Carnegie Institute of Technology

Pittsburgh 13, Pennsylvania

GRADUATE SCHOOL of INDUSTRIAL ADMINISTRATION

William Larimer Mellon, Founder

18

Management Sciences Research Report No. 98

SEARCH-REDUCTION TESTS for a COMBINATORIAL PRODUCTION SEQUENCING ALGORITHM

by

Fred Glover *

December 1966

*University of California, Berkeley.

This report was prepared as part of the activities of the Management Science Research Group, Carnegie Institute of Technology, under Contract NONR 760(24) NR 047-048 with the U. S. Office of Naval Research.

Management Sciences Research Group
Carnegie Institute of Technology
Graduate School of Industrial Administration
Pittsburgh, Pennsylvania 15213

1. Introduction

Recently a number of tree search algorithms have been developed for solving combinatorial optimizational problems (see, for example, [1, 2, 4]). One such approach by J. F. Pierce and D. J. Hatfield [3] is particularly interesting because it applies to a wide range of important problems in production sequencing. A key feature of tree search approaches in general, and the Pierce and Hatfield algorithm in particular, is the use of tests to exclude dominated alternatives from consideration. The purpose of this paper is to develop several relations that lead to tests other than those proposed by Pierce and Hatfield in order to reduce the number of solutions examined by their algorithm (and other tree search methods for the same problem).

2. The Sequencing Problem

Consider a set of n+1 jobs, designated 0, 1, ..., n, to be processed on a single machine, where a'_{ij} represents the (non-negative) set up time for job j immediately after processing job i, and p'_j represents the (nonnegative) time required to process j once it is ready.

Assuming that job 0 must go on the machine first and job n last (0 and n can be dummy jobs), the problem is to find a way of sequencing the remaining jobs so that each one is processed exactly once[+] and the total machine time is minimized. In addition,

---

[+]Thus, by convention we may let $a'_{jj}=a'_{jo}=a'_{nj}=\infty$ for all j to indicate that no job is permitted to precede itself or job 0, or to follow job n.

each job j must be completed by its deadline d_j ($d_n \geq \max_{j<n} (d_j)$).

Following the formulation of [3], we seek a permutation $(i_o, i_1, \ldots, i_n)$ of $(0, 1, \ldots, n)$ to

$$\text{Minimize } p_{io} + \sum_{k=1}^{n} (a'_{i_{k-1}i_k} + p'_{i_k}) \qquad (1)$$

$$\text{subject to } t_{i_j} \equiv p_{i_o} + \sum_{k=1}^{j} (a_{i_{k-1}i_k} + p'_{i_k} \leq d_{i_j} \qquad (2)$$

where $i_o=0$, $i_n=n$, and t_{i_j} is the actual completion time of the j th job in the sequence.

Pierce and Hatfield note that p_j defined by $p_j=p'_j+\min_{i<n} (a'_{ij})$ provides a lower bound on the total processing and set up time for job j, regardless of which job i precedes it (to accommodate j=0, let $p_o=p'_o$). Thus, if

$$\sum_{j=o}^{n} p_j > d_n$$

the problem has no feasible solution. More generally, replacing $a'_{ij}+p'_j$ by p_j, a result of Smith [5] states that maximum lateness, $\max_{1\geq j} (t_j - d_j)$, is minimized when $d_1 \leq d_2 \leq \ldots \leq d_n$.

Consequently, assuming that the jobs are ordered in this way, it follows that the problem has a feasible solution only if

$$\sum_{j=o}^{k} p_j \leq d_k \text{ , } k=1, \ldots, n \qquad (3)$$

Pierce and Hatfield use this result to construct several of the tests of their algorithm. We will propose somewhat more restrictive tests and suggest a way of reorganizing the problem that makes it possible to exploit Smith's result in a more effective way. In addition, we will give a slightly more general result that also permits a more limiting feasibility test than (3).

3. Recasting the Problem

It is convenient to restate the problem in the following form. Represent the n+1 jobs by n+1 nodes, where $a_{ij}=a'_{ij}+p'_j$ represents the length of arc (i, j) from node i to node j. The problem then is to find the shortest (directed) path from node 0 to node n that goes through each node j exactly once, and for which the distance from node 0 to each node j does not exceed d_j.

The process of enumerating alternative paths is conveniently accomplished by enumerating sets of arcs (i, j). As observed in [3] (in a slightly different framework), if arc (i, j) is chosen to be included in the path, then (i, j) essentially reduces to a node v for which $a_{hv}=a_{hi}+a_{ij}$ and $a_{vh}=a_{jh}$ for all $h \neq i, j$. Also, the permissible length of the path to node v cannot exceed $d_v = \text{Min}\,(d_j, d_i+a_{ij})$.

Because the reduced problem (in which node v replaces (i, j)) has exactly the same form as the original, the feasibility test (3) can be repeated at each stage in the enumeration.

As a first step toward developing more restrictive tests, identify for each $j > 0$ an index j' such that

$$a_{j'j}=p_j = \underset{i<n}{\text{Min}}\,(a_{ij}) \quad .$$

Also, define Δ_j to be the (nonnegative) difference between $p_j(a_{j'j})$ and the "second largest" a_{ij}; that is

$$\Delta_j = \underset{\substack{i<n \\ i\neq j'}}{\text{Min}}\,(a_{ij}) - p_j \quad .$$

Then, for a given permutation $(i)=(i_o, i_1, \ldots, i_n)$ (representing a sequence of nodes in a path from 0 to n), define†

$$S_k^{(i)} = \left(j : j < k \text{ and } i'_j = i'_k \right)$$

$$\delta_k^{(i)} = \begin{cases} 0 \text{ if } S_k^{(i)} \text{ is empty} \\ \max\limits_{j \in S_k^{(i)}} (\Delta i_j) \text{ otherwise} \end{cases}$$

$$\gamma_k^{(i)} = \text{Min } (\Delta i_k, \delta_k^{(i)}) .$$

(For completeness, we specify that $\Delta_o = 0$ and $\gamma_o^{(i)} = 0$.)

Remark 1. For any permutation $(i) = (i_o, i_1, \ldots, i_n)$, $i_o = 0$, $i_n = n$,

$$\sum_{j=o}^{k} (p_{i_j} + \gamma_j^{(i)}) \leq \sum_{j=1}^{k} a_{i_{j-1} i_j} , \tag{4}$$

for all $k = 1, \ldots, n$.

Proof: Let V be the set of the k numbers $i_o, \ldots, i_k$ and let $V_q^{(i)} = \left(j : i_j \in V \text{ and } i'_j = q. \right)$ Then we may write

$$\sum_{j=o}^{k} \gamma_j^{(i)} = \sum_{j \in V_o^{(i)}} \gamma_j^{(i)} + \sum_{j \in V_1^{(i)}} \gamma_j^{(i)} \ldots + \sum_{j \in V_{n-1}^{(i)}} \gamma_j^{(i)}$$

Let q^* be defined so that $\Delta i_{q^*} = \max\limits_{j \in V_q^{(i)}} (\Delta i_j)$.

Then it may easily be verified by the definition of $\gamma_j^{(i)}$ that

$$\sum_{j \in V_q^{(i)}} \gamma_j^{(i)} = \sum_{j \in V_q^{(i)} - (q^*)} \Delta i_j . \tag{5}$$

Thus,

$$\sum_{j \in V_q^{(i)}} p_{i_j} + \sum \gamma_j^{(i)} = \sum_{j \in V_q^{(i)} - q^*} (a_{i'_j i_j} + \Delta i_j) + a_{i'_{q^*} i_{q^*}}$$

and since $i_{j-1} = i'_j$ can hold for at most one $j \in V_q^{(i)}$, we have

† We use the subscript i'_j to denote the primed subscript defined on page 3; i.e., $a_{i'_j i_j} = p_{i_j}$.

$\sum_{j\in V_q^{(i)}} (p_{i_j} + \gamma_j^{(i)}) \leq \sum_{j\in V_q^{(i)}} a_{i_{j-1}i_j}$, and the remark follows at once.

By Remark 1 we see that

$$\sum_{j=1}^{k} (p_{i_j} + \gamma_j^{(i)}) > d_{i_k} \tag{6}$$

implies that the path given by permutation (i) is infeasible. To obtain a useful test from this result, we must specify a permutation (i) for which (6) implies that <u>all</u> paths from 0 to n are infeasible. We now show that a permutation with this property occurs by indexing the nodes so that $d_1 \leq d_2 \leq \ldots \leq d_n$, which is the same indexing given by Smith's result when $\gamma_j^{(i)} + p_{i_j}$ depends only on the index i_j, but not on the permutation (i).

<u>Remark 2</u>. Let $(i)=(i_o, i_1, \ldots, i_n)$ and $(h)=(h_o, h_1, \ldots, h_n)$ be two paths from 0 to n such that $(i_1, i_2, \ldots, i_k)$ and $(h_1, h_2, \ldots, h_k)$ are permutations of the same k numbers. Then

$$\sum_{j=1}^{k} \gamma_j^{(i)} = \sum_{j=1} \gamma_j^{(h)}$$

<u>Proof</u>: Define the set V as in the proof of Remark 1. Then the sets $V_q^{(h)}$ may be defined relative to (h) exactly as they are defined relative to (i); i.e., so that

$$\left\{i_j : j\in V_q^{(i)}\right\} = \left\{h_j : j\in V_q^{(h)}\right\}.$$

Hence, by (5) it follows that

$\sum_{j\in V_q^{(i)}} \gamma_j^{(i)} = \sum_{j\in V_q^{(h)}} \gamma_j^{(h)}$ for all q. This completes the proof.

<u>Theorem 1</u>. Assume that $d_1 \leq d_2 \leq \ldots \leq d_n$, and let

$t_k^{(i)} = \sum_{j=1} (p_{i_j} + \gamma_j^{(i)})$. Then for any numbers $\gamma_j^{(i)}$ satisfying the property of the preceding remark and $\gamma_j^{(i)} + p_{i_j} \geq 0$, the quantity

$$\operatorname*{Max}_{1 \leq k \leq n} (t_k^{(i)} - d_{i_k}) \tag{7}$$

is minimized by the permutation (i) for which $i_j = j$ for all j.

<u>Proof</u>: Suppose a permutation (h) minimizes (7) such that for some j, $h_j > h_{j+1}$. Define (i) to be the same permutation as (h) except that $i_{j+1} = h_j$ and $i_j = h_{j+1}$. By the property of $\gamma_j^{(i)}$ it follows that $t_k^{(i)} = t_k^{(h)}$ for all k except k=j. But

$$t_j^{(i)} \leq t_{j+1}^{(i)} = t_{j+1}^{(h)} \text{ and } d_{i_{j+1}} = d_{h_j} \geq d_{i_j} = d_{h_{j+1}}.$$

Thus, $t_j^{(i)} - d_{i_j} \leq t_{j+1}^{(h)} - d_{h_{j+1}}$ and $t_{j+1}^{(i)} - d_{i_{j+1}} \leq t_{j+1}^{(h)} - d_{h_{j+1}}$.

Consequently (i) minimizes (7) as well as (h). If (i) doesn't have the property specified by the theorem we let (i) take the role of (h) and derive a new (i) as above. Eventually, we must obtain an (i) for which the theorem is satisfied, or conclude that the initial (h) already satisfied the theorem (i.e., $h_j > h_{j+1}$ was false).

Assume that the nodes are indexed as specified in Theorem 1. (The permutation (i) can thus be disregarded.) Then, to simplify the application of the above results, we note that each γ_k can be given by comparing exactly two quantities, without having to compute δ_k as a maximum over several $a_{j'j}$. For let $h=\underset{j\in S_k}{\mathrm{Max}}(j)$. Then $\delta_k=\mathrm{Max}(\Delta_h, \delta_h)$ and $\gamma_h=\mathrm{Min}(\Delta_h, \delta_h)$. Thus, to compute γ_h it suffices to select the smaller of Δ_h and δ_h for γ_h and record the other as δ_k. (Note $k>h$.)

4. Additional Sequencing Relations and Tests

Continuing to view the sequencing problem as a constrained shortest path problem, suppose now that all arcs (i, 1) are "cut off" at a distance s_1 from node 1, where $s_1=\underset{i<n}{\mathrm{Min}}(a_{i1})$. Since every path must go through 1, the length of the path must be at least s_1.

Next consider cutting off all arcs (2, j) at a distance $r_2\leq\underset{o<j}{\mathrm{Min}}(a_{2j})$ from node 2. Since arc (2,1) has already been cut off at a distance s_1 from 1, we can't permit $r_2>a_{21}-s_1$ without going past the first cut. However, as long as $r_2+s_1\leq a_{21}$, then since every path must go through node 2 as well as node 1, we are assured that the path length is at least r_2+s_1.

In general, if we cut off all arcs entering node j at a (nonnegative) distance s_j from j, and all arcs leaving j at a distance r_j from j, then if $r_k+s_j\leq a_{kj}$ for all nodes k and j, it is clear that

$$\sum_{j=o}^{n}(r_j+s_j)=L \tag{8}$$

where L is a lower bound on the length of every path from 0 to n. (We stipulate by convention that $s_o=r_n=0$ since no arcs enter node 0 or leave node n).

The foregoing demonstrates a well known fact customarily proved algebraically (but given little intuitive justification) in the context of solving transportation and assignment problems. Because (8) gives a lower bound on path lengths, Little et al [2] use it in the test procedure of the tree search algorithm for the traveling salesman problem.

Following [2], Pierce and Hatfield also use (8) to determine whether a feasible path previously found is better than one currently being generated. We propose a different way to use (8), yielding generally stronger tests than given in [3].

Evidently, for any permutation $(i)=(i_o, i_1, \ldots, i_n)$†

$$\sum_{j=o}^{k-1}(r_{i_j}+s_{i_j})+s_{i_k} \leqq \sum_{j=1}^{k} a_{i_{j-1}i_j}$$

and hence

$$u_k^{(i)}=\sum_{j=o}^{k-1}(r_{i_j}+s_{i_j})+s_{i_k} \leqq d_{i_k} \tag{9}$$

gives a necessary requirement for the feasibility of (i). Again, we seek to minimize $\underset{1\leqq k\leqq n}{\mathrm{Max}} (u_k^{(i)} - d_{i_k})$ and hence make (9) into a useful test.

Remark 3. The quantity $\underset{1\leqq l\leqq n}{\mathrm{Max}} (u_k^{(i)} - d_{i_k})$ is minimized by any permutation (i) such that

$$d_{i_{j-1}}+r_{i_j} \leqq d_{i_j} + r_{i_j} \tag{10}$$

for j=2, ..., n-1.

---

† As before, we continue to assume here and throughout the paper that $i_o=0$ and $i_n=n$.

Proof: $u_k^{(i)}$ d_{i_n} is a constant independent of (i) (provided $i_n=n$), hence it is permissible to restrict j in (10) to $j\leq n-1$. Then, rewrite (9) so that

$$u_k^{(i)} + r_{i_k} = \sum_{j=o}^{k} (r_{i_j} + s_{i_j}) \leq d_{i_k} + r_{i_k} \tag{11}$$

Since r_{i_j} and s_{i_j} depend only on the subscript i_j and not on the permutation (i), the remark follows directly from Smith's theroem [5].

We note that the tests developed by Pierce and Hatfield from (3) can also be applied to (11). We derive additional tests based on (11) below.

In accordance with Remark 3, assume that the nodes are indexed so that

$$d_1 + r_1 \leq d_2 + r_2 \leq \dots \leq d_{n-1} + r_{n-1}$$

and let $\displaystyle\sum_h^k = \sum_{j=h}^{k} (r_j + s_j)$.

Also, define

$$\alpha_h^k = \min_{h\leq j\leq k} (d_j + r_j - \sum_o^j) .$$

Theorem 2. If $j>i$ ($j\neq n$, $i\neq o$), then (j,i) is a permissible arc in the path from o to n only if

$$\alpha_i^{j-1} \geq a_{ji} + s_j - s_i \tag{12}$$

and

$$\alpha_j^n \geq a_{ji} - (s_i + r_j) \tag{13}$$

Proof: Suppose arc (j, i) is in the path. Imagine a fictitious node v placed at node i, replacing nodes j and i. Each arc (h, j) formerly entering j ($h\neq i$) is extended by arc (j, i) to become a new arc (h,v), where $a_{hv}=a_{hj}+a_{ji}$. Similarly, each arc (i, h) for

$h \neq j$ becomes (v,h). Thus, we may let $v=r_i$ and $s_v=s_j+a_{ji}$ (possibly larger values are also permissible). To assure feasibility, $d_v=\mathrm{Min}\ (d_i,\ d_j+a_{ji})$. Since $d_i+r_i \leq d_j+r_j$, we have $d_i \leq d_j+r_j-r_i \leq d_j+a_{ji}$. Hence $d_v=d_i$.

Let α_h^{k*} denote the new value for α_h^k based on the addition of node v as above and the deletion of nodes i and j. However, since $d_v+r_v=d_i+r_i$, we may conceive v to be (indexed) the same as i. Also, deleting the index j, we may allow the other indices to remain unchanged. Then, clearly $\alpha_i^{k*} = \alpha_i^k$ for $k<i$. Also, since $\sum_o^k$ is decreased by r_i+s_i and increased by r_v+s_v for all k satisfying $i \leq k<j-1$, we have $\alpha_i^{k*} = \alpha_i^k + (r_i+s_i)-(r_v+s_v)$, and hence $\alpha_i^{j-1*} = \alpha_i^{j-1} + s_i-(a_{ji}+s_j)$. If there is a feasible path using (j, i) then $\alpha_i^{j-1*} \geq 0$, which is the same as (12).

For $k>j$, $\sum_o^k$ is decreased by $(r_i+s_i) + (r_j+s_j)$ and increased by r_v+s_v. Hence,

$$\alpha_{j+1}^{n*} = \alpha_{j+1}^{n} + r_j + s_i - a_{ji} \quad .$$

Also, from (12) we have

$$\alpha_{j-1}^{j-1} \geq a_{ji} + s_j - s_i$$

which in turn implies

$\alpha_j^j \geq a_{ji} - (r_i+s_i)$. This together with $\alpha_{j+1}^{n*} \geq 0$ gives (13) and completes the proof.

Theorem 3. Assume $j<i$, and let $d=\text{Min}\ (d_i,\ d_j+a_{ji})$ and

$$v = \underset{h>j}{\text{Min}}\ (h:\ d_h+r_h \geq d+r_i).$$

Then $(j,\ i)$ is a permissible arc in the path from 0 to n only if

$$\sum_{o}^{v-1} \leq d_j + r_j \qquad (14)$$

$$\alpha_v^{i-1} \geq a_{ji} + r_i - r_j \qquad (15)$$

$$\alpha_i^{n} \geq a_{ji} - (r_j+s_i) \qquad (16)$$

Moreover, if $d=d_i$, then (14) and (15) are irrelevant.

Proof: As in the proof of Theorem 2, create the node v to replace i and j where $r_v^*=r_i$ and $s_v^*=a_{ji}+s_j$.† Also, let α_h^{k*} denote the new values of α_h^k by this replacement. For $k<j$, clearly $\alpha_1^{k*}=\alpha_1^k$, and for $k>j$, $\alpha_k^{v-1*}=\alpha_k^{v-1}+r_j+s_j=0$.

(We assume that the index j has been deleted so that the old indices remain unchanged through v-1. Also for convenience we assume that the other indices likewise remain unchanged, with node v simply "inserted" between the old indices v-1 and v.) To obtain a feasible path from 0 to the new node v we require

$$\sum_{o}^{v-1} - (r_j+s_j) + s_v^* \leq d.$$

Substituting in the values for s_v^* and d yields (14). Similarly, for $v \leq k<i$ it is required that

$$\sum_{o}^{k} - (r_j+s_j) + (r_v^*+s_v^*) \leq d_k$$

and substitution directly implies (15). Finally, (16) follows by

---

† We use an asterisk here to distinguish r_v^* and s_v^* for the new node v from r_v and s_v for the old v, which may not be deleted.

the same argument that justified (13). That (14) and (15) are irrelevant when $d=d_i$ is immediate. This completes the proof.

We now consider how to take advantage of the relations given by the two preceding theorems. We will show that it is possible to test arcs for exclusion without explicitly computing values of α_h^k. To see this, let $\beta_j = d_j + r_j - \sum_o^j$ and define the list $(m_1, m_2, \ldots, m_n)$ of the indices $1, 2, \ldots, n$ so that $h<k$ implies $\beta_{m_h} \leq \beta_{m_k}$.

Then, beginning with m_1, consider the arcs (j, i) such that $j>m \geq i$. For all such j and i, $\alpha_i^{j-1} = \beta_{m_1}$. Hence (12) can be applied to β_{m_1} without requiring separate determinations of α_i^{j-1} (for $j>m_1 \geq i$). Likewise for all $j=m_1$, we have $\alpha_j^n = \beta_{m_1}$, and (13) can be applied without separate determinations of α_j^n for these j.

In general, for any m_k let $k_1 = \underset{\substack{j<k \\ m_j<m_k}}{\mathrm{Max}} (m_j)$ and $k_2 = \underset{\substack{j<k \\ m_j<m_k}}{\mathrm{Min}} (m_j)$. If k_1 (or k_2) is not meaningfully defined, we let $k_1=0$ ($k_2=n+1$). Then $\alpha_i^{j-1} = \beta_{m_k}$ for all $i>k_1$ and all $j \leq k_2$ such that $j>m_k \geq i$. Likewise, $\alpha_j^n = \beta_{m_k}$ for all j such that $m_k \geq j>k_1$ unless $k_2 \leq n$, in which case α_j^n is given by β_{m_k} for a smaller value of k.

From these observations the checking of relations (12) and (13) can be considerably facilitated, and similar remarks apply to checking (15) and (16). (Note (16) and (13) are disposed of simultaneously by letting j and i interchange roles in α_j^n and α_i^n).

5. Applying Relaxed Tests

Since $a_{ji} \geqq r_j + s_i$, we note that (12) implies the less restrictive relation $\alpha_i^{j-1} \geqq r_j + s_j$. This information can be exploited with the following test procedure.+

Suppose that a (second) list $Q=(q_1, q_2, \ldots, q_n)$ of indices is created so that $r_{q_1} + s_{q_1} \leqq r_{q_2} + s_{q_2} \leqq \ldots \leqq r_{q_n} + s_{q_n}$. Given that $\alpha_i^{j-1} = \beta_{m_k}$ for j satisfying $k_2 \geqq j > m_k$ (and i satisfying $m_k \geqq i > k_1$) consider the least h such that $k_2 \geqq q_h > m_k$. Then if $\beta_{m_k} < r_{q_h} + s_{q_h}$, it follows that the arcs (j, i) are inadmissible for all j, i satisfying $k_2 \geqq j > m_k$ and $m_k \geqq i > k_1$.

More generally, by reference to the list Q, one can readily determing the indices $j_1, j_2, \ldots, j_p$ consisting of those j satisfying $k_2 \geqq j > m_k$, and $r_{j_1} + s_{j_1} \leqq r_{j_2} + s_{j_2} \leqq \ldots \leqq r_{j_p} + s_{j_p}$. For any q such that $1 \leqq q \leqq p$, if $\beta_{m_k} < r_{j_q} + s_{j_q}$, then the arcs (j_h, i) are inadmissible for all j_h, i for which $m_k \geqq i > k_1$ and $q \leqq h \leqq p$. Similarly, if $\beta_{m_k} \geqq r_{j_q} + s_{j_q}$, then this relation will hold for all j_h such that $h=q$. Thus, regardless of the outcome of the test, it will not have to be reapplied for a subset of the j_h (either for $h \leqq q$ or $h \geqq q$.)

An analogous test procedure can be used to exploit the relation $\alpha_v^{i-1} \geqq r_i + s_i$ (implied by (15)).

+ Similar (slightly less restrictive) information is treated in a different manner by Pierce and Hatfield in [3].

References

1. Glover, Fred, "Truncated Enumeration Methods for Solving Pure and Mixed Integer Linear Programs", WP-27, Operations Research Center, University of California, Berkeley, May 1966.

2. Little, J. D. C., Murty, K. G., Sweeney, D. W., and Karel, C., "An Algorithm for the Traveling Salesman Problem," Operations Research II, 1963.

3. Pierce, J. F., and Hatfield, D. J., "Production Sequencing by Combinatorial Programming," Cambridge Scientific Center, IBM, Cambridge, Massachusetts, 1966.

4. Roy, B., and Bertier, P., "Une Procedure de Resolution pour une Classe de Problems Pouvant Avoir Un Charactere Combinatoire," ICC Bulletin, Vol. 4, 1965.

5. Smith, W. E., "Various Optimizers for Single Stage Production," Naval Research Logistics Quarterly, 3, 59-66, 1956.

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) Graduate School of Industrial Administration Carnegie Institute of Technology	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP Not applicable

3. REPORT TITLE

SEARCH-REDUCTION TESTS FOR A COMBINATORIAL PRODUCTION SEQUENCING ALGORITHM

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

Technical Report

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

Glover, Fred

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December, 1966	14	5

8a. CONTRACT OR GRANT NO. NONR 760(24) b. PROJECT NO. NR 047-048 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) Management Sciences Research Report No. 98 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) none

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited

11. SUPPLEMENTARY NOTES None	12. SPONSORING MILITARY ACTIVITY Logistics and Mathematical Statistics Branch Office of Naval Research Washington, D. C. 20360

13. ABSTRACT

Recently a number of tree search algorithms have been proposed for solving combinatorial optimization problems. The practical value of such approaches depends heavily on the tests used to identify and exclude dominated alternatives from consideration. An interesting study by Pierce and Hatfield gives a useful tree search algorithm for a certain class of production sequencing problems, and further motivates the search for new theoretical results on which more effective tests can be based. In this paper we attempt to provide such results and to lay a foundation for exploiting them in an efficient way.

Unclassified

Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
production sequencing combinatorial optimization						

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.

2a. 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.

2b. 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.

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

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

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

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

9a. 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.

9b. 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.

Unclassified

Security Classification