W. Pierskalla

Canadian Operational Research Society Journal
Vol. 5, No. 2 July 1967.

19

THE TRI-SUBSTITUTION METHOD FOR THE THREE-DIMENSIONAL ASSIGNMENT PROBLEM*

WILLIAM P. PIERSKALLA
Case Institute of Technology, Cleveland, Ohio

Abstract

A tri-substitution method is proposed for obtaining a close-to-optimal (if not the optimal) solution to moderately large three-index assignment problems. The method inserts three new variables, each at value one, into the current basic feasible solution and at the same time assigns a value of zero to three of the old variables which previously had value one. In this way integer feasibility is maintained at each step of the method. Some other advantages of the method are that it is an additive method and it appears to work well on moderately large three-index assignment problems.

Résumé

Un algorithme permettant de trouver la solution du problème d'assignation dans un système à plusieurs dimensions est exposé. Cet algorithme suit la méthode du développement en faisceau avec embranchements et nœuds. On fait usage de deux sous—problèmes pour déterminer facilement les limites du problème principal d'assignation.
Toutefois, on peut se demander si l'algorithme proposé est vraiment une méthode efficace de solution pour certains problèmes complexes d'assignation dans plusieurs dimensions. On présente en plus, une méthode qui offre dans le cas de problèmes d'assignation tri-dimensionnelle de grandeur moyenne, une solution assez rapide et quasi-optimum (si non optimum).

Description of the Problem

In recent years much interest has concentrated on integer linear programing problems, and in particular several papers have focused on the zero-one integer LP problem. This paper is also concerned with the zero-one integer problem, but in regard to the special case of an optimal assignment in three dimensions. The three-dimensional assignment problem is the problem of assigning one item to one job at one point or interval in time in such a way as to minimize the total costs of the assignment. Mathematically, if it is assumed that $p \leq q \leq r$, then we want to:

$$\text{minimize } \Sigma_{i=1}^{p} \Sigma_{j=1}^{q} \Sigma_{k=1}^{r} c_{ijk} x_{ijk} \tag{1}$$

$$\text{subject to } \Sigma_{j=1}^{q} \Sigma_{k=1}^{r} x_{ijk} = 1, \quad \text{for } i = 1, \ldots, p \tag{2}$$

$$\Sigma_{i=1}^{p} \Sigma_{k=1}^{r} x_{ijk} \leq 1, \quad \text{for } j = 1, \ldots, q \tag{3}$$

$$\Sigma_{i=1}^{p} \Sigma_{j=1}^{q} x_{ijk} \leq 1, \quad \text{for } k = 1, \ldots, r \tag{4}$$

$$x_{ijk} = 0, 1. \quad \text{for all } i, j, k. \tag{5}$$

c_{ijk} is the cost of assigning item i to location j at time k. $x_{ijk} = 1$ means the i^{th} item is assigned to the j^{th} location at time k. $x_{ijk} = 0$ means the i^{th} item is

*Received December 14, 1966.

not assigned to location j at time k. Equality holds in the first set of constraints since it is presumed that a total assignment of the p items is required.

Some examples of how this type of problem might arise are:

(1) Consider a rolling mill scheduling problem in a steel plant. Hot steel ingots are formed in molds. If there is space available, the hot ingot can be placed in one of p available soaking pits. If space is not available, the hot ingot must wait (consequently it cools). Conversely, if no hot ingots are available for an unoccupied soaking pit, it is always possible to fill the pit with a cold ingot from an auxiliary pile. Thus we consider that there are r ingots of hot and cold variety available. Now the purpose of the soaking pits is to bring the ingot to a uniform temperature throughout. Any time after the ingot has reached the uniform temperature it may then go to the rolling mill for final processing. The problem is to find an optimal manner of scheduling the ingots into the soaking pit in order that the rolling mill has a minimum down time while waiting for ingots.

(2) Another example is the minimum cost scheduling of new capital investment projects into different possible physical locations over some time horizon. In particular a firm may wish to construct four new plants over the next six years. If ten sites are available for these plants then we could let $p = 4$ $q = 6$ and $r = 10$ to find the optimal construction schedule.

(3) A third example of a three-dimensional assignment problem would be the following: The army wishes to move new battalions of troops into certain strategic locations over the next six months. They want to deploy these troops in such a way as to maximize the defensive and/or offensive capability. If the number of months $\leq$ number of battalions $\leq$ number of possible locations, then in this model, number of months $= p$, number of battalions $= q$, and number of locations $= r$.

A more general formulation of this problem as a three dimensional transportation type problem has been considered by E. Schell[9] and K. B. Haley,[5] with the exception that they do not consider constraint (5), but rather $x_{ijk} \geq 0$. Since the polyhedron defined by (2)–(4) and $x_{ijk} \geq 0$ has in general a great number of non-integer extreme points, the approach taken by Schell and Haley will most often yield a non-integer solution (for large p, q, and r). A similar comment applies to the simplex method when (5) is replaced by $x_{ijk} \geq 0$.

In two very interesting papers Balas[1] and Glover[4] solve this 0-1 programing problem, as well as the general 0-1 integer programing problem, but because of the general approach taken in these two papers, it is possible to find alternative methods for highly structured problems such as we have here.

In this paper a method is presented which achieves a close-to-optimal (if not the optimal) solution in a reasonable amount of time for moderately large three-index assignment problems. This method is called the tri-substitution method.

The tri-substitution method works on a principle similar to the simplex method of linear programing. However, it differs from the simplex method in that we insert three new variables and we remove three variables at each step.

Furthermore, it is an additive method in the sense that at any step only additions and subtractions are performed.

Some of the advantages of the tri-substitution method have already been mentioned: it has solved some moderately large problems in a reasonable amount of time (more will be said about this later); it is additive—thus requiring simple programing steps; it always maintains primal feasibility, that is, (2), (3), (4), and (5) are satisfied at each step—hence the program may be stopped at any point, and the best current, feasible solution is at hand; and it provides a very good initial feasible solution to any optimal algorithm for (1) to (5).

The main disadvantage of the method is that a good general stopping rule has not been found. The method could find the optimal or near-optimal solution but would not know whether a better solution exists or where to find it. Although some particular stopping rules are available, they do not cover all possible cases. One such rule is to compute a lower bound l_1 for $\Sigma\Sigma\Sigma\, c_{ijk}x_{ijk}$. Thus if $\Sigma\Sigma\Sigma\, c_{ijk}x^*_{ijk} = l_1$ and if (2), (3), (4), and (5) are satisfied, then x^* is optimal. Another rule is the simple one that if all of the relative cost factors ~~simplex multipliers~~ have $z_t - c_t \leq 0$ for all $t = 1, \ldots, pqr$, then the current solution is optimal. Unfortunately it need not be the case that the lower bound l_1 is tight, nor that all $z_t - c_t \leq 0$ for the current solution to be optimal. A third rule is that if the linear programing solution $\bar{x}$ to (1), (2), (3), and (4) with $x_{ijk} \geq 0$ is known and the c_{ijk} are integers (if fractions we will scale them to integers first), then if x_I is the current feasible integer solution, and if $c \cdot (x_I - \bar{x}) < 1$, then x_I is optimal for (1)–(5).

Before going further with a more explicit description of the method we will state some interesting and pertinent results in lemma and theorem form. Proofs of these results are available in the appendix.

Lemma 1

(1) Feasible integer solutions to (2), (3), (4), and (5) exist.

(2) A vector x is a feasible integer solution if and only if (a) there are precisely p co-ordinates of x with value one and the remaining co-ordinates have value zero, and (b) the p elements with value one consist of x_{ijk}'s with the property that the i subscripts are a permutation of the integers $1, \ldots, p$, the j subscripts are a permutation of p of the integers $1, \ldots, q$, and the k subscripts are a permutation of p of the integers $1, \ldots, r$.

For example, the solution $x_{iii} = 1$ for all $i = 1, \ldots, p$ and $x_{ijk} = 0$ otherwise is a feasible integer solution.

Theorem 1

(1) An optimal solution to (1), (2), (3), (4), and (5) exists although not necessarily uniquely.

(2) An upper bound on $\Sigma\Sigma\Sigma\, c_{ijk}x_{ijk}$ is given by

$$M_1 = \Sigma_{i=1}^{p} \max_{k,j} (c_{ijk}). \tag{6}$$

(3) A lower bound on $\Sigma\Sigma\Sigma\, c_{ijk}x_{ijk}$ is given by (7)

$$m_1 = \Sigma_{i=1}^{p} \min_{k,j} (c_{ijk}).$$

(4) Another lower bound on $\Sigma\Sigma\Sigma\, c_{ijk}x_{ijk}$ is given by

$$\sigma_1 = \Sigma_{\mu=1}^{p} \bar{y}_\mu - \Sigma_{\mu=p+}^{p+q+r} \bar{y}_\mu,$$

where

$$\bar{y}_{p+q+k} = \max(0, -c_{pqk}), \qquad \text{for } k = 1, \ldots, r$$

$$\bar{y}_{p+j} = \max_{k=1,\ldots,r} (0, -c_{p,p+j,k} - \bar{y}_{p+q+k}), \quad \text{for } j = 1, \ldots, q-1$$

$$\bar{y}_{p+q} = 0,$$

$$\bar{y}_i = \min_{\substack{j=1,\ldots,q \\ k=1,\ldots,r}} (c_{i,j,k} + \bar{y}_{p+j} + \bar{y}_{p+q+k}) \quad \text{for } i = 1, \ldots, p.$$

In general, since we have not specified any conditions for the c_{ijk}, m_1 will differ from σ_1. Furthermore, many lower bounds of the m_1 or σ_1 variety can be formed by considering other feasible solutions to the dual problem generated by the primal problem where we do not consider the primal integer constraints. However, if an optimal solution to the dual problem is found (denote this optimal solution by d_1), then clearly $d_1 \geq m_1$ and $d_1 \geq \sigma_1$. Thus d_1 provides the best lower bound which can be achieved by considering the dual problem. For large problems the calculation of d_1 can be rather time consuming, whereas the calculation of m_1 or σ_1 is very fast. Thus we let l_1 be the best available lower bound which we have calculated, that is, $l_1 = \max(m_1, \sigma_1, \ldots)$. If d_1 has been calculated then $l_1 = d_1$.

The Tri-Substitution Method

In this method a simplex-like approach is used. However, at every change of basis three new variables are introduced and three old variables removed, all of which have value one. In this manner it is possible to maintain primal feasibility. As was shown above, the problem given by (1) to (5) has an optimal solution, and any feasible solution must have precisely p of the x_{ijk}'s equal to one. Furthermore, these x_{ijk}'s which equal one must have the i subscripts a permutation of the integers $1, \ldots, p$, the j subscripts a permutation of p of the integers $1, \ldots, q$, and the k subscripts a permutation of p of the integers $1, \ldots, r$.

Equations (2) to (5) can be written in more general form which will occasionally be convenient for describing the steps of the method:

$$\text{minimize } c \cdot x \tag{1}$$

subject to

$$Ax + Iv = 1 \tag{8}$$

$$x_{ijk} = 0, 1 \tag{9}$$

$$w_i \geq 0, \tag{10}$$

where $v = (0, \ldots, 0, w_1, \ldots, w_{q+r})'$ is a column vector whose first p elements are zero, and the w_i are slack variables. The elements of the matrix A will be denoted $a_{mn}(m = 1, \ldots, p + q + r; n = 1, \ldots, pqr)$.

Lemma 2

If the variables are ordered $x_{111}, x_{112}, \ldots, x_{11r}, x_{121}, x_{122}, \ldots, x_{12r}, \ldots, x_{1q1}, \ldots, x_{1qr}, x_{211}, x_{212}, \ldots, x_{21r}, \ldots, x_{p11}, x_{p12}, \ldots, x_{p1r}, x_{p21}, \ldots, x_{pqr}$, then there is exactly one basic x_{ijk} variable with value one in each group of successive columns 1 to qr, $qr + 1$ to $2qr$, $2qr + 1$ to $3qr, \ldots, (p - 1)qr + 1$ to pqr in any basic feasible solution to (8) to (10).

In addition to the detailed description of the method below, a flow-chart of the main aspects of the method is presented on the following page.

Tri-Substitution Method:

(1) Start with the initial basic feasible solution, x_B, given by $x_{iii} = 1$ for $i = 1, \ldots, p$ and $x_{ijk} = 0$ otherwise. Furthermore, we will pivot on the first 1 we come to in each of the columns for the initial basic solution starting with x_{111} and continuing to the right in the tableau A. [Actually tableau A need never be changed in the entire method. It is only necessary to compute B^{-1} which can be done easily. In fact it is never necessary to have A stored in the computer at any time.] Thus we have a basic variable at the one level in each of the first p rows. In addition, the elements a_{mn} of A for $m = 1, \ldots, p - 1$ and $n = mqr + 1, \ldots, pqr$ are all zero and remain zero in the tableau after the initial basis is formed. If $c_B \cdot x_B = l_1$ or if any of the other stopping conditions given previously are met then go to step 8.

B^{-1} is the inverse of the current basis matrix, c_B are the elements of c related to the basis B.

$z = c_B B^{-1} A$

(2) Use the standard LP simplex method to determine the incoming $x_{i_0j_0k_0}$: that is, since the problem is to minimize $c \cdot x$, we choose the $x_{i_0j_0k_0}$ to come in which has $z_{i_0j_0k_0} - c_{i_0j_0k_0} > 0$, where $z = c_B B^{-1}A$. If all $z_{ijk} - c_{ijk} \leq 0$, go to step 8.

(3) The $x_{i_0j_0k_0}$ chosen in step 2 affects at most three x_{ijk}'s in the current basic solution which have the value one. Denote the current basic solution by x_B, that is, the $x_{i_0j_0k_0}$ coming in affects

$$x_{i_0j_1k_1} = 1 \text{ for some } j_1 \text{ and } k_1, \text{ where } x_{i_0j_1k_1} \in x_B$$

$$x_{i_1j_0k_2} = 1 \text{ for some } i_1 \text{ and } k_2, \text{ where } x_{i_1j_0k_2} \in x_B$$

$$x_{i_2j_2k_0} = 1 \text{ for some } i_2 \text{ and } j_2, \text{ where } x_{i_2j_2k_0} \in x_B$$

It is possible that two or three of these x_{ijk}'s could be the same. For example:

Three affected		*Two affected*	*One affected*
x_{345} affects		x_{345} affects	x_{345} affects
$x_{324} = 1$	in current x_B	$x_{242} = 1$	$x_{316} = 1$
$x_{641} = 1$		$x_{371} = 1$	
$x_{275} = 1$			

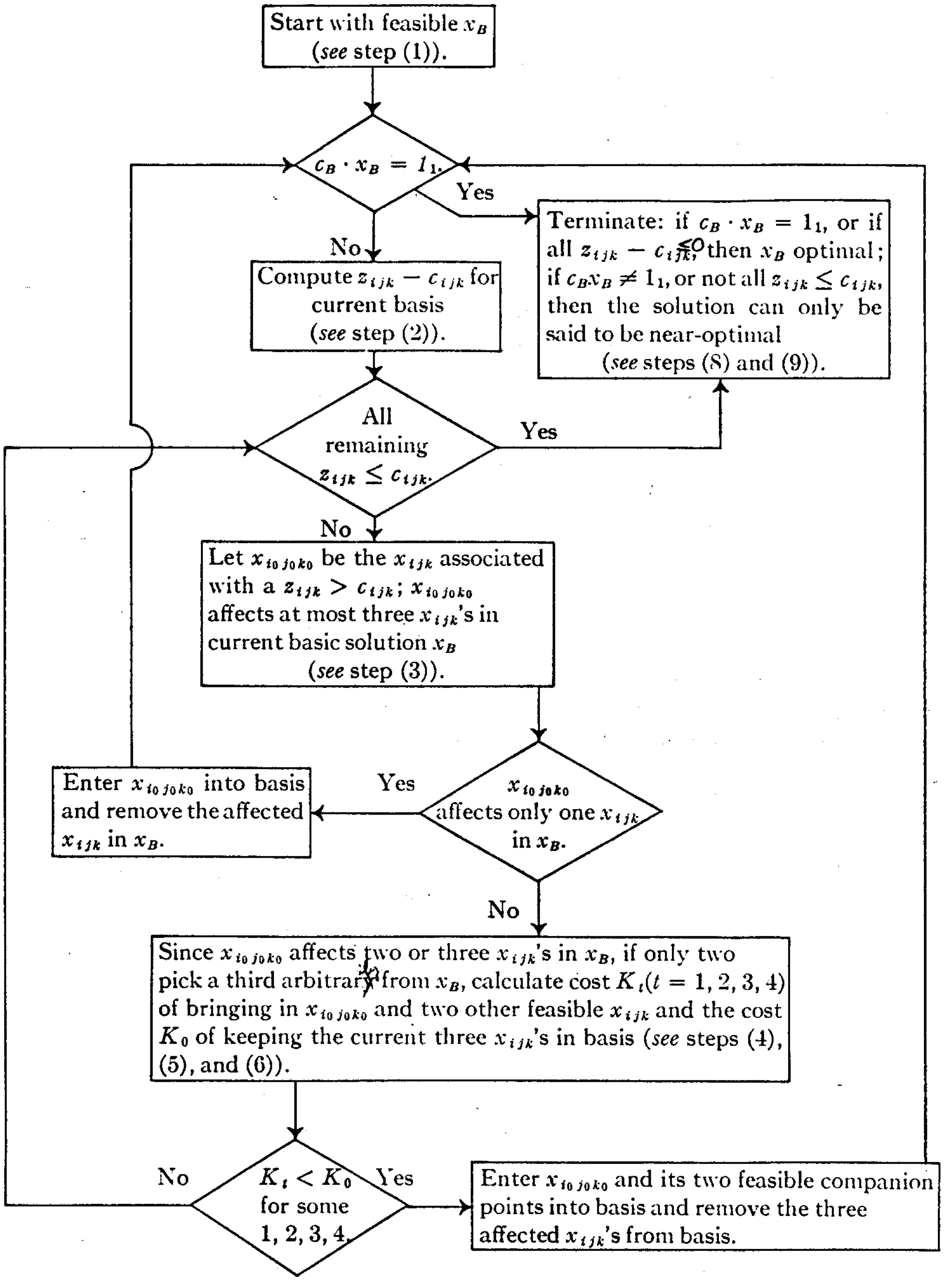
Start with feasible x_B (*see* step (1)).
$c_B \cdot x_B = 1_1$.
Yes
No
Terminate: if $c_B \cdot x_B = 1_1$, or if all $z_{ijk} - c_{ijk} \leq 0$ then x_B optimal; if $c_B x_B \neq 1_1$, or not all $z_{ijk} \leq c_{ijk}$, then the solution can only be said to be near-optimal (*see* steps (8) and (9)).
Compute $z_{ijk} - c_{ijk}$ for current basis (*see* step (2)).
All remaining $z_{ijk} \leq c_{ijk}$.
Yes
No
Let $x_{i_0 j_0 k_0}$ be the x_{ijk} associated with a $z_{ijk} > c_{ijk}$; $x_{i_0 j_0 k_0}$ affects at most three x_{ijk}'s in current basic solution x_B (*see* step (3)).
Enter $x_{i_0 j_0 k_0}$ into basis and remove the affected x_{ijk} in x_B.
Yes
$x_{i_0 j_0 k_0}$ affects only one x_{ijk} in x_B.
No
Since $x_{i_0 j_0 k_0}$ affects two or three x_{ijk}'s in x_B, if only two pick a third arbitrary from x_B, calculate cost $K_t (t = 1, 2, 3, 4)$ of bringing in $x_{i_0 j_0 k_0}$ and two other feasible x_{ijk} and the cost K_0 of keeping the current three x_{ijk}'s in basis (*see* steps (4), (5), and (6)).
No
$K_t < K_0$ for some 1, 2, 3, 4.
Yes
Enter $x_{i_0 j_0 k_0}$ and its two feasible companion points into basis and remove the three affected x_{ijk}'s from basis.

FIG. 1. Flow chart of the tri-substitution method.

In the above example where two of the variables are affected we are saying that there is no variable in the current basic solution with its k index position value $k = 5$. In the "one affected" case there are no basic variables in the current basic solution with the j and k index position values $j = 4$ and $k = 5$.

Now if $x_{i_0 j_0 k_0}$ affects only one $x_{i_0 jk}$ in the current basic solution, then replace that $x_{i_0 jk} = 1$ by $x_{i_0 j_0 k_0} = 1$ and we obtain a strict decrease in the objective function. Make the pivot for this replacement on the first one in the $i_0 j_0 k_0$ column~~s~~. If three x_{ijk}'s are affected go to step 4; if only two are affected go to step 6.

(4) $x_{i_0 j_0 k_0}$ and $x_{i_0 j_1 k_1}$, $x_{i_1 j_0 k_1}$, and $x_{i_2 j_2 k_0}$ determine a cube of 27 variables of the original pqr variables. This cube is all of the 27 different orderings of the indices i_0, i_1, i_2, j_0, j_1, j_2, k_0, k_1, k_2, for example, the cube associated with $x_{i_0 j_0 k_0} = x_{345}$ and x_{324}, x_{641}, x_{275} is

$$x_{324}, x_{321}, x_{325}, x_{344}, x_{341}, x_{345}, x_{374}, x_{371}, x_{375}$$

$$x_{624}, x_{621}, x_{625}, x_{644}, x_{641}, x_{645}, x_{674}, x_{671}, x_{675}$$

$$x_{224}, x_{221}, x_{225}, x_{244}, x_{241}, x_{245}, x_{274}, x_{271}, x_{275}.$$

Go to step 5.

(5) Now consider all possible paths through this cube such that $x_{i_0 j_0 k_0}$ is included in each path (a path consists of any three non-coplaner x_{ijk}'s). There are exactly four such paths:

$$\text{path } 1 = (x_{i_0 j_0 k_0}, x_{i_1 j_1 k_1}, x_{i_2 j_2 k_2}),$$
$$\text{path } 2 = (x_{i_0 j_0 k_0}, x_{i_1 j_1 k_2}, x_{i_2 j_2 k_1}),$$
$$\text{path } 3 = (x_{i_0 j_0 k_0}, x_{i_1 j_2 k_1}, x_{i_2 j_1 k_2}),$$
$$\text{path } 4 = (x_{i_0 j_0 k_0}, x_{i_1 j_2 k_2}, x_{i_2 j_1 k_1}).$$

Calculate $c_{i_0 j_0 k_0} + c_{i_1 j \gamma k \delta} + c_{i_2 j \alpha k \beta} = K_t$, $t = 1, 2, 3, 4$ for each path t above. If for any K_t we have

$$K_t < c_{i_0 j_1 k_1} + c_{i_1 j_0 k_2} + c_{i_2 j_2 k_0} = K_0,$$

then introduce all the variables in path t for the three variables $x_{i_0 j_1 k_1}$, $x_{i_1 j_0 k_2}$, $x_{i_2 j_2 k_0}$. However, introduce the triple by pivoting on the first 1 we come to in each of the columns for the triple. Go to step 8. If none of the $K_t < K_0$, then go to step 7.

Lemma 3

A basic variable at the one level will always appear in each of the first p rows of the tableau. Furthermore, the a_{mn} of A for $m = 1, \ldots, p - 1$ and $n = mqr + 1, \ldots, pqr$ and for $m = 2, \ldots, p$ and $n = 1, \ldots, (m - 1)qr$ remain zero in each tableau after each pivot operation above.

In addition we can state:

Lemma 4

If a triple is replaced as above, (1) the new variables will each have value one and (2) the objective function will change in value by an amount which is the sum of the relative cost factors for the new variables.

(6) Since only two x_{ijk}'s determined by $x_{i_0 j_0 k_0}$ are affected, then choose any other $x_{ijk} \epsilon x_B$ such that $x_{ijk} = 1$ and form the cube of 27 variables formed by these four x_{ijk}'s. As in step 5, calculate K_t and determine whether $K_t < K_0$. If yes, then remove the three old x_{ijk}'s and insert the new x_{ijk}'s in the manner described in step 5 and go to step 8. If no, then choose a different $x_{ijk} \epsilon x_B$ with $x_{ijk} = 1$ and test the new cube for an improved path. Continue this process until an improved path (i.e., $K_t < K_0$) is found or until all of the $p - 2 x_{ijk}$'s ϵx_B with $x_{ijk} = 1$ have been exhausted.

NOTE: In this step the $x_{i_0 j_0 k_0}$ and its two x_{ijk}'s belong to each cube tested. If an improved path has been found, remove the three old variables and insert the three new variables in the manner described in step 5 and go to step 8. If not, then go to step 7.

(7) Now choose the next x_{ijk} which satisfies $z_{ijk} - c_{ijk} > 0$ and go to step 3. If all z_{ijk}'s with $z_{ijk} - c_{ijk} > 0$ have been exhausted and no improvement in the objective function has been found, then go to step 9.

Lemma 5

If there exists a change of variables which leads to an improvement in the objective function at any iteration based on steps (3), (5), or (6) above, then this set of variables making the improvement may be found by considering only the $z_{ijk} - c_{ijk} > 0$.

(8) Let x_B now denote the new basic feasible solution. If $c_B \cdot x_B = l_1$, then the new basic feasible solution is optimal. If all $z_{ijk} - c_{ijk} \leq 0$, then the new basic feasible solution is optimal. If $c_B \cdot x_B > l_1$ and if some $z_{ijk} - x_{ijk} > 0$ then go to step 2.

(9) By entering step 9, we are saying that there does not exist any path of triples which will lead to a strict improvement in the objective function. At this stage it is obvious that if we examined all quadruple, quintuple, . . . , p-tuple replacements that we would find a replacement which would lead to the optimal solution. However, examining all such replacements would defeat the purpose of rapid computation, and in fact the computations would go up at least factorially with p.

It should be mentioned that it is not a sufficient condition for the optimality of x that all $K_t > K_0$ for all possible triples which may enter at the one level at the current solution point. A simple counter example exhibits the property that all $K_t > K_0$ for these triples and x is not optimal. Let $p = q = r = 4$ and let $c_{111} = 0$, $c_{123} = c_{234} = c_{341} = 1$, $c_{333} = c_{444} = 2$, $c_{322} = c_{342} = c_{442} = c_{422} = 3$, $c_{ij1} = C_{ij2} = 2$ otherwise and $c_{ij3} = c_{ij4} = 3$ otherwise. Then starting with the initial feasible solution $x_{111} = x_{222} = x_{333} = x_{444} = 1$, $x_{ijk} = 0$ otherwise, we obtain $c \cdot x = 6$ and all K_t found by the above algorithm has $K_t > K_0$ however, the optimal solution x^* is given by $x^*_{123} = x^*_{234} = x^*_{341} = x^*_{412} = 1$, $x^*_{ijk} = 0$ otherwise and $c \cdot x^* = 5$.

SOME COMPUTATIONAL RESULTS

In applying this method to various problems using a Univac 1107, we obtained the following results:

TABLE 1

Problem number	p	q	r	Number of variables	Number of equations	Solution achieved by this method	Optimal solution	Machine* running time (seconds)
1	4	4	4	64	12	4	4	15
2	5	5	5	125	15	5	5	27
3	6	6	6	216	18	6	6	62
4	7	7	7	343	21	7	7	125
5	8	8	8	512	24	8	8	261
6	5	5	5	125	15	5	4	20

*Does not contain the compilation time of approximately 24 seconds.

We are currently in the process of testing this method on large problems where p, q, and r are not equal. In addition we have formulated an algorithm for achieving the optimal solution to (1) to (5) and even for the more general multi-index assignment problem. Since this algorithm is still untested computationally, it will be presented at a later date.

Appendix

Proofs

Lemma 1 part (1) is obvious: part (2) ($\Rightarrow$). Let x be a feasible solution to (2) to (5). Since $\Sigma_j \Sigma_k x_{ijk} = 1 \, \forall \, i = 1, \ldots, p$, then upon adding these p equality constraints $\Sigma_{i=1}^{p} \Sigma_{j=1}^{q} \Sigma_{k=1}^{r} x_{ijk} = p$. But $x_{ijk} = 0, 1 \, \forall \, i, j, k$, thus there are exactly px_{ijk}'s equal to one. Now assume that the x_{ijk}'s which equal one do not satisfy part (b). Thus there is at least one $x_{ijk} = 1$ with at least one common index element with another $x_{ijk} = 1$, say $x_{i_0 j_0 k_0} = x_{i_1 j_1 k_0} = 1$. But by (2) we must have $\Sigma_{i=1}^{p} \Sigma_{j=1}^{q} x_{ijk_0} \leq 1$ since x is feasible; thus we have a contradiction since $\Sigma_{i=1}^{p} \Sigma_{j=1}^{q} x_{ijk_0} \geq 2$. Clearly (5) is satisfied, and since each inequality of (2) to (4) can contain at most one $x_{ijk} = 1$ and each equality must contain one $x_{ijk} = 1$ because of the permuting of the indices, then (2) to (4) hold. Hence by definition x is feasible.

Theorem 1 We first prove part (3).

Part (3) $\Sigma_{i=1}^{p} \Sigma_{j=1}^{q} \Sigma_{k=1}^{r} c_{ijk}x_{ijk} \geq \Sigma_{i=1}^{p} \min \{c_{ijk}\} \Sigma_{j=1}^{q} \Sigma_{k=1}^{r} x_{ijk} = \Sigma_{i=1}^{p} \min \{c_{ijk}\} = m_1$

Part (1) This part follows immediately from part (3) above and lemma 1.

Part (2) This proof is *mutatis mutandis* the same as part (3) above.

Part (4) The dual of the primal problem where we omit the integer constraints and use instead $x_{ijk} \geq 0$ is given by: max $\Sigma_{i=1}^{p} y_i - \Sigma_{i=p+1}^{p+q+r} y_i$ subject to $y_i - y_{p+j} - y_{p+q+k} \leq c_{ijk}$ for all $i = 1, \ldots, p, j = 1, \ldots, q; k = 1, \ldots, r, y_i$ unconstrained for $i = 1, \ldots, p, y_i \geq 0$ for $i = p + 1, \ldots, p + q + r$.

Now the solution $\{\bar{y}\}$ given in part (4) of theorem 1 satisfies these dual constraints since $\bar{y}_i - \bar{y}_{p+j} - \bar{y}_{p+q+k} \leq c_{ijk} + \bar{y}_{p+j} + \bar{y}_{p+q+k} - \bar{y}_{p+j} - \bar{y}_{p+q+k} = c_{ijk}$. Hence this solution is feasible for this dual problem. Now from linear programing if $\bar{x}$ is optimal for the primal LP problem and if x^* is optimal for (1) to (5) then $cx^* \geq c\bar{x} \geq \sigma_1$.

Lemma 2

This result follows directly from the fact that a basic feasible solution has the property that for all x_{ijk}'s with value one, the i position index numbers are a permutation of the integers *1*, . . . , p.

Lemma 3

This result is obvious from equation (2) and the statement given in the proof of lemma 1 above.

Lemma 4

Part (1) follows immediately from lemma 3 since there are no replacement pivot operations in the first p rows of A because the $a_{mn} \epsilon A$ for $m = 1, \ldots, p - 1$ and $n = mqr + 1, \ldots, pqr$ and for $m = 2, \ldots, p$ and $n = 1, \ldots, (m - 1)qr$ are all zero. Therefore the first p co-ordinates of the resource vector remain at the 1 level at all times in the tri-substitution method, and the new variables are brought into the first p rows.

Part (2) follows from the fact that the slack variables w_t have zero cost in the objective function, and since all of the slacks remain in every feasible basis, the relative cost factors for a given feasible basic solution $x_{i_1 j_1 k_1}, \ldots, x_{i_p j_p k_p}$ are merely $z_{ijk} - c_{ijk} = c_{i_t j_t k_t} - c_{ijk}$. Now if $x_{i_1 j_1 k_1}, x_{i_2 j_2 k_2}$ and $x_{i_3 j_3 k_3}$ are replaced by $x_{i_1 u_1 v_1}, x_{i_2 u_2 v_2}$ and $x_{i_3 u_3 v_3}$ then the change in the objective function is $c_{i_1 j_1 k_1} x_{i_1 j_1 k_1} + c_{i_2 j_2 k_2} x_{i_2 j_2 k_2} + c_{i_3 j_3 k_3} x_{i_3 j_3 k_3} - c_{i_1 u_1 v_1} x_{i_1 u_1 v_1} - c_{i_2 u_2 v_2} x_{i_2 u_2 v_2} - c_{i_3 u_3 v_3} x_{i_3 u_3 v_3} = c_{i_1 j_1 k_1} - c_{i_1 u_1 v_1} + c_{i_2 j_2 k_2} - c_{i_2 u_2 v_2} + c_{i_3 j_3 k_3} - c_{i_3 u_3 v_3}$, which is just the sum of the relative cost factors for $x_{i_1 u_1 v_1}, x_{i_2 u_2 v_2}$, and $x_{i_3 u_3 v_3}$.

Lemma 5

From lemma 4 above we merely add the relative cost factors. Thus, if none of them in a change of basis being considered are >0, then the change of basis could not lead to an improvement. Therefore, at least one of them must be >0, but then that change of basis would have been considered when that factor >0 was considered; hence it is not necessary to consider all possible changes of basis for relative cost factors ≤ 0.

References

(1) Balas, Egon, "An additive algorithm for solving linear programming with zero-one variables," Operations Research, vol. 13, no. 0, 1965, 517–45.

(2) Balinski, M. L., "Integer programming: Methods, uses, computation," Management Science, vol. 12, no. 3, Nov., 1965, 253–313.

(3) Geoffrion, A. M., "Integer programming by implicit enumeration and Balas' method," Rand Corporation Memorandum RM-4783-PR, Feb., 1966.

(4) Glover, Fred, "A multiphase-dual algorithm for the zero-one integer programming problem," Operations Research, vol. 13, no. 0, 1965, 879–919.

(5) Haley, K. B., "The existence of a solution to the multi-index problem," Operations Research Quarterly, vol. 16, no. 4, Dec., 1965, 471–74.

(6) ———, "The multi-index problem," Operations Research, vol. 11, no. 0, 1963, 368–79.

(7) ———, "The solid transportation problem," Operations Research, vol. 10, no. 0, 1962, 448–63.

(8) Little, J. D. C., K. G. Murty, D. W. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem," Operations Research, vol. 11, no. 0, 1963, 972–89.

(9) Schell, E., "Distribution of a product by several properties," Proceedings of the Second Symposium in Linear Programming, Washington, DC, January, 27–29, 1955, vols. 1–2, 615–42.