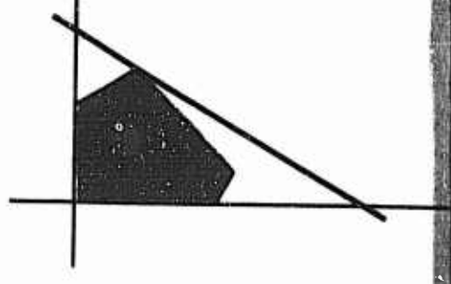
ORC 66-30
October 1966

SCHEDULING SEVERAL PRODUCTS ON ONE MACHINE TO MINIMIZE CHANGE-OVERS

by

C. R. Glassey



OPERATIONS RESEARCH CENTER

COLLEGE OF ENGINEERING

UNIVERSITY OF CALIFORNIA - BERKELEY

SCHEDULING SEVERAL PRODUCTS ON ONE MACHINE TO MINIMIZE CHANGE-OVERS

by

C. R. Glassey
Operations Research Center
University of California, Berkeley

October 1966 ORC 66-33

This research has been partially supported by the Office of Naval Research under Contract Nonr-222(83) and the National Science Foundation under Grant GP-4593 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Abstract

Given a finite horizon delivery schedule for n products we wish to schedule production on a single machine to meet deliveries and minimize the number of change-overs of the machine from one product to another.

A state space is defined and in it a network is constructed such that the shortest distance through the network corresponds to the minimum number of production change-overs. Certain properties of the optimal path are deduced from the dynamic programming formulation of the shortest route problem, and these properties are utilized in the construction of an algorithm that finds the optimal path. A numerical example illustrates the method.

Definition of the Problem

Consider, for example, a manufacturer of ice cream who produces 28 flavors (in general, n products), and has received orders for delivery of specified quantities of each flavor on certain days of the next month (of length T working days). He has but a single ice cream making machine, and desires a production schedule that will minimize the number of change-overs from one flavor to another while enabling him to meet his delivery commitments. To simplify the discussion, we assume that the unit quantity of an order is one day's output of the machine (or, alternatively, that we break our planning period into discrete time units corresponding to the time to produce a unit of product).

With this simplification, we introduce the following notation:

$x(t)$ = is the state of the system at time t, a lattice point in n - space, in which

$x_i(t)$ = cumulative production of product i,

$d(t)$ = cumulative requirement function,

$d_i(t)$ = cumulative requirement for product i as of time t,

e_i = ith unit vector (all elements zero except the ith which is one),

I_n = the set of integers $1, 2, \ldots, n$.

We will assume that the machine never shuts down (since our objective function is not improved by doing so) and we neglect the time required for a product change-over. Thus

$$t = \sum_{i=1}^{n} x_i(t) \qquad (1)$$

as a consequence of the assumption of unit production rate, and

$$x(t) - x(t-1) = e_i \qquad \text{(for some } i\text{)} \tag{2}$$

meaning that the system state advances one unit parallel to the ith coordinate axis during the interval $(t-1, t)$ in which time one unit of the ith product was made. Note that a product change-over corresponds to a change in direction of a path in state space. If we define the norm of a vector x

$$||x|| = \sum_{i=1}^{n} |x_i| \tag{3}$$

and the second difference vector

$$\Delta^2 x(t) = [x(t+1) - x(t)] - [x(t) - x(t-1)] \tag{4}$$

we see that

$$\begin{aligned} ||\Delta^2 x(t)|| &= 2 \quad \text{for change-over at time } t \\ &= 0 \quad \text{otherwise .} \end{aligned} \tag{5}$$

Let p denote the time at the end of the planning period and

$$T = \sum_{i=1}^{n} d_i(p) \tag{6}$$

be the time at which all required product can be completed under non-stop production. We will formulate an equivalent problem in which we leave the requirements $d(t)$ unchanged for $t < T$ but set

$$d(T) = d(p) \ . \tag{7}$$

The precise statement of the equivalent problem is

$$\text{Minimize} \qquad C = \frac{1}{2} \sum_{t=1}^{T-1} ||\Delta^2 x(t)|| \tag{8}$$

Subject to $x(t) \geq d(t)$, $t \in I_T$ (9)

and $\Delta x(t) = e_i$ for some i and all $t \in I_T$ (10)

This appears to be a rather formidable mathematical programming problem. After some discussion of the properties of feasible solutions, we will give an equivalent network formulation and develop a computationally feasible solution method.

A Conjecture and a Counter Example

An obvious conjecture arrived at independently by the author and several others is that the optimal policy is characterized by a rule of the type "don't change-over until you have to." This is correct for two products but not for three, as shown by the example of Table 1.

Table 1

Time	Product	Requirement
1	1	1
3	2	1
4	3	1
5	2	1
6	3	1
7	2	1
8	3	1
9	2	1
10	3	1
11	1	1

All feasible schedules begin with product 1 . According to the conjecture, we should produce two units of product 1 , but then we switch between 2 and 3 every day for the next 8 days. The optimal policy is 1 unit of product 1 followed by 2 units each of 2 and 3 (repeated once), and finally 1 unit of product 1 .

We shall see after some investigation of the structure of the problem that the optimal policy has a property similar to that conjectured, but with the time orientation reversed.

The Feasible Region

We define a feasible schedule $\{x(t)\}$ as a sequence of points satisfying (9) and (10). The feasible region F is the set of all points belonging to some feasible sequence. This region has certain properties that will be useful in what follows, and we list them as a series of lemmas.

Lemma 1: The feasible region F is non-empty iff

$$\sum_{i=1}^{n} d_i(t) \leq t \quad , \text{ all } \quad t \in I_T \ .$$

Proof: If the condition is not satisfied, it is clear that cumulative demand exceeds productive capacity at some time hence no feasible schedule exists. If the condition is satisfied, one can construct the obvious "earliest due date first" feasible schedule. This is done by letting $\delta_q(t)$ be the amount of product q required on day t [thus $d_q(t) = \sum_{\tau=1}^{t} \delta_q(\tau)$] .

We first make product i , the product for which $\delta_i(t_i) > 0$ for the smallest t_i , until $\delta_i(t_i)$ produced, then shift to the product j having the next earliest non-zero delivery requirement and produce $\delta_j(t_j)$ of it, etc.

Lemma 2: If F is non-empty and x satisfies $x \geq d(\tau)$ where $\tau = ||x||$, there exists a sequence $\{x(t)\}$ which is feasible up to time τ in the sense that (9) and (10) are satisfied for $t \leq \tau$.

Proof: We formulate a related problem in which the delivery requirements are unchanged for $t = 0, 1, \ldots, \tau - 1$, but we set $d(\tau) = x$. Then the condition of Lemma 1 is satisfied for this related problem.

Lemma 3: Given the point x satisfying the conditions of Lemma 2, if there is a point y in F such that $y > x$, then x is in F.

Proof: From the definition of F, we need to show the existence of a feasible path passing through x. A feasible segment from 0 to x is established by Lemma 2, and a segment from y to x^T exists because $y \in F$. We will show that a feasible segment from x to y exists by constructing an auxilliary problem in which x plays the role of the origin. Let $t_1 = ||x||$, $z = y - x$, $t_2 = ||z||$. Construct the feasible region F' defined by the requirement functions

$$d'_i(t) = \text{Max}\{0, d_i(t + t_1) - x_i\}, \quad i \in I_n, \; t \in I_{t_2}$$

Since $z_i = y_i - x_i \geq \text{Max}\{0, d_i(t_1 + t_2) - x_i\} = d'_i(t_2)$, we conclude from Lemma 2 that there exists a path $\{z(t), t \in I_{t_2}\}$ from 0 to z which is in F'. Now consider the path $\{x(t + t_1) = z(t) + x, t \in I_{t_2}\}$ which connects x to y. Then

$$x(t + t_1) = z(t) + x \geq d'(t) + x \geq d(t + t_1) \quad \text{for } t \in I_{t_2},$$

hence this segment is feasible.

A Network Formulation

A convenient way to structure this problem is by means of a network.[1] The nodes are the lattice points in F. From each node x we construct arcs of unit length to each point y in F having the property that

$$y = x + ke_i \ , \ k > 0 \ , \text{ for some } \ i \in I_n \tag{11}$$

The interpretation is that y is a state reachable from x by producing k units of product i at the cost of a single product change-over. We note that $z = x + k_1 e_i$ is also in F for all $0 < k_1 \leq k$, since the path in state space consisting of the points x, $x + e_i$, $x + 2e_i$, ... , y is the only possbile path from x to y and hence lies in F by Lemma 3. Thus the arc (x , y) in the network corresponds to a segment of a feasible schedule.

We define the distance between nodes of the network to be the length of the shortest path joining them (which is <u>not</u> the same as the distance between points in state space, which would be $||x - y||$). Let x^T denote the terminal node of the network,

$$x^T = d(T) \ ; \tag{12}$$

and the shortest path from 0 to x^T is then the optimal production schedule. The shortest path may not, of course, be unique; but we will nevertheless speak of the optimal solution while realizing that there may be many such solutions.

In principle, a simple labelling scheme could be used to find the shortest path. Define $v(x)$ as the distance of node x from the origin. Each node x will be given a label of the form $[v(x) , y]$ where the arc (y , x) is the last

[1] This formulation was suggested by G. Dantzig.

arc in the shortest path from 0 to x. Labels can be applied in a systematic way by defining $S_j = \{x | x \in F , v(x) = j\}$, and we label all points in S_{j-1}, then S_j, etc. The set $S_0 = 0$ by definition, and the general step is: for each point y in S_{j-1}, assign the label (j , y) to each point x in F of the form $x = y + ke_i$, $k > 0$. The procedure terminates when x^T is labelled.

This procedure suffers from two practical limitations. First, it is not easy to determine if a point is in the feasible region. Second, the network may contain a very large number of nodes [an upper bound is $\prod_{i=1}^{n} d_i(T)$].

These difficulties may be at least partially overcome by using two properties of the optimal path to reduce considerably the number of nodes that must be labelled, and by starting the labelling procedure from x^T rather than 0. These properties are made explicit in the theorems of the next section.

Theoretical Foundations of an Improved Algorithm

One useful result is contained in

Theorem 1: The shortest path length function $v(x)$ is monotone non-decreasing in each argument, i.e., $y > x$ implies $v(y) \geq v(x)$.

In terms of production schedules, this expresses the very plausible result that, given an optimal schedule that achieves a cumulative production vector y, there is a way to produce less of some products at no greater cost in number of change-overs.

A rigorous mathematical proof of this result can be obtained from a dynamic programming approach to the problem. To do this, we introduce the notion of the set of predecessors of a node y which we define by

$$P(y) = \{x | x \in F ; x = y - ke_i , k > 0 , i \in I_n\} .$$

Thus, $P(y)$ is the set of states from which y can be reached by making any one product; in the network language, the set of nodes directly connected to y by single arcs. With this definition, it is clear that the shortest path length function satisfies

$$v(y) = \underset{x \in P(y)}{\text{Min}} \{1 + v(x)\} \tag{13}$$

These sets of predecessors have an interesting property which we state in

Lemma 4: If a point y in F has two predecessors w and x lying in different directions, these have in turn a common predecessor.

Proof: Let $w = y - k_1 e_i$, $x = y - k_2 e_j$, $k_1 > 0$, $k_2 > 0$, $i \neq j$. We show that $z = w - k_2 e_j = x - k_1 e_i$ is in F . Let $t = ||y||$, $t_1 = ||w|| = t - k_1$, $t_2 = ||x|| = t - k_2$, and $t_3 = ||z|| = t_2 - k_1 = t_1 - k_2$. Now w , $x \in F \Rightarrow w \geq d(t_1) \geq d(t_3)$ and $x \geq d(t_2) \geq d(t_3)$, since the requirement functions $d(t)$ are non-decreasing. Hence

$$z_q = w_q \geq d_q(t_3) \qquad \text{for } q \neq i \text{ , and}$$

$$z_q = x_q \geq d_q(t_3) \qquad \text{for } q \neq j \text{ .}$$

Thus z satisfies the conditions of Lemma 3 and hence belongs to F . We are now in a position to complete the

Proof of Theorem 1 (by induction): We state an equivalent theorem: $y \in S_k$, $x \in P(y)$ implies $v(x) \leq v(y) = k$. For $k = 1$, $P(y)$ is the single point 0 which is also S_0 . In general step, we define x^* as the optimal predecessor of y so $v(y) = 1 + v(x^*)$, and $x^* \in S_{k-1}$. We now consider three possible cases.

Case A: x differs from y in the same direction as x^* and $x < x^*$. Then $x \in P(x^*)$ so, by the induction hypothesis, $v(x) \leq v(x^*) = v(y) - 1$.

Case B: x differs from y in the same direction as x^*, but $x > x^*$. Then $x^* \in P(x)$, so by equation (13), $v(x) \leq 1 + v(x^*) = v(y)$.

Case C: x differs from y in a different direction than x^*. Then by Lemma 4, x and x^* have a common predecessor, call it w, and $v(w) \leq v(x^*)$ by the induction hypothesis. Furthermore, by equation (13), $v(x) \leq v(w) + 1 \leq v(x^*) + 1 = v(y)$.

The key idea in the design of an improved algorithm is stated in

Theorem 2: For each y in F, there exists an optimal predecessor $x = y - ke_i$ with the property that $z = y - (k + 1)e_i$ is not in F.

Proof: If $x = y - ke_i$ is an optimal predecessor of y and $z = y - (k + 1)e_i$ is in F then $v(z) \leq v(x)$ as was shown in Case A of Theorem 1, (the equality must hold, otherwise x would not be an optimal predecessor of y) hence z is also an optimal predecessor. But, since $P(y)$ is bounded from below, there must be a largest k for which the point $y - ke_i$ is in $P(y)$.

An Improved Algorithm

Theorem 2 suggests that we construct and label a subnetwork G^* as follows: from the terminal node x^T, construct the set $P^*(x^T)$ of points y having the property of Theorem 2, thus

$$P^*(x^T) = \{y \mid y \in F \; ; \; y = x^T - ke_i \; ; \text{ and } \; z = x^T - (k + 1)e_i \notin F\} .$$

There are at most n points in $P^*(x^T)$, and they are easily constructed as follows: for $j = 1, 2, \ldots, n$:

set $y_i = x_i(T)$ for $i \neq j$, and

set $y_j = d_i(T - k)$ where k satisfies the two conditions

(a) $x_i(T) - k - 1 < d_i(T - k - 1)$ and

(b) $x_i(T) - k = d_i(T - k)$.

By Theorem 2, $P^*(x^T)$ contains an optimal predecessor of x^T . Each point in $P^*(x^T)$ is assigned the label $(1 , x^T)$. We define the set $R_1 = P^*(x^T)$.

In general, R_k is the set of nodes in G^* at distance k from x^T . The kth step in the procedure is the construction and labelling of the set R_k by constructing the set $P^*(x)$ for each x in R_{k-1} . In general the set $P^*(x)$ is constructed in the same way as $P^*(x^T)$ except we replace the time T in conditions (a) and (b) with $t = \sum_{i=1}^{n} x_i$, and we exclude from $P^*(x)$ any nodes that have been previously labelled. The procedure terminates when the origin appears in the set R_k for some value of k , which is the length of the shortest path from 0 to x^T .

Reduction of Computational Effort

It will be seen that the arcs joining pairs (x , y) where y $P^*(x)$ form a tree structure, so that the search for the shortest route through the network G^* can be viewed as a tree searching problem. It may be possible to reduce the number of branches to be explored by applying Theoreml. We define $u(x)$ as the length of the shortest path from x to x^T and $C(x) = u(x) + v(x)$ as the length of the shortest path from 0 to x^T passing through x . Now suppose $y > x$ but $u(x) \leq u(y)$. By Theorem 1, $v(y) \geq v(x)$ so that $C(y) \geq C(x) \geq v(x^T)$, and in our search for the shortest path we can safely neglect all paths through the point y . After constructing the set R_k as described above, we eliminate from R_k all points y such that $y > x$ for some x in the set $R_k \cup R_{k-1} \cup \ldots \cup R_1$.[2]

[2]It is easy to show that y need only be compared with x in R_k .

Example:

Data for a simple three-product problem are given in Table 2.

Table 2

Time	Product	Quantity Required
2	3	1
3	1	2
5	2	1
8	2	1
8	3	1
10	1	1
11	3	2
12	2	1
16	1	3
17	2	1
18	3	2
20	1	1
		17

The total required is 17 so we let $T = 17$, and let $\delta_1(17) = 1$, $\delta_3(17) = 3$. The resulting cumulative requirement functions are shown in Figure 1, which also shows the optimal production schedule. The tree generated in obtaining this schedule is shown in Figure 2, with the set R_k appearing in the kth row. Points deleted from R_k by applying Theorem 2 are underlined.

From the terminal state (7 , 4 , 6) at time 17 , we examine each of the three possibilities for the last production segment. Considering product 1 , as shown on Figure 1, we arrive at the point (3 , 4 , 6) at time 13 , the point where the 45° line representing cumulative production of product 1 intersects the requirement line. The other two states which are possible optimal predecessors of x^T are (7 , 3 , 6) at time 16 and (7 , 4 , 4) at time 15 and are similarly found. These three points constitute the set R_1 and are shown on the first level of the tree in Figure 2. The first point generated in R_2 is a member.

of $P(3 , 4 , 6)$ and is $(3 , 2 , 6)$ as shown in Figure 1. This point is determined by extending the 45 production line of product 2 from the point $x_2 = 4 , t = 13$ until it intersects the demand line at $x_2 = 2 , t = 11$. After the remainder of the points in R_2 are similarly constructed, it is noted that $(3 , 4 , 4)$ $(3 , 4 , 2)$ and hence $(3 , 4 , 4)$ is deleted under the rule of the previous section. The origin $(0 , 0 , 0)$ is generated by this procedure as an element of R_6 , so the optimal path contains six arcs. Retracing this path shows that the optimal sequence of products is: 3 , 1 , 2 , 3 , 2 , 1 .

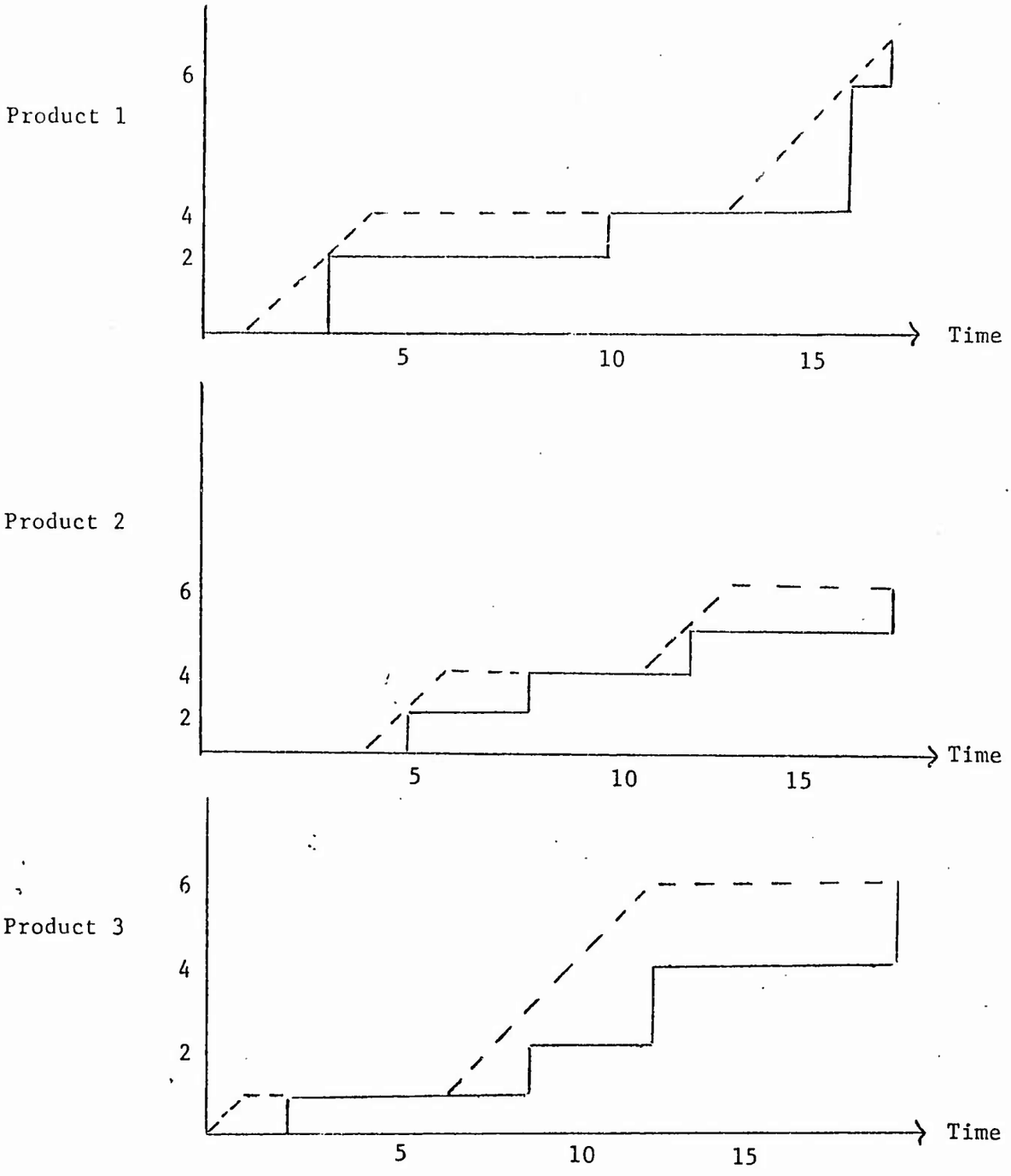


Figure 1

Cumulative requirements (solid lines) and optimal production schedule (dashed) for three products.

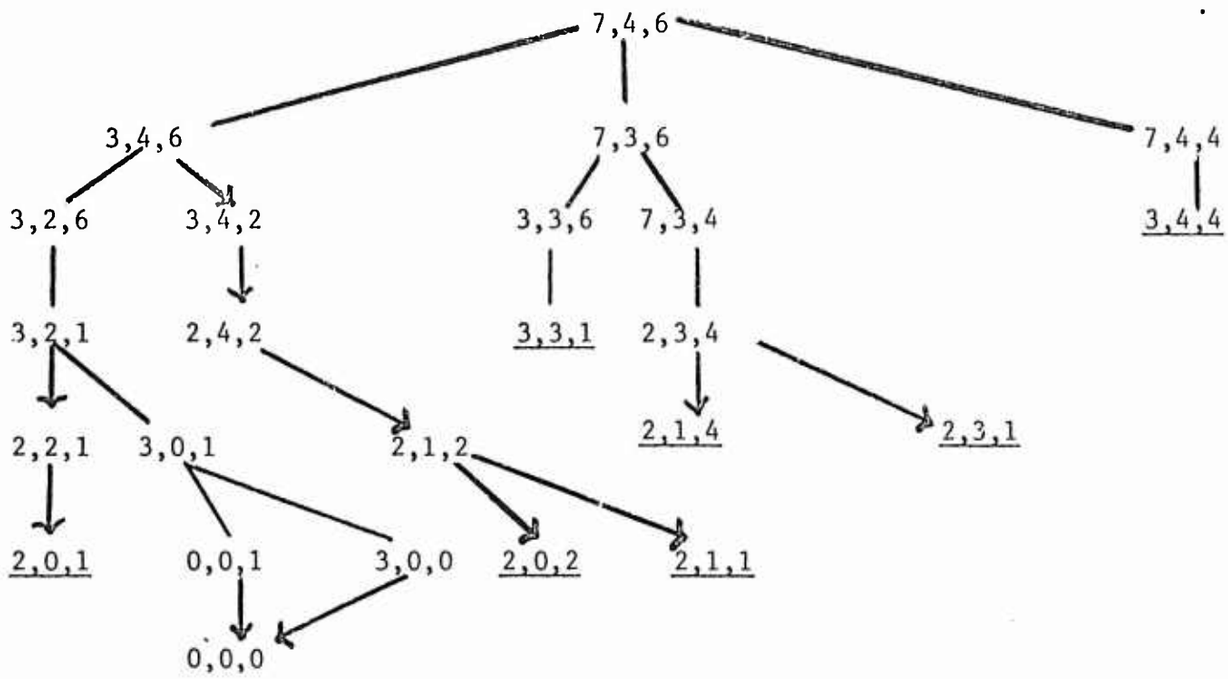


Figure 2

Tree structure for finding optimal production schedule

Note that this tree contains 24 nodes. If we had not used the elimination procedure, but generated the two predecessors of each node at each level after the first, we would have generated $3 \times 2^4 = 48$ nodes in the first 5 levels.

Extensions and Generalizations

We can easily drop the retrictions of discrete time and integer demands imposed in the first section, provided demands occur only at a finite number of points in the time interval $[0 , p]$. The obvious modifications in the development are to replace sequences of points with vector valued functions of time, finite differences become derivatives, sums over time become integrals, etc. The statement of Theorem 2 becomes

Theorem 2': Each y in F has an optimal predecessor x^* which is a limit point of the set $\{x | x = y + \gamma(x^* - y) , \gamma > 0\}$ F .

Of course the network formulation results in an infinite number of nodes, but the reduced network G^* is still finite. With certain other minor changes, the proofs of the previous sections can be recast in this continuous time language.

The restriction of demand values to integers was necessary in the discrete time model to insure that change-overs occur at integer values of time, but plays no other role.

It would appear that this approach can be used in scheduling of products on several identical machines. Theorem 1 still holds since the structure of the feasible region F is not altered in any radical way. The set of predecessors of point y still consists of the intersection of a finite number of rays through y with the region F , and the optimal predecessor is a limit point of this set, so a version of Theorem 2 can be stated. However, the details of carrying out the computations are somewhat more complicated.

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
University of California, Berkeley	Unclassified
	2b GROUP

3. REPORT TITLE

SCHEDULING SEVERAL PRODUCTS ON ONE MACHINE TO MINIMIZE CHANGE-OVERS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Research Report, October 1966

5. AUTHOR(S) (Last name, first name, initial)

Glassey, C. R.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1966	17	

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
Nonr-222(83)	
b. PROJECT NO. Nr 047 033	ORC 66-33
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
Also supported by National Science Foundation, Grant Number GP-4593	Mathematical Science Division

13. ABSTRACT

Given a finite horizon delivery schedule for n products we wish to schedule production on a single machine to meet deliveries and minimize the number of change-overs of the machine from one product to another.

A state space is defined and in it a network is constructed such that the shortest distance through the network corresponds to the minimum number of production change-overs. Certain properties of the optimal path are deduced from the dynamic programming formulation of the shortest route problem, and these properties are utilized in the construction of an algorithm that finds the optimal path. A numerical example illustrates the method.

DD FORM 1 JAN 64 1473

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Scheduling Sequencing Multiple-product Single-machine Shortest route Dynamic Programming						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2*a*. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2*b*. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7*a*. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7*b*. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8*a*. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8*b*, &c, & 8*d*. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9*a*. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9*b*. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, roles, and weights is optional.