ALGORITHM 27

ASSIGNMENT

ROLAND SILVER

MIT Lincoln Laboratory, * Lexington, Massachusetts

procedure Assignment(d, n, x) ; value n ; integer n ;
array d ; integer array x ;
comment: Assignment determines that permutation x of the
integers [1:n] for which the sum (i := 1(1)n) of the
elements d[i, x[i]] of the n x n matrix d is a
minimum. n ≥ 2. For more complete information
see: An Algorithm for the Assignment Problem,
Roland Silver, Comm. ACM, Nov. 1960, p. 605 ;

begin
switch Switch := NEXT, LI, NEXT1, MARK ;
array a[1:n, 1:n] ;
integer array e[1:n], eb[1:n], lambda[1:n], mu[1:n],
r[1:n], y[1:n] ;
integer eb, el, c0, i, j, k, l, rl, rs, sw ;

comment: INITIALIZE ;
for i := 1 step 1 until n do
begin min := d[i, 1] ;
for j := 2 step 1 until n do if d[i, j] < min then
min := d[i, j] ;
for j := 1 step 1 until n do a[i, j] := d[i, j] − min
end i ;
for j := 1 step 1 until n do
begin min := a[1, j] ;
for i := 2 step 1 until n do if a[i, j] < min then
min := a[i, j] ;
for i := 1 step 1 until n do a[i, j] := a[i, j] − min
end j ;
for i := 1 step 1 until n do x[i] := y[i] := 0 ;
for i := 1 step 1 until n do
begin
for j := 1 step 1 until n do
begin
if a[i, j] ≠ 0 ∨ y[i] ≠ 0 ∨ x[i] ≠ 0 then go to J1 ;
x[i] := j ; y[i] := 1
J1: end j ;
end i ;
START: comment: Start labeling ;
rl := cl := 0 ;
r[i] := 1 ;
for i := 1 step 1 until n do
begin mu[i] := lambda[i] := 0 ;
if x[i] ≠ 0 then go to LI ;
rl := rl + 1 ; r[rl] := i ; mu[i] := −1
end i ;
LI: Label: comment: Label and scan ;
i := [r[el]] ; rs := rs + 1 ;
for j := 1 step 1 until n do
begin
if a[i, j] ≠ 0 or lambda[j] ≠ 0 then go to J2 ;
lambda[j] := i ; cl := cl + 1 ; c[cl] := j ;
if y[j] ≠ 0 then go to MARK ;
rl := rl + 1 ; r[rl] := y[j] ; mu[y[j]] := i
J2: end j ;

if rs ≤ rl then go to LABEL ;
comment:
RENORMALIZE ;
sw := 1 ; cb := cl ; cb := 0 ;
for j := 1 step 1 until n do
begin
if lambda[j] ≠ 0 then go to J3 ;
cli := cb[j + 1] ; cb[cb[j]] := j
J3: end j ;
min := a[ri, cb[ri]] ;
for k := 1 step 1 until rl do
begin
for l := 1 step 1 until ebl do if a[rl, k] ≤ min
then min := a[rl, k] ; cb[rl] := k
end k ;
for i := 1 step 1 until n do
begin
if mu[i] ≠ 0 then go to L2 ;
for l := 1 step 1 until ebl do a[i, c[l]] := a[i, c[l]] + min ;
go to L3 ;
L2: for i := 1 step 1 until ebl do
begin
a[j, cb[j]] := a[j, cb[j]] − min ;
go to Switch[sw] ;
NEXT: if a[i, cb[i]] ≠ 0 ∨ lambda[cb[i]] ≠ 0 then go to LI ;
lambda[cb[i]] := i ;
if y[cb[i]] ≠ 0 then
begin
cl := cb[i] ; sw := 2 ; go to L1 end ;
e[i] := cl + 1 ; c[e[i]] := cb[i] ;
rl := rl + 1 ; r[rl] := y[cb[i]] ;
L1: end 1 ;
L3: end i ;
go to Switch[sw + 2] ;
NEXT1: if cb0 = cl then go to LABEL ;
for i := cb0 + 1 step 1 until cl do
mu[y[e[i]]] := e[i] ; go to LABEL ;
MARK: comment: mark new column and permute ;
y[i] := 1 := lambda[j] ;
if x[i] = 0 then begin x[i] := j ; go to
START end ;
k := j ; j := x[i] ; x[i] := k ;
go to MARK
end Assignment

* Operated with support from the U. S. Army, Navy and Air
Force.

[Note: The reader should distinguish between the letter
and the figure 1, both of which appear in the above
algorithm.—Ed.]

CERTIFICATION OF ALGORITHM 27

ASSIGNMENT [Roland Silver, Comm. ACM, Nov. 1960]

ALBERT NEWHOUSE

University of Houston, Houston, Texas

The ASSIGNMENT algorithm was translated into MAD and
successfully run on the IBM 709/7994 after the following corrections were made:
All references to array \(a\) and \(d\) refer to the same array, i.e. change all \(a[i, j]\) to \(d[i, j]\). Furthermore:

(a) 3rd line after \(LABEL\): \text{comment}: Label and scan;
begin if \(d[i, j] \neq 0 \lor \lambda \lambda [j] \neq 0\) then do
(b) first line after \(J3\): \text{end} \(j\);
should read
\[ \min := d[r[1], cb[1]]; \]
(c) line \(J2\):
should read
\[ J2: \text{for } i := 1 \text{ step } 1 \text{ until } cb[i] \text{ do} \]
Since there is no provision made for this algorithm to end the following additions were made:
(1) in the integer declaration add the variable: \(flag\)
(2) first line after \(START\): \text{comment}: \(\cdots\)
add the line
\[ \text{flag} := n; \]
(3) first line before \(J1\): \text{end} \(i\);
change to read
\[ rl := rl + 1; \ rrl[i] := i; \ \text{mn}[i] := -1; \ \text{flag} := \text{flag} - 1 \]
(4) add a line after \(J1\): \text{end} \(i\);
\[ \text{if } \text{flag} = n \text{ then go to } FINI; \]
(5) change the last line of the algorithm to read:
\[ FINI: \text{end Assignment} \]
In order to obtain the minimum value of the \(\sum_{i=1}^{n} a_{i}\) (in the following called total) the following additions may be made:
Add a real variable \(total\) and
\(A\) new line after \(INITIALIZE\);
\[ total := 0; \]
\(B\) new line after the first \text{end} \(i\);
\[ total := total + \min; \]
\(C\) new line after the first \text{end} \(j\);
\[ total := total + \min; \]
\(D\) after the line \text{end} \(k\); after \(J3\) \text{end} \(i\);
\[ total := total + (rl + cb[i] - n) \times \min; \]

CERTIFICATION OF ALGORITHM 27
ASSIGNMENT [Roland Silvers, Comm. ACM 3, Nov. 1960].
Robert D. Witty
Burroughs Corp., Detroit, Mich.
Assignment was successfully run on the Burroughs B5000 using Burroughs extended Algol 60.

\begin{verbatim}
Input Array
  60  0  0  70  0  0
  0  40 18  0  60 24
  60 16  2  4  0  40
  0  27 18  3  55 75
  0  40  2 16 11 53
 28  4 10  84  0 16
\end{verbatim}

\textit{Solution Vector}: \(X(6, 4, 3, 1, 5, 2)\)

The following changes were made in the algorithm prior to its successful run:
\begin{verbatim}
FROM \(MIN := a[r[i, cb[i]]];\)
TO \(MIN := a[r[1], cb[1]];\)
FROM \(\text{if } X[i] = 0 \text{ then begin } X[i] := j; \text{go to START} \text{ end;} \)
TO \(\text{if } X[i] = 0 \text{ then begin } X[i] := j; \text{for } i := 1 \text{ step } 1 \text{ until } N \text{ do begin if } X[i] = 0 \text{ then go to START;} \text{end; go to EXIT;} \text{end;}\)
\end{verbatim}