1970

The assignment problem and a suboptimal solution technique

William Barry Gregory

THE ASSIGNMENT PROBLEM AND A
SUBOPTIMAL SOLUTION TECHNIQUE

BY

WILLIAM BARRY GREGORY, 1943-

A

THESIS

submitted to the faculty of

UNIVERSITY OF MISSOURI - ROLLA

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Rolla, Missouri

1970

Approved by

C. Y. Ho (advisor) B E Gillett

[signature]

ABSTRACT

A technique is presented which finds a fast suboptimal solution to the assignment problem. This same technique is then applied to two large dynamic programming problems for which the optimal solution is not known. These examples illustrate how easily this technique can be applied and that it is better than most optimizing techniques because it is fast, cheap and only minor hand calculations are needed.

The effect of different initial solutions and their value are compared and it is found that the initial solution is not as significant in a suboptimal technique as in an optimizing technique. The initial solution is significant enough, however, to warrant some care in choosing it.

ACKNOWLEDGEMENT

I wish to acknowledge Professor Ralph E. Lee for the use of the Computer Center facilities and for the assistance of his fine staff.

I wish to acknowledge all the people who encouraged me to write this thesis and all those who helped me while I was writing it by their thoughtful criticism and much needed guidance.

Most of all, I wish to thank my wife, Kay, who nagged me until I finished it so she could correct my spelling and type this final copy.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Page

I. INTRODUCTION

The investigation of this thesis is just a special case of the transportation problem, which is a special case of the general linear programming problem.

The general linear programming problem, as stated mathematically, is to find x_i, $i = 1,2,\ldots,n$ which minimizes $Z = \sum_i c_i x_i = \langle C, X \rangle$ subject to the constraints

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leqslant b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leqslant b_2$$
$$\vdots$$
$$a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leqslant b_m$$

and

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ \ldots,\ x_n \geqslant 0$$

or more concisely, the constraints can be written as $AX \leqslant B$ and $x_i \geqslant 0$, $i = 1,2,\ldots,n$ where A is the m x n coefficient matrix and B is an m element long column vector. Z is called the objective function and the x_i's are called decision variables. A, B, and C are usually given constants for any particular problem. The Simplex Method is the most popular technique for solving this problem.

The transportation problem, as stated previously, is a special case of the general linear programming problem. The transportation problem, in mathematical terms, is to find x_{ij}, $i=1,2,\ldots,m$ and $j=1,2,\ldots,n$ which minimizes $Z = \sum_i \sum_j c_{ij} x_{ij} = \langle C,X \rangle$ subject to the constraints

$$\sum_j x_{ij} = a_i \text{ for all } i,\ i = 1,2,\ldots,m$$
$$\sum_i x_{ij} = b_j \text{ for all } j,\ j = 1,2,\ldots,n$$

and

$$x_{ij} \geqslant 0 \text{ for all } i \text{ and } j.$$

This problem has a feasible solution only if $\sum_i a_i = \sum_j b_j = \sum_i \sum_j x_{ij}$. That is to say, the system is in balance. The Simplex Method could obviously be used to solve this problem. However, like many other problems which are a special case of a more general problem, there are better techniques for solving the transportation problem than the Simplex Method.

The subject of this investigation is a special case of the transportation problem commonly referred to as the assignment problem. If $m = n$, $\sum_i x_{ij} = 1$ for all j and $\sum_j x_{ij} = 1$ for all i in the transportation problem, then it is reduced to the assignment problem. Which, more concisely, is to find x_{ij}, $i=1,2,\ldots,n$ and $j=1,2,\ldots,n$ which minimizes $Z = \sum_i \sum_j c_{ij} x_{ij} = \langle C,X \rangle$ subject to the constraints

$$\sum_i x_{ij} = 1 \text{ for all } j$$
$$\sum_j x_{ij} = 1 \text{ for all } i$$

and

$$x_{ij} = \begin{cases} 1 \text{ if } j^{th} \text{ element in } i^{th} \text{ row is assigned} \\ 0 \text{ otherwise.} \end{cases}$$

This means that in the solution one and only one element in any row or column of the X matrix will have a value of one. All other elements in that row or column will have a value of zero. That is to say, the solution will be a row or column permutation of an nth order Identity matrix.

As before, this problem is solvable by the Simplex Method, but the nature of the problem lends itself to other techniques. These other techniques are the major subject of this thesis.

A typical problem which could be solved by developing it as an assignment problem might be the assignment of jobs to persons in a job shop. Knowing that each person could do certain jobs at a certain cost and would require a certain amount of time, the problem would probably be to assign all the jobs to all the persons in the shop to minimize the cost or time or perhaps a combination of both.

A technique which will minimize a function then certainly could be modified to maximize a similar function.

Another typical problem might be the assignment of machines to certain functions to maximize production.

Perhaps these problems are a little understated. Probably the most difficult problem to solve of any that will be stated here is the problem of obtaining the cost coefficients as realistic values. This, however, is not the problem which is under consideration.

Therefore, I submit that there are many problems which arise in this area that need solutions. I feel that many of the current techniques being used to solve the problems are too involved to be practical. Many problems are being solved irresponsibly by persons who can only guess at the solutions. If a problem requires an urgent solution, it is often solved with very little regard to any planned technique but based upon intuitive guessing. Often this is the only tool available. However, I do believe there is a technique which is more logical than intuitive guessing and yet simple enough to be used to solve large problems with a minimum of effort.

A Review of Literature follows this section as Section II.

The technique I have devised and its effectiveness is discussed in Section III, titled A Fast Solution Technique.

Some Examples and Practical Problems will constitute Section IV. Section V will state the Conclusions drawn from this study.

II. REVIEW OF LITERATURE

There exist several methods for solving the assignment problem. The problem itself is known by many different names. Dantzig[3] calls it the Marriage Problem and it is described as a problem of deciding who marries who when mail order brides are brought into a mining town. When each prospective bride chooses a different prospective groom, the problem is easy to solve. It only becomes difficult after more than one girl chooses a particular boy. And, of course, bigamous, and generally, polygamous marriages are ruled out. Dantzig applies dynamic programming to solve the problem. Niederjohn[15] felt that removing some of the steps of Dantzig's Method would make it more usable.

Dynamic Programming lends itself to the solution of the assignment problem very nicely except that it requires considerable core storage when implemented on a digital computer and, therefore, the size of problem solvable is somewhat restricted by the equipment available.

A second, and perhaps more popular technique, is the Hungarian Method first described by Kuhn[7] and considerably refined by Munkres[13]. Perhaps the reason this technique is more popular is the fact that the person using the technique develops more of a feel for the problem and the computation required is just simple addition and subtraction.

Although this is a popular hand calculation method, its implementation on a digital computer would be somewhat restricted by computations required not in solving a problem, but in determining when a solution is optimal. It is popular because it is a hand calculation method, but the necessary re-writing of the cost matrix makes it somewhat restrictive as to size of problem solvable. In most problems small enough to solve by hand, the optimum solution can be determined by inspection as the last step of the algorithm. Yaspan[19] refined the last step of this algorithm to make it usable for larger problems although he felt that it was still too cumbersome for a digital computer. Murty[14] felt that more than one solution might be needed and, in fact, a ranking of solutions could be useful.

The Simplex Method has always been popular as a technique for solving the assignment problem and a number of papers have been written by very knowledgeable people who feel that the background of information available about the Simplex Method make it more usable because the proof of an optimum solution in a finite number of steps has already been found. Szwarc[18] attempted to improve it by finding a better initial solution and allowing mixed-integer linear solutions. Although he was more concerned with the transportation problem, his modifications could surely be applied to the assignment problem. This leads, however, to the same problem of efficiency. Methods, in general, tailor-made for the transportation problem could hardly be expected to perform as well for a particular sub-problem such as the assignment problem.

The fourth method is relatively new by comparison to other methods already discussed. This method is branch and bound, or backtracking, and is discussed in its entirety by Little, et al.[9] as applied to a traveling salesman problem. Dantzig, of course, formulated the assignment problem and the traveling salesman or shortest route problem into a common form and used dynamic programming to solve both types of problems. This formulation is restrictive and allows only a dynamic programming solution. It does not appear to be a good method for converting when dynamic programming is not being used. Glover[6] presents a combination of backtracking and dual-simplex. Most of his ideas come from Balas[2] who used the dual-simplex and considered the speed at which a solution was found to be of prime importance. These papers are more concerned with the general problem and not with problems of a particular nature. Gavett and Plyter[5] and Efroymson and Ray[4], on the other hand, consider branch and bound a good technique for optimal assignment of facilities to locations which is more closely related to the classical assignment problem. Lawler and Wood[8] are responsible for a very good survey in which they compare branch and bound very favorably to dynamic programming for solving integer linear programming and traveling salesman problems. They feel, however, that dynamic programming offers the best direct approach.

There are several other partial enumeration methods to be found in the literature. In fact, Reiter and Sherman[16] considered four such partial enumerations schemes with just a slightly new twist. They incorporated random search in addition to whatever intelligently directed search could be employed. The random search was intended to prevent the directed search from stopping at a local optimum which was not the optimum solution. Their first two algorithms are similar to the one presented in this paper, but are significantly different only in the action taken after an initial solution is found.

There are, then, generally four uniquely different techniques for solving the assignment problem. Dynamic programming, while it will solve the problem, is restricted as to the size of problem that can be solved. Hungarian methods are difficult for large problems, but are nice to give the user a feel for the problem. With this method, it is difficult to know when the problem is solved. The Simplex Method is usually thought of first because it is the most popular method for solving linear programming problems. However, the constraints of an assignment problem are generally too numerous to be solved because of the matrix inversion employed by the Simplex Method. The partial enumeration schemes of Reiter and Gordon may be the best approach with branch and bound adding support to the theory behind it. Branch and bound was originally developed for the traveling salesman problem and the direct application to the assignment problem is less than ideal.

All of the above techniques are concerned with an exact solution. Manne[11] believes that in a job-shop scheduling problem, Monte Carlo methods would require much less computer time. Balas[1] seems to think that aggregation or solving some subsets of the larger problem before solving the larger problem might be a still better approach. Saaty[17] applies partitions to acquire an approximate solution to the general linear programming problem. Machol[10] solves a practical problem with a suboptimal technique and feels that his is the best solution that could be found under the circumstances. And so, there are perhaps many applications where the expense of obtaining an optimal solution is not considered worthwhile.

III. A FAST SOLUTION TECHNIQUE

The fast solution technique basically consists of 4 steps, as follows, assuming the problem is to minimize the function:

1. Find the smallest element in a row.
2. Temporarily assign this element and remove all other elements in that element column from consideration.
3. Proceed to another row and follow the procedure of steps 1 and 2 above being careful not to temporarily assign more than one element in any one row or column. For most small problems, $n \leq 4$ the solution is apparent when all the rows and columns have been temporarily assigned.
4. Having temporarily assigned n elements of the n x n coefficient matrix C, the assignments will be tested to determine if an improvement can be made. The idea is to compare the sum of two assignments with the sum of their alternative elements. If the alternative element sum is smaller than the assignment pair sum, then the assignment should be modified to include the alternative pair in the assignment, which means, of course, to omit the pair previously assigned. For example,

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & & & c_{1n} \\ \cdot & \cdot & & & & \cdot \\ & & c_{ij} & \cdots & c_{ik} & \\ & & \cdot & & \cdot & \\ & & c_{mj} & \cdots & c_{mk} & \\ \cdot & \cdot & & & & \cdot \\ c_{n1} & c_{n2} & \cdots & & & c_{nn} \end{bmatrix}$$

assume c_{ij} and c_{mk} are temporarily assigned. Step 4 means if $c_{ij} + c_{mk}$ is larger than $c_{ik} + c_{mj}$ then c_{ik} and c_{mj} should be assigned instead of c_{ij} and c_{mk}. Or, in more precise mathematical terms, $x_{ij} = x_{mk} = 0$ and $x_{ik} = x_{mj} = 1$ as the new assignment. Any procedure which provides for the comparison of every assigned element pair with their alternative pair is entirely satisfactory. If an assignment is modified, then the procedure should begin again in order to insure that all comparisons are made.

A flow chart of this technique appears in Appendix A.

One such procedure for making all required comparisons might be to compare each row's assigned element and all the subsequent assignments with their alternative element sums. That is to say, the first row's element would be compared to the second row's element and then to the third row's element and eventually to the nth row's element. Then the second row's element is compared to the 3rd row's element and then with the 4th row's element and finally with the nth row's element. This procedure should be followed until the (n-1)th row's element is compared to the nth row's element. If no reassignments were made, the solution is found. If a reassignment has been made, this procedure should be repeated.

Another suitable procedure might be to compare the largest assigned element and the next largest. And then compare largest with 3rd largest on down to comparing the largest and smallest with their alternative element sums. Then go back and compare the 2nd largest with the 3rd largest until all the assigned elements have been compared pair-wise to all other assigned elements. If a reassignment is made, then the procedure should be repeated.

Note the form of the decision matrix X. It is a row or column permutation of an nth order Identity matrix.

The C matrix or cost matrix, as it is often called, is a matrix of coefficients of the minimizing function. Throughout the solution of the problem, C's elements remain constant.

One final point must be made about this technique. This technique does not find the optimum solution. If the procedure were extended to comparing inner products of row or column permutations of nth order Identity matrices times the cost coefficient matrix, then the optimum solution could be found by enumeration. This is obviously impractical for a problem of any magnitude. In fact $n!$ inner products, where n is the order of the coefficient matrix, would have to be calculated to solve by enumeration. As an example, consider a 100th order coefficient matrix.

To solve this problem by enumeration would require 9.3326×10^{157} inner products. This problem is not unsolvable by any means, however, to attempt a solution by hand is not even thinkable.

For most practical problems, an optimum solution is not required because of the time required to obtain an optimum solution or because of the expense of obtaining the optimum solution. In many cases, the cost of getting the correct answer to a problem will outweigh the profit of using it as opposed to an easily obtained near correct answer. The only trouble with using a near correct answer is that in most cases, the cost of obtaining the correct answer is not known and, of course, the amount of improvement or profit gain is not known.

For these reasons, a discussion of these differences is included in Section IV under Some Examples and Practical Problems.

IV. SOME EXAMPLES AND PRACTICAL PROBLEMS

The reader should understand the differences between optimal and suboptimal solutions and when one is required or more desirable than the other. If the problem is a continuing one, such as the traveling salesman traveling the same route time and time again, the error is magnified every time a suboptimal solution is used. If the problem is not continuous, as in a job-shop where persons are assigned to do different jobs on a job to job basis, a suboptimal solution is perhaps more desirable because it can be obtained much more cheaply and quickly.

At this point, it should be noted that anyone can obtain a suboptimal solution. However, some suboptimal solutions may be worse than no solution at all because the suboptimal or feasible solution set contains the maximum as well as the minimum of the objective function. The traveling salesman problem is a problem which requires an optimal solution, especially if the route is to be used several times. In general, if the solution is to be applied only once, a suboptimal solution will be adequate and a very close to optimal solution may be even more desirable than the optimal solution.

The following are a series of examples where a suboptimal solution might well be applicable.

The next two examples are classic dynamic programming problems which will be solved by a non-dynamic method.

Example 1:

Consider the problem of assigning more than one person to a job to improve the overall performance of that job. Table I gives the effectiveness (reciprocal of time) of doing each of eight jobs, assuming twenty persons are used to work on the jobs and if a job does not get done at all there is no penalty. The objective is clearly to maximize the effectiveness.

Clearly, all twenty persons could be assigned to either job 1 or job 5 and the effectiveness is 76. Just as clearly, nineteen persons could be assigned to job 5 and one to job 4 and the effectiveness is boosted to 96, which is a pretty good start. This last also suggests a fast solution technique which is in all likelihood suboptimal, but the ease with which such a large increase was found also adds to the intrigue. Doing this personnel swapping just as far as it will go yields an effectiveness total of 141 with 1, 1, 7, 1, 10, 0, 0 and 0 persons assigned to jobs one to eight respectively. This may be the optimal solution, but enumeration or dynamic programming would have to be used to tell for sure. Both of these are out of the question because dynamic programming would require too much core and enumeration would take too long. There are four alternative solutions found also and they are 2, 0, 7, 1, 10, 0, 0, and 0 or 1, 0, 7, 1, 10, 1, 0, and 0 or 1, 0, 7, 1, 10, 0, 1, and 0 or 1, 0, 7, 1, 10, 0, 0, and 1, all of which yield an effectiveness total of 141. The zeros in the table indicate that zero men will accomplish zero jobs and that at least four men are required to

TABLE I EFFECTIVENESS COEFFICIENTS

Personnel	Job Number 1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	10	2	0	20	0	2	2	2
2	12	3	0	20	0	2	2	4
3	14	3	0	20	0	2	3	6
4	16	3	6	21	25	2	4	8
5	18	4	12	21	27	4	5	10
6	20	6	18	22	27	6	6	12
7	22	10	36	22	27	6	7	14
8	24	18	37	23	42	8	10	16
9	26	26	37	24	49	10	11	18
10	28	40	37	24	71	12	11	20
11	31	44	38	24	71	13	18	22
12	33	45	38	24	71	14	23	24
13	37	46	39	24	71	14	29	26
14	39	47	40	25	71	15	36	28
15	43	49	40	25	72	15	48	30
16	47	50	50	26	73	22	59	32
17	55	51	60	26	74	31	70	34
18	60	51	61	26	75	41	71	36
19	66	51	62	26	76	42	72	38
20	76	52	63	26	76	42	73	40

do jobs 3 and 5. The numbers in the table may not be as realistic as some other numbers, but the method for finding a solution would remain the same.

Note that if the number of personnel were reduced, the effectiveness values would remain the same and a new solution could be found quickly with just a minimum of effort.

Example 2:

The St. Louis Post Mortam News Agency* distributes newspapers and magazines to 500 different newsstands in the Missouri-Illinois area. They handle 44 different news items. Table II gives the probability that each magazine up to 20 units will be purchased daily. Table III gives the normalized profit and weight of each magazine. The regular delivery truck has sufficient capacity to handle all the magazines needed by all the newsstands, but today it is out of service with a broken rear axle. The only other truck available is a rental truck that is considerably smaller than the usual delivery truck. The problem, then, is to maximize expected profit while minimizing the weight on the rental truck.

The first step to consider is to obtain a cost matrix. A function must be defined that will describe the cost coefficient matrix adequately, not only as a function of expected profit, but as a function of weight. If the maximum weight allowed on the vehicle is exceeded, then the vehicle may break down or the driver be arrested for exceeding weight limitations. For this problem, we will use a cost coefficient as follows:

$$c_{ij} = \left(1 - \sum_{l=0}^{j-1} p_{lj}\right) P_j / W_j$$

where:

p_{lj} = probability that the lth unit of the jth magazine will be sold and $p_{0j} = 0$.

P_j = normalized profit on each unit of the jth magazine.

* The name of this Agency is of a ficticious nature as is all the data involved in this and all other examples.

TABLE II PROBABILITY DISTRIBUTIONS FOR MAGAZINE SALES

	Magazine Number										
	1	2	3	4	5	6	7	8	9	10	11
1	.01	.02	.20		.01	.50					.20
2	.01	.01	.20		.01	.50					.20
3	.01	.01	.25		.01			.99			.10
4	.01	.01	.20	.20	.10		.10	.01		.20	.10
5	.01	.01	.15	.25	.10		.10		.02	.20	.20
6	.02	.01		.20	.10		.10		.25	.20	.20
7	.02	.01		.20	.18		.10		.40	.20	
8	.05	.01		.10	.18		.30		.20	.20	
9	.06	.01		.05	.18		.20		.09		
10	.30	.02			.09		.10		.04		
11	.30	.02			.02						
12	.06	.02			.01						
13	.05	.02			.01						
14	.02	.05									
15	.02	.05									
16	.01	.11									
17	.01	.11									
18	.01	.11									
19	.01	.25									
20	.01	.15									

TABLE II (Continued)

	Magazine Number 12	13	14	15	16	17	18	19	20	21	22
1							1.0				
2									.25	.15	
3						.01		.50		.15	.10
4				1.0		.01				.25	.10
5					.80	.02			.25	.25	.20
6	.25	1.0			.10	.02		.50		.10	.20
7	.50				.10	.04				.10	.15
8	.25					.04			.25		.15
9						.20					.10
10						.25					
11						.30					
12						.11			.25		
13											
14											
15											
16											
17											
18											
19											
20			1.0								

TABLE II (Continued)

	Magazine Number										
	23	24	25	26	27	28	29	30	31	32	33
1					.20	.05	.10	.20	.40	.20	.40
2			.50	.20		.05	.10	.20	.40	.40	.20
3		.50			.20	.05	.10	.20	.20	.40	.40
4	.50		.25	.20		.05	.10	.20			
5		.25			.20	.05	.10	.20			
6	.50		.15	.20		.05	.10				
7		.25			.20	.05	.10				
8			.10	.20		.05	.10				
9					.20	.05	.10				
10				.20		.05	.10				
11						.05					
12						.05					
13						.05					
14						.05					
15						.05					
16						.05					
17						.05					
18						.05					
19						.05					
20						.05					

TABLE II (Continued)

	Magazine Number 34	35	36	37	38	39	40	41	42	43	44
1										.01	
2	.25	.25	.50		.20			.20		.02	.03
3	.25			.40	.20			.10		.03	
4					.40	.40	.30	.20		.04	.07
5					.20	.10	.10	.20	.10	.05	
6	.25					.10	.10	.10	.20	.06	.11
7	.25	.25				.30	.10	.20	.10	.07	
8		.25		.40		.10	.10		.20	.08	.15
9							.30		.10	.09	
10									.20	.05	.14
11		.25							.10	.05	
12										.09	.14
13				.10						.08	
14				.10						.07	.15
15										.06	
16										.05	.11
17										.04	
18										.03	.07
19			.50							.02	
20										.01	.03

TABLE III NORMALIZED PROFITS, WEIGHTS, AND RATIOS

Magazine #	1	2	3	4	5	6	7	8	9	10	11
Profit (P)	.10	.15	.30	.50	.50	1.00	.50	1.00	.50	.50	.50
Weight (W)	.01	.01	.20	.25	.25	.95	.40	.98	.25	.30	.40
P/W	10.0	15.0	1.5	2.0	2.0	1.05	1.25	1.02	2.0	1.67	1.25

Magazine #	12	13	14	15	16	17	18	19	20	21	22
Profit (P)	.60	.60	.50	.75	.50	.25	1.00	.50	.50	.75	.50
Weight (W)	.50	.50	.08	.75	.20	.15	1.00	.25	.35	.10	.10
P/W	1.20	1.20	6.25	1.0	2.5	1.67	1.0	2.0	1.43	7.5	5.0

Magazine #	23	24	25	26	27	28	29	30	31	32	33
Profit (P)	.50	.50	.50	.50	.50	.25	.50	.75	1.00	1.00	1.00
Weight (W)	.05	.05	.05	.05	.05	.01	.02	.05	.50	.50	.50
P/W	10.0	10.0	10.0	10.0	10.0	25.0	25.0	15.0	2.0	2.0	2.0

Magazine #	34	35	36	37	38	39	40	41	42	43	44
Profit (P)	.50	.50	1.00	.75	.50	.50	.50	.50	.25	.30	.50
Weight (W)	.30	.30	.90	.50	.40	.30	.30	.35	.20	.01	.02
P/W	1.67	1.67	1.11	1.5	1.25	1.67	1.67	1.43	1.2	30.0	25.0

and

W_j = normalized weight of each unit of the jth magazine.

And we will assume that the weight allowed is one half that which would fulfill all the requirements of the usual sales. The weight put on the usual delivery truck is 95.92 units. The weight allowed on the rental truck is therefore 47.96 units. Table IV contains the newly computed cost coefficient matrix from which the optimum content of the rental truck can be computed. Let us start with an initial solution of all the magazine units for the first twenty-four magazines plus three units of the twenty-fifth magazine. This combined weight is 47.92 units, which is within the allowed weight. The procedure will then be the same as in Example 6. Lower profit-weight ratios will be subtracted and higher profit-weight ratios will be added. The weight restriction must not be violated. The initial solution in terms of units of magazines is 20, 20, 5, 9, 13, 2, 10, 4, 10, 8, 6, 8, 6, 20, 4, 7, 12, 1, 6, 12, 7, 9, 6, 7, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, and 0 for each of the 44 magazines respectively.

The next step should be to increase profit while maintaining the correct weight. Examine magazine number 8. If we reduced the number of units of this magazine by one, we would reduce the weight by .98 units. This weight could then be filled by a magazine with a larger profit-weight ratio. Magazine number 43 is ideal for this exchange. All 20 units of magazine number 43 should be included. This exchange did not increase the weight and, in fact, reduced it. This lost weight can be replaced by several units of other magazines. All 20 units of magazine number 44 can be added, as can all 20 units of magazine number 28.

TABLE IV COST COEFFICIENTS

	Magazine Number 1	2	3	4	5	6	7	8	9	10	11
1	10.0	15.0	1.50	2.0	2.0	1.05	1.25	1.02	2.0	1.67	1.25
2	9.9	14.85	1.20	2.0	1.98	.525*	1.25	1.02	2.0	1.67	1.00
3	9.8	14.70	.90	2.0	1.96		1.25	1.02	2.0	1.67	.75
4	9.7	14.55	.525	2.0	1.94		1.25	.011*	2.0	1.67	.625
5	9.6	14.40	.225*	1.6	1.74		1.125		2.0	1.33	.500
6	9.5	14.25		1.1	1.54		1.000		1.96	1.00	.25*
7	9.3	14.10		.7	1.34		.875		1.46	.67	
8	9.1	13.95		.3	.98		.750		.66	.33*	
9	8.6	13.80		.1*	.62		.375		.26		
10	8.0	13.65			.26		.125*		.08*		
11	5.0	13.35			.08						
12	2.0	13.05			.04						
13	1.4	12.75			.02*						
14	.9	12.45									
15	.7	11.70									
16	.5	10.95									
17	.4	9.30									
18	.3	7.65									
19	.2	6.00									
20	.1*	2.25*									

*Indicates initial solution

TABLE IV (Continued)

	Magazine Number										
	12	13	14	15	16	17	18	19	20	21	22
1	1.2	1.2	16.0	1.0	2.5	1.67	1.0*	2.0	1.43	7.5	5.0
2	1.2	1.2	16.0	1.0	2.5	1.67		2.0	1.43	7.5	5.0
3	1.2	1.2	16.0	1.0	2.5	1.67		2.0	1.07	6.375	5.0
4	1.2	1.2	16.0	1.0*	2.5	1.65		1.0	1.07	5.250	4.5
5	1.2	1.2	16.0		2.5	1.63		1.0	1.07	3.375	4.0
6	1.2	1.2*	16.0		.5	1.60		1.0*	.72	1.500	3.0
7	.9		16.0		.25*	1.57			.72	.75*	2.0
8	.3*		16.0			1.50			.72		1.25
9			16.0			1.44			.36		.5*
10			16.0			1.11			.36		
11			16.0			.69			.36		
12			16.0			.18*			.36*		
13			16.0								
14			16.0								
15			16.0								
16			16.0								
17			16.0								
18			16.0								
19			16.0								
20			16.0*								

*Indicates initial solution

TABLE IV (Continued)

	Magazine Number										
	23	24	25	26	27	28	29	30	31	32	33
1	10.0	10.0	10.0	10.0	10.0	25.0	25.0	15.0	2.0	2.0	2.0
2	10.0	10.0	10.0	10.0	8.0	23.75	22.5	12.0	1.2	1.6	1.2
3	10.0	10.0	5.0*	8.0	8.0	22.50	20.0	9.0	.4	.8	.8
4	10.0	5.0	5.0	8.0	6.0	21.25	17.5	6.0			
5	5.0	5.0	2.5	6.0	6.0	20.00	15.0	3.0			
6	5.0*	2.5	2.5	6.0	4.0	18.75	12.5				
7		2.5*	1.0	4.0	4.0	17.50	10.0				
8			1.0	4.0	2.0	16.25	7.5				
9				2.0	2.0	15.00	5.0				
10				2.0		13.75	2.5				
11						12.50					
12						11.25					
13						10.00					
14						8.75					
15						7.50					
16						6.25					
17						5.00					
18						3.75					
19						2.50					
20						1.25					

*Indicates initial solution

TABLE IV (Continued)

	Magazine Number										
	34	35	36	37	38	39	40	41	42	43	44
1	1.67	1.67	1.11	1.5	1.25	1.67	1.67	1.43	1.2	30.0	25.0
2	1.67	1.67	1.11	1.5	1.25	1.67	1.67	1.43	1.2	29.7	25.0
3	1.25	1.25	.555	1.5	1.00	1.67	1.67	1.15	1.2	29.1	24.25
4	.83	1.25	.555	.9	.75	1.67	1.67	1.01	1.2	28.2	24.25
5	.83	1.25	.555	.9	.25	1.00	1.17	.72	1.2	27.0	22.50
6	.83	1.25	.555	.9		.83	1.00	.43	1.08	25.5	22.50
7	.42	1.25	.555	.9		.67	.83	.29	.84	23.7	19.75
8		.83	.555	.9		.17	.67		.72	21.6	19.75
9		.42	.555	.3			.50		.48	19.2	16.00
10		.42	.555	.3					.36	16.5	16.00
11		.42	.555	.3					.12	15.0	12.50
12			.555	.3						13.5	12.50
13			.555	.3						10.8	9.00
14			.555	.15						8.4	9.00
15			.555							6.3	5.25
16			.555							4.5	5.25
17			.555							3.0	2.50
18			.555							1.8	2.50
19			.555							1.8	2.50
20										.3	.75

Only 9 units of magazine number 29 can be added to get back to the original weight of the initial solution. However: one additional unit of number 29 will be allowed as this will still be within the original weight constraint of 47.96. The new solution is then 20, 20, 5, 9, 13, 2, 10, 3, 10, 8, 6, 8, 6, 20, 4, 7, 12, 1, 6, 12, 7, 9, 6, 7, 3, 0, 0, 20, 10, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20, and 20. The number of units is increased by 70 and the expected profit changes from 72.37 to 86.39 normalized units.

Note how the magazine to be removed was located. The smallest available cost coefficient in the initial solution is .011, which corresponds to magazine number 8. The largest coefficient not in the initial solution is 30.0, which corresponds to magazine number 43. If the exact solution were required, only one unit of magazine number 43 would have been chosen at a time until a larger coefficient than the next one in number 43 were found. But, since this is not an exact algorithm, this discrepancy will be ignored. This, incidentally, corresponds to total enumeration.

Follow the above procedure until all of the entries chosen in the initial solution have been checked and possibly reduced. Then check those that were entered into the solution to see if they should be removed.

The final solution is 13, 20, 2, 6, 7, 1, 5, 3, 7, 5, 1, 6, 6, 20, 0, 5, 10, 0, 3, 5, 6, 8, 6, 7, 7, 10, 9, 20, 10, 5, 2, 2, 2, 3, 7, 2, 3, 2, 4, 5, 3, 5, 19, and 19 for the 44 magazines respectively. And the expected profit is 119.84. The weight on the rental truck will be 47.96 units.

The expected profit is computed using the normalized profits from Table III and the probability distributions from Table II. It is the sum of the expected profits from each magazine unit. Each magazine's expected profit is the sum of the probability that the magazines will be sold times profit.

In this problem, as in most others of its general type, the most difficult part is obtaining meaningful coefficients. More often than not, there is no mathematics involved in choosing coefficients as in the case of personnel assignments where employees are rated as to effectiveness in job performance.

The previous examples hint at a large variety of problems encountered in real life that might be solved by a better method than intuitive guessing. Some of these other examples follow.

Example 3:

A small computer center has purchased a somewhat larger and faster computer. The only problem being that they do not have sufficient funds to purchase a new keypunch that codes the correct punch combinations for the new equipment. The old keypunch machine has all the required equipment to punch the new code except that when the plus (+) key is punched and a plus (+) is printed on the top of the card, the new machine interprets the punch combination as an ampersand (&). There are 48 punch combinations on the old machine and 60 punch combinations that the new machine can interpret. This means, of course, that 12 punch combinations will have to be handled on an individual basis, but 48 punches can be handled directly if the old machine can be converted to punch the correct combination for the new equipment when various keys are depressed. This is only a theoretical problem, but is the same problem one would encounter if it were necessary to convert an IBM 026 keypunch to an IBM 029 as would be necessary if an IBM-360 computer were purchased to replace an old IBM-1620. This example is similar to Machol's[10] paper tape punch conversion problem.

Example 4:

Consider the job-shop problem. Assume the shop foreman has 200 jobs to be done in the next four days and he knows that his fifty men can do a separate job a day on the average of 7 1/2 hours per job. Then each man must average at least one job a day for four days. How does he assign the jobs to get each job finished by the last day? If you were an experienced foreman who knew each of the fifty men well, then you would probably work it out so that there wasn't any problem. If, however, your job as foreman hangs in the balance, it might be nice to have a scheme which indicates a definite plan of attack to solve the problem, and to know in advance that the jobs will all be done, or if, indeed, they can be done.

Example 5:

Assume for a moment you are a doctor and that you carry a black bag with you on your house calls. What do you carry in your bag to minimize the probability that you will arrive at your patient's bedside without the proper medication for the patient's ailment? If, on the other hand, you are the patient, you might consider the possibility of the doctor increasing the size of his bag and decide that a near solution just is not adequate in this problem. This is actually a modified version of the knapsack problem which was mentioned frequently throughout the literature.

Example 6:

Minty[12] gives us an interesting variation to the shortest route problem which might well be an application for an approximation method. Suppose that an airline traveler presents himself at the ticket counter and requests the quickest route to another city. Then his route would be dependent upon when he arrived at the ticket counter, as well as where his origin is and where his destination might be. This problem has some very interesting aspects that might make it complicated enough to keep one busy a long time.

Example 7:

Suppose that the chairman of a department at a small midwestern college must assign his personnel to teach the courses offered by his department. All of the personnel can teach all of the courses, however, some can teach some courses better than others. How then does the chairman assign his people to get the best teaching effort from them? Are there any methods being used at the present time to solve this problem? Actually, this is not as difficult a problem as some because the chairman obviously knows the personnel well enough to make a better-than-average guess. And, in fact, may find the optimal solution without any mathematical method.

These last five over-simplified examples should be sufficient to suggest many more problems which might be solved by suboptimal techniques.

V. CONCLUSION

The initial solution used in a suboptimal algorithm is not as important as it would be in an optimizing procedure. If an optimum solution is required, choosing the proper initial solution may reduce the number of steps appreciably. Figure 1 compares two different initial solutions with an absolute minimum and an actual minimum. Table V lists all the values used in the figures below.

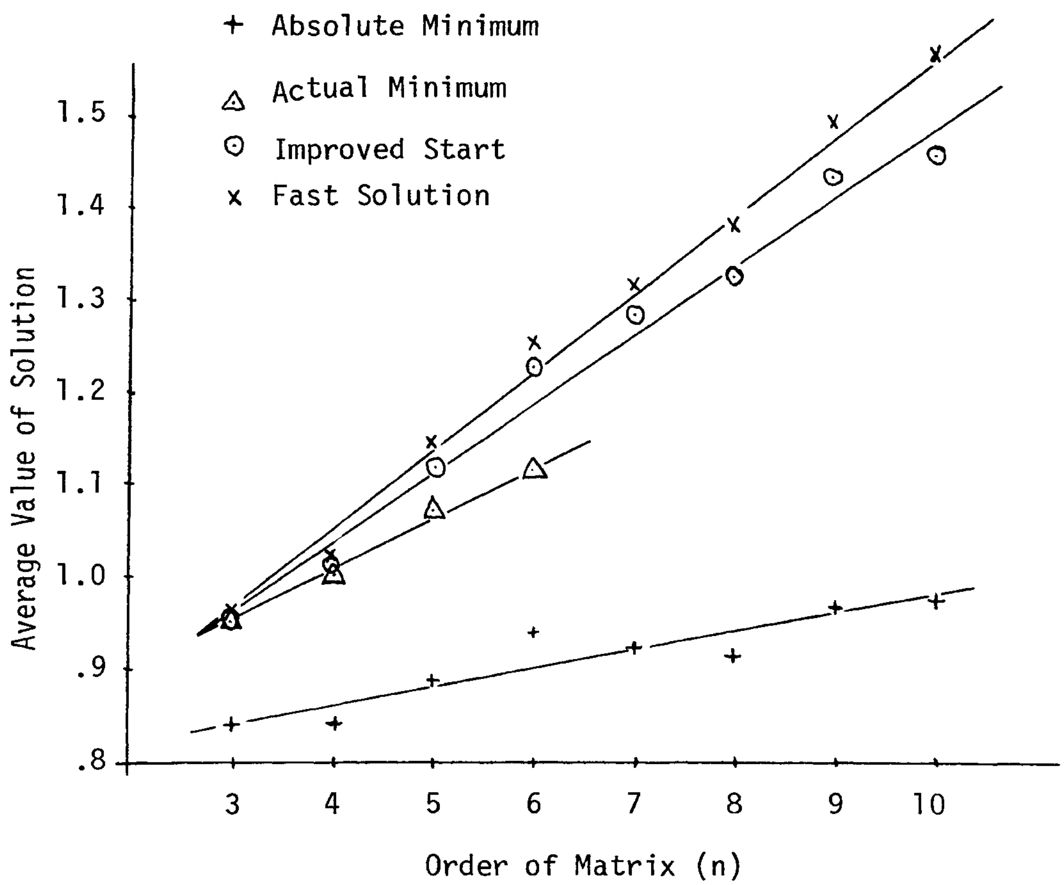


Figure 1 SOLUTION VALUE

TABLE V DATA OBTAINED FROM COMPUTER RUNS

	Order of Matrix (n)	3	4	5	6	7	8	9	10	100
Average Solution Value	Absolute	.8349	.8430	.8858	.9346	.9223	.9133	.9656	.9717	0.9964
	Fast	.9636	1.0185	1.1437	1.2528	1.3205	1.3817	1.4962	1.5701	4.490
	Improved	.9590	1.0159	1.1145	1.2252	1.2826	1.3281	1.4382	1.4593	3.2475
	Actual	.9559	.9987	1.0740	1.1171	-	-	-	-	-
Average Entry Size	Absolute	.2783	.2108	.1772	.1558	.1318	.1142	.1073	.0972	.0100
	Fast	.3212	.2546	.2287	.2088	.1886	.1727	.1662	.1570	.0449
	Improved	.3197	.2539	.2229	.2042	.1832	.1660	.1598	.1459	.0325
	Actual	.3186	.2497	.2148	.1862	-	-	-	-	-
Computer Time In Seconds	Fast and Absolute	.17	.21	.25	.30	.38	.42	.55	.60	47.0
	Improved	.11	.21	.23	.29	.35	.42	.53	.61	91.0
	Actual	.16	.24	1.11	15.01	-	-	-	-	-
%Average Error*	Fast	.81	1.99	6.49	12.16					
	Improved	.32	1.75	2.84	9.68					

*% Average Error = $\frac{\text{this solution - Actual}}{\text{Actual}}$

The Absolute Minimum is computed as the

$$\max\left(\sum_{i=1}^{n}\min_{j}(c_{ij}),\ \sum_{j=1}^{n}\min_{i}(c_{ij})\right).$$

The Actual Solution is found by enumeration. Both Improved Start and Fast Solution follow the algorithm of this paper except that Fast Solution starts with the Identity Matrix as the initial solution and Improved Start uses a combination of minimum search and elimination. Each of these programs will be listed in Appendix B. Absolute Minimum and Fast Solution were run as one program.

Figure 2 is a comparison of average entry size to order of matrix for all of the above routines.

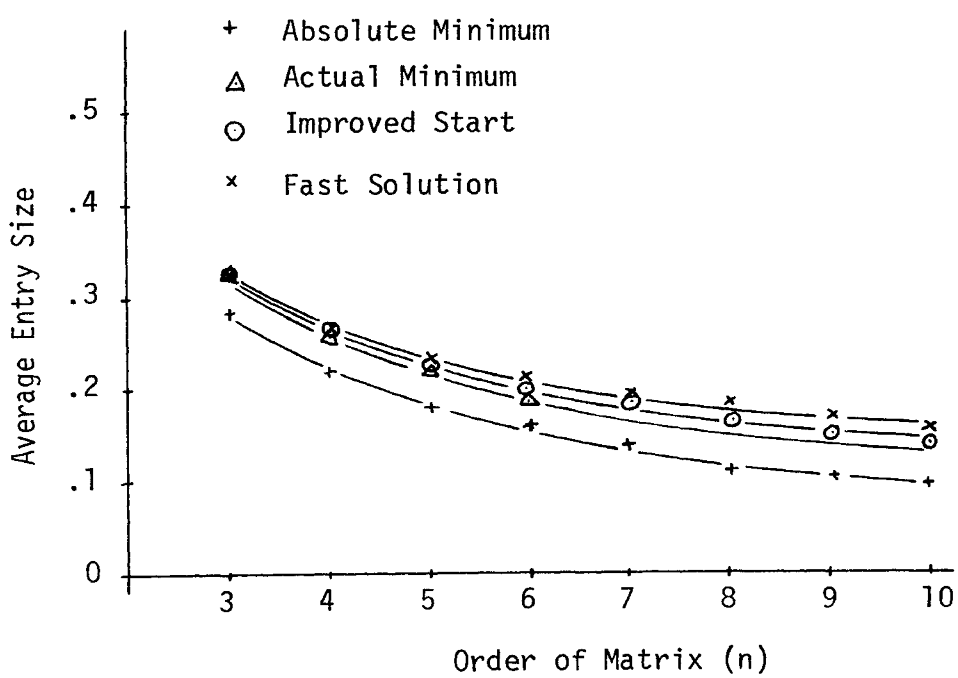


Figure 2 ENTRY SIZE

Note that Actual Minimum is projected beyond n = 6 because there were no results computed. Since 15 seconds of computer time were required to solve only one 6 x 6 as compared to 1.11 seconds for a 5 x 5, all enumeration runs for n>6 were abandoned. Their value was somewhat questionable compared to the computer time for n = 6. All of the data points illustrated involved 100 nth order matrices whose coefficients were generated by a uniformly distributed random number generator except Actual Solution data point n = 6 where only 36 matrices were used.

Note that the initial solution is significant even in this suboptimal technique. It becomes more significant as the order of the cost coefficient matrix increases. Examine the experimental runs for n = 100. This data alone indicates at lease a 27% error in the solution as found by Fast Solution compared only to Improved Start.

Figure 3 compares computer time in seconds to order of matrix for Improved Start and total enumeration (Actual Minimum). This figure illustrates that computer time for total enumeration is exponential, however, the algorithm presented here appears to be almost a linear function of order of matrix. In fact, one 100 x 100 was evaluated requiring 91.0 seconds of computer time. This is by no means linear, but this time is so small compared to what would have been required for enumeration that the cost comparisons for finding solutions is not realistic.

By the time publishers have edited many articles by very good writers their meaning and usefulness are almost obliterated. The value of the new development is either unsubstanciated by mathematical development or lacking in satisfactory examples or both. As the

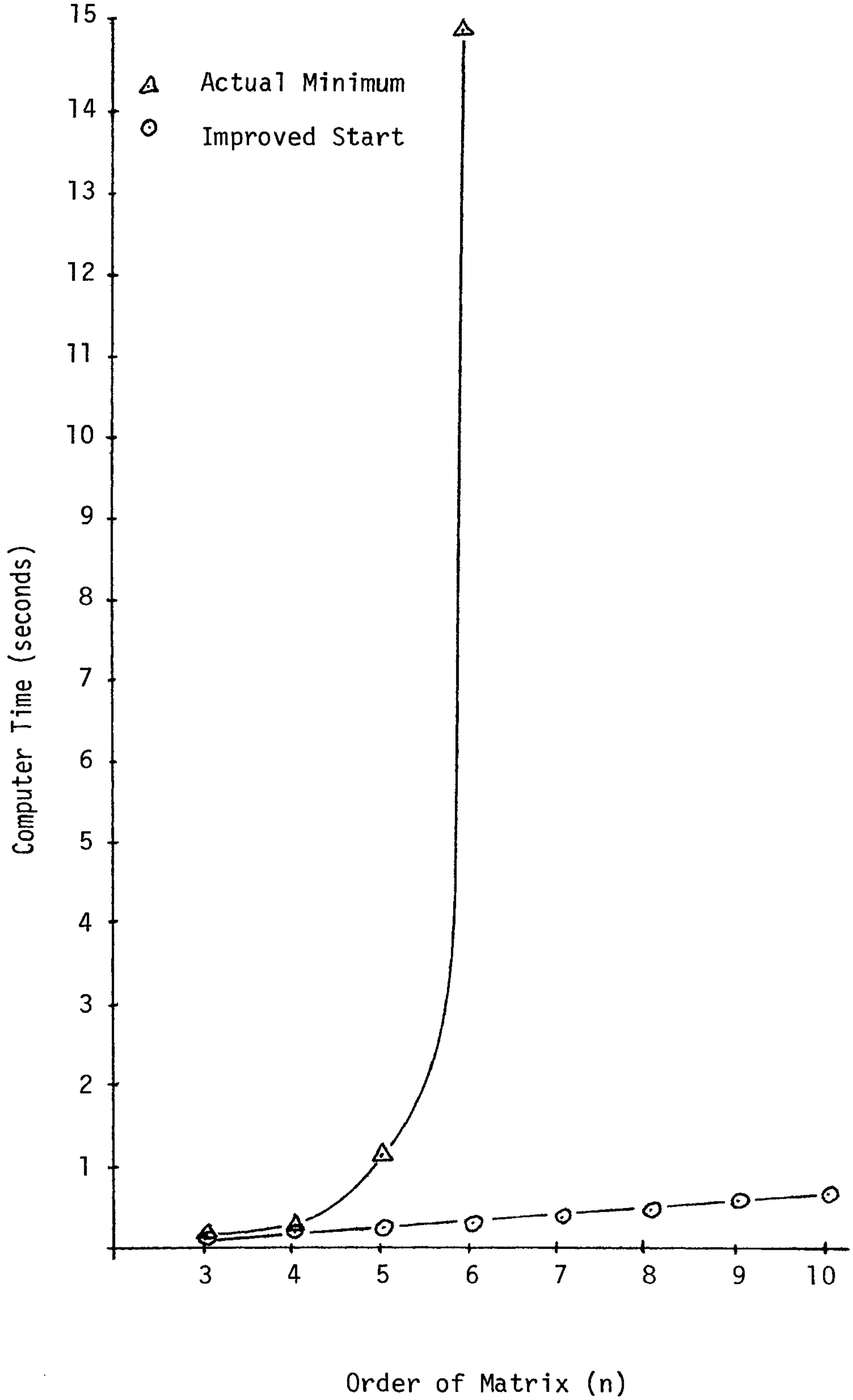


Figure 3 COMPUTER TIME

situation now stands, an Operations Research Analyst has a better chance to develop his own algorithms than of understanding many would-be, incomplete algorithms in the literature.

Branch and bound, although it is a good method for finding an exact solution, was not used for comparison purposes because it was not necessary to illustrate the desired points of the investigation.

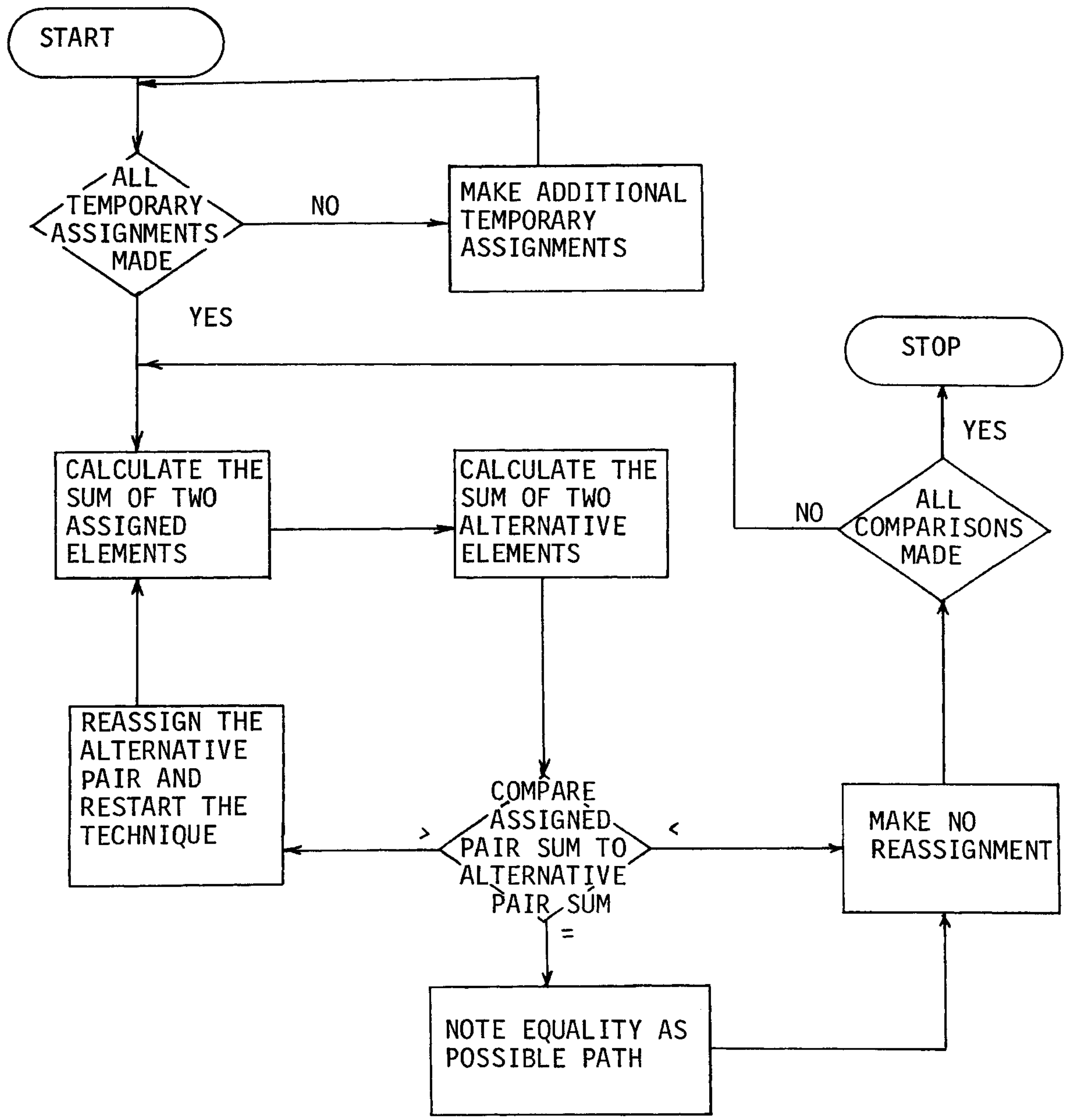
With respect to solving this problem by hand, some pairwise comparisons are not necessary because of previous comparisons or at least the additions are not necessary because one pair is obviously larger than the other. All of these things make it more usable.

In conclusion, the initial solution is most important when an exact optimum solution is required or when the order of the coefficient matrix is large. Otherwise, the initial solution should necessarily only be feasible. An exact optimum solution is only important when it can be obtained cheaply enough to warrant its use over a suboptimal, easily obtained, cheap solution. The method of this investigation gives such a suboptimal, easily obtained, cheap solution. Illustrations of its use are included to assist the reader in understanding when a cheap solution is a good solution. Other applications of suboptimal solutions are too numerous to mention. Perhaps other areas of Operations Research could benefit by a better understanding of the expense of obtaining a solution as opposed to the benefit of the solution. How good is the cure if the patient dies waiting for its application?

VI. APPENDICES

Appendix A

Flowchart of the suboptimal technique.



Appendix B

Computer Program Listings

1. Program Listing to solve by total enumeration.

```
SOLVER:  PROC OPTIONS(MAIN);  /*SOLVES BY ENUMERATION*/
    DCL  (IX,IY)  FIXED BINARY(31),  C(10,10);
    IX=28801;
    N=_;  /* ORDER OF MATRIX TO BE CONSIDERED */
    M=100; /* NO OF MATRICES TO BE INVESTIGATED */
    SUM=0; /* SUM OF SOLUTIONS */
    DO K=1 TO M;
    DO I = 1 TO N;
    DO J=1 TO N;
    CALL RANDU(IX,IY,R); /* CALL TO FORTRAN RANDOM NUMBER GENERATOR */
    IX=IY;
    C(I,J)=R;
    END; END;
    /* C MATRIX IS GENERATED */
    ANS=N;
LOOP:DO I1=1 TO N;
     DO I2=1 TO N;
     DO I3=1 TO N;  /* DO LOOPS REQUIRED FOR I4, I5, ... I10 */
     IF I1¬=I2 & I1¬=I3 & I2¬= I3 THEN
     DO;
     SANS=C(1,I1)+C(2,I2)+C(3,I3);
     IF SANS ANS THEN ANS=SANS;
     END;
     END LOOP;
     PUT EDIT('COST MATRIX',((C(I,J) DO I=1 TO N) DO J=1 TO N))
             (SKIP(3),X(10),A,(N)(SKIP,(N)(F(10,6))));
     PUT EDIT('THE SOLUTION BY ENUMERATION = ',ANS)
             (SKIP,X(5),A,F(10,4));
     SUM=SUM+ANS;
     END;
     SANS=SUM/M;
```

Appendix B (Continued)

1. Continued

```
PUT EDIT('THE AVERAGE TRUE SOLUTION =',SANS,'WITH M=',M)
        (SKIP(3),X(10),A,F(10,6),X(5),A,F(5));
END SOLVER;
```

2. Program Listing to find absolute minimum and fast solution.

```
MINIMUM:  PROC OPTIONS(MAIN);
     DCL C(1C,10),IA(10),IB(10,10),(IX,IY) FIXED BIN(31); IX=28801;
     /* N IS ORDER OF MATRIX.
        C IS MATRIX OF RANDOM NUMBERS UNIFORMLY DISTRIBUTED 0-1 INT.
        SRI  IS SMALLEST ELEMENT IN ITH ROW OF C.
        SCI  IS SMALLEST ELEMENT IN ITH COLUMN OF C.
        SROW IS SUM OF SMALLEST ROW ELEMENTS.
        SCOL IS SUM OF SMALLEST COLUMN ELEMENTS.
        SMIN IS LARGER OF SROW OR SCOL.
        SMIN IS THE ABSOLUTE MINIMUM OF POSSIBLE ASSIGNMENT. */
     N=_;  /* ORDER OF MATRIX TO BE CONSIDERED  */
     M=100; /* M IS NUMBER OF MATRICES TO BE INVESTIGATED */
     SUM=0; /* SUM IS SUM OF ABSOLUTE MINIMUMS */
     TSUM=0;  /* MOVE */
     DO K=1 TO M;
     DO I=1 TO N;
     DO J=1 TO N;
     CALL RANDU(IX,IY,R);  /* CALL TO FORTRAN RANDOM NUMBER GENERATOR */
     IX=IY;
     C(I,J)=R;
     END; END;
 /* C HAS BEEN GENERATED */
     SROW,SCOL=0;
     DO I=1 TO N; SRI, SCI=1.0;
     DO J=1 TO N;
     IF C(I,J) < SRI THEN SRI=C(I,J);
```

Appendix B (Continued)

2. Continued

```
     IF C(J,I) < SCI THEN SCI=C(J,I);
     END;
     SROW=SROW+SRI;
     SCOL=SCOL+SCI;
     END;
     IF SROW  >  SCOL THEN SMIN=SROW;
                      ELSE SMIN=SCOL;
     PUT EDIT(K,'TH MATRIX ABSOLUTE MINIMUM= ',SMIN)
             (SKIP,F(10),A,F(10,4));
     SUM=SUM+SMIN;
     PUT EDIT('THE COST MATRIX IS',((C(I,J) DO I=1 TO N) DO J=1 TO N))
             (SKIP, X(10),A,(N)(SKIP,(N)(F(10,6))));
     PUT SKIP;
     /*  USE IDENTITY MATRIX AS FIRST ASSIGNMENT */
     IB = 0;  /*SOLUTION MATRIX */
     DO J=1 TO N;
     IB(J,J)=1;
     IA(J)=J;
     END;  /*TEMPORARY ASSIGNMENT COMPLETED */
     /* TEST TEMPORARY ASSIGNMENT FOR ANY PAIRWISE IMPROVEMENT */
AGAIN:
     ISW=1;  /* COMPLETION SWITCH  */
     /* CHECK FOR PROPER SOLUTION */
AG:  DO I=1 TO N-1;
     DO J=I + 1 TO N;
     IF (C(I,IA(I))+C(J,IA(J)))>(C(J,IA(I))+C(I,IA(J)))
        THEN DO; IB(J,IA(I)),IB(I,IA(J))=1;
                 IB(I,IA(I)),IB(J,IA(J))=0;
     ISW=0;
        LL=IA(I);  IA(I)=IA(J);  IA(J)=LL;
             END;
     END; END;
```

Appendix B (Continued)

2. Continued

```
IF ISW=0 THEN GO TO AGAIN;
XSUM=0;
DO I=1 TO N; DO J=1 TO N;
XSUM=XSUM+C(I,J)*IB(I,J); END; END;
PUT EDIT('BY FAST SOLUTION MINIMUM= ',XSUM)(X(5),A,F(10,4));
TSUM=TSUM+XSUM;
PUT SKIP(3);
END;
ANS=SUM/M;
PUT EDIT('THE AVERAGE ABSOLUTE MINIMUM =',ANS,'WITH M=',M)
        (SKIP(3),X(10),A,F(10,6),X(1),A,F(5));
TANS=TSUM/M;
PUT EDIT('THE AVERAGE MINIMUM BY FAST SOLUTION =',TANS)
        (SKIP(2),X(20),A,F(10,6));
END MINIMUM;
```

3. Program Listing to find solution by improved start.

```
IMPROVE: PROC OPTIONS(MAIN);
DCL C(100,100), IA(100), IB(100,100), (IX,IY) FIXED BINARY(31);
    IX=28801;
N=___; /*ORDER OF COEFFICIENT MATRIX */
M=1;
SUM=0;
DO K=1 TO M;
DO I = 1 TO N;
DO J = 1 TO N;
CALL RANDU(IX,IY,R);
IX=IY;
C(I,J)=R;
END; END;
 IB=0; IA=0;
DO I=1 TO N;       TEST=1.1;
```

Appendix B (Continued)

3. Continued

```
DO J=1 TO N;
IF C(I,J)  TEST THEN DO; DO L=1 TO N;
    IF IA(L)=J THEN GO TO PASS; END;
TEST=C(I,J); JJ=J; END;
PASS: END;
   IA(I)=JJ;
IB(I,JJ)=1; END;
/* TEMPORARY ASSIGNMENT IS NOT NECESSARILY THE IDENTITY MATRIX */
AGAIN: ISW=1;
DO I=1 TO N-1;
DO J=I+1 TO N;
     IF (C(I,IA(I))+C(J,IA(J)))> (C(J,IA(I))+C(I,IA(J)))
     THEN DO;
IB(J,IA(I)),IB(I,IA(J))=1;
IB(I,IA(I)),IB(J,IA(J))=0;
  ISW=0; II=IA(I); IA(I)=IA(J); IA(J)=II;
END; END; END;
IF ISW=0 THEN GO TO AGAIN;
XSUM=0;
DO I=1 TO N;
DO J=1 TO N;
XSUM=XSUM+C(I,J)*IB(I,J);
END; END;
PUT EDIT('BY IMPROVED START  MINIMUM=',XSUM,'FOR MATRIX',K)
(SKIP(3),X(10),A,F(10,4),X(5),A,F(5));
     PUT EDIT('COST MATRIX')(PAGE,X(20),A);
PUT EDIT(C) (SKIP,10 F(10,5));
   PUT EDIT('SOLUTION MATRIX')(PAGE,X(20),A);
PUT EDIT(IB)(SKIP,10 F(10,5));
SUM=SUM+XSUM; END; ANS=SUM/M;
PUT EDIT('THE AVERAGE SOLUTION BY IMPROVED START=',ANS)
   (SKIP(3),X(10),A,F(10,6));  END IMPROVE;
```

VII. BIBLIOGRAPHY

1. Balas, Egon, "Solution of the Large Scale Transportation Problem Through Aggregation", Journal of the Operations Research Society of America, Vol. 13, No. 1, Jan.-Feb. 1965, pp 82-93.

2. Balas, Egon, "An Additive Algorithm for Solving Linear Programs With Zero-One Variables", Journal of the Operations Research Society of America, Vol. 13, No. 4, July-August 1965, pp 517-549.

3. Dantzig, George B., "Discrete-Variable Extremum Problems," Symposium on Extremum Problems, Journal of the Operations Research Society of America, Vol. 5, No. 2, April 1957, pp 266-277.

4. Efroymson, M.A., and Ray, T. L., "A Branch-Bound Algorithm for Plant Location," Journal of the Operations Research Society of America, Vol. 14, No. 3, May-June 1966, pp 361-368.

5. Gavett, J. W., and Plyter, Norman V., "The Optimal Assignment of Facilities to Locations by Branch and Bound", Journal of the Operations Research Society of America, Vol. 14, No. 2, March-April 1966, pp 210-232.

6. Glover, Fred, "A Multiphase-Dual Algorithm for the Zero-One Integer Programming Problem," Journal of the Operations Research Society of America, Vol. 13, No. 6, Nov.-Dec. 1965, pp. 879-919.

7. Kuhn, H. W., "The Hungarian Method for the Assignment Problem", Naval Research Logistics Quarterly, 2 (1955), pp 83-97.

8. Lawler, E. L., and Wood, D. E., "Branch-and-Bound Methods: A Survey", Journal of the Operations Research Society of America, Vol. 14, No. 4, July-Aug. 1966, pp 699-719.

9. Little, John D. C., Murty, Katta G., Sweeney, Dura W., and Karel, Caroline, "An Algorithm for the Traveling Salesman Problem," Journal of the Operations Research Society of America, Vol. 11, 1963, pp 972-989.

VII. BIBLIOGRAPHY (Continued)

10. Machol, Robert E., "An Application of the Assignment Problem", Journal of the Operations Research Society of America, Vol. 9, April 1961.

11. Manne, Allan S., "On the Job-Shop Scheduling Problem", Journal of the Operations Research Society of America, Vol. 8, No. 2, March-April 1960, pp. 219-223.

12. Minty, George J., "A Varient on the Shortest-Route Problem", Journal of the Society of Industrial and Applied Mathematics, Vol. 6, No. 6, Nov.-Dec. 1958, pp 882-883.

13. Munkres, James, "Algorithm for the Assignment and Transportation Problems," Journal of the Society of Industrial and Applied Mathematics, Vol. 5, No. 1, March, 1957, pp 32-38.

14. Murty, Katta G., "An Algorithm for Ranking All the Assignments in Order of Increasing Cost," Journal of the Operations Research Society of America, Vol. 16, No. 4, July-August 1968, pp 682-687.

15. Niederjohn, James A., "The Use of Row Values in Solution of the Transportation Problem," Letters to the Editor, Journal of the Operations Research Society of America, Vol. 8, No. 4, July-August 1960, pp 570-572.

16. Reiter, Stanley, and Sherman, Gordon, "Discrete Optimizing", Journal of the Society of Industrial and Applied Mathematics, Vol. 13, No. 3, Sept. 1965, pp 864-889.

17. Saaty, Thomas L., "Approximation to the Value of the Objective Function in Linear Programming by Method of Partition", Journal of the Operations Research Society of America, Vol. 4, January 1956.

18. Szwarc, Wlodgimierz, "The Initial Solution to the Transportation Problem", Letters to the Editor, Journal of the Operations Research Society of America, Vol. 8, No. 5, Sept.-Oct. 1960, pp. 727-729.

19. Yaspan, A., "On Finding a Maximal Assignment", Journal of the Operations Research Society of America, Vol. 14, No. 4, July-August, 1966, pp 646-651.

VIII. VITA

William Barry Gregory was born on January 21, 1943, in Carthage, Missouri where he also received his primary and secondary education. He received a Bachelor of Science degree in Mathematics from the University of Missouri-Rolla in May 1968.

He was married to his wife, Kay, in August, 1963 and they have a daughter named Kathleen, aged 5.

He was employed by McDonnell Aircraft Corporation for two years as a Technical Analyst and Computer Programmer. He has been employed by the University of Missouri-Rolla Computer Center since September, 1967, as a Scientific Programmer and Systems Analyst.

He has been enrolled in the Graduate School of the University of Missouri-Rolla since September, 1968.