KM 11(1983)
pag 78 - 98

A dual method for the Assignment Problem

J.N.M. van Loon
Koninklijke Militaire Academie
Breda

Abstract

The linear assignment problem can be solved by finding the row- and column reduction constants such that their sum is maximal. We do this with the aid of the simplex method applied to the dual problem. For the administration of the process we need only a straightforward assignment tableau format, containing the reduced costs. Results of application of the algorithm to some small problems are compared with the behaviour of the Hungarian and the Tomizawa approach.

1. Introduction

We consider the linear assignment problem:

$$\text{Min} \sum_{ij}\sum c_{ij}x_{ij}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{ij}=1 \qquad i=1,\ldots,n$$

(1) $$\sum_{i=1}^{n} x_{ij}=1 \qquad j=1,\ldots,n$$

$$x_{ij}\geqslant 0 \qquad i=1,\ldots,n;\ j=1,\ldots,n$$

The 0-1 conditions on the variables x_{ij} can be replaced by non-negativity constraints, because every basic feasible solution of (1) automatically has this property. For our purpose we need the dual formulation of problem (1):

$$\text{Max} \sum_{i=1}^{n} u_i + \sum_{j=1}^{n} v_j$$

(2) $$\text{s.t. } u_i + v_j \leqslant c_{ij} \qquad i=1,\ldots,n;\ j=1,\ldots,n$$

$$u_i,\ v_j \text{ unrestricted!}$$

We will show that this dual problem can be solved by the simplexmethod, using only an assignment tableau format for updating the "reduced-cost"-matrix. This matrix can function as "right hand side-column" containing the values of the basic feasible solution. Also the non-basic variables of the corresponding simplex tableau are indicated in the assignment tableau. The relative costs of the non-basic variables can be determined from primal equations and the appropriate pivotcolumn is generated in an easy way by using "reduction" arguments.

2. Row- and columnreduction

Most of the efficient algorithms for solving the assignment problem make use of the following theorem:

Theorem 1: The optimal solution of the linear assignment problem does not change under addition of a constant to each element of a row or column of the cost matrix.

Proof: Suppose $c'_{ij}=c_{ij}+c$ for $j=1,\ldots,n$ in row i of the cost matrix. Then the objective function is:
$\mathrm{Min}[\sum_{ij}\sum c_{ij}x_{ij} + c\sum_j x_{ij}] = \mathrm{Min} \sum_{ij}\sum c_{ij}x_{ij} + c$, because

$\sum_j x_{ij}=1$ should hold for every feasible solution.

Throughout this study we will use Th. 1 in the following order: 1.Reduce all costs c_{ij} in row i with

(3) $$u_i = \mathrm{Min}_j \{c_{ij} | j=1,\ldots,n\} \text{ for } i=1,\ldots,n$$

2.Next diminish all reduced costs in column j with

(4) $$v_j = \mathrm{Min}_i \{c_{ij}-u_i | i=1,\ldots,n\} \text{ for } j=1,\ldots,n$$

Example 1 As an illustration we use the example from [2]:
Cost matrix:

$$\{c_{ij}\} = \begin{bmatrix} 7 & 12 & 9 & 11 & 5 \\ 5 & 10 & 7 & 8 & 12 \\ 14 & 15 & 13 & 12 & 8 \\ 8 & 13 & 11 & 14 & 7 \\ 10 & 9 & 7 & 6 & 13 \end{bmatrix}$$

Table 1.

after reduction:

						u_i	Δu_i
	2	4	3	6	$\underline{0}$	5	
	$\underline{0}$	2	1	3	7	5	
	6	4	4	4	$\underline{0}$	8	
$\bar{c}_{ij}$:	1	3	3	7	$\underline{0}$	7	
	4	$\underline{0}$	$\underline{0}$	$\underline{0}$	7	6	-1
v_j	$\underline{0}$	3	1	$\underline{0}$	$\underline{0}$	35	
Δv_j		1	1	1			2

Table 2

The numbers u_i and v_j represent a basic feasible solution of the dual problem (2). The corresponding values of the slack variables are:

(5) $$y_{ij} = \bar{c}_{ij} = c_{ij} - u_i - v_j \geqslant 0 \quad i=1,\dots,n;\ j=1,\dots,n \quad ^{x)}$$

and the objective value: $\sum_i u_i + \sum_j v_j = 35$

The 0-elements corresponding to non-basic variables will be underlined in the assignment tableau. The elements $v_j = 0$ are included whenever the cells where $\min_i \{c_{ij} - u_i\}$ and $\min_j \{c_{ij}\}$ coincide. So in total there are 2n non-basic variables. The remaining numbers $\bar{c}_{ij}$ and u_i, v_j are the values of the basic variables in the dual simplex tableau.

In Table 3 and 4 resp. the initial dual simplex tableau and the tableau corresponding to the solution after reduction in Ex. 1. are given. In fact 7 simplex iterations are needed.

x) For ease of notation we use y_{ij}, referring to the corresponding primal variables x_{ij}.

	$-u_1$	$-u_2$	$-u_3$	$-u_4$	$-u_5$	$-v_1$	$-v_2$	$-v_3$	$-v_4$	$-v_5$	
	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
y_{11}	1					1					7
y_{12}	1						1				12
.	1							1			9
.	1								1		11
.	1									1	5
y_{21}		1				1					5
.		1					1				10
.		1						1			7
.		1							1		8
.		1								1	12
y_{31}			1			1					14
.			1				1				15
.			1					1			13
.			1						1		12
.			1							1	8
y_{41}				1		1					8
.				1			1				13
.				1				1			11
.				1					1		14
.				1						1	7
y_{51}					1	1					10
.					1		1				9
.					1			1			7
.					1				1		6
y_{55}					1					1	13

Table 3

	$-y_{15}$	$-y_{21}$	$-y_{35}$	$-y_{45}$	$-y_{54}$	$-v_1$	$-y_{52}$	$-y_{53}$	$-v_4$	$-v_5$	
	1	1	1	1	-1	0	1	1	-2	2	35
y_{11}	-1					1				-1	2
y_{12}	-1				1		-1		1	-1	4
y_{13}	-1				1			-1	1	-1	3
y_{14}	-1								1	-1	6
u_1	1									1	5
u_2		1				1					5
y_{22}		-1			1	-1	-1		1		2
y_{23}		-1			1	-1		-1	1		1
y_{24}		-1				-1			1		3
y_{25}		-1				-1				1	7
y_{31}			-1			1				-1	6
y_{32}			-1		1		-1		1	-1	4
y_{33}			-1		1			-1	1	-1	4
y_{34}			-1						1	-1	4
u_3			1							1	8
y_{41}				-1		1				-1	1
y_{42}				-1	1		-1		1	-1	3
y_{43}				-1	1			-1	1	-1	3
y_{44}				-1					1	-1	7
u_4				1						1	7
y_{51}					-1	1			-1		4
v_2					-1		1		-1		3
v_3					-1			1	-1		1
u_5					1				1		6
y_{55}					-1				-1	1	7

Table 4

The cost matrix $\{\bar{c}_{ij}\}$ can be reduced further by changing u_i and v_j in an appropriate way. These alterations are represented by Δu_i and Δv_j. Because of (3) and (5) we must choose $\Delta u_i \leqslant 0$ for $i=1,\dots,n$. *)

E.g. in Table 2 let us take $\Delta u_5 = -1$, so add 1 to each element of row 5. Then we can choose $\Delta v_2 = \Delta v_3 = \Delta v_4 = 1$ giving an improvement $\sum_i \Delta u_i + \sum_j \Delta v_j = 2$ for the objective value. The new assignment tableau is:

						u_i
	2	3	2	5	0	5
	0	1	0	2	7	5
$\bar{c}_{ij}$	6	3	3	3	0	8
	1	2	2	6	0	7
	5	0	0	0	8	5
v_j	0	4	2	1	0	37

Table 5

Further improving combinations of Δu_i and Δv_j can be selected, until $\Sigma u_i + \Sigma v_j$ is maximal. This "reduction process" can be controlled completely by the simplex method applied to the dual problem.

3. Preparation for the simplex method

In using the simplex method for solving problem (2) we should be aware of the fact, that the variables u_i and v_j are unrestricted in sign. The optimal solution is not necessarily non-negative.

*) See also Th. 5.

Example 2

$\{c_{ij}\} =$

				u_i	Δu_i
	1	1	0	0	
	0	0	0	0	-1
	1	1	0	0	
v_j	0	0	0		
Δv_j	1	1			

Table 6

$\{\bar{c}_{ij}\} =$

				u_i
	0	0	0	0
	0	0	1	-1
	0	0	0	0
v_j	1	1	0	1

Table 7

Optimal solution:
$\underline{u} = (0,-1,0)$
$\underline{v} = (1,1,0)$
$\Sigma\, u_i + \Sigma\, v_j = 1$
Primal solution:
e.g. $x_{11} = x_{22} = x_{33} = 1$

Repeated application of Th. 1 can give relief:

Theorem 2: Addition of a big number M to all elements of the cost matrix can give a non-negative solution of the dual problem.

Proof: Let the optimal solution be $\{u_i\}$, $\{v_j\}$ with $u_r = \underset{i}{\text{Min}}\, u_i < 0$ and $v_s = \underset{j}{\text{Min}}\, v_j < 0$. Then $\{u_i + v_s\}$, $\{v_j - v_s\}$ is also a feasible solution with the same object value $\Sigma\, u_i + \Sigma\, v_j$. If we take $M \geqslant -u_r - v_s$ then $\{M + u_i + v_s\}$, $\{v_j - v_s\}$ is a non-negative feasible solution with $\sum_i u_i + \sum_j v_j + n.M$, hence optimal.

Example 3 Add M= 10 to the cost matrix of Ex. 2: an optimal solution is then: $\underline{u}^1 = (10,10,9)$, $\underline{v} = (1,1,0)$.
$\Sigma\, u_i + \Sigma\, v_j = 31 - 30 = 1$

4. Simplex method

The content of the next theorems allows us to use the simplex method without carrying the burden of the dual simplex tableaux.

Theorem 3: The relative costs of the non-basic variables in the dual simplex tableau can be determined by solving the equation set:

$$(6) \quad \sum_{j=1;NB}^{n} R_{ij} = 1 \text{ for } i=1,\dots,n \text{ and } \sum_{i=1;NB}^{n} R_{ij} - R_{n+1,j;NB} = 1 \text{ for } j=1,\dots,n$$

where: NB means summation only over non-basic elements

: R_{ij} = relative cost of non-basic variable y_{ij}

: $R_{n+1,j}$ = relative cost of non-basic variable v_j^{p} *)

Proof: In a simplex tableau the relative costs of the non-basic variables are equal to the values of the corresponding primal basic variables, whereas dual basic variables have corresponding primal value zero. The constraints of the primal problem can equivalently, be written as follows:

$$\sum_j x_{ij} \leqslant 1 \quad , \quad \sum_i x_{ij} \geqslant 1$$

or

$$\sum_j x_{ij} + u_i^p = 1 \quad , \quad \sum_i x_{ij} - v_j^p = 1$$

$$u_i^p \geqslant 0 \qquad\qquad v_j^p \geqslant 0$$

For any simplex tableau, where all u_i are basic variables and so $u_i^p = 0$ and all $x_{ij} = 0$, resp. $v_j^p = 0$ whenever y_{ij} resp. v_j is basic (6) must hold. Because these 2n equations are independent the 2n variables can be determined uniquely.

*) u_i^p and v_j^p refer to the corresponding dual variables u_i and v_j.

Example 4 Let us start from the basic feasible solution in Ex 1:

					u_i
2	4	3	6	$\underline{0}$	5
$\underline{0}$	2	1	3	7	5
6	4	4	4	$\underline{0}$	8
1	3	3	7	$\underline{0}$	7
4	$\underline{0}$	$\underline{0}$	$\underline{0}$	7	6
v_j: $\underline{0}$	3	1	$\underline{0}$	$\underline{0}$	

Table 8

The equations (6) become:

$R_{15} = 1$ $\qquad R_{21} - R_{61} = 1$

$R_{21} = 1$ $\qquad R_{52} = 1$

$R_{35} = 1$ $\qquad R_{53} = 1$

$R_{45} = 1$ $\qquad R_{54} - R_{64} = 1$

$R_{52} + R_{53} + R_{54} = 1$ $\qquad R_{15} + R_{35} + R_{45} - R_{65} = 1$

Solution: $R_{61} = 0$, $R_{65} = 2$, $R_{54} = -1$ and $R_{64} = -2$ etc.

Notice that $\sum_{j,NB} R_{6j} = 0-2+2 = 0$

In Th. 4 it will be shown that this follows automatically for each tableau.

From these relative costs it can be decided which variable should go into the basis by determining

(7) $\qquad \text{Min } \{R_{ij} \mid i = 1,\dots,6 \; ; \; j = 1,\dots,5\} = R_{rs}$

The corresponding y_{rs} or v_s leaves the set of non-basic variables. In the example we find $R_{rs} = R_{64} = -2$, so we make $v_4 > 0$.

Theorem 4: In each assignment tableau the sum of the relative costs of the non-basic v_j-variables is zero.

Proof: The relative costs of the non-basic variables v_j are $R_{n+1,j} = v_j^p$, whilst $u_i^p = 0$ for all i, so from the primal problem it follows that

$\sum_j x_{ij} = 1$ for $i = 1,\dots,n$ or $\sum\sum_{ij} x_{ij} = n$

and $\sum_i x_{ij} - v_j^p = 1$ for $j = 1,\dots n$ or $\sum\sum_{ij} x_{ij} - \sum_{j,NB} v_j^p = n$

Hence $\sum_{j,NB} v_j^p = \sum_{j,NB} R_{n+1,j} = 0.$

5. Pivotcolumn determination

After determination of the new basic variable we need the corresponding pivotcolumn in the dual simplex tableau. This column can be found with the aid of the following "reduction interpretation". We distinguish two cases:

5.1. Some v_j becomes basic.

Suppose we take $\Delta v_k = 1$. So each element c_{ik} in the corresponding column of the reduced-cost matrix becomes $c_{ik}-1$. In order to maintain non-negativity of the reduced costs we should make $\Delta u_i = -1$ in each row containing a non-basic element in that column. Next for each new row with $\Delta u_i = -1$ we must mark columns containing non-basic elements with $\Delta v_j = 1$ and so on. This marking process is finished if no new marks are needed.

Example 5 Consider the tableau of Ex. 4 and make $\Delta v_4 = 1$.

						Δu_i
	.	.	.	.	x	
	x	.	.	.	.	
	.	.	.	.	x	
	.	.	.	.	x	
	.	x	x	x	.	-1
	x	.	.	x	x	
Δv_j		1	1	1		2

Table 9

Then $\Delta u_5 = -1$, next $\Delta v_2 = \Delta v_3 = 1$ are needed.

Notice that $\sum_i \Delta u_i + \sum_j \Delta v_j = 2$, fitting in with the meaning of relative costs.

x= NB-element

From the influence of the alterations Δu_i and Δv_j on the cost matrix and thus on the values of the basic variables of the dual tableau we can deduce the pivotcolumn.

Example 6 The pivotcolumn in Ex. 5 is found by taking the reverse of each element in Table 10.

						u_i
	0	-1	-1	-1	x	0
	x	-1	-1	-1	0	0
	0	-1	-1	-1	x	0
	0	-1	-1	-1	x	0
	1	x	x	x	1	-1
v_j	x	1	1	1	x	

Table 10.

Compare this with the v_4-column of Table 4: e.g. making $v_4 = 1$ gives: $y_{12} = 4 - v_4 = 4-1 = 3$

The only difficulty in this "reduction" argument is that besides v_k other variables v_j could be forced to become basic. However from Th. 2 we know that, eventually after introduction of M at the start, there is a non-negative dual solution, to be found with the simplex method. Therefore this situation can not be met, because then two variables would become simultaneously basic, which is impossible in the simplex method.

5.2. Some y_{ij} becomes basic.

In this case we can use similar "reduction" arguments. We start with $\Delta u_i = -1$, next marking all columns k with non-basic elements in row i, except column j, with $\Delta v_k = 1$ and so on. In contrast with case 5.1. now another way of reasoning is possible: begin with $\Delta v_j = -1$ and so on. But then we get some $\Delta v_j < o$ and the new solution could become infeasible. In Th. 5 we consider this seperately.

Example 7

						Δu_i
	.	.	.	.	x	
	x	.	.	.	.	
	.	.	.	.	x	
	.	.	.	.	x	
	.	x	x	x	.	-1
	x	.	.	x	x	
Δv_j		1	1			1

Table 11

Suppose the variable y_{54} in Ex. 5 should become basic. Then take $\Delta u_5 = -1$, next $\Delta v_2 = = \Delta v_3 = 1$ with $R_{54} = \sum_i \Delta u_i + \sum_j \Delta v_j = 1$

In this example starting with $\Delta v_4 = -1$ is impossible.

<u>Theorem 5</u>: For a non-basic element with $R_{ij} < 0$ the "reduction"-interpretation with some $\Delta u_i = -1$ and some $\Delta v_j = 1$ can always be used in order to determine the pivot column, provided that we start the algorithm with row reduction followed by column reduction.

Proof: Suppose an appropriate M is added to $\{c_{ij}\}$, such that a non-negative optimal solution $\{u_i^o, v_j^o\}$ exists. This solution can be found with the simplex method. In the first reduction phase we determine the row minima: $u_i \geq M$ for i=1,..,n and next some $v_j > 0$ (at most n-1), say for $j \in J$ with $J \subset \{1,\ldots,n\}$ are found. In the following step we can consider the reduced matrix $\{c_{ij}-u_i-v_j\}$ as the initial cost matrix of a new problem. This problem has the same optimal primal solution (by Th. 1) and corresponding non-negative dual solution:

$$\{u_i^{o\prime}\}=\{u_i^o\} \text{ and } v_j^{o\prime} = v_j^o \text{ for } j \notin J \text{ and } v_j^{o\prime} = v_j^o - v_j \text{ for } j \in J.$$

In applying the simplex method to the new problem we meet the same simplex tableau after the first reduction phase, with the same non-basic elements, relative costs and reduction constants $u_i' = u_i$, but with all $v_j' = 0$. Now the only way to improve the solution is making some $\Delta u_i < 0$ and some $\Delta v_j > 0$, otherwise the solution becomes infeasible, leading to a contradiction. The same argument can be used in further steps.

Theorem 5 does not hold in general for non-basic elements with $R_{ij} \geq 0$.

Example 8

						Δu_i
	.	.	.	.	x	-1
	x	.	.	.	.	
	.	.	.	.	x	
	.	.	.	.	x	
	.	x	x	x	.	
	x	.	.	x	x	
Δv_j						-1

'Table 12

Suppose in the solution of Ex. 4 we want the column belonging to NB-variable y_{15} with $R_{15} = 1$. Then we conclude that $\Delta u_1 = -1$. For variable y_{52} with $R_{52} = 1$ follows $\Delta v_2 = -1$.

Remark: In fact it is always possible to use the reduction argument starting with $\Delta u_i = -1$. When a second variable in the v_j-row should become basic we can compensate this by the following move:

make $\begin{cases} \Delta v'_j = -1 \text{ for } j=1,\dots,n \text{ and} \\ \Delta u'_i = 1 \text{ for } i=1,\dots,n \end{cases}$

Example 9

						Δu_i	$\Delta u'_i$	$\Delta u''_i$
	.	.	.	.	x		1	-1
	x	.	.	.	.		1	-1
	.	.	.	.	x		1	-1
	.	.	.	.	x		1	-1
	.	x	x	x	.	-1	1	
	x	.	.	x	x			
Δv_j			1	1				
$\Delta v'_j$	-1	-1	-1	-1	-1			
$\Delta v''_j$	1				1			

Table 13

Suppose we want $y_{52} = 1$ in Table 13. Then the reasoning is as follows:

$\Delta u_5 = -1 \Rightarrow \Delta v_3 = \Delta v_4 = 1 \Rightarrow$

$\Delta u'_i = 1$ for $i=1,\dots,5 \Rightarrow$

$\Delta v'_j = -1$ for $j=1,\dots,5$,

because otherwise $v_4 > 0$ too $\Rightarrow \Delta v''_1 = \Delta v''_5 = 1$ for feasibility $\Rightarrow$

$\Delta u''_1 = \Delta u''_2 = \Delta u''_3 = \Delta u''_4 = -1$.

Final result: $\Delta v_2 = -1$.

6. Determination of the leaving basic variable

From the simplex criterion

$$\bar{c}_{pq} = \text{Min}\{\bar{c}_{ij} \mid \Delta v_j = 1 \text{ and } \Delta u_i = 0 \ ;\ i=1,\ldots,n;\ j=1,\ldots,n\}^{*}$$

it follows, that y_{pq} becomes non-basic. Justification is easy because each pivotcolumn element that is involved equals unity. Therefore $\bar{c}_{pq}$ gives also the amount by which the same reduction can be performed.

Example 10

2	3	2	5	<u>0</u>	5
<u>0</u>	1	<u>0</u>	2	7	5
6	3	3	3	<u>0</u>	8
1	2	2	6	<u>0</u>	7
5	<u>0</u>	<u>0</u>	<u>0</u>	8	5
<u>0</u>	4	2	1	<u>0</u>	37

Table 14

From Tables 8 and 10 we conclude: $\bar{c}_{pq} = \bar{c}_{23} = 1$, so (2,3) is the new non-basic element. After reduction, amount 1, the assignment tableau of Table 14 is reached. Object value: $\Sigma u_i + \Sigma v_j = 37$.

Remark: The reduction amount $\bar{c}_{pq}$ could be zero. In that case the composition of the basis changes, the matrix $\{\bar{c}_{ij}\}$ does not.

7. Relative costs

The next simplex step starts with the calculation of the relative costs of the non-basic variables according to Th. 3. However this can be done more efficiently by using the next theorem:

Theorem 6: The relative costs of the non-basic variables can be determined by

a. Constructing a loop, involving cell (p,q) and cell (r,s) and other non-basic cells (in the v_j-row as well)

* The elements of the u_i-column are supposed to be big enough, eventually after addition of M, to remain positive.

b. Recalculation of R_{ij}'s:

$$R_{ij} = R_{ij} + (-1)^{k+1} R_{rs},$$ where k=1 in cell (p,q) moving around the loop and temporary reversal of signs in the $(n+1)^{st}$ row.

c. Leaving the remaining R_{ij} unchanged.

Proof: The new values of R_{ij} satisfy the equations of Th. 3, because the row- and columnsums are unchanged. This holds also for the v_j-row, where $\Sigma R_{n+1,j} = 0$.

Further, in the primal tableau the non-basic cells can be considered as a basic solution in a transportation tableau (m=n+1, n=n; number of basic elements: 2n), provided that we take $-R_{n+1,j}$.

In a transportation scheme such loops are unique.

Example 11

						u_i
	2	4	3	6	$\underline{0}^{1}$	5
	$\underline{0}^{1}$	2	$\boxed{1}$	3	7	5
	6	4	4	4	$\underline{0}^{1}$	8
	1	3	3	7	$\underline{0}^{1}$	7
	4	$\underline{0}^{1}$	$\underline{0}^{1}$	$\underline{0}^{-1}$	7	6
v_j	$\underline{0}^{0}$	3	1	$\underline{0}^{-2}$	$\underline{0}^{2}$	

Table 15

						u_i
	2	3	2	5	$\underline{0}^{1}$	5
	$\underline{0}^{-1}$	1	$\underline{0}^{2}$	2	7	5
	6	3	3	3	$\underline{0}^{1}$	8
	1	2	2	6	$\underline{0}^{1}$	7
	5	$\underline{0}^{1}$	$\underline{0}^{-1}$	$\underline{0}^{+1}$	8	5
v_j	$\underline{0}^{-2}$	4	2	1	$\underline{0}^{2}$	

Table 16

The relative costs of Ex. 4 are given in Table 15. From Ex. 10 we know that y_{23} is leaving the basis. The loop is passing the elements (2,3): $R_{23} = 0 + 2 = 2$; (2,1): $R_{21} = 1 - 2 = -1$; (6,1): $R_{61} = 0 + 2 = 2$; (6,4): $R_{64} = 2 - 2 = 0$ (disappears); (5,4): $R_{54} = -1 + 2 = 1$ and (5,3): $R_{53} = 1 - 2 = -1$. After these adjustments the sign of R_{61} is reversed again, so $R_{61} = -2$.

8. Stopping rule

The algorithm terminates when all relative costs of the non-basic variables are non-negative. Obviously the elements with $R_{ij} = 1$ then represent a feasible and optimal assignment. The corresponding primal variables x_{ij} have the same value and satisfy the constraints of the primal problem (1). Moreover by Th. 4 we know that all relative costs in the v_j-row must be zero. The optimal objective value is: $\Sigma\Sigma x_{ij} = \Sigma u_i + \Sigma v_j$.

9. Algorithm

Define the matrix $\{NB(i,j)\}$, with $NB(i,j) = 1$ if the element (i,j) is non-basic
$= 0$ if not.

Define $J=\{1,\dots,n\}$.

Start:

0. If necessary: $\bar{c}_{ij} = c_{ij} + M$ for $i=1,\dots,n$; $j=1,\dots,n$
   Set $NB(i,j)=0$ for $i=1,\dots,n+1$; $j=1,\dots,n$

1. Calculate $u_i = \bar{c}_{is} = \text{Min}\{\bar{c}_{ij} \mid j=1,\dots,n\}$ for $i=1,\dots,n$.
   Make $s \notin J$; $NB(i,s)=1$; $NB(n+1,s)=1$
   and reduce the cost matrix:
   $$\bar{c}_{ij} = \bar{c}_{ij} - u_i \quad i=1,\dots,n;\ j=1,\dots,n$$

2. Calculate $v_j = \bar{c}_{rj} = \text{Min}\{\bar{c}_{ij} \mid i=1,\dots,n\}$ for $j \in J$.
   Make $NB(r,j)=1$
   and reduce the cost matrix:
   $$\bar{c}_{ij} = \bar{c}_{ij} - v_j \quad i=1,\dots,n;\ j=1,\dots,n$$

3. Determine Min cost= $\sum_i u_i + \sum_j v_j$

Determination of relative costs:

4. Solve
   $$\sum_{j=1}^{n} NB(i,j)R_{ij} = 1 \quad i=1,\dots,n$$

$$\sum_{i=1}^{n+1} NB(i,j)R_{ij} = 1 \quad j=1,\ldots,n$$

Determination of new basic variable:

5. Make $R_{n+1,j} = -R_{n+1,j}$ if $NB(n+1,j) = 1$

   Determine $\text{Min}\{R_{ij} | NB(i,j)=1 \;;\; i=1,\ldots,n+1 \;;\; j=1,\ldots,n\} =$

   R_{rs} If $R_{rs} \geq 0$ then stop: optimal solution: $x_{ij}=1$ iff $R_{ij}=1$

Calculation of "pivotcolumn"

6a. If $r=n+1$: Mark column s with $\Delta v_s=1$. Next mark each row $i \neq n+1$ having $NB(i,s)=1$ with $\Delta u_i=-1$. For all new marked rows i successivily mark unlabeled columns having $NB(i,j)=1$ with $\Delta v_j=1$. Then again for all just marked columns mark rows, and so on until no marks can be given. Go to 7

6b. If $r \neq n+1$: Mark row r with $\Delta u_r=-1$. Next mark each column, except column s, where $NB(r,j)=1$ with $\Delta v_j=1$ etc, see 6[a].

Determination of reduction amount

7. Calculate $\theta=\bar{c}_{pq}=\text{Min}\{\bar{c}_{ij} | \Delta v_j=1 \text{ and } \Delta u_i=0;\; i=1,\ldots,n;\; j=1,\ldots,n\}$

Reduction of the cost matrix:

8. If $\theta \neq 0$ then make $\bar{c}_{ij}= \bar{c}_{ij}-(\Delta u_i+ \Delta v_j)\cdot\theta$;

   $u_i= u_i+ \Delta u_i\cdot\theta$ for $i=1,\ldots,n$

   $v_j= v_j+ \Delta v_j\cdot\theta$ for $j=1,\ldots,n$

   Min cost= Min cost + θ

Recalculation of relative costs:

9. Make $R_{n+1,j}= -R_{n+1,j}$ for $j=1,\ldots,n$ whenever $NB(n+1,j)=1$

   Construct a loop involving cell (p,q), cell (r,s) and other NB-cells. Recalculate $R_{ij}= R_{ij} + (-1)^{k+1}R_{rs}$ moving around the loop, starting from cell (p,q) with k=1.

Changing NB elements:

10. $NB(p,q) = 1 \quad NB(r,s) = 0$

11. Go to 5.

10. <u>Example</u>

We apply the algorithm to the problem of Ex. 1. After step 3 the reduced cost matrix is given by Table 17.2. The solution of the equation set in step 4 was found in Ex. 4 and is entered in the non-basic cells (underlined) of the tableau. In step 5 we find: Min $R_{ij} = R_{64} = -2$ and v_4 goes into the basis. In Ex.5 we already determined the appropriate set $\{\Delta u_i, \Delta v_j\}$ and in step 7 we calculate $\theta = \bar{c}_{23} = 1$, indicated by a rectangle. After reduction, step 8, the tableau is represented by Table 17.3. The relative costs are changed, moving around the loop, indicated in the tableau, starting from cell (2,3).

In the next tableau we determine: Min $R_{ij} = R_{61} = -2$, so v_1 goes into the basis. The order of the alterations in u_i and v_j is: $\Delta v_1 = 1 \Rightarrow \Delta u_2 = -1 \Rightarrow \Delta v_3 = 1 \Rightarrow \Delta u_5 = -1 \Rightarrow \Delta v_2 = \Delta v_4 = 1$

Next we find $\text{Min}\{\bar{c}_{ij} \mid \Delta u_i = 0 \; ; \Delta v_j = 1\} = \bar{c}_{41} = 1$. As a consequence y_{41} becomes non-basic and $\theta = 1$.

In Table 17.4 we choose y_{53} for entering the basis. As leaving variable we choose y_{42} because $\text{Min}\{c_{ij}\} = \bar{c}_{22} = \bar{c}_{42} = 1$ with $\theta = 1$.

Finally in Table 17.5 we can improve the object value again by one unity getting Table 17.6, where all $R_{ij} \geqslant 0$.

An optimal solution is $x_{11} = x_{23} = x_{35} = x_{42} = x_{54} = 1$ with value $\sum_i u_i + \sum_j v_j = \sum_{ij}\sum x_{ij} = 41$.

1

						u_i
	7	12	9	11	5	0
	5	10	7	8	12	0
	14	15	13	12	8	0
	8	13	11	14	7	0
	10	9	7	6	13	0
v_j	0	0	0	0	0	0

2

						u_i	Δu_i	
	2	4	3	6	$\underline{0}^{1}$	5		
	$\underline{0}^{1}$	2	$\boxed{1}$	3	7	5		
	6	4	4	4	$\underline{0}^{1}$	8		
	1	3	3	7	$\underline{0}^{1}$	7		
	4	$\underline{0}^{1}$	$\underline{0}^{1}$	$\underline{0}^{-1}$	7	6	-1	
v_j	$\underline{0}^{0}$	3	1	$\underline{0}^{2}$	$\underline{0}^{-2}$	35		$\Sigma u_i + \Sigma v_j$
Δv_j		1	1	1			2	$\Sigma \Delta u_i + \Sigma \Delta v_j$

3

						u_i	Δu_i
	2	3	2	5	$\underline{0}^{1}$	5	
	$\underline{0}^{-1}$	1	$\underline{0}^{2}$	2	7	5	-1
	6	3	3	3	$\underline{0}^{1}$	8	
	[1]	2	2	6	0^{1}	7	
	5	$\underline{0}^{1}$	$\underline{0}^{-1}$	$\underline{0}^{1}$	8	5	-1
v_j	$\underline{0}^{2}$	4	2	1	$\underline{0}^{-2}$	37	
Δv_j	1	1	1	1			2

4

						u_i	Δu_i
	1	2	1	4	$\underline{0}^{1}$	5	
	$\underline{0}^{-1}$	1	$\underline{0}^{2}$	2	8	4	
	5	2	2	2	$\underline{0}^{1}$	8	
	$\underline{0}^{2}$	[1]	1	5	$\underline{0}^{-1}$	7	
	5	$\underline{0}^{1}$	$\underline{0}^{-1}$	$\underline{0}^{1}$	9	4	-1
v_j	1	5	3	2	$\underline{0}^{0}$	39	
Δv_j		1		1			1

5

						u_i	Δu_i
	[1]	1	1	3	$\underline{0}^{1}$	5	
	$\underline{0}^{0}$	0	$\underline{0}^{1}$	1	8	4	-1
	5	1	2	1	$\underline{0}^{1}$	8	
	$\underline{0}^{1}$	$\underline{0}^{1}$	1	4	$\underline{0}^{-1}$	7	-1
	6	$\underline{0}^{0}$	1	$\underline{0}^{1}$	10	3	-1
v_j	1	6	3	3	$\underline{0}^{0}$	40	
Δv_j	1	1	1	1			1

6

						u_i	
	$\underline{0}^{1}$	0	0	2	$\underline{0}^{0}$	5	
	$\underline{0}^{0}$	0	$\underline{0}^{1}$	1	9	3	
	4	0	1	0	$\underline{0}^{1}$	8	
	$\underline{0}^{0}$	$\underline{0}^{1}$	1	4	1	6	
	6	$\underline{0}^{0}$	1	$\underline{0}^{1}$	11	2	
v_j	2	7	4	4	$\underline{0}^{0}$	41	

Table 17.1-6

11. Computational Results

The algorithm is applied to some small problems, by hand. The results, given in Table 18, are compared with the performance of the Hungarian method and the revised Tomizawa approach from [2].

All methods start with row-reduction of the cost-matrix followed by column-reduction. The dual method always uses $\Delta u_i \leqslant 0$ and $\Delta v_j \geqslant 0$. The Hungarian method, as described in [1], always gives $\Delta u_i \geqslant 0$ and $\Delta v_j \leqslant 0$. Therefore we also applied the following "vertical" version of the Hungarian algorithm (notation according to algorithm 10.1, p. 172 in [1]), enabling a better comparison:

Start: a. reduction on rows, next on columns

b. simple determination of a set of independent zeroes, row by row

part A: } rows are replaced by columns and vice versa.
part B:

part C: $u_i := \begin{cases} u_i - \eta & i \in I \\ u_i & \text{otherwise} \end{cases}$

$$v_j := \begin{cases} v_j + \eta & j \in J \\ v_j & \text{otherwise} \end{cases}$$

$$\bar{c}_{ij} := \begin{cases} \bar{c}_{ij} + \eta & i \in I \text{ and } j \notin J \\ \bar{c}_{ij} - \eta & i \notin I \text{ and } j \in J \\ \bar{c}_{ij} & \text{otherwise} \end{cases}$$

In most problems the number of iterations in the vertical method was less than in the horizontal version!
Also in the Tomizawa algorithm we started with a simple determination of independent zeroes, row by row.
We distinguished two different kinds of iteration steps:

a. steps where also the matrix was to be reduced.
b. steps where only the number of assignments or the composition of NB-elements (dual method) changed.

The first number of an entrance-pair in the table gives the sum total of (a)-steps, the second one the sum total of (b)-steps.
Note that the total number of steps in the Tomizawa column automatically also gives the number of rows without an independent zero in the start.
Comparison of dual and Hungarian method on one side and the Tomizawa algorithm on the other side can better be based on C.P.U.-running times, computerprograms are being written at the moment.
Some remarks on the problems: nr. 1 is the example we used in the preceding pages, nr. 5 and nr. 6 contain random integers between 0 and 100. In this last problem some alternative relative costs were chosen, which gave the mentioned numbers of iterations. Nr. 7 was constructed such that only one independent zero was found in the beginning and Nr. 8 is very much dual degenerate.

For the present some conclusions are: the dual method differs from and can compete with the Hungarian method, not with the Tomizawa algorithm. In cases of few independent zeroes the dual method could be the best method, because it chooses the best relative cost. In small problems these costs do not variate much, in larger problems this aspect could become important. This will be investigated with the aid of the computer.

Problem nr.	size n	Algorithm				Remarks
		Dual	Hor.Hung.	Vert.Hung.	Rev.Tomizawa	
1	5	4+0	4+2	4+2	2+0	from [2]
2	7	2+1	2+2	2+2	2+0	from [1] , p. 169
3	4	2+0	2+2	2+2	2+0	from [3] , p. 78
4	7	1+1	2+1	1+1	1+0	from [4]
5	10	4+1	5+2	3+2	2+0	random numbers
6	10	3-4-5-6	5+2	3+2	2+0	random numbers
7	5	2+1	2+4	3+4	3+1	one independent zero
8	5	0+1	0+1	0+1	0+1	dual degenerate

Table 18

References

[1] W. Domschke, Logistik, Transport (Oldenbourg, München, 1981)

[2] B. Dorhout, Experiments with some algorithms for the linear assignment problem, Report BW 39/77 (M.C., Amsterdam 1977)

[3] H. Eiselt, H. von Frajer, Operations Research Handbook (de Gruyter, Berlijn, 1977)

[4] M. Hung, W. Rom, Solving the assignment problem by relaxation, OR, Vol 28, nr.4 (1980).