Management, UCLA, September 1975 (available as Working Paper No. 240, Western Management Science Institute, UCLA).
10. A. C. Williams, "Some Modeling Principles for M.I.P.'s," presented at the VIII International Symposium on Mathematical Programming, Stanford University, August 1973.