WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

TREE-SEARCH ALGORITHMS FOR QUADRATIC ASSIGNMENT PROBLEMS*

J. F. Pierce and W. B. Crowston

May 1969

389-69

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

TREE-SEARCH ALGORITHMS FOR QUADRATIC ASSIGNMENT PROBLEMS*

J. F. Pierce and W. B. Crowston

May 1969

389-69

*The authors gratefully acknowledge the support of this research received from the Boeing Research Project and from the National Science Foundation under grant No.70985.

TREE-SEARCH ALGORITHMS FOR QUADRATIC ASSIGNMENT PROBLEMS

J. F. Pierce and W. B. Crowston

Sloan School of Management

Massachusetts Institute of Technology

Cambridge, Massachusetts

Abstract

Problems having the mathematical structure of a quadratic assignment problem are found in a diversity of contexts: by the economist in assigning a number of plants or indivisible operations to a number of different geographical locations; by the architect or industrial engineer in laying out activities, offices or departments in a building; by the human engineer in arranging the indicators and controls in an operators control room; by the electronics engineer in laying out components on a backboard; by the computer systems engineer in arranging information in drum and disc storage; by the production scheduler in sequencing work through a production facility, and so on.

In this paper we discuss several types of algorithms for solving such problems, presenting a unifying framework for some of the existing algorithms, and describing some new algorithms. All of the algorithms discussed proceed first to a feasible solution and then to better and better feasible solutions, until ultimately one is discovered which is shown to be optimal.

In a subsequent paper we shall discuss our computational experience with a number of these algorithms.

TABLE OF CONTENTS

I. Introduction 1

II. Single-Assignment Algorithms 9

III. Extensions of the Single-Assignment Algorithms 23

IV. Pair-Assignment Algorithms 39

V. Pair-Exclusion Algorithms. 45

VI. Concluding Remarks 63

References 66

LIST OF FIGURES

Figure 1 Illustrative tree with each level representing a unique plant. 10

Figure 2 Illustrative tree with each level representing a unique location 11

Figure 3 Flow and distance data for illustrative problem of Gavett and Plyter 17

Figure 4 Tree elaborated for problem of Figure 3 using Gilmore-Lawler algorithm with bounds of equation (8). 21

Figure 5 Tree elaborated for problem of Figure 3 using Gilmore-Lawler algorithm with bounds of equation (9). 22

Figure 6 Tree elaborated for problem of Figure 3 with alternative single-assignment algorithm 34

Figure 7 Tree elaborated for alternative single-assignment algorithm together with testing for mandatory assignments 36

Figure 8 Data for problem of Gavett and Plyter represented in terms of pairs of assignment. 39

Figure 9 Tree elaborated for problem by Gavett and Plyter algorithm 42

Figure 10 Optimal linear assignment matrices to problem with data in Figure 8: (a) all assignments admissable, (b) (AB,14) inadmissable. 46

Figure 11 Partial tree elaborated by pair-exclusion algorithm . . 48

Figure 12 Conflicts between pairs of assignments in optimal linear assignment solutions to illustrate problems in Figure 8 51

Figure 13 Illustrative, alternate ways to partition set of quadratic assignment solutions into subsets in pair-exclusion algorithms 52

Figure 14 Tree elaborated for problem of Figure 3 with illustrative pair-exclusive algorithm. 58

I. Introduction

The quadratic assingment problem is one which arises in a diversity of contexts and has been investigated by a number of researchers. Formally, the problem may be stated simply as follows: given n^4 cost coefficients S_{ijkq}, $(i,j,k,q = 1,2,3,\ldots,n)$ determine values of the n^2 variables $x_{ij}(i,j = 1,2,3,\ldots,n)$ so as to:

$$\text{Minimize} \quad Z = \Sigma_{i,j}\Sigma_{k,q} S_{ijkq} x_{ij} x_{kq} \tag{1}$$

$$\text{Subject to:} \quad \sum_{i=1}^{n} x_{ij} = 1 \quad j = 1,2,3,\ldots,n \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \quad i = 1,2,3,\ldots,n \tag{3}$$

$$\text{and} \quad x_{ij} = 0,1 \quad i,j=1,2,\ldots,n \tag{4}$$

Historically its name derives from the fact that mathematically its structure is identical to that of the classical linear assignment problem concerning the assignment of n indivisible entities to each of n mutually exclusive classes, one entity per class, except that in the present case the objective function (1) contains terms which are quadratic in the decision variables.

Commencing in the field of economics, Koopmans and Beckman [15] identified this name with the structure of problems which concern the assignment of n indivisible plants to n locations. Suppose the cost of establishing and operating plant i at location j plus the cost of

supplying prespecified product demand to customers from this location is $c_{ij}, i,j = 1,2,\ldots,n$, these costs being independent of other plant-location assignments. Also suppose that between plants i and k there is a commodity flow of f_{ik} units (e.g. weight) which is independent of plant location, and that the cost per unit flow between locations j and q is d_{jq}, independent of plant assignments. Then in this context (1) becomes:

$$Z = \sum_{i,j} c_{ij} x_{ij} + \sum_{i,j} \sum_{k,q} f_{ik} d_{jq} x_{ij} x_{kq} \tag{5}$$

where

$$S_{ijkq} = \begin{cases} f_{ik} \cdot d_{jq}, & \text{if } i \neq k \text{ or } i \neq q \\ c_{ij} + f_{ii} d_{jj}, & \text{if } i = k \text{ and } j = q \end{cases}$$

As a generalization to this assignment problem, Lawler [18] discusses the multicommodity case in which there is a flow f_{ik}^t for each commodity t and a cost per unit flow between locations j and q of d_{jq}^t. As another generalization, Graves and Whinston [12] point out the possibility in this model of a cost component w_{ijkq} that depends on a pair of assignments, such as might be illustrated by the cost of laying a pipe line between two plants. Combining these we thus have the more general cost expression:

$$Z = \sum_{i,j} c_{ij} x_{ij} + \sum_{i,j} \sum_{k,q} w_{ijkq} x_{ij} x_{kq} + \sum_{i,j} \sum_{k,q} \sum_{t} f_{ik}^t d_{jq}^t x_{ij} x_{kq} \tag{6}$$

where

$$S_{ijkq} = \begin{cases} w_{ijkq} + \sum_t f_{ik}^t d_{jq} & \text{if } i \neq k \text{ or } j \neq q \\ c_{ij} + f_{ii} d_{jj} & \text{if } i = k \text{ and } j = q \end{cases}$$

In the event there are no inter-plant flows $f_{ik}^{t} = o$ for all i,k,t, and the problem in (5) reduces to the linear assignment problem. When c_{ij} = 0 for all i, j and

$$f_{ik} = \begin{cases} 1 & k = i + 1,\ i < n \\ 1 & i = n,\ k = 1 \\ 0 & \text{otherwise} \end{cases}$$

the problem reduces to the traveling salesman problem.

At a more micro-economic level this problem arises in the context of locating department of offices within a plant or store to minimize the cost of transporting product, the total distance walked, or some similar measure [1,24,31,32]. At a still more micro level it is the problem of locating operator dials and indicators on a display and control panel. In other contexts (1) - (4) is the problem of minimizing "latency" in magnetic drum or disc storage computers [19], minimizing total wire length in the placement of electronic components in assemblies [2,7,30], or minimizing total flow time or total variable production and inventory carrying cost in various production sequencing problems [22].

In some contexts there may be constraints applicable to the problem which are not represented in the statement as embodied in (1) - (4). For example, there may be a restriction that plant i not be located at j, or a restriction that plants i and k be not more than distance d apart, or that i and k be closer than d. All single and pairwise constraints

of this kind are readily accommodated in (1) - (4) by setting $s_{ijkq} = M$, $M \to \infty$. However, more difficult to include are constraints involving three or more assignments unless it be possible to derive an equivalent set of pairwise constraints. While the algorithms to be discussed can be adapted for such cases the resulting algorithms may not be as efficient.

From a problem-solving point of view there may in practice, be fewer than n plants, $m < n$, but with no loss of generality we may assume $m = n$ by introducing dummy plants $m + 1, m + 2, \ldots, n$ with $c_{ij} = 0$ and $f_{ik} = 0$ for all $i,k > m$. Also it is noted that, stated in terms of a plant i and its location 1(i), problem (1) - (4) and its variations is the problem of finding a permutation $\{1(1), 1(2), 1(3), \ldots, 1(n)\}$ of the integers $\{1,2,3,\ldots,n\}$ so as to minimize:

$$Z = \sum_{i,k} s_{ik1(i)1(k)}$$

This representation will sometimes be used in the following discussion.

For solving quadratic assignment problems a number of procedures of both the reliable[1] and the unreliable type have been reported in the literature. Reliable procedures for determining optimal solutions with objective function (5) have been presented by Gilmore [10] and Lawler [18], and for the symmetric case[2] of (5) by Land [16] and Gavett and Plyter [8]. For the problem with the general objective function (6) a reliable algorithm has been given by Lawler [18]. On the other hand unreliable procedures have been reported for various quadratic assignment problems by

---

[1] By a reliable problem-solving procedure we shall mean one which, if carried through to completion, guarantees the discovery of an optimal solution.

[2] A symmetric distance matrix $d_{jq} = d_{qj}$ for all j and q allows the flow between activities to be summed, $f'_{ik} = f_{ik} + f_{ki}$, thereby possibly simplifying the problem-solving process.

Armour and Buffa [1], Gaschutz and Ahrens [7], Gilmore [10], Graves and Whinston [12], Hillier [13], Hillier and Connors [14], Nugent, Vollman, and Ruml [23], Pegels [24], Steinberg [30], Whitehead and Eldors [32] and by Wimmert[3] [33]. An interesting experimental comparison of a number of these latter procedures is presented in the paper by Nugent et al [23].

From a computational point of view, the present status can perhaps be succinctly summarized as follows. Existing reliable algorithms can essentially be classified into three groups: the integer programming approach of Lawler [18]; the semi-enumerative procedures of Lawler [18] and Gilmore [10]; and the semi-enumerative approaches of Gavett and Plyter [8] and Land [16]. With presently available integer programming algorithms the first appraoch is impractical even for small problems, in light of the size of the programming problem which results. For the second group we know of no actual computational experience with the algorithms, but as stated by Gilmore [10] his reliable algorithms are "probably not computationally feasible for n much larger than 15." For the third group Gavett and Plyter [8] report that with their algorithm as programmed in Fortran on an IBM 7044, a problem with n=7 required 14 minutes and one with n=8, 42 minutes. In short, in the words of Nugent et al, "one is forced to conclude that no computationally feasible optimal-producing procedure exists at present. Interest must focus on suboptimal procedures."

In the present paper we re-direct attention back to reliable procedures for solving quadratic assignment problems. The methods to be considered are

---

[3]The algorithm of Wimmert was originally presented as yielding optimal solutions but was subsequently shown by Conway and Maxwell [3] to yield suboptimal solutions.

those which have equivalently been referred to as branch and bound procedures [20], back-track programming procedures [11], implicit enumeration procedures [9], reliable heuristic programming procedures [25], and others. Essentially these are the types of methods that were used in the algorithms of Gavett and Plyter [8], Gilmore [10], Land [16], and Lawler [18]. In the following sections we present a unified framework in which to compare the existing algorithms, and discuss some alternative search strategies and other means by which it may be possible to devise more efficient procedures.

Before turning to the algorithms in detail, however, we shall comment briefly on the nature of the methods to be considered and the reasons for our inclination toward them. The most common name for the procedures to be investigated is "branch and bound," the name given to the ideas employed by Little et al. [20] in their algorithms for solving the traveling salesman problem. Their work has made demonstrably clear the great potential of these methods for the solution of complex combinatorial problems. The "branch" notion stems from the fact that in terms of a tree of alternate potential solutions to the problem the procedure is continually concerned with choosing a next branch of the tree to elaborate and evaluate. The "bound" term denotes their emphasis on, and effective use of, means of bounding the value of the objective function at each node in the tree, both for eliminating dominated paths and for selecting a next branch for elaboration and evaluation.

Perhaps the essence of the procedures to be considered is most succinctly captured in the meaning given to "combinatorial programming" by its authors Rossman and Twery [26]. By combinatorial programming we mean procedures developed on the basis of two principal concepts: the use of a controlled enumerative technique for (implicitly) considering all potential solutions; and the elimination from explicit consideration of particular potential solutions which are known from dominance, bounding and feasibility considerations to be unacceptable. All of the equivalent terms for these methods will be used interchangeably thoughout.

As will become apparent many of the feasibility, dominance and bounding considerations presented in the following sections are also applicable in other combinatorial programming algorithms as well as in other types of problem-solving procedures. In the following sections, attention will be focused on combinatorial programming algorithms in which problem-solving proceeds first to the discovery of a feasible solution and then to successively better feasible solutions until ultimately one is discovered which is shown to be optimal. We direct attention to these procedures principally because they have the following three desirable attributes.

First, with such procedures there is a possibility of obtaining usable solutions and terminating problem-solving prior to the ultimate completion of the problem-solving process. This feature is obviously important for quadratic assignment problems.

Second, these procedures exploit in an efficient manner information that is available beforehand pertaining to the value of an optimal solution, as is always the case for instance when a feasible solution is known from experience or has been derived with the aid of a heuristic (unreliable) procedure. That is, since in the procedures to be investigated subsequent search is always directed toward solutions with a value better than the best known so far and will terminate if a solution is discovered attaining a known lower bound, use of a priori knowledge of upper and/or lower bounds serves to reduce the region that need be searched. Therefore, in contexts where good heuristic procedures are available, for example, a system of problem-solving procedures may prove advantageous in which the reliable, direct algorithm is employed as an adjunct to the heuristic procedures, an adjunct to be employed when in a given instance the economics of the problem and problem-solving effort together with environmental considerations warrant the added search for a solution better than that yielded by the heuristic procedures.

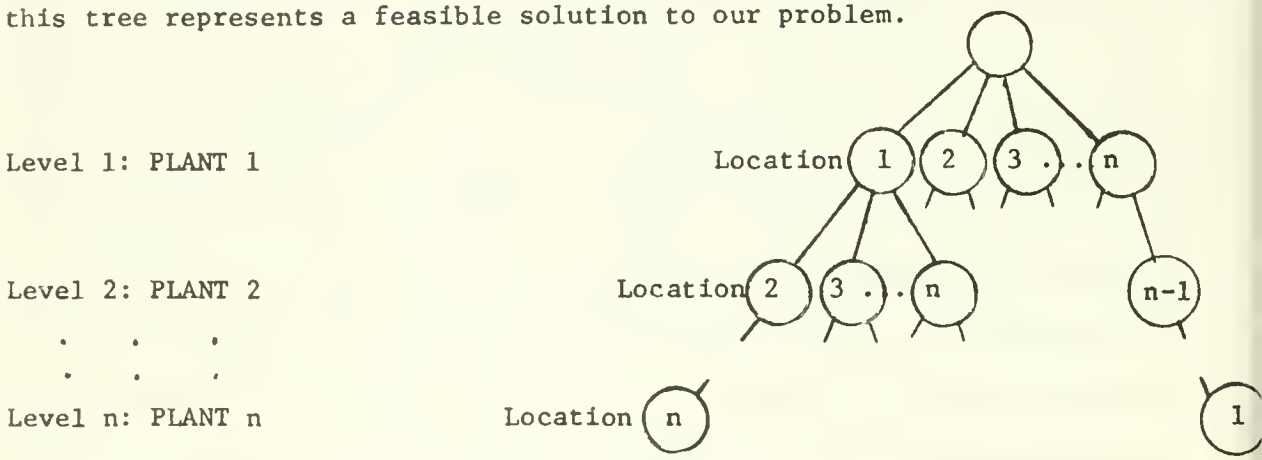
Thirdly, these algorithms are attractive in that with slight modification they can be employed to find not only an optimal solution, but all optimal solutions, or a specified number of most preferred solutions, or all solutions having a value within a specified interval of the optimal value, and so on. Such possibilities may be of interest in contexts in which there are attributes of the problem of importance which are not directly represented in the model of the problem being solved.

II. Single-Assignment Algorithms

As noted earlier, a first principle of combinatorial programming is the use of a controlled enumeration procedure for systematically considering, at least implicitly, all potential solutions. For quadratic assignment problems there are at least two general procedures; one based on the systematic consideration of single assignments, x_{ij}, and one based on the systematic consideration of pairs of assignments, $x_{ij}\ x_{kq}$. Both types have appeared in reliable algorithms to date, Gilmore [10] and Lawler [18] using the former, and Land [16] and Gavett and Plyter [8] using the latter. In this section we consider algorithms of the former type.

A property of a feasible solution to the problem (2)-(4) is that with the variables x_{ij} arranged in an n x n matrix $\underline{X} = \|x_{ij}\|$ there exists exactly one variable in each row and column of the row assignment matrix $\underline{X}$ having unit value. To satisfy the requirements for considering all potential solutions, we therefore need a controlled enumeration procedure for generating all possible ways of selecting one element from each row and column of $\underline{X}$. One possible procedure, for example, is to successively select elements from successive rows of $\underline{X}$ and to select within a given row the first element (when scanned from left to right, say) which will result neither in a nonfeasible solution nor in a solution already generated. Ultimately upon making a selection from row n and hence completing the specification of a solution, the procedure backs up to row n-1, selects the next admissable element and steps forward to row n again. The results may be

represented in a tree structure with the i th level of nodes representing the permissable assignments for plant i, l(i), in the permutation {l(1), l(2), ..., l(n)} as shown in Figure 1. Note that each path in this tree represents a feasible solution to our problem.



Illustrative tree with each level representing a unique plant.

Figure 1

The procedure described above would thus elaborate the tree shown from left to right. Enumeration may thus be equivalently viewed as entailing the successive row by row selection of an element from matrix $\underline{X}$ to include in the permutation $\theta = \{l(1), l(2), \ldots, l(n)\}$ or as entailing the successive level by level selection of branches in a tree (one branch per level) until a terminal node is reached at level n. Upon reaching a terminal node, the corresponding assignment is evaluated and the tree-evaluation process backtracks to the lowest node on the path for which all branches have not been elaborated, selects the next and resumes. When the process has backtracked to the origin node and all its branches have been enumerated, generation and hence problem solving is complete.

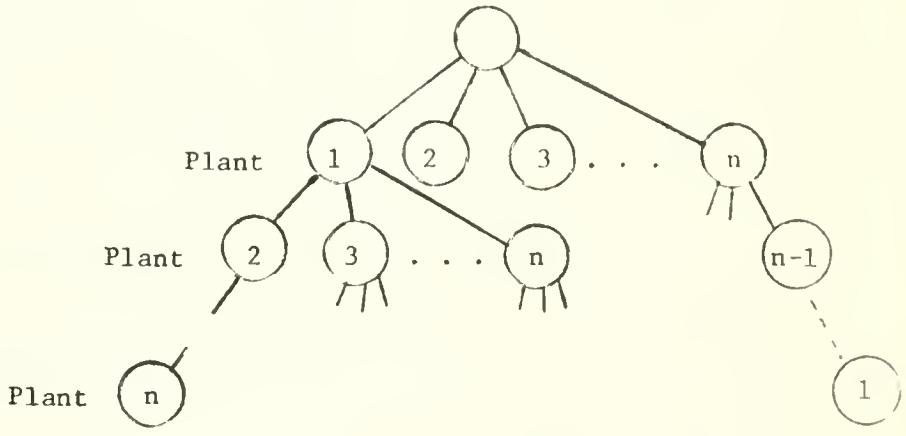
In addition to this illustrative enumeration scheme there are many others for systematically selecting an element from each row and column of $\underline{X}$. For example, if we interchange the words "row" and "column" in the cited procedure, we have a tree with levels corresponding to locations rather than plants, as shown in Figure 2.

Level 1: Location 1

Level 2: Location 2

. . .

Level n: Location n



Illustrative tree with each level representing a unique location.

Figure 2

If in the process of exhaustive enumeration, it becomes known with certainty for a particular node assignment, that all paths which pass through this node represent potential solutions which are nonfeasible or are dominated by a feasible solution already discovered, then the enumeration and evaluation of all branches emanating from this node can be eliminated without imparing the reliability of the problem-solving procedure. Let us consider possibilities for reducing search based on dominance considerations. For any two feasible solutions, Θ_i and Θ_j, Θ_i dominates Θ_j if $Z_{\Theta_i} \leq Z_{\Theta_j}$.

If in an exhaustive procedure Θ_i denotes the ith feasible solution discovered then in general $Z_{\Theta_i} \gtreqless Z_{\Theta_{i+1}}$. Through dominance consideration we seek to

reduce enumeration and evaluation of feasible solutions to a subset in which

$$Z_{\Theta_1} > Z_{\Theta_2} > \ldots > Z_{\Theta_{(u)}}$$

where $Z_{\Theta_{(u)}}$ is an optimal solution.[4] In effect, this is accomplished by affixing to the problem throughout the search process a constraint of the form $Z_{\Theta_j} < Z_{\hat{\Theta}}$ where Θ_j is solution discovered so far. In essence, the optimization problem is hereby transformed into a sequence of u feasibility problems for purposes of problem-solving. To implement this type of consideration a lower bound B is developed on Z_{Θ_j} at each node in the tree for all Θ_j whose paths pass through the given node, that is, $B \leq Z_{\Theta j}$ for all Θ_j; if $B \geq Z_{\hat{\Theta}}$ then no branches emanating from th node need be explicitly considered

For the quadratic assignment problem these bounds can be determined in number of ways. Suppose we have arrived at a node on level v of a tree of th problem (v=o,1,...,n-1) having made assignments ($\bar{i}$, $1(\bar{i})$) and we now wish to chose a next assignment, x_{ij}. Let a_{ij}^{v} be a lower bound on the sum $S_{ijij} + \sum_{k \in I} S_{ijkl(k)} + \frac{1}{2} \sum_{p \notin I} S_{ijpl(p)}$, where I is the set of assigned plants $\bar{i}$. Since the first two terms are known exactly we have

$$a_{ij}^{v} = S_{ijij} + \sum_{k \in I} S_{ijkl(k)} + \frac{1}{2} \bar{a}_{ij}^{v} \qquad (7)$$

where $\bar{a}_{ij}^{v}$ is a lower bound on the sum of (n-v) terms, $\sum_{p \notin I} S_{ijpl(p)}$. As noted by Lawler a minimum bound $\bar{a}_{ij}^{v}$ can be obtained by solving the linear assignment problem of dimension (n-v). In the special case of (5) where $S_{ijkq} = (f_{ik}d_{jq} + f_{ki}d_{qj})$, both Lawler and Gilmore point out that a lower bound is more easily obtained by matching the largest value of f_{ik} with the smallest d_{jq}, the next to largest f_{ik} with the second smallest

---

[4] It is possible that more than one optimal solution could exist, e.g., $Z_{\Theta_{(u)}} = Z_{\Theta_{(u+1)}} = \ldots = Z_{\Theta_{(v)}}$.

d_{jl} and so on. For the symmetric case of $d_{jq}=d_{qj}$ and $t=1$, $S_{ijkq}=(f_{ik}+f_{ki})d_{jq}$ so that the sum of the products of these values actually gives a minimum value. This applies equally to the case $S_{ijkq} = \sum_t (f_{ik}{}^t d_{jq}{}^t + f_{ki}{}^t d_{qj}{}^t)$.

Thus by determining an appropriate bound $\bar{a}_{ij}^{v}$ a value can be obtained for each unassigned plant i and location j remaining at level v. Let $\underline{A}^{v}$ denote the resulting (n-v) matrix $\| a_{ij}{}^{v} \|$. If we denote by $Z_v^* = \text{Min.}\{ \sum_{ij} a_{ij}^{v} x_{ij} \}$ the value of an optimal solution to a linear assignment problem defined by $\underline{A}^{v}$, then a lower bound B^{v} on all feasible solutions whose path passes through the node is:

$$B^{v} = Z_v^* + \sum_{i,k \in I} S_{il(i)kl(k)} \qquad (8)$$

Thus if $B^{v} \geq Z_\Theta$ then the search process can be backtraced immediately without considering any of the branches emanating from the node.

An alternative to this bound which requires less computation (but is also less stringent) is suggested by Gilmore for the Koopmans-Beckman problem which he investigates.

With the objective function

$$Z = \sum_{k \in I} c_{kl(k)} + \sum_{i,k \in I} f_{ik}\ d_{l(i)l(k)}$$

he suggests at level v the bound:

$$B_v = \sum_{k \in I} c_{kl(k)} + \sum_{i,k \in I} f_{ik}\ d_{l(i)l(k)} + S_1+S_2+S_3 \qquad (9)$$

where S_1 is the value of an optimal solution to the (n-v)- dimensional assignment problem defined by $\| c_{rt} \|$ $r \neq i$ and $t \neq l(i)$ for any $i \in I$,

$$S_2 = \sum_{\substack{i,k \in I \\ j \neq l(q),\ q \in I}} (\hat{f}_{ik} \cdot \hat{d}_{l(i),j} + \hat{\hat{f}}_{ki} \cdot \hat{\hat{d}}_{jl(i)}$$

where the largest element $\hat{f}_{ik}$ is matched with the smallest element $\hat{d}_{1(i)j}$ and so on, and likewise for $\hat{\hat{f}}_{ki}$ and $\hat{\hat{d}}_{j1(i)}$, and finally where

$$S_3 = \sum_{i,k \notin I} f_{ik}^* \, d_{jg}^*$$

$$j,g \neq 1(w)$$

$$\text{any } w \in I$$

where the largest element of f_{ik}^* is paired with the smallest element d_{jg}^*, etc.

Let us now return to the discussion of controlled enumeration procedures. The controlled enumeration procedures mentioned previously are data-independent with respect to the order in which potential solutions are investigated; for every problem having the same number n of plants the ord is identical regardless of the characteristics of the particular problem being solved.[5] In such procedures little problem-solving time is invested in determining a next branch in the tree for investigation and (at least in the procedures discussed)[6] in keeping track of the part of the tree investigated so far. Perhaps, however, more efficient combinatorial programs may result by expending additional time on these functions and making the ordering of search more dependent on the particular features of the problem being solved.

There are at least two basic search patterns of a general data-dependent nature wherein at each point in the search process a branch is selected for elaboration to the next level which has associated with it a most preferred value of a measure W. A natural characteristic to employ

---

[5] While perhaps making it possible to eliminate from explicit investigation particular subsets of potential solutions in a given problem, feasibility and/or dominance considerations do not change the order of consideration.

[6] Bookkeeping for the portions of the tree investigated so far would be considerably more extensive, for instance, for a level-by-level type search pattern such as with dynamic programming where all nodes-on-level j of tree are elaborated before proceeding to level j + 1 (or j-1), etc.

as a measure W, for instance, is a lower bound on the total cost B_v of all potential feasible solutions passing through the node.

In the "flooding" type pattern a branch is always selected from among all branches in the current tree requiring elaboration to a next level. In the second type pattern, search is directed towards the enumeration of complete paths so that in turn one branch is selected at level 1, then one at level 2, etc.: at level j all branches emanating from the single node selected at level j-1 are evaluated in terms of the measure B and a most preferred one selected for elaboration to level j + 1. Upon reaching a terminal node or one for which it is known that all paths passing through it are dominated or nonfeasible, the process backs up as usual to the lowest node for which all branches have not been considered. We will consider principally this latter type wherein search can be directed first to the discovery of a feasible solution and then to better and better feasible solutions, although the discussion will generally be equally applicable to the other basic strategies and to mixtures thereof as well.

At this point we can now summarize the approaches of Lawler and Gilmore in the following way. Both approaches employ a search strategy wherein the j^{th} level in the tree corresponds to the assignment of some plant to the j^{th} location, as suggested by the tree of Figure 2. The ordering of locations $j_1, j_2, \ldots, j_n$ is arbitrary or, perhaps as suggested by Gilmore, in accord with some heuristic ordering rule such as by decreasing sums $\sum_{q \neq j} (d_{qj}+d_{jq})$. Given this ordering, both employ the data-dependent level-by-level search strategy wherein at level v the node chosen for elaboration to the next level is one for which the bound B_{v+1} is lowest among those not yet elaborated. Both approaches explicitly elaborate the

(n-v) nodes branching from the node, evaluate B_{v+1} for each, and repeat the process. Should $B_k \geq Z^{\blacktriangle}_{\Theta}$ for all nodes at any level k the process backtracks to the lowest level in the tree for which there exists an unelaborated node which is not dominated, and resumes. The difference in the algorithms lies in the bounds B_v used. Lawler being concerned with the general quadratic assignment problem with objective function (1) solves a linear assignment problem to get each $\bar{a}_{ij}^{v}$ in (7), and then a single linear assignment problem for $\| a_{ij}^{v} \|$ to get B_v, as given in (8); when his problem specializes to the single-commodity Koopmans-Beckman problem, he proceeds in the same manner except that he gets each $\bar{a}_{ij}^{v}$ directly by ordering elements as described earlier. Gilmore focuses on the single-commodity Koopmans-Beckman problem and develops B_v either in the manner described for Lawler or in accordance with (9).

As an illustration, we will solve the Koopmans-Beckman problem of Gavett and Plyter [8] shown in Figure 3, computing bounds according to (8). In our solution to this problem we will examine the locations in the sequence A,B,C,D, so that at the first level for instance, we investiga A-1, A-2, A-3, and A-4, and so on. We begin however at the 0 level with no assignments. Therefore for

$$\text{A-1: } I=\emptyset \quad , \quad \sum_{i,\, k \in I} S_{il(i)kl(k)} = 0$$

$$a_{ij}^{o} = S_{ijij} + \sum_{k \in I} S_{ijkl(k)} + 1/2\ \bar{a}_{ij}^{v}$$

$$= 1/2\ \bar{a}_{ij}^{1}$$

$$\bar{a}_{A-1}^{0} = (7)\ (13) + (6)\ (25) + (2)\ (28) = 297$$

Using this procedure we establish matrix $2\ \| a_{ij}^{1} \|$ and as a result all

d_{jq}

	A	B	C	D
A	—	6	7	2
B	6	—	5	6
C	7	5	—	1
D	2	6	1	—

f_{ik}

	1	2	3	4
1	—	10	20	5
2	18	—	9	4
3	8	6	—	8
4	8	0	15	—

$f_{ik} + f_{ki}$

	1	2	3	4
1	—	28	25	13
2	28	—	15	4
3	25	15	—	23
4	13	4	23	—

All $s_{ijij} = 0$

Flow and distance data for illustrative problem of Gavett and Plyter.

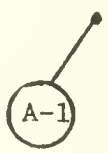
Figure 3

	1	2	3	4
A	297	174	293	152
B	368	254	353	217
C	244	131	245	116
D	156	82	161	73

bounds determined in the following calculations will be exactly twice as large as the true bound. Solving the linear assignment problem defined by this matrix, gives us the resulting matrix with total reduction of 792. The lower bound on the problem is therefore 792/2 = 396.

	1	2	3	4
A	14	0	5	0
B	20	15	0	0
C	4	0	0	7
D	0	35	0	48

Reduction 792



A-1: $2\,a_{ij}^{1} = 2S_{ijij} + 2\sum_{k\epsilon I} S_{ijkl(k)} + \bar{a}_{ij}^{1}$

for cell B-2 we have,

$$S_{ijij} = 0$$

$$\sum_{k\epsilon I} S_{ijkl(k)} = (f_{1,2})(d_{A,B}) = (28)(6) = 168$$

$\sum_{k\epsilon I} S_{ijkl(k)} =$

	2	3	4
B	168	150	78
C	196	175	91
D	56	50	26

This term represents the interaction of the previous assignment (A-1) with the possible new assignment (B-2)

$$\bar{a}_{ij}^{1} = (d_{B,C})(f_{2,3}) + (d_{B,D})(f_{2,4})$$

$$= (5)(15) + (6)(4) = 99$$

where $d_{B,C} \geq d_{B,D}$, and $f(2,4) \leq f(2,3)$

$\Sigma \bar{a}_{ij}^{1} =$

	2	3	4
B	99	205	139
C	35	98	43
D	39	113	47

and we now have

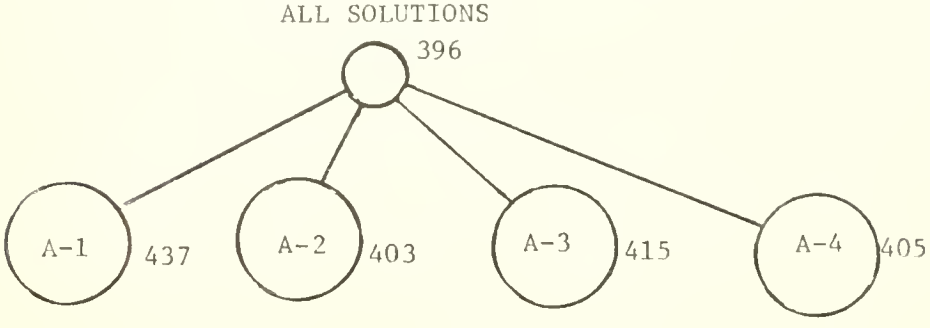
$2\ a_{ij}^{1} =$

	2	3	4
B	435	505	295
C	427	448	225
D	151	213	99

Solving this linear assignment problem we obtain a reduction of 873 with the resulting matrix. To get the desired bound we now divide by two (since the elements in the matrix were $2a_{ij}^{1}$) and round the resulting fraction up since only integer solutions to the problem are feasible.

	2	3	4
B	0	8	0
C	62	21	0
D	0	0	88

Similarily we develop level 1 of the enumeration tree.



We select the smallest of these, A-2 to develop first at level two. We now develop cell C-3 given previous selections of A-2, B-1

$$\sum_{k\in I} S_{ijkl(k)} = (f_{3,2})(d_{C,A}) + (f_{3,1})(d_{cb}) + (f_{1,2})d_{(A,B)}$$

$\sum_{k\in I} S_{ijkl(k)}$ =

	3	4
C	338	369
D	288	194

$$\Sigma\bar{a}_{ij}^2 = (d_{CD})(+34) = 23$$

$\Sigma\bar{a}_{ij}^2$ =

	3	4
C	23	23
D	23	23

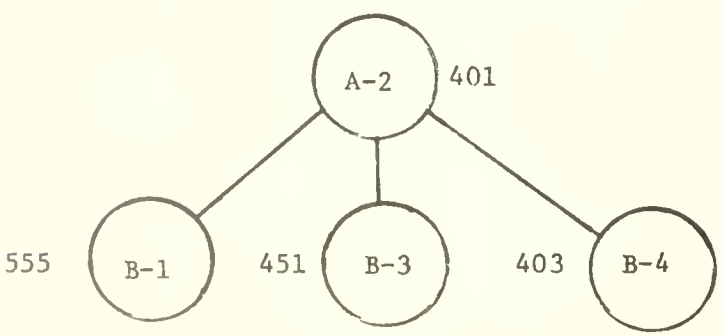
$2\ a_{ij}^2$ =

	3	4
C	699	761
D	599	411

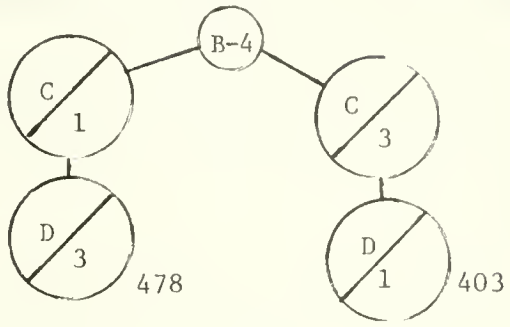
	3	4
C	0	62
D	138	0

$$\text{Reduction } \frac{1110}{2} = 555$$

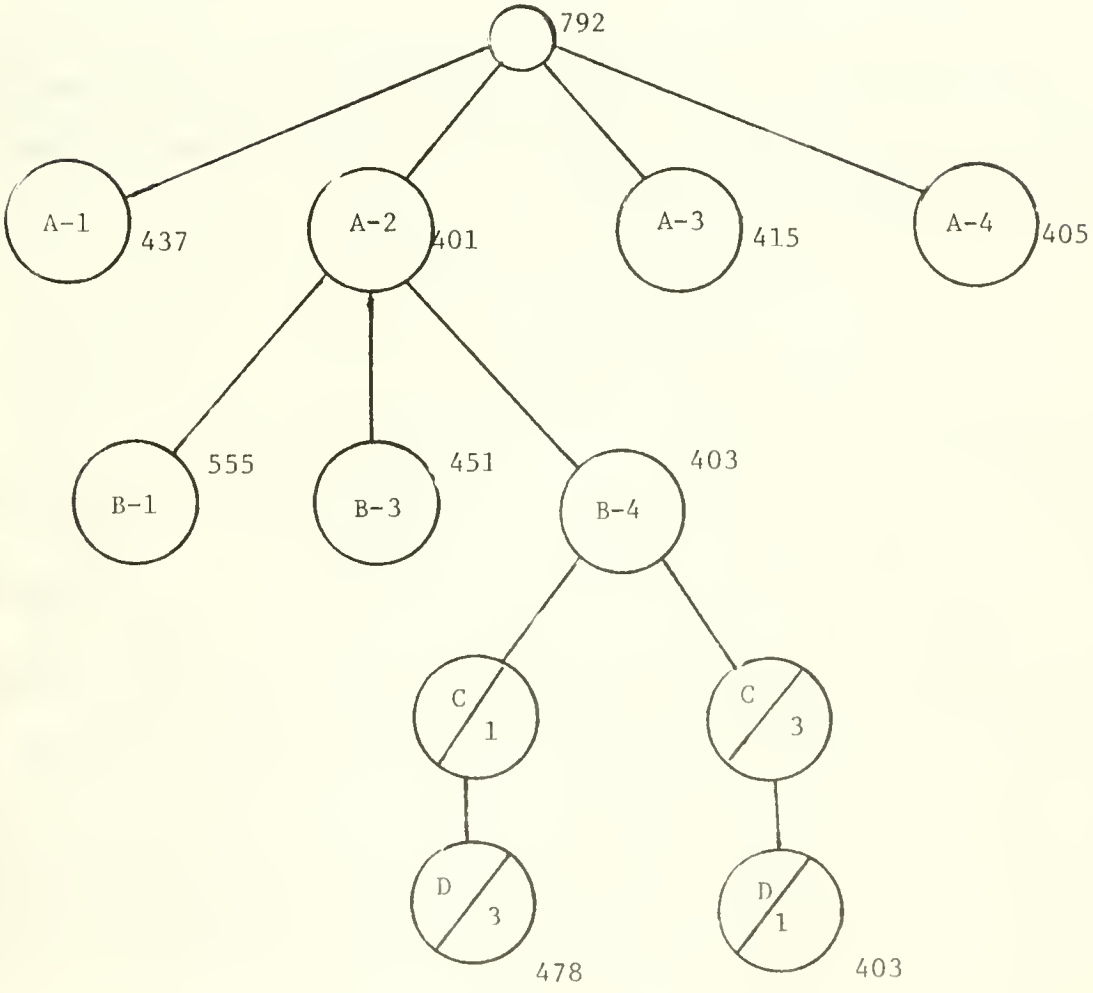
The same set of operations for A-2,B-3 and A-2, B-4



The information for level 3, and 4 are obtained by complete enumeration which we designate with the symbol ⊘. We begin with B-4,



The value of 806 for A-2,B-4,C-3,D-1, is shown to be the optimal solution.



Tree elaborated for problem of Figure 3 using Gilmore-Lawler algorithm with bounds of equation (8).

Figure 4

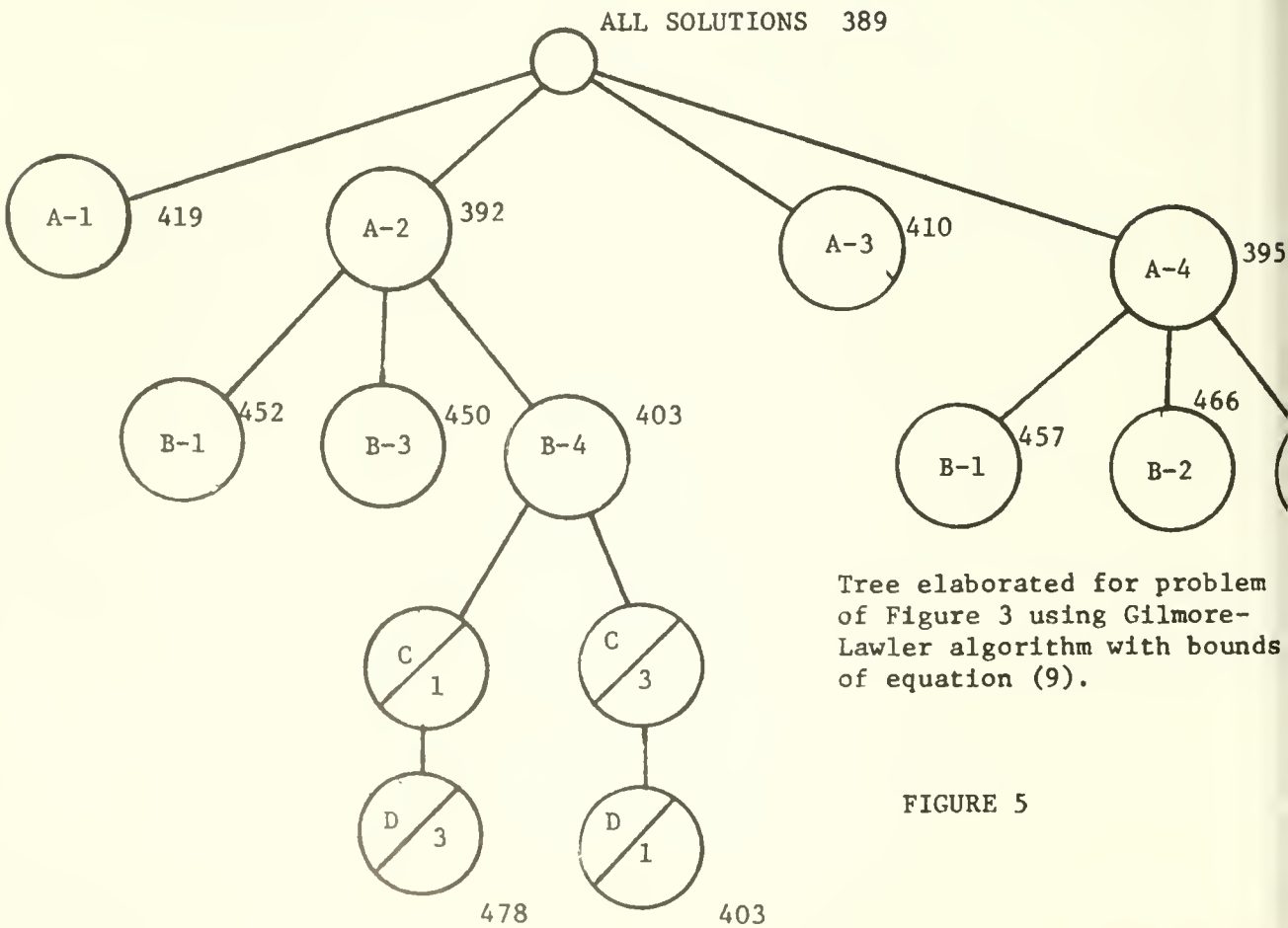
In contrast using the less stringent (but more easily evaluated) bounds of Gilmore (9) the result is as shown by the tree of Figure 5. As illustrations, we will now demonstrate the calculations for a bound on (i) all solutions (ii) solutions with A-4 and (iii) solutions with A-4, B-3 are as follows:

(i) $7 \cdot 4 + 6 \cdot 13 + 6 \cdot 15 + 5 \cdot 23 + 2 \cdot 25 + 1 \cdot 28 = 389$

(ii) $7 \cdot 4 + 6 \cdot 13 + 2 \cdot 23 + 6 \cdot 15 + 5 \cdot 25 + 1 \cdot 28 = 395$

(iii) $6 \cdot 23 + 7 \cdot 4 + 2 \cdot 13 + 6 \cdot 15 + 5 \cdot 25 + 1 \cdot 28 = 435$

The resulting tree is seen to have a greater number of nodes than the former but since the evaluation of each is less time-consuming the total problem-solving time could be smaller.



Tree elaborated for problem of Figure 3 using Gilmore-Lawler algorithm with bounds of equation (9).

FIGURE 5

III. Extensions of the Single-Assignment Algorithm

Turning to prospective improvements in the problem-solving procedures which have been discussed, let us review the steps in the Gilmore-Lawler algorithm at a node on level (v-1) in the tree. For each of the (n-v+1) assignments (i,j) that can be made a lower bound B_v is determined according to (8). To determine each of the values B_v requires the formation of an (n-v+1)x (n-v+1) matrix $\underline{A}^v$ and the solution of the linear assignment problem which it defines. To get each of the elements a_{ij}^v requires in turn the solution of an (n-v) dimensional assignment problem (which in the Koopmans-Beckman problem can be accomplished by simply sequencing the relevant flow and distance values and forming the inner product.) To make an assignment at this node thus entails the solution of (n-v+1) assignment problems of dimension (n-v) and $(n-v+1)(n-v)^2$ problems of dimension (n-v-1). By expending less computation effort in making an assignment at each stage it may, however, be possible to achieve overall improvement in problem solving. In the following discussion we shall continue to employ the same level by level search strategy, choosing at each level a node with a lowest bound, but shall consider alternate ways of assessing the lower bounds.

In (8) the value Z_v^* for an optimal assignment solution to the problem defined by matrix $\underline{A}_v$ was employed in developing bound B_v, but in general any value Z_v constituting a lower bound on Z_v^* may

also be used. One such bound less stringent than Z_v^* can be computed with little effort by the matrix reduction method used by Little, et al, [20]. This method rests on the fact that if T(g) is the cost of an assignment with respect to a matrix A and if T'(g) is the cost of that assignment with respect to matrix A' which is formed by subtracting the constant b from each element of one row or column, then T(g) = T'(g) + b, and the optimal assignments under both matrices are the same. By subtracting appropriate constants from each row and column, a matrix A" of non-negative elements with at least one zero in each row and column can be obtained.[7] Such a matrix they have termed a "reduced matrix" and the sum of the constraints subtracted in forming the matrix, the "amount of reduction." If T"(g) is the total cost of an assignment with respect to the reduced matrix A", and R is the amount of reduction incurred in reducing A, then T(g)=T"(g) + R. Since all elements in A" are non-negative, $T"(g) \geq 0$ for all assignments g, and therefore the amount of reduction R constitutes a lower bound on the optimal value of the assignment problem defined by A.

We will denote by A_v'' a reduced matrix for A_v and by Z_v'' the reduction achieved in reducing it. $Z_v"$ may be used in place of Z_v in determining B_v.

Moreover, between Z_v'' and Z_v^* there are a number of values Z_v' which may be used. Of special interest are those derived during solution of

---

[7] This can be accomplished, for instance, by first subtracting from each column the smallest element in the column and then subtracting from each resulting row the smallest element in the row. In general, however, the reduced matrix and the amount of reduction are not unique but may be dependent on the order in which rows and column are reduced.

the linear assignment problem by a dual algorithm since, for successive iterations t,t + 1, ..., of the algorithm, the value of the objective function Z_v^t is non-decreasing: $Z_v^t \leq Z_v^{t+1} \leq \ldots \leq Z_v^*$. With such an algorithm problem-solving can terminate should the condition

$$Z_v^t + \sum_{i,k\in I} S_{il(i)kl(k)} \geq Z_{\hat{\Theta}}$$

become satisfied for any t, since all paths passing through the node associated with A_v must then be dominated. Algorithms of this type include, for example, the Hungarian method [4], the network flow algorithm of Ford and Fulkerson [6] and the flow algorithm as improved by Sprague [29]. At each iteration in these dual algorithms Z_v^t is the amount of reduction associated with a matrix of non-negative coefficients A_v^t derived from the original, a matrix in which $a_{ij}^t = 0$ for each $x_{ij} = 1$ in the optimal solution.

Besides the choice of the amount of reduction to perform on a matrix $A_v = ||a_{ij}^v||$ there are numerous alternatives for selecting the elements a_{ij}^v to be used in assessing B_v. As was noted earlier, in general any value a_{ij}^v which results from use of an appropriate lower bound in (7) for a_{ij}^{-v} is permissible. Thus, for example, in cases where in developing $A_v = ||a_{ij}^v||$ it is not possible to determine a_{ij}^v simply by sequencing the flow and distance elements and forming their inner product, it may prove efficient to determine lower bounds on the a_{ij}^v in this same way, and then proceed to solve the resulting matrix A_v as discussed. Another possibility is to simply set $A_v = ||a_{ij}^{v-1}||$, $i \neq k$ and $j \neq q$ where (k,q) is the assignment made in passing from level v-1 to level v, and then to employ the bound:

$$B^v = Z_v^t + a_{kl}^{v-1} + \sum_{i,k\in I} S_{il(i)kl(k)}$$

where Z_v^t is a lower bound on the problem defined by A_v and I is the set of assignments existing when the elements a_{ij}^{v-1} were determined. Or, in general:

$$B_x^v = Z_v^t + \sum_{i=x}^{v-1} a^x_{j_i 1(j_i)} + \sum_{i,k \in I} S_{il(i)kl(k)} \qquad (10)$$

where[8] $A_{v,x} = \| a_{ij}^x \|$, and $a^x_{j_i 1(j_i)}$ are the coefficients of the assignments $(j_x, 1(j_x))$, $(j_{x+1}, 1(j_{x+1}))$,....., $(j_{v-1}, 1(j_{v-1}))$ which have been made and Z_v^t is a lower bound in the problem defined by $A_{v,x}$. And between these extreme alternatives of determining a minimum value for every a_{ij}^v according to (7) at level v and of simply using a_{ij}^x from a previous stage there is, for example, the alternative of re-computing only selected a_{ij} perceived to be critical[9], together with others.

---

[8] The bound in (10) follows directly from the fact that the sum of the coefficients a_{ij} in A_v for any feasible linear assignment solution constitutes a valid lower bound on the cost of that assignment in the quadratic problem.

[9] As the potential variability in a cost coefficient a_{ij} diminishes with successive assignments the potential importance of updating its value may also diminish. For example, referring to (7) it is seen that the only variability in the coefficient a_{ij} from level to level derives from the product $\sum_{p \in I} f_{ip} d_{jl(p)}$ which decreases with increasing v. Letting

$$f_i^* = \min_j \{f_{ij}\} \ , \ d_j^* = \min_k \{d_{jk}\} \ , \ f_{ij} = f_i^* + f'_{ij}, \text{ and } d_{ij} = d_j^* + d'_{ij},$$

the sum becomes:

$$\sum_{p=1}^{n} f_{ip} d_{jl(p)} = \sum_{p=1}^{n} (f_i^* + f'_{ip})(d_j^* + d_{jl(p)})$$

$$= \text{Constant} + \sum_p f'_{ip} d'_{jl(p)}$$

The maximum variability is thus $\sum_R (f_{i(k)} - f_{i(n+1-k)}) d_{j(n+1-k)}$ which can perhaps be used to assess the potential importance of updating the coefficient a_{ij}.

Continuing further the discussion of alternate means of bounding, recall that the situation we have been discussing was that in which we had arrived at a node at level (v-1) and, in the manner of Lawler and Gilmore, were making each of possible (n-v+1) assignments (j,i) at level v and evaluating through means of an appropriate matrix A_v a bound B_v for each of the (n-v+1) nodes. Any of the ways for getting the elements for the matrices A_v and any degree of reduction could be employed in each case. However, while perhaps resulting in less stringent bounds, a value for each B_v at level v can be assessed at level (v-1) without first generating each of the matrices A_v, hence reducing the computational effort preparatory to making a next assignment. For if A_{v-1} is an appropriate assignment matrix for the problem at level (v-1) and $A^t_{v-1}=|| a^{t,v-1}_{ij} ||$ is any matrix with nonnegative elements derived from it through one or more stages of reduction, then a lower bound on solutions passing through the node at level v which results from the assignment (i,j) is:

$$B^t_{v-1}(i,j) = a^{t,\ v-1}_{ij} + Z^t_{v-1} + \sum_{i,k\in I} S_{il(i)kl(k)} \qquad (11)$$

where Z^t_{v-1} is the amount of reduction incurred in reducing matrix A_{v-1} to A^t_{v-1}. In practice A^t_{v-1} would most likely be the reduced matrix which results from simply reducing rows and columns, or the matrix A associated with an optimal assignment solution in which $a_{ij} = 0$ for all $x_{ij} = 1$ in the optimal solution. To facilitate discussion, we will assume at least the former so that there exists at least one zero element in each row and column of A^t_{v-1} although this in no way limits the generality of the discussion.

To employ in this framework the search strategy of Gilmore and Lawler which selects an assignment (i,v) which has a lower bound, we simply choose an assignment corresponding to a zero element in the column of A_{v-1} representing location v. We then proceed to formulate a matrix A_v for this one node, generating some or all of the remaining (n-v) matrices at this level at a later point in the search process only if not dominated.

To generalize somewhat beyond the search strategy of Gilmore and Lawler and make search more dependent on the data, we may select from all candidate assignments (i,j) at level (v-1) a next assignment, not limiting choice to location v. Assuming A_{v-1} is a reduced matrix, however, there are at least (n-v-1) zero elements, at least one for each row and column. Therefore, additional criteria are required for choosing among the zero elements.

A very effective criterion is that of Little et al. [20] which employs what is termed an alternate cost. At each point throughout the search process the selection of an assignment (i,j) partitions the set of all potential solutions into two subsets, one of all potential solutions which includes (i,j) and the other of all potential solutions which does not. At level v-1 a lower bound on the cost of potential solutions in the first subset is

$$Z^t_{v-1} + \sum_{p,k \in I} S_{pl(p)kl(k)}$$

since (i,j) is a zero element. On the other hand, since one element must eve

be selected from each row and each column of the assignment matrix, a lower bound for potential solutions in the second subset is

$$E^t_{v-1}(i,j) = Z^t_{v-1} + \sum_{p,k\in l} S_{p1(p)k1(k)} + \min_{x\neq i}\{a^t_{(x),(j)}\} + \min_{s\neq j}\{a^t_{(i),(s)}\} \quad (12)$$

We will refer to the quantity $E^t_v(i,j)$, a lower bound on the objective function for all potential alternatives to the pair (i,j), as simply the alternate cost for the pair (i,j). According to the criterion of Little, et al., the zero element is chosen for which the alternate cost is the greatest. Thus,search proceeds stage by stage selecting elements according to their cost criterion until either a terminal node is reached or one for which it is known that all potential solutions passing through it are dominated. At this point the search process backtracks to the first node for which the alternate cost is less than the total completion time of the best feasible sequence discovered so far, sets $a_{kp} = M$ for the assignment just investigated, and then resumes.

As a computational consideration it is noted that if $E^t_{v-1}(i,j) \geq Z_\Theta$ then the assignment (i,j) must necessarily be included in every non-dominated path passing through the node. If this is true for two or more assignments, then it is unnecessary to explicitly consider nodes for each of these, but rather make all such assignments immediately (jumping levels in the tree) and then proceed to establish matrix A_v for the remaining choices.[10] In the special event that this is true for all zero elements in an optimal assignment solution at a given node, then the only further consideration that need be given the node is to evaluate the quadratic assignment solution defined by this assignment.

---

[10] Note that before proceeding to establish the new matrix it might prove worthwhile to re-evaluate the alternate costs of the remaining zero elements and re-check whether or not $E^t_{v-1}(i,j) \geq Z_\Theta$ for any additional zero element.

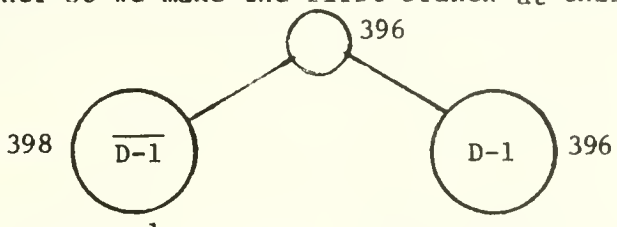
In summary, there are many bounding alternatives that may be employed within the tree search algorithm. Basically, as we have seen, these concern alternative ways for determining the elements in matrix A_v at a given node; alternative degrees of reduction to be applied to it; and choices regarding the dominance tests to be made on the basis of the resulting bounds. Further, as has been discussed, there are a number of alternative search strategies ranging from fixed, data-independent strategies to the level-by-level strategy with pre-specified levels of Gilmore and Lawler, and to the general level-by-level strategy with variable levels. In addition, there is the alternative of stopping to evaluate the quadratic cost of a feasible solution which results whenever a feasible solution to the linear assignment problem is determined at any node: for if $Z_\Theta < Z_{\hat{\Theta}}$ a better feasible solution has been discovered and the lower value of Z may be used to make potentially more stringent the dominance tests in reducing subsequent search.

To illustrate these extensions we again solve the problem of Figure 3 In the algorithm to be used the search strategy is the general level-by-level strategy except that when we reach level (n-2) we shift to a data-independent strategy of exhaustively enumerating all feasible assignments. Beginning at level 0 and at every level thereafter, we establish matrix A_v by determining optimal values of the elements a_{ij} and then reducing fully to an optimal assignment solution.

We then solve a second time the same problem, illustrating the possibility of evaluating the feasible quadratic assignment solution defined by the optimal linear assignment. In this procedure we also

review the alternate costs at every node to identify variables x_{ij} which must necessarily have value $x_{ij}=1$. For all such variables a level in the tree is jumped.

We begin with the matrix developed previously and the lower bound on all solutions of $\frac{792}{2} = 396$. An examination of the matrix shows that cells A-2, A-4, B-3, B-4, C-2, C-3, D-1, and D-3 may be selected at zero incremental cost. The alternate cost of D-1, that is 4, is higher than any other so we make the first branch at this cell with alternate cost $\frac{792+4}{2} = 398$.



We then calculate a^1_{ij} as before. The elements are

$\sum_{k\in I} S_{ijkl(k)} =$

	2	3	4
A	56	50	26
B	168	150	78
C	28	25	13

$\Sigma \bar{a}^1_{ij} =$

	2	3	4
A	118	243	166
B	99	205	139
C	103	220	143

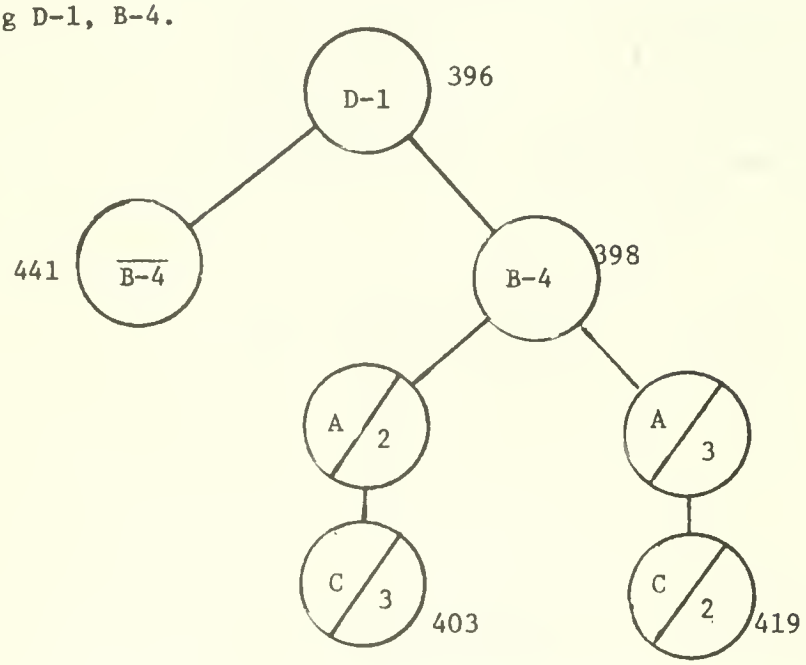
$2 \quad a^1_{ij} =$

	2	3	4
A	230	343	218
B	435	505	295
C	159	270	169

Solving the linear assignment problem associated with this matrix we obtain the following solution with a total reduction of $\frac{795}{2} = 398$.

	2	3	4
A	0	2	0
B	128	87	0
C	0	0	22

The assignment with the largest alternate cost is B-4 with 87. We therefore branch on B-4, then enumerate all possible alternatives following D-1, B-4.

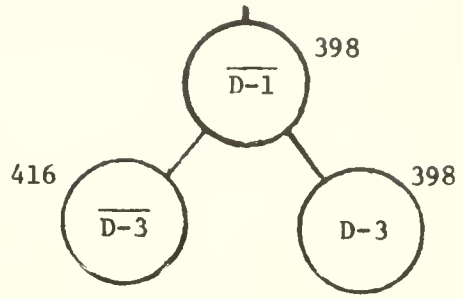


At this point all completions are bounded by the solution D-1,B-4, A-2, and C-3, with the exception of $\overline{D-1}$. We next modify the initial matrix to reflect the condition $\overline{D-1}$, by adding M (a large number) to the D-1 element. The resulting matrix is solved as a linear assignment problem below, with a reduction of $\frac{796}{2} = 398$.

	1	2	3	4
A	4	0	40	0
B	10	15	35	0
C	0	6	41	13
D	M	0	0	13

The highest alternate cost in the matrix is 35 on D-3.

We branch as below



The matrix given D-3 is

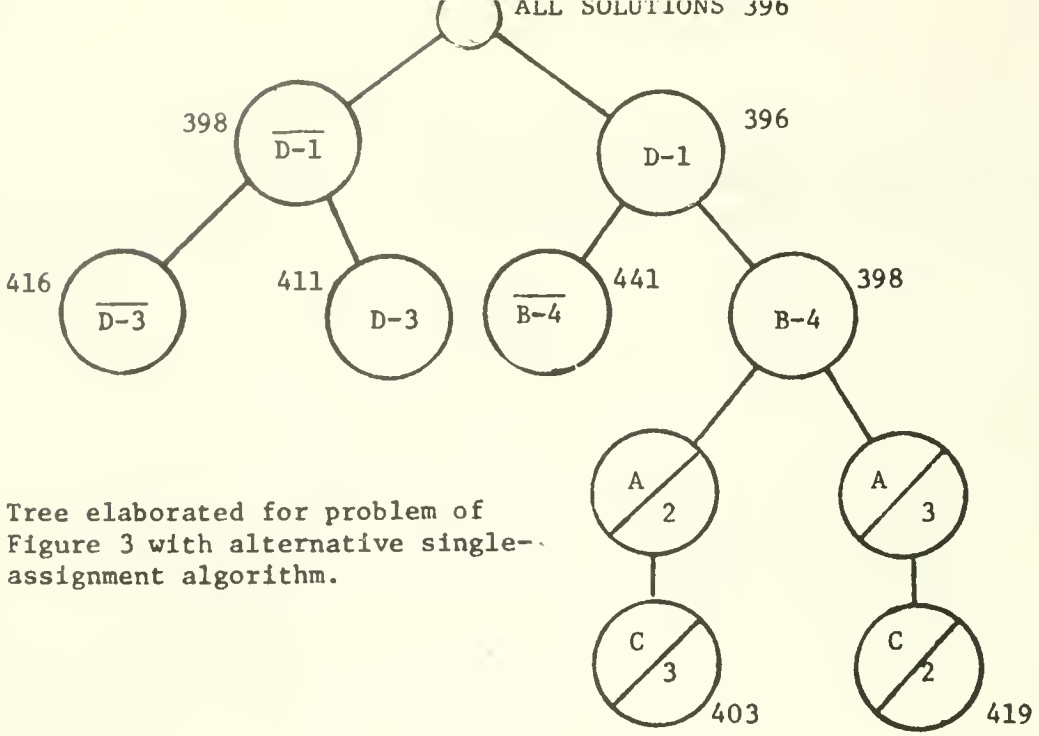
$2\ a_{ij}^{1} =$

	1	2	4
A	359	256	198
B	518	344	365
C	281	198	139

Simple row and column reduction gives a total reduction of $\frac{822}{2} = 411$ and the following matrix:

	1	2	4
A	20	0	1
B	91	0	80
C	0	0	0

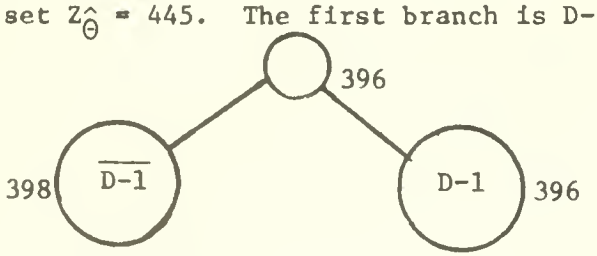
This is adequate to show that all solutions containing D-3 are bounded, and that we have discovered the optimal solution. We have now completely elaborated the enumeration tree as shown in Figure 6.



Tree elaborated for problem of Figure 3 with alternative single-assignment algorithm.

FIGURE 6

As noted earlier if $E_v^t(i,j)$ in (12) is greater than or equal to $Z_{\hat{\theta}}$ then the assignment (i,j) must necessarily be included in every non-dominated path passing through the node. We will now modify the calculations in our sample problem to include this test. We begin again with the initial assignment solution D-1,C-2,B-3,A-4, with a total reduction of $\frac{792}{2} = 396$ and an actual cost of $\frac{890}{2} = 445$. At this point we set $Z_{\hat{\theta}} = 445$. The first branch is D-1 as before and we

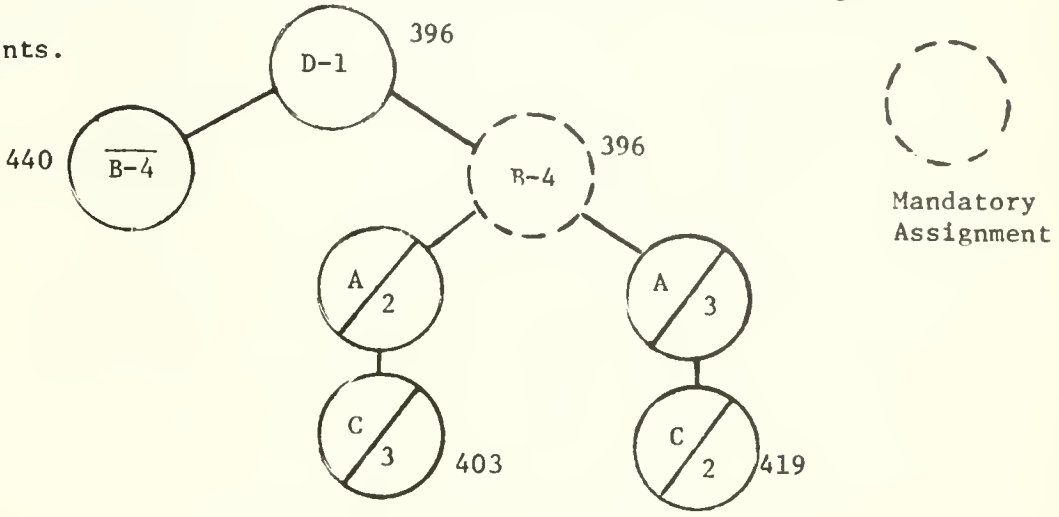


form and reduce the second matrix and solve for an optimal linear assignment.

	2	3	4
A	0	2	0
B	128	87	0
C	0	0	22

Reduction = 796

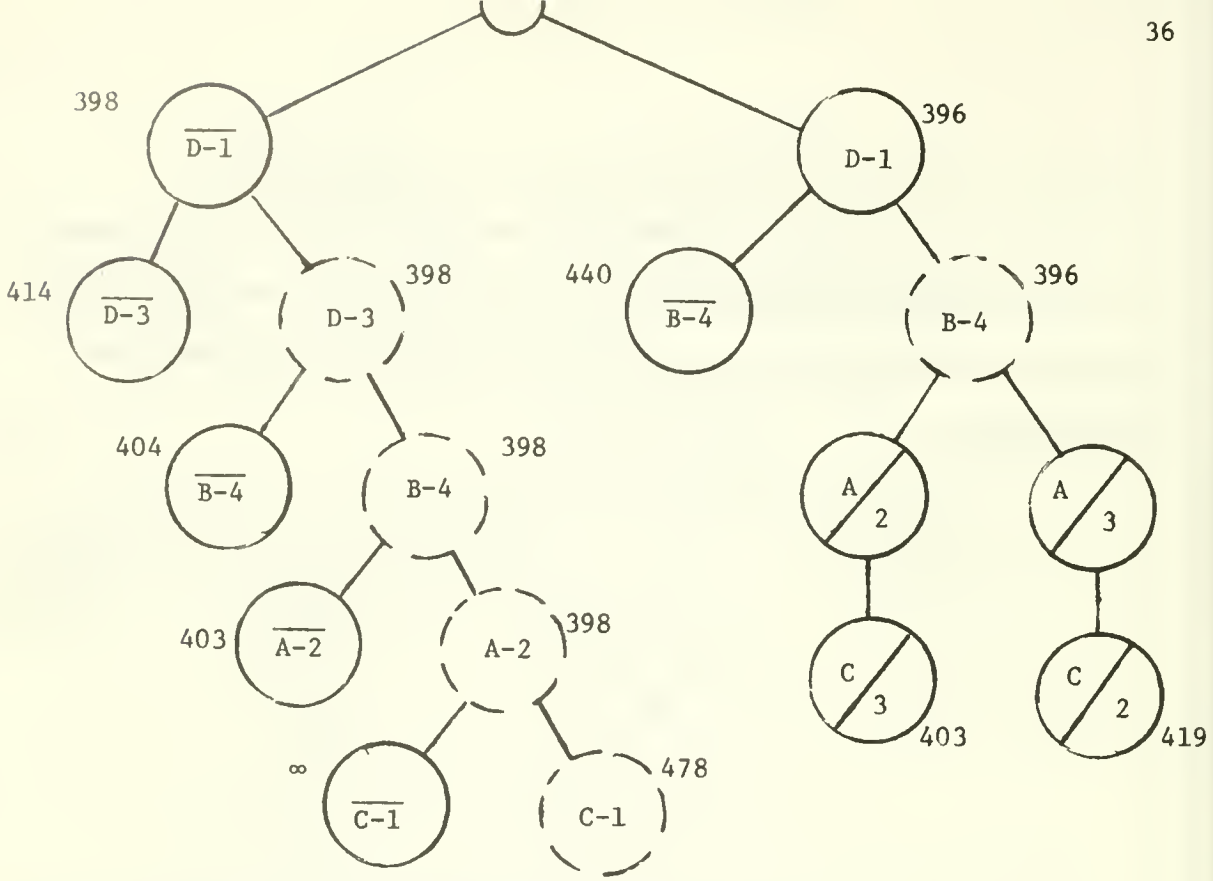
The exact cost of the current assignment solution, D-1, A-2, B-4, C-3 is 403 and this becomes the new value for $Z_{\hat{\theta}}$. An evaluation of the alternate costs shows that B-4, with a cost of 87 must be in an optimal solution if D-1 is. We then enumerate the remaining two assignments.



An evaluation of the $\overline{D\text{-}1}$ branch in the original reduced matrix as shown below

	1	2	3	4
A	14	0	5	0
B	20	15	0	0
C	4	0	0	7
D	M	35	0	48

now indicates that the alternate cost of D-3 is 35 and since $\frac{792+35}{2} > Z_{\hat{\theta}} = 403$ then D-3 must be in any solution which is superior to $Z_{\hat{\theta}}$. If the alternate cost of B-4 is updated we find it is now $(\frac{792+15}{2}) > 403$ and therefore B-4 must be in any superior solution. By the same argument A-2 and C-1 are also assigned. We now can complete the tree based solely on information in the first matrix created and on an evaluation of the complete solution D-3,B-4,A-2,C-1. The resulting enumeration tree is shown in Figure 7.



Tree elaborated for alternative single-assignment algorithm together with testing for mandatory assignments.

Figure 7

We will now show an alternate method for bounding the D-1 path which involves the updating of particular cells in the cost matrix. To begin we solve the assignment problem for the best solution not including D-1, getting the following matrix.

	1	2	3	4
A	4	0	40	0
B	10	15	35	0
C	0	6	41	13
D	M	0	0	13

Reduction $\frac{796}{2} = 398$

Since D-3 has the highest alternate cost we select that assignment and then update only the other elements of the assignment solution, C-1,A-2, B-4, on the assumption that D-3 is selected.

This results in the following matrix.

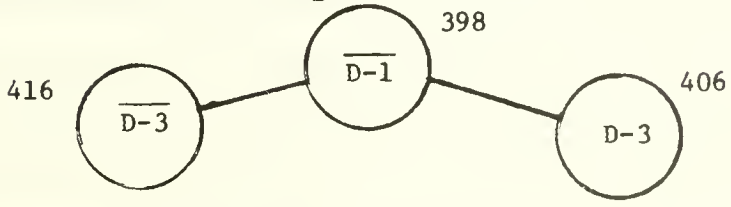
	1	2	3	4
A	297	256	293	152
B	368	254	353	365
C	281	131	245	116
D	M	82	161	73

UPDATED

with assignment solution

	1	2	3	4
A	46	89	47	0
B	0	79	0	94
C	66	0	35	0
D	M	0	0	6

and total reduction of $\frac{812}{2} = 406$. Therefore the solution with D-3 is bounded.



For this particular problem this method resulting in the solution of Figure 7 is the most efficient of the methods discussed in this section. Although the tree is larger than that of Figure 6 it is generated completely from the first matrix and from two completely enumerated solutions.

In concluding discussion of single-assignment algorithms it should be emphasized that while we have been illustrating ideas by means of a simple Koopmans-Beckman problem (5) with all costs $c_{ij} = 0$, the ideas and algorithms are perfectly general and apply equally as well to the more generalized Koopmans-Beckman problem with cost function (6) as well as to the general quadratic assignment problem with cost function (1).

As remarked in the introduction, there are sometimes constraints to be satisfied in addition to those of (2)-(4). To the extent that the conditions of these constraints can be completely stated by setting $s_{ijkq} = M$ for appropriate i,j,k and q, the algorithms as discussed can be utilized without change. For algorithms which, for problems where $s_{ijkq} = f_{ik}d_{jq}$, determine elements a_{ij}^{v} simply by appropriately sequencing the elements f_{ik} and d_{jq} there are now at least three options: First, continue to determine a_{ij}^{v} in the same way, the result being a lower bound on the true minimum value; second, determine the true minimum by solving an assignment problem with $s_{ijkq} = M$ where appropriate; or third, determine the sequencing of f_{ik} and d_{jq} in the present way and make adjustments for inadmissable pairings $f_{ik}d_{jq}$ which result. In each case when $s_{ijkq} = M$ we will set $a_{ij}^{v} = M$ whenever assignment (kq) is made, and $a_{kq}^{v} = M$ whenever (ij) is made. When constraints involving three or more assignments are present, we can proceed in the same way except that the constraints become explicitly represented in the problem through the s_{ijkq} and a_{ij}^{v} only when a sufficient number of assignments have already been made to enable identification of these nonfeasible assignments.

IV Pair-Assignment Algorithms

In contrast to the algorithms which have been discussed in the previous sections, both Land [16] and Gavett and Plyter [8] have developed algorithms in which search proceeds on the basis of a controlled enumeration of the variables $y_{ijkq} = x_{ij} \cdot x_{kq}$ where as before each variable x_{ij} denotes the locating of plant i at location j. These authors too view the underlying problem as a linear assignment problem but one of assigning a pair of plants i and k to locations j and q In both instances [8,16] the algorithms developed apply to the symmetric Koopmans-Beckman problem with $S_{ijkq} = S_{iqkj} = f_{ik}d_{jq}$.

In Figure 8 is shown the relevant assignment matrix for the problem of Figure 3. In general there are n(n-1)/2 pairs of plants and pairs of locations in the problem.

LOCATION PAIR / PLANT PAIR	2-1 1-2	3-1 1-3	4-1 1-4	3-2 2-3	4-2 2-4	4-3 3-4
A-B	168	150	78	90	24	138
A-C	196	175	91	105	28	161
A-D	56	50	26	30	8	46
B-C	140	125	65	75	20	115
B-D	168	150	78	90	24	138
C-D	28	25	13	15	4	23

Data for problem of Gavett and Plyter
represented in terms of pairs of assignment.

Figure 8

However, there are many feasible solutions to this assignment problem which are not feasible solutions to the original quadratic assignment problem. For example it is entirely acceptable in the linear assignment problem for plants A and B to be assigned locations 1 and 2 and plants A and C locations 3 and 4, a solution clearly infeasible for the original quadratic problem. For a feasible solution to the original problem we must therefore, affix to the linear assignment problem the following additional constraints:

$$
\begin{array}{lll}
\text{If} & y_{ijkq} = 1 & \\
\text{Then} & y_{ivpl} = 0 \qquad y_{vipl} = 0 & \\
 & y_{ujpl} = 0 \qquad y_{iupl} = 0 & \\
 & y_{uvkl} = 0 \qquad y_{uvlk} = 0 & \\
 & y_{uvpq} = 0 \qquad y_{uvqp} = 0 & \\
\text{where} & i \neq u \neq j,\ i \neq v \neq j,\ k \neq p \neq q \text{ and } k \neq l \neq q. &
\end{array}
\tag{13}
$$

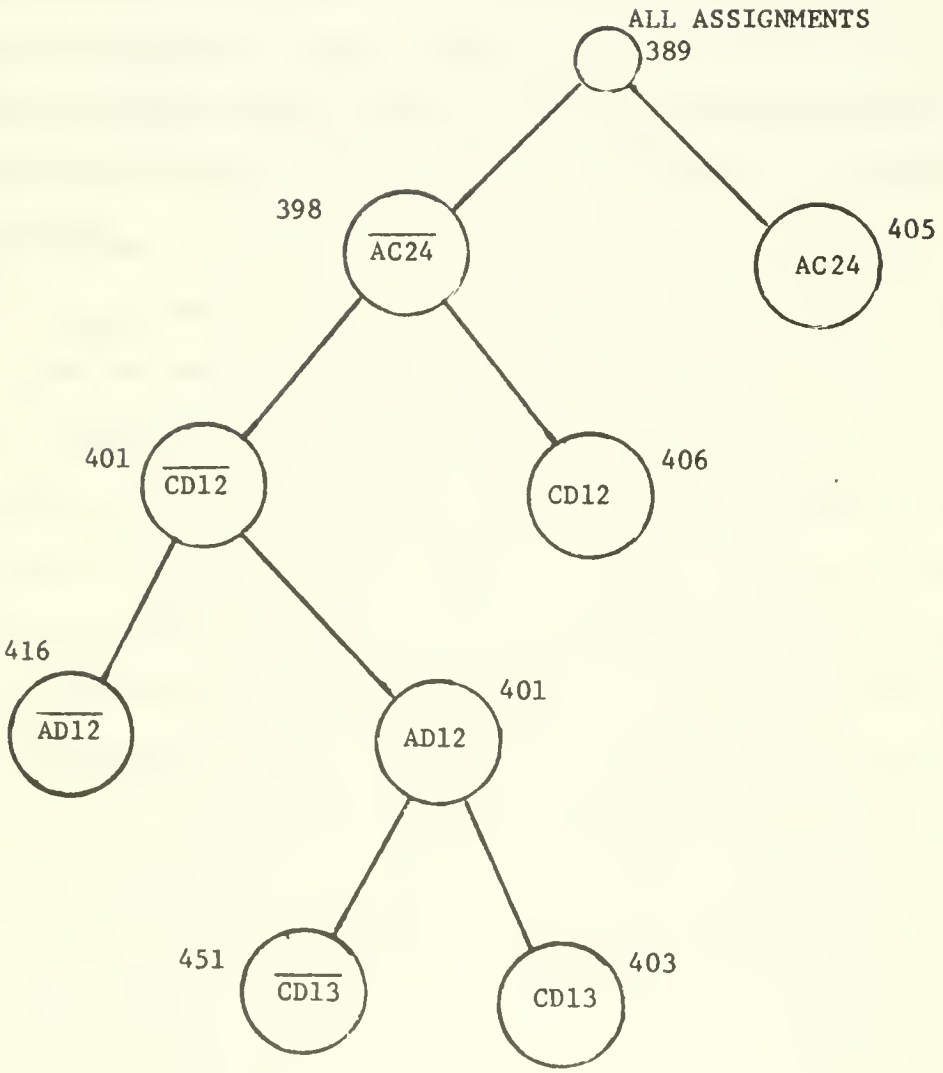
Operationally both the algorithm of Land and that of Gavett and Plyter commence by determining an optimal linear assignment solution for the matrix A_o, and determining a reduced matrix A''_o with nonnegative entries in which $a''_{ijkq} = 0$ for all variables $y_{ijkq} = 1$ in the optimal solution. Thereafter Gavett and Plyter employ only a row and column-reduced matrix A_v at each node, and Land employs only a column-reduced matrix at each node. As in the procedures discussed in the previous sections, both of their algorithms proceed level by level in the tree, committing one new pair (i.e., se ting $y_{ijkq} = 1$) to the solution at each level, and backtracking to the

lowest level in the tree having an unevaluated branch. In selecting the pair to be committed at a given level in the tree Gavett and Plyter use the alternate cost method of Little et al. [20], while Land [16] always selects from the column having the fewest number of feasible elements in the column-reduced matrix A_v a zero element having the largest alternate cost (based only on alternate costs in the same column). After committing a pair to the solution at a given node in the tree (i.e., setting $y_{ijkq}=1$) feasibility condition (13) is invoked by setting the cost $c_{efgh}=M$ for all $y_{efgh}=0$ specified in (13), and the resulting matrix used as matrix A_{v+1} at the next level. In a variation of the search procedures discussed heretofore, however, Gavett and Plyter, after selecting the assignment pair and hence the variable y_{ijkq} at node v, apply (13) for the branch $y_{ijkq}=1$ and reduce the resulting matrix A_{v+1} to get a lower bound on the cost of solutions with $y_{ijkq}=1$; if the resulting bound exceeds the alternate cost of the assignment (ijkq) they will at this point in the search pursue the branch for $y_{ijkq}=0$ rather than that[11] for $y_{ijkq}=1$.

This latter point is readily illustrated, for instance, in Figure 9 which shows the tree elaborated by the Gavett and Plyter algorithm for the example of Figure 3. Since the detailed calculations underlying the development of this tree are contained in [8] we shall omit them here.

---

[11] In comparison with the other search strategies that have been discussed, this strategy may entail the elaboration of a longer path in the decision tree and require longer problem-solving time to determine a first feasible solution.



Tree elaborated for problem
by Gavett and Plyter algorithm.

Figure 9

As in the case of single-assignment algorithms there are a number of alternatives to these pair-assignment algorithms which may result in more efficient algorithms. For example, all of the alternatives discussed earlier concerning the extent to which a linear assignment matrix is reduced at a node in the tree are applicable in the present problem. Similarly the use of alternate costs to identify mandatory assignments and the jumping of levels in the tree on the basis thereof, is equally appropriate in the present case. Of course in the present problem the alternatives for determining the costs a^{v}_{ijkq} are inapplicable since $A_v = a^{v-1}_{ijkq}$ for all v, i.e. the value a^{v}_{ijkq} is the actual cost of assigning plant i to location j and plant k to location q rather than simply a lower bound, and hence need not be updated.

In concluding discussion of this class of algorithms we comment on their extension to the nonsymmetric quadratic assignment problem in which $s_{ijkq} \neq s_{iqkj}$. For this problem we have the associated linear problem:

$$\text{Minimize} \quad \sum_{(ik),(jq)} (s_{ijkq}y_{ijkq} + s_{iqkj}y_{iqkj})$$

$$\text{Subject to:} \quad \sum_{(jq)}^{n(n-1)/2} (y_{ijkq} + y_{iqjk}) = 1 \text{ all } (ik) \qquad (14)$$

$$\sum_{(ik)}^{n(n-1)/2} y_{ijkq} + \sum_{(ik)}^{n(n-1)/2} y_{iqjk} = 1 \text{ all } (jq)$$

$$\text{and} \quad y_{ijkq}, y_{iqjk} = 0, 1 \text{ for all } i,j,k,q. \qquad *$$

upon which are imposed, as before, constraints (13).

As discussed in more detail for a related, multi-facility production requiring problem (or multi-salesman traveling-salesman problem) in [25], when formulated as a linear programming problem (14) has a total of n(n-1) variables and activity vectors, the vectors for each pair y_{ijkq} and y_{iqkj} being identical except for their cost. Being linearly dependent, at most one of these vectors in each pair can appear in an optimal feasible solution to (14), this necessarily being the vector in the pair with the smaller cost. Therefore setting $\bar{s}_{ijkq} = \min(s_{ijkq}, s_{iqkj})$ it follows that an optimal solution to (14) can be obtained by solving the $n(n-1)/2 \times n(n-1)/2$ linear assignment problem with costs $||\bar{s}_{ijkq}||$.

Operationally, then, problem-solving for the nonsymmetric problem can proceed as for the symmetric except that at each node in the process the cost matrix to be used is composed of the presently minimum elements, $||\min(s_{ijkq}, s_{iqkj})||$. To this matrix can be applied any degree of reduction as in the symmetric problem. Upon selecting a pair of assignments to commit to the quadratic problem solution, <u>all</u> costs (both for a variable y_{ijkq} and its interchange y_{iqkj}) are updated as required to reflect the feasibility conditions in (13); for any pair therefore, $\min(s_{ijkq}, s_{iqkj})$ may increase for the next node. Otherwise the only difference between the symmetric and the nonsymmetric problems concerns the alternate cost of an assignment: in the nonsymmetric case a valid alternative to the assignment y_{ijkq} may be the interchanged assignment y_{iqkj} so that in this case the alternate cost is the minimum of that as evaluated for the symmetric problem and the cost of the interchanged assignment.

V. Pair-Exclusion Algorithms

In all of the algorithms discussed up to this point, problem-solving has proceeded on the basis of a stage by stage commitment of assignments to the solution of the problem, each such assignment representing a level in the decision tree. Upon backtracking a particular assignment would then be excluded from the solution and the forward, assignment process resumed. This has been the nature of the process for both the single-assignment and the pair-assignment algorithms. In this section we conclude the paper with an algorithm in which problem-solving proceeds on the basis of a stage-by-stage exclusion of assignments from a solution to the problem.

More specifically, let us consider the quadratic assignment problem as formulated in the previous section. Suppose for this problem an optimal assignment has been determined for the linear assignment portion of the problem. If for this assignment conditions (13) are satisfied for every $y_{ijkq}=1$ in the solution (i.e. all pairs result in each plant being assigned to one location, and no location having more than one plant assigned to it) then this solution represents an optimal , feasible solution to the original quadratic assignment problem and problem-solving is complete. Otherwise there exists one or more conflicting assignments in this solution rendering infeasible the solution to the quadratic assignment problem.

As an example Figure 10a shows a reduced matrix for the illustrative problem in Figure 8 in which the optimal linear assignment is indicated by the cells with alternate costs represented in the upper right hand corner. As is readily verified, this optimal linear assignment is not feasible for the

COST = 78 + 28 + 50 + 115 + 90 + 28 = 389

	1-2 2-1	1-3 3-1	1-4 4-1	2-3 3-2	2-4 4-2	3-4 4-3
AB	31	16	0^{0}	0	0	8
AC	55	37	9	11	0^{9}	27
AD	3	0^{0}	32	24	68	0
BC	18	6	2	0	11	0^{0}
BD	31	16	0	0^{0}	0	8
CD	0^{3}	0	44	34	89	2

(a)

	1-2 2-1	1-3 3-1	1-4 4-1	2-3 3-2	2-4 4-2	3-4 4-3
AB	31	16	M	0^{0}	0	8
AC	55	37	9	11	0^{9}	27
D	3	0^{0}	32	24	68	0
BC	18	6	2	0	11	0^{0}
BD	31	16	0^{2}	0	0	8
CD	0^{3}	0	44	34	89	2

(b)

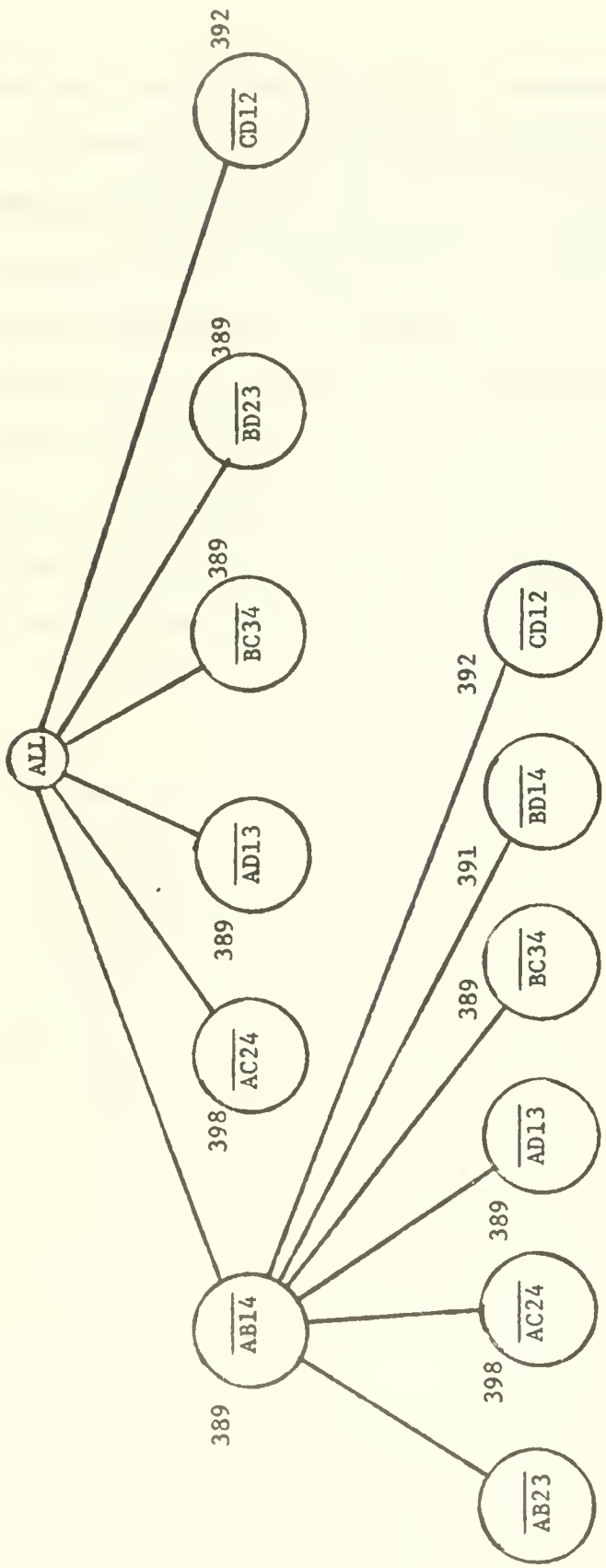
Optimal linear assignment matrices to problem with data in Figure 8: (a) all assignments admissable, (b) (AB,14) inadmissable.

Figure 10

quadratic problem, e.g. assignment (AB,14) is inconsistent with (BD,23), (BC,34) with (AD,13) and (CD,12). In an optimal feasible quadratic assignment, then, it must be true that at least one of the assignments in this optimal linear assignment will not be present. We can therefore subdivide the total set of feasible quadratic assignments into those that do not include the assignment (AB,14), those that do not include (AC,24), and so on, for each of the present assignments. The result, in terms of a tree, is as shown as the first level of nodes in Figure 11. If for each subset we now determine the best feasible quadratic assignment among solutions in that subset, the best among these is an optimal solution to the original problem.

Beside each of the nodes on level 1 in Figure 11 is shown a lower bound on the cost of solutions in the subset equal to the cost of the optimal solution in Figure 10a.(the amount of the reduction) plus the alternate cost of the particular assignment which, as indicated by the node, is to be excluded. Suppose we now choose for elaboration one of these nodes for which this bound is minimum, say $\overline{AB14}$. Making inadmissable this assignment in the cost matrix in Figure10a(i.e. giving a large cost of M) and solving for an optimal assignment to the resulting problem, there results the matrix in Figure 10b. Checking the assignment which results, this solution is not feasible for the quadratic problem either; the result is the tree with the new level of nodes as shown in Figure 11.

In a similar manner, we can now proceed to select any of these nodes with lowest bound, solve the assignment problem and check it for feasibility,



Partial tree elaborated by pair-exclusion algorithm.

Figure 11

continuing until a node is reached for which the optimal linear assignment is a feasible quadratic assignment. At this point we would then backtrack and resume with a node whose lower bound was less than the value of the quadratic assignment solution, continuing in this manner until the complete tree has been considered.

In general it is difficult to anticipate the performance of this type of algorithm relative to the commitment types as discussed in the earlier sections. For the related, basic traveling salesman problem this general approach has proved significantly more efficient than stage-by-stage commitment algorithms [4, 28, 29]. Undoubtedly this is due at least in part to the fact that, in the words of Shapiro [28], the optimal traveling salesman solution is frequently quite "close" to the optimal linear assignment solution in the respect that a large majority of assignments in the former are present in the latter, so that relatively small decision trees need be explicitly elaborated. In addition it is due in part to the existence of an efficient dual algorithm for solving the linear assignment problems at each node [29]. In the present problem this latter element will be equally important but, on the other hand, it is not apparent that the optimal quadratic assignment will be "close" to the optimal linear assignment, i.e., that in optimal linear assignments the conditions in (13) will commonly be automatically satisfied.

To pursue discussion in greater detail, there are a number of choices which must be made in specifying a particular "pair-exclusion" algorithm. What search strategy is to be employed in selecting a node in the tree to elaborate next? Given the conflicts in an optimal linear assignment solution at a given node how should solutions be subdivided into subsets for further evaluation? Which branch emanating from a node (i.e. which subset) should be considered first?

For specificity let us assume for discussion purposes that the search strategy used is the same as that which has been assumed in all of the other algorithms discussed. That is, we proceed downward in the tree one level at each successive stage, choosing at each stage for elaboration the node having the smallest lower bound on the cost of solutions represented by the node. Upon reaching the bottom of the tree or reaching a node for which there exists no feasible, nondominated solutions the process backtracks to the lowest level in the tree for which there is an unevaluated node and resumes. There remains, then, the determination of subsets at nodes and the assessment of lower bounds for the solutions contained in the resulting subsets.

While subdividing the subsets into the n subsets on the basis of the n assignments in the linear solution is perhaps the easiest subdivision to specify at a node (as was done in the illustration) it is by no means the only possibility nor probably the most desirable subdivision. In general, any subdivision into subsets at a node is permissable which excludes at least one conflict present among the present assignments (and hence renders inadmissable the present solution) and for which the union of the subsets contains at least one[12] feasible quadratic assignment which is optimal for the set being subdivided.

Figure 12 shows all conflicts in the optimal assignment solution of Figure 10a arising between pairs of the assignments. Any of these conflicts could be used as the basis for subdividing. For example, Figure 13a illustrates subdivision of the basis of the conflicts between the assignments (AC24) and (AD13). Of course, in the optimal linear assignment at the resulting nodes there may persist conflicts which were present at the parent node; thus at node $(\overline{AD13})$it

[12]The subsets need not be mutually exclusive.

might be necessary to resolve at a next level the conflict between (AC24) and (BD23) should it be present in the new solution at that node, as illustrated in Figure 13b. On the other hand, the conflict may not appear in subsequent linear assignment solutions and hence not have to be considered explicitly.

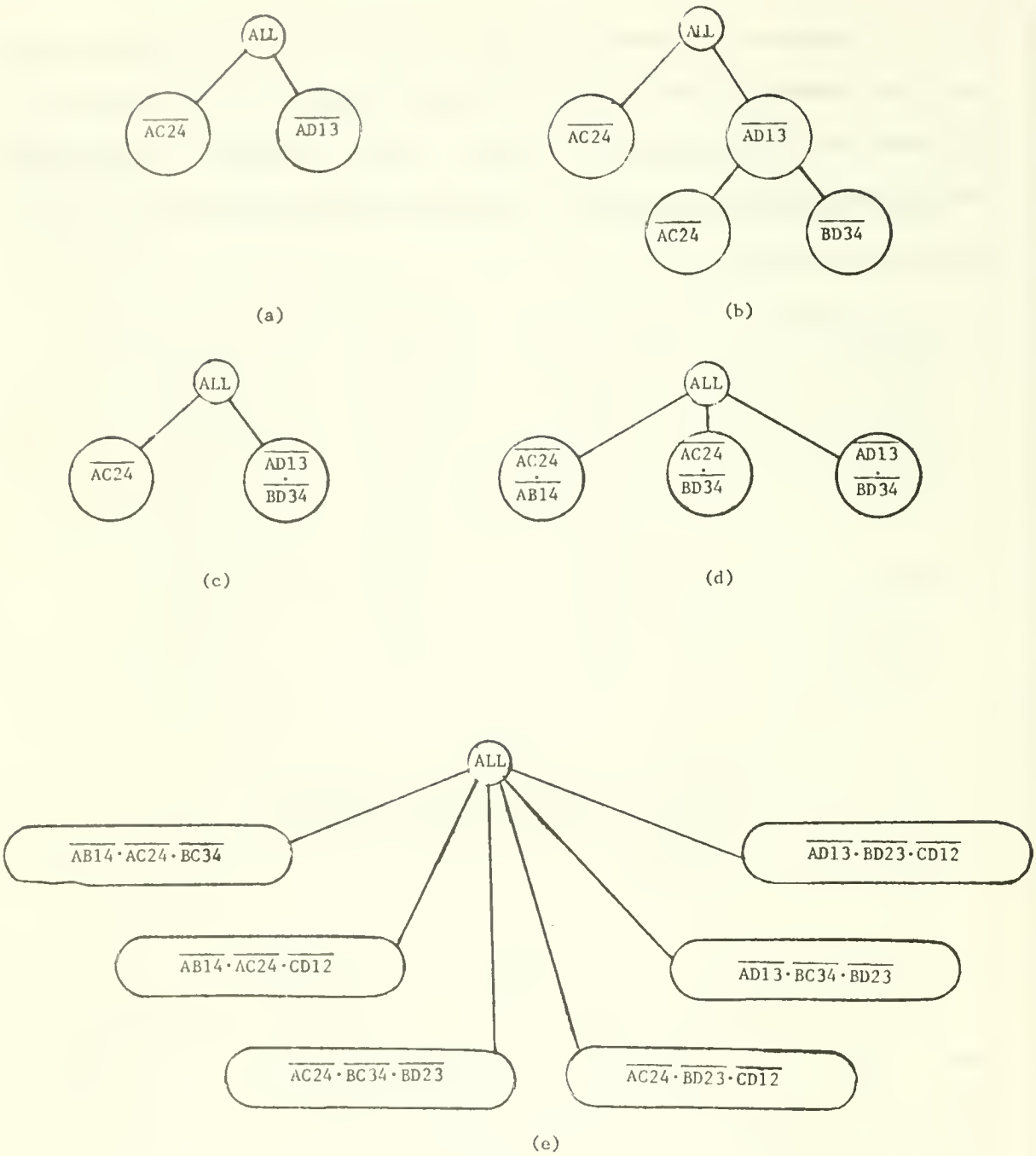
Assignment	Alt. Cost	Conflict	Alt. Cost
AB14	389	BD23	389
AB14	389	CD12	392
AC24	398	AD13	389
AC24	398	BD23	389
AD13	389	BC34	389
BC34	389	CD12	392

Conflicts between pairs of assignments in optimal linear assignment solutions to illustrate problems in Figure 8.

Figure 12

More stringent subdivisions may result, perhaps, by using more than a single conflict. Letting $\overline{AC24}$ denote the event "not assignment AC24", AC24 the event "assignment AC24", "$\oplus$" logical "or" (disjunction) and "$\cdot$" logical "and" (conjunction), we can represent the resolution of the conflict between AC24 and AD13, for example, as simply:

$$\overline{AC24} \oplus \overline{AD13} \qquad (15)$$



Illustrative, alternate ways to partition sets of quadratic assignment solutions into subsets of pair-exclusion algorithms.

FIGURE 13

meaning simply that a necessary condition for feasibility is the event "not assignment AC24" or "not assignment AD13," or both. Suppose we now consider the conflict between, say, AC24 and BD23. For resolution of this conflict we must have $\overline{AC24} \oplus \overline{BD23}$, so that in conjunction with (15) we must have for resolution of both:

$$(\overline{AC24} \oplus \overline{AD13}) \cdot (\overline{AC24} \oplus \overline{BD23}) = \overline{AC24} \oplus (\overline{AD13} \cdot \overline{BD23}) \qquad (16)$$

as represented in Figure 9c. If desired we could consider in conjunction with these two,say, the pair of conclicts AD13 and BC34, with the result:

$$(\overline{AC24} \oplus \overline{AD13}) \cdot (\overline{AC24} \oplus \overline{BD23}) \cdot (\overline{AD13} \oplus \overline{BC34}) = \qquad (17)$$

$$\overline{AC24} \cdot \overline{AD13} \oplus \overline{AC24} \cdot \overline{BC34} \oplus \overline{AD13} \cdot \overline{BD23}$$

as represented in Figure13d,or perhaps these in conjunction with the pair BC34 and CD12 as represented in Figure13e. Similarly, any combination of the constraints can be considered.[13]

For the resulting nodes we proceed just as before to determine an optimal linear assignment, where now every assignment appearing in the expression defining a node is made inadmissable. For a lower bound for a node probably the easiest is to simply use the largest alternate cost of the assignments to be excluded at the node. A more stringent bound of course would result by actually making the elements inadmissable and reducing the resulting assignment matrix or by obtaining an optimal assignment solution.

---

[13]Note that conflicts involving assignments not actually committed in the optimal linear assignment can be used in conjunction with these as well. Thus, for example, since the cost of assigning AB to 23 in Figure 6a is zero this could well appear as an assignment in the solution at a subsequent level. Since this assignment, however, would conflict with, say, assignment CD12, one could if desired explicitly consider that conflict at the present node in forming the subdivisions.

These illustrations serve to indicate the nature of the choices to be made at a node in the tree. Referring to Figure13 it is clear that (c) is preferable to (b) since the resulting subdivision is obtained at a single level of the tree, and with no increase in the number of nodes at that level. Similarly it can be argued that the subdivision in Figure13e. is preferable to that on the first level in Figure11 since the total number of subdivisions or nodes to be considered is the same while the size of each of the subsets in the former is smaller. Unclear, however, is the choice among, say, those in Figures 13(c), (d), and (e), a choice involving a larger number of subsets but each of smaller size. On the one hand it is necessary to determine a bound and/or optimal assignment for each node, but on the other, the subset being smaller the greater is the possibility the node will be bounded by an existing feasible solution and hence not require any further consideration. Similarly with regard to the evaluation of lower bounds at a node: reducing the assignment matrix and/or determining an optimal assignment yields a more stringent bound and enhances the liklihood that the node will be bounded, but to do either requires establishing and manipulating the appropriate assignment matrix for that node in contradistinction to the use of the alternate cost information which requires no explicit consideration of that node's matrix. These choices remain subjects for empirical study.

In concluding discussion it is noted that in practice it may be efficient within this exclusion type of algorithm to be looking at each node for mandatory assignments as well as for assignments to be excluded. As before this could be done simply by checking the alternate costs of the assignments which occur in the optimal linear assignment. For each

mandatory assignment discovered the appropriate related assignments would at that time be made inadmissable, thereby making subsequent bounding and search potentially more effective.

As an illustration we again consider the problem in Figure 8, and solve it with the following algorithm. For a search strategy we use the same level-by-level strategy used for illustration throughout, choosing at each subdivision an unexplored node having the smallest lower bound. In each optimal linear assignment the alternate costs of the assignments are all checked to see if the assignment can be shown to be mandatory. Whenever a node is encountered for which the optimal linear assignment results in a nonfeasible quadratic assignment, a subdivision is formed in the following way. All pairs of assignments in the optimal solutions are investigated for conflict and those so found are noted together with their alternate costs (as was done in Figure12). From this list is selected the pair for which the smaller of the two alternate costs is largest among the minimum of all pairs: ultimately every feasible quadratic assignment which resolves all of these conflicts must have a cost at least as large as this value. Should there be more than one pair with this same minimum, a pair is selected for which the other alternate cost is maximum. (In Figure12 we thus select either the pair AC24-AD13 or AC24-BD23.) For the selected pair we then search for other pairs which have an assignment identical to the assignment in the selected pair having the larger alternate cost, and use these pairs in conjunction with the selected pair. (In Figure12 we would thus form from the pairs AC24-AD13 and AC24-BD23 the subdivision: $(\overline{AC24} \oplus \overline{AD13} \cdot \overline{BD23})$.) The resulting subdivision will thus insure that at least the necessary minimal increase in cost that eventually must be incurred will in fact be incurred

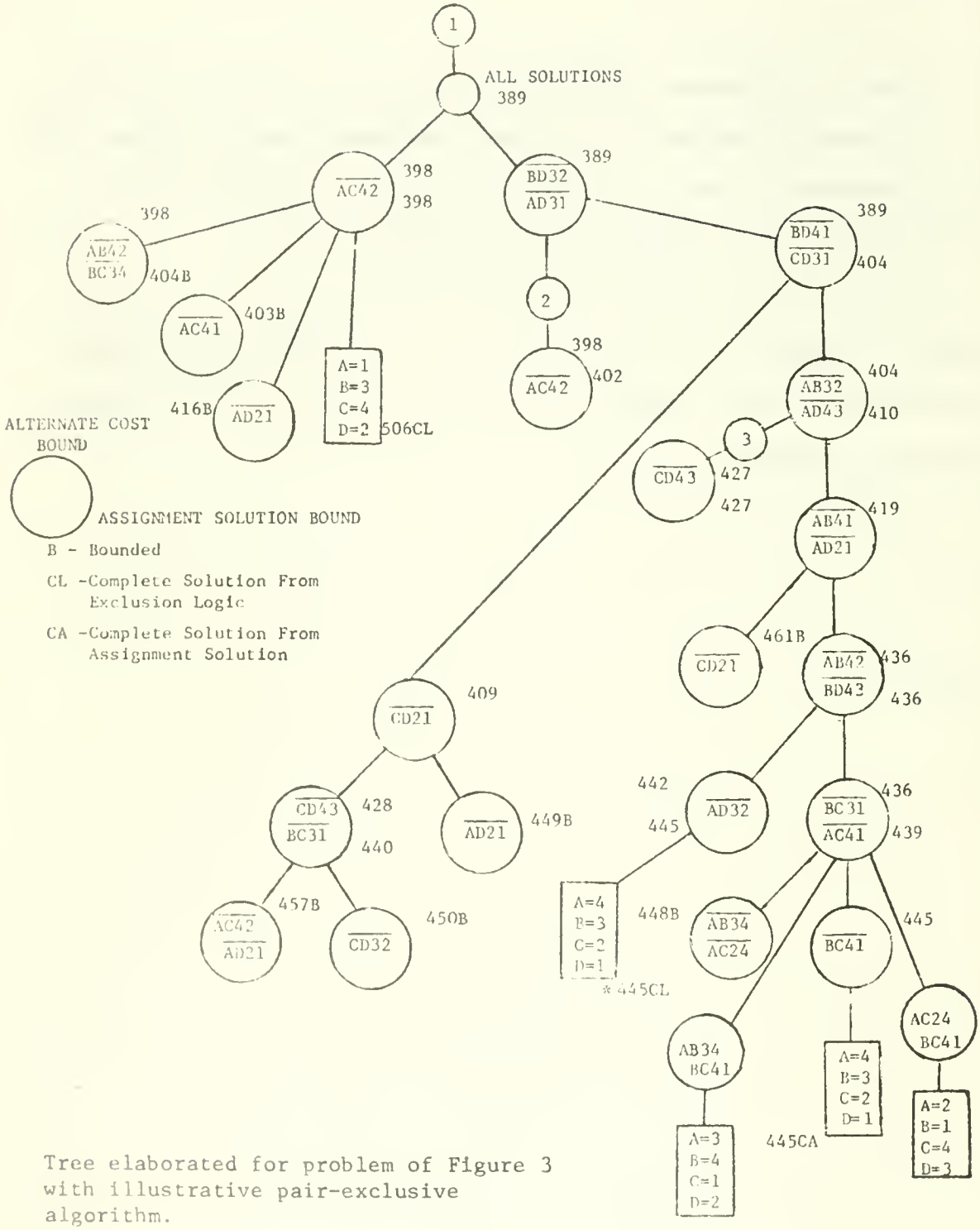
now, and possibly a greater increase -- but without proliferating the number of individual subsets to be considered at this level in the tree. Finally, other assignments are sought having exactly the same conflicting assignments as this assignment in the original selection pair with the higher alternate cost (in the example, all assignments having the conflicts with the same assignments AD13 and BD23 as does AC24) and these conjugated with the present set of conflicts (there are no such assignments in the example). The result is a subdivision consisting of two subsets.

In the event there is conflict in an optimal linear assignment solution but no conflict among simple pairs of assignments we simply choose the first subset of the assignments discovered to be in conflict, remove the assignments in the subset not contributing to the conflict, and subdivide on the basis of the remaining assignments -- each assignment defining one subdivision.

Occasionally in the development of the enumeration tree it can be shown at a node that a particular pair of assignments is mandatory in the same sense as in Section III. For example, in Figure 14a we have a solution 445 CL labelled with a * which was obtained as follows: The solution to the linear assignment problem obtained after adding $\overline{AD32}$ had a reduction of 445 (equal to the current best feasible solution). The alternate cost of assignment CD12 was 19 and therefore CD12 would have to be in any optimal solution to the probelm. Similarly the updated alternate costs of AB34 and BD13 force them into a solution. These assignments taken jointly give A=4, B=3, C=2 and D=1 for an actual cost of 445. It is interesting to

note that the assignments which are obtained in this manner may result in a nonfeasible solution to the linear assignment problem at that point in the enumeration tree. For example, we might have arrived at the above solution even if BC23 were specifically excluded.

For the Gavett and Plyter problem in Figures 3 and 8, the tree which is elaborated is shown in Figure 14. At least for this problem the optimal quadratic assignment solution is not "close" to the optimal quadratic assignment solution.



Tree elaborated for problem of Figure 3 with illustrative pair-exclusive algorithm.

FIGURE 14a

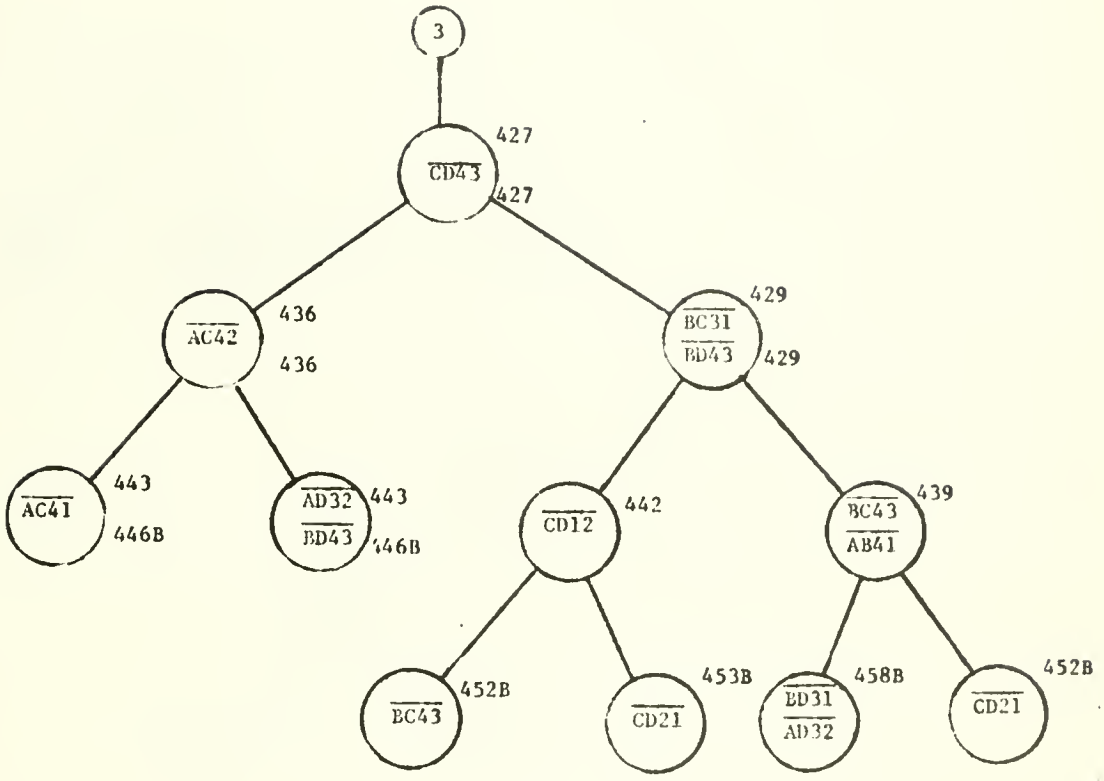

3
427
$\overline{CD43}$
427
436
$\overline{AC42}$
436
429
$\overline{BC31}$
$\overline{BD43}$
429
443
$\overline{AC41}$
446B
443
$\overline{AD32}$
$\overline{BD43}$
446B
442
$\overline{CD12}$
439
$\overline{BC43}$
$\overline{AB41}$
452B
$\overline{BC43}$
453B
$\overline{CD21}$
458B
$\overline{ED31}$
$\overline{AD32}$
452B
$\overline{CD21}$

FIGURE 14b

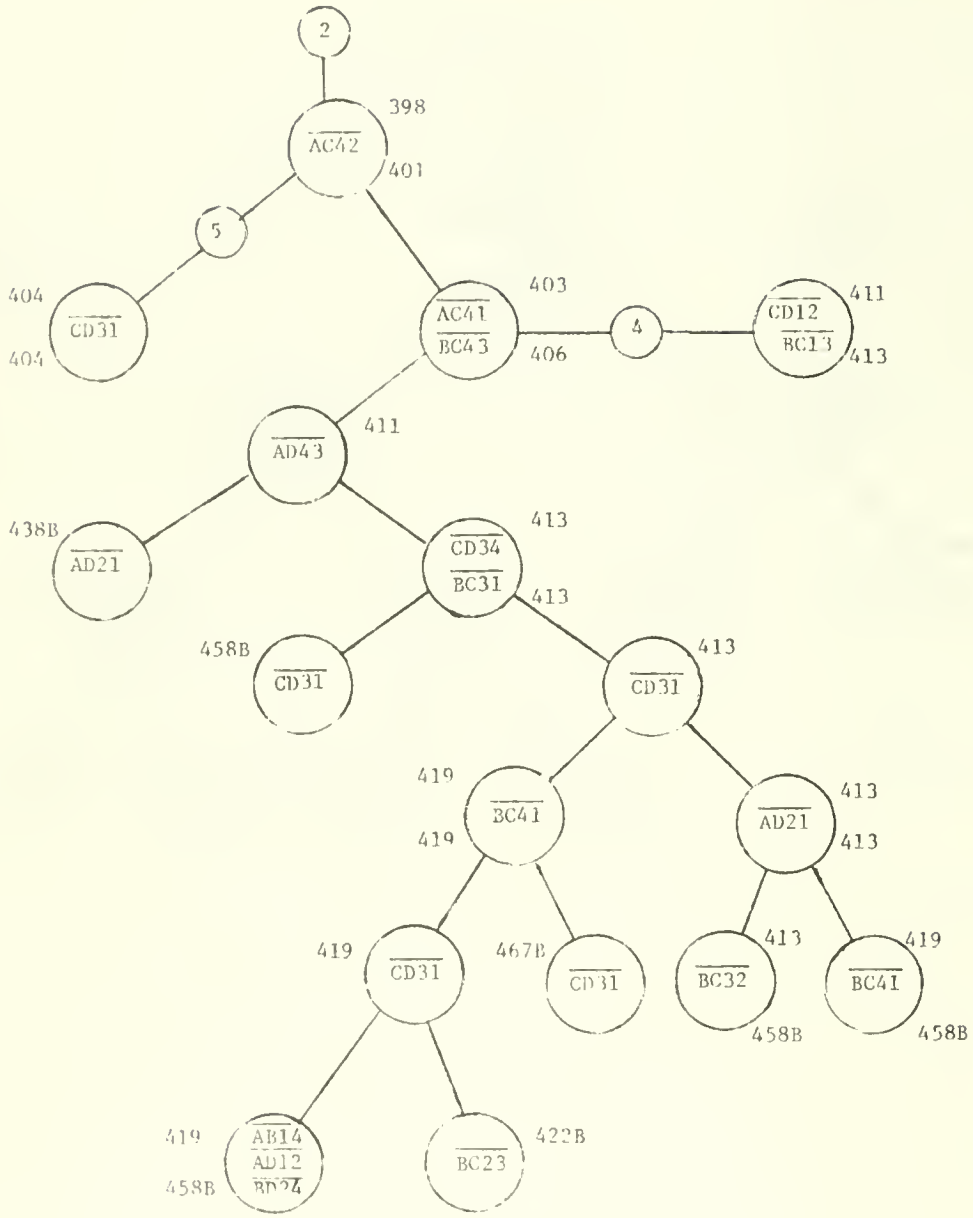

2
398
$\overline{AC42}$
401
5
404
$\overline{CD31}$
404
403
$\overline{AC41}$
$\overline{BC43}$
406
4
411
$\overline{CD12}$
$\overline{BC13}$
413
411
$\overline{AD43}$
438B
$\overline{AD21}$
413
$\overline{CD34}$
$\overline{BC31}$
413
458B
$\overline{CD31}$
413
$\overline{CD31}$
419
$\overline{BC41}$
419
413
$\overline{AD21}$
413
419
$\overline{CD31}$
467B
$\overline{CD31}$
413
$\overline{BC32}$
458B
419
$\overline{BC41}$
458B
419
$\overline{AB14}$
$\overline{AD12}$
$\overline{BD24}$
458B
422B
$\overline{BC23}$

FIGURE 14c

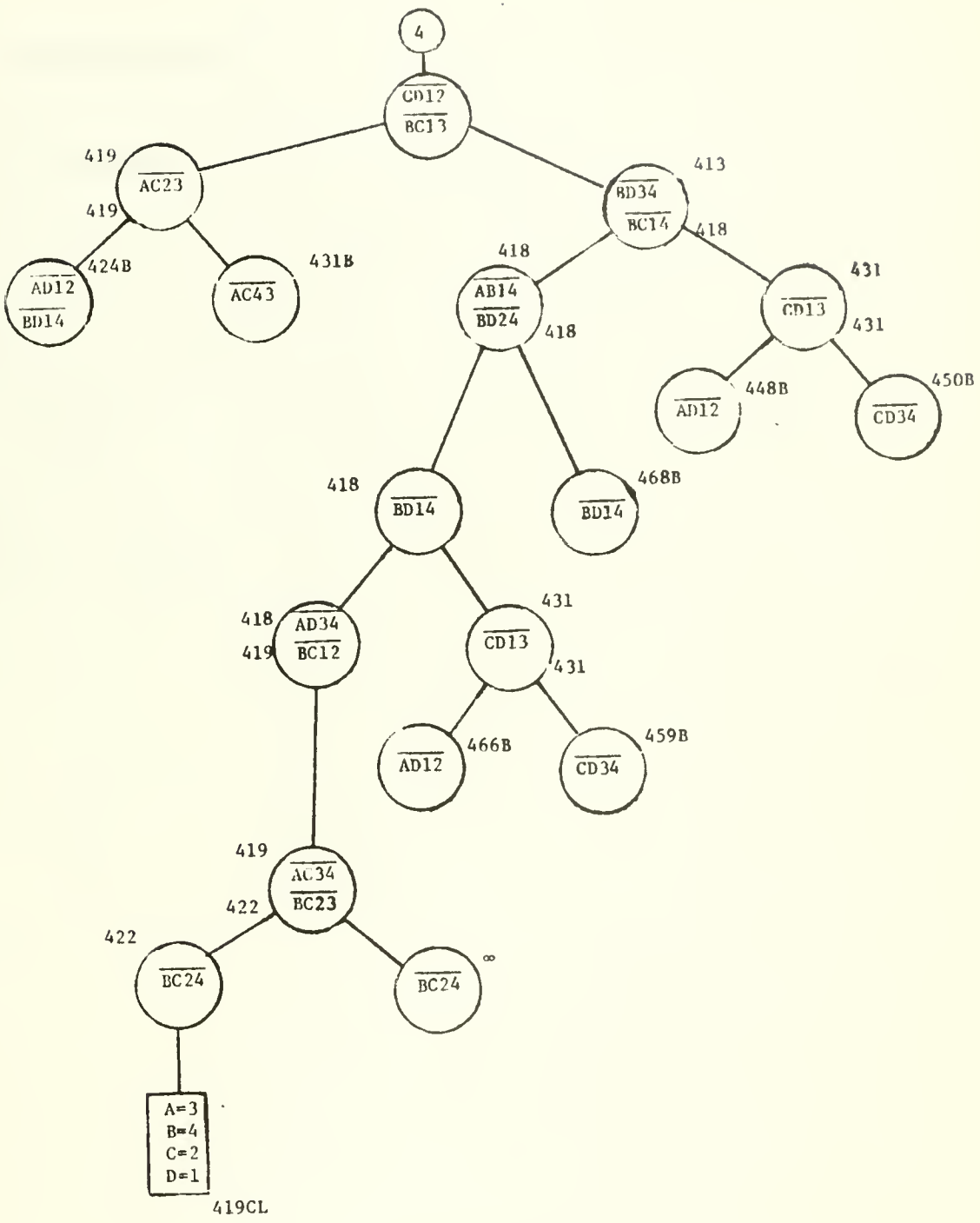

4
CD12
BC13
419
AC23
419
AD12
BD14
424B
AC43
431B
413
BD34
BC14
418
418
AB14
BD24
418
431
CD13
431
AD12
448B
CD34
450B
418
BD14
BD14
468B
418
AD34
BC12
419
431
CD13
431
AD12
466B
CD34
459B
419
AC34
BC23
422
422
BC24
BC24
∞
A=3
B=4
C=2
D=1
419CL

FIGURE 14d

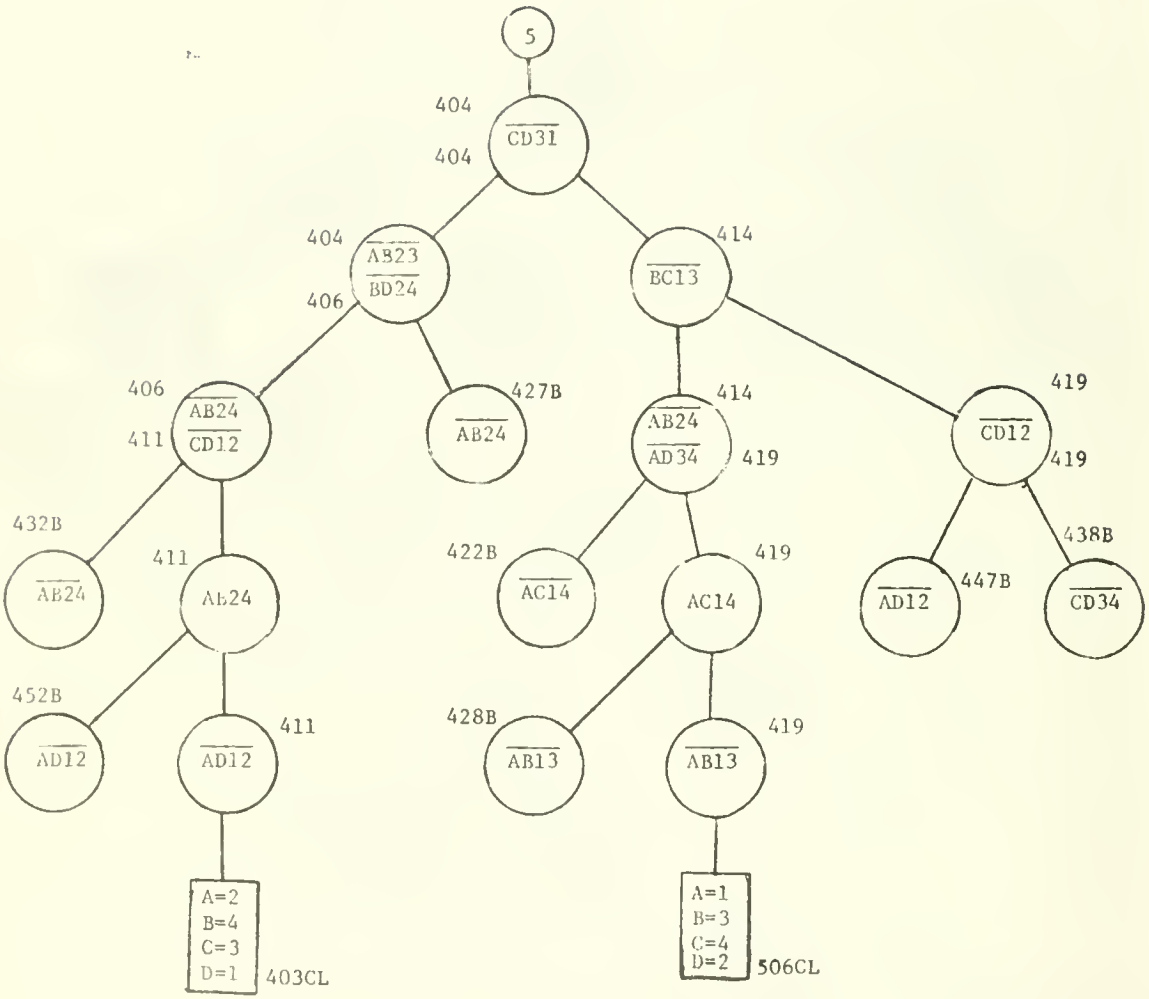

5
404
$\overline{CD31}$
404
404
$\overline{AB23}$
$\overline{BD24}$
406
414
$\overline{BC13}$
406
$\overline{AB24}$
$\overline{CD12}$
411
427B
$\overline{AB24}$
414
$\overline{AB24}$
$\overline{AD34}$
419
419
$\overline{CD12}$
419
432B
$\overline{AB24}$
411
AB24
422B
$\overline{AC14}$
AC14
419
438B
$\overline{CD34}$
447B
$\overline{AD12}$
452B
$\overline{AD12}$
$\overline{AD12}$
411
428B
$\overline{AB13}$
$\overline{AB13}$
419
A=2
B=4
C=3
D=1
403CL
A=1
B=3
C=4
D=2
506CL

FIGURE 14e

VI. Concluding Remarks

In this paper three classes of algorithms have been discussed for solving quadratic assignment problems. Regardless of its class each algorithm which has been considered is reliable in the respect that if carried to completion it guarantees the discovery of an optimal solution. Furthermore in finding an optimal solution it proceeds first to a feasible solution and then to better and better feasible solutions so that, if desired, problem-solving can be terminated prematurely with a usable, if not optimal, solution. In addition, these procedures can all efficiently exploit information available beforehand regarding the value of a known feasible solution and hence, for example, can be readily used in conjunction with heuristic procedure which gives good sub-optimal solutions. Moreover, if desired, all of the algorithms discussed can be used with slight modification to determine all optimal solutions, or a specified number of the most preferred solutions, and so on.

Common to all three classes of algorithms is the structuring of the quadratic assignment problem in terms of a related linear assignment problem. In each this latter problem is then used in directing the tree-search process and in bounding and dominance considerations designed to reduce search. However, between the linear assignment problems used for the single-assignment algorithms and the pair-assignment algorithm there are major differences.

In the single-assignment case the linear assignment problem is only of dimensions n x n and has the property that a feasible solution always represents a feasible solution to the quadratic assignment problem. Its shortcoming, however, lies in the fact that the cost structure embodied in the linear assignment problem is not in general an exact representation of the true cost structure of the quadratic problem but only an approximation. The effect is to diminish the stringency of the bounds and dominance tests, and to necessitate the periodic expenditure of problem-solving time in updating the representation of the cost structure.

In the pair-assignment or pair-eclusion cases, on the other hand, the associated linear assignment problem is significantly larger, being of dimensions n(n-1)2 x n(n-1)/2, but in this larger problem it is possible to represent exactly the cost structure of the quadratic problem. However, the shortcoming of this representation lies in the fact that a feasible solution to the linear problem need not bonstitute a feasible solution to the quadratic problem.

Even from our experience with the one sample problem in this paper it is clear that the different algorithms can give rise to the elaboration of quite different partial trees of solutions with quite differing numbers of nodes (see Figures 4,5,6,7,9 and 14). However, in light of the fact that the time required to elaborate and evaluate a single node in a tree can differ markedly among the algorithms, it is difficult to assess the relative efficiency of the different algorithms even for this one, single

problem. In practice, moreover, the relative efficiency may well turn out to be highly dependent on the particular form of the quadratic assignment being solved. For example, were the coefficients c_{ij} in a problem with objective function (5) to predominate, the approximate cost structure in the single-assignment methods might in fact be quite "close" to the true cost structure and hence that class of algorithms be quite efficient. On the other hand, were an architect to pose a problem with a large number of pairwise[13] constraints on the permissable assignments, the ability to reflect directly their implications in the cost representation of the pair-assignment or pair-exclusion classes might render these approaches more efficient. Hopefully, it will be possible to glean information pertaining to these questions from the computational results to be reported in the subsequent paper.

---

[13] Were he to present constraints involving, say, triplets of assignments, one might then wish to formulate the problem in terms of an $(n(n-1)(n-2)/6) \times (n(n-1)(n-2)/6)$ linear assignment problem, and proceed with a triplet-assignment or triplet-exclusion algorithm akin to algorithms discussed in Sections IV and V.

REFERENCES

1. Armour, G. C. and E. S. Buffa, "A Heuristic Algorithm and Simulative Approach to Relative Location of Facilities," Management Science, Vol. 9, No. 2 (January 1963) p. 294-309.

2. Breuer, M. A., "The Formulation of Some Allocation and Connection Problems as Integer Programs," Naval Research Logistics Quarterly, Vol. 13, No. 1 (March 1966) p. 83-95.

3. Conway, R. W. and W. L. Maxwell, "A Note on the Assignment of Facility Location," Journal of Industrial Engineering, Vol. 12, No. 1 (January-February 1961) p. 34-36.

4. Eastman, W. L., "Linear Programming with Pattern Constraints," Ph.D. Thesis, Computation Laboratory Report No. Bl. 20, Harvard University, 1958.

5. Flood, M. M., "The Traveling-Salesman Problem," Operations Research, Vol. 4, No. 1 (February 1956) p. 61-75.

6. Ford, L. R., Jr. and D. R. Fulkerson, Flows in Networks, Princeton University Press, New Jersey, 1962.

7. Gaschutz, G. K. and J. H. Ahrens, "Suboptimal Algorithms for the Quadratic Assignment Problem," Naval Research Logistics Quarterly, Vol. 15, No. 1 (March 1968) p. 49-62.

8. Gavett, J. W. and N. V. Plyter, "The Optimal Assignment of Facilities to Locations by Branch and Bound," Operations Research, Vol. 14, No. 2 (March-April 1966) p. 210-232.

9. Geoffrion, A., "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review, Vol. 9, No. 2 (1967) p. 178-190.

10. Gilmore, P. C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," Journal Soc. for Industrial and Applied Mathematics, Vol. 10, No. 2 (June 1962) p. 305-313.

11. Golomb, S. W. and L. D. Baumert, "Backtrack Programming," J. Assoc. for Computing Machinery, Vol. 12, No. 4 (October 1965), p. 516-524.

12. Graves, G. W. and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," Working Paper No. 110, Western Management Service Institute, UCLA, November 1966.

13. Hillier, F. S., "Quantitative Tools for Plant Layout Analysis," Journal of Industrial Engineering, Vol. 14, No. 1 (January-February 1963) p. 33-40.

14. Hillier, F. S. and M. M. Connors, "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities," Management Science, Vol. 13, No. 1 (September 1966) p. 42-57.

15. Koopmans, T. C. and M. Beckmann, "Assignment Problems and the Location of Economic Activities," Econometrica, Vol. 25, No. 1 (January 1957) p. 53-76.

16. Land, A. H., "A Problem of Assignment with Interrelated Costs," Operational Research Quarterly, Vol. 14, No. 2 (June 1963) p. 185-198.

17. Land, A. H. and A. Doig, "An Automatic Method of Solving Discrete Programming Problems," Econometrica, Vol. 28, No. 3 (July 1960) p. 497-520.

18. Lawler, E. L., "The Quadratic Assignment Problem," Management Science, Vol. 9, No. 4 (July 1963) p. 586-599.

19. _____, "Notes on the Quadratic Assignment Problem," Harvard Computation Laboratory, unpublished paper, April, 1960.

20. Little, J. D. C., K. G. Murty, D. W. Sweeney and K. Caroline, "An Algorithm for the Traveling Salesman Problem," Operations Research, Vol. 14, No. 6 (November-December 1963) p. 972-989.

21. Markowitz, H. M. and A. S. Manne, "On the Solution of Discrete Programming Problems," Econometrica, Vol. 25, No. 1 (January 1957) p. 84-110.

22. Maxwell, W. L., "The Scheduling of Single Machine Systems: A Review," The International Journal of Production Research, Vol. 3, No. 3 (1964) p. 177-199.

23. Nugent, C. E., T. E. Vollman, and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," Operations Research, Vol. 16, No. 1 (January-February 1968) p. 150-173.

24. Pegels, C. C., "Plant Layout and Discrete Optimizing," International Journal of Production Research, Vol. 5, No. 1 (1966) p. 81-92.

25. Pierce, J. F. and D. J. Hatfield, "Production Sequencing by Combinatorial Programming," Operations Research and the Design of Management Information Systems (J. F. Pierce, Ed.), Tech. Association of the Pulp and Paper Industry, New York, 1967.

26. Rossman, M. J. and R. J. Twery, "Combinatorial Programming," Unpublished paper presented at 6th Annual meeting of Operations Research Society of America, Boston, 1958.

27. Rutman, R. A., "An Algorithm for Placement of Interconnected Elements Based on Minimum Wire Length," Proc. AFIPS Spring Joint Computer Conference, 1964, p. 477-491.

28. Shapiro, D., "Algorithms for the Solution of the Optimal Cost Traveling Salesman Problem," Sc.D. Thesis, Washington University, St. Louis, 1966.

29. Sprague, C. R., "Computational Experience with Variants of the Eastman Algorithm for the Traveling Salesman Problem," M.I.T. Sloan School Working Paper, forthcoming.

30. Steinberg, L., "The Backboard Wiring Problem: A Placement Algorithm," Soc. for Ind. and Applied Mathematics, Vol. 3, No. 1 (January 1961) p. 37-50.

31. Vollman, T. E. and E. S. Buffa, "The Facilities Layout Problem in Perspective", Management Science, Vol. 12, No. 10 (June 1966) p. 450-468.

32. Whitehead, B. and M. Z. Elders, "An Approach to the Optimum Layout of Single-Story Buildings," Architect's Journal, Vol. 139, No. 25 (June 1964) p. 1373-1380.

33. Wimmert, R. J., "A Mathematical Model of Equipment Location," Journal of Industrial Engineering, Vol. 9, No. 6 (November-December 1958) p. 498-505.