

J. KUCHARCZYK (Wrocław)

## QUASI-OPTIMAL ORDERING OF OBJECTS

**1. Procedure declaration.** The procedure *goptord* realizes the quasi-optimal ordering of objects using the method presented in [1].

Data:

- $p$  — number of objects;
- $d[1 : p \times (p-1) \div 2]$  — lower half of the distance matrix between objects;
- init* — Boolean variable whose value should be **true** on entry if an initial ordering  $(a_1, a_2, \dots, a_p)$  of the objects is given; otherwise the procedure assumes the initial ordering  $(1, 2, \dots, p)$ ;
- $a[1 : p]$  — integer array holding the initial ordering of the objects (valid only if *init* has the value **true** on entry).

Results:

- $d[1 : p \times (p-1) \div 2]$  — rearranged distance matrix, i.e. the row  $i$  contains the distances of the object  $a[i]$ ;
- $a[1 : p]$  — integer array holding the quasi-optimal ordering of the objects;
- $q$  — goodness-of-ordering index (formula (2) of [1]).

**2. Method used.** The procedure *goptord* has been programmed after the algorithm given in [1], p. 26-27. However, rather than using similarities, distances between objects are used and, therefore,  $\min_{s,t} q(s, t)$  is sought.

Also formula (3) from [1] has been transformed to the equivalent formula

$$q(s, t) = (t-s) \left[ \sum_{v=1}^{s-1} (d_{tv} - d_{sv}) + \sum_{v=t+1}^p (d_{vs} - d_{vt}) \right] + \sum_{v=s+1}^{t-1} [2v - (s+t)] (d_{vs} - d_{tv}),$$

more convenient for programming.

The distance matrix is transposed together with the objects. On exit the element  $d_{st}$  contains the distance between objects  $a[s]$  and  $a[t]$ . This approach has been profitable with respect to computer running time.

```

procedure qoptord(p,d,init,a,q);
  value p;
  integer p;
  real q;
  Boolean init;
  array d;
  integer array a;
  begin
    integer h,h1,i,j,s,s1,s2,t,t1,t2,ts,v;
    real minq,qr;
    procedure transpose(i,j);
      value i,j;
      integer i,j;
      begin
        real t;
        integer i1,i2,j1,j2,k,k1,k2;
        procedure exch(a,b);
          value a,b;
          integer a,b;
          begin
            t:=d[a];
            d[a]:=d[b];
            d[b]:=t
          end exch;
        i1:=i-1;
        i2:=(i-2) $\times$ i1 $\div$ 2;
        j1:=j-1;
        j2:=(j-2) $\times$ j1 $\div$ 2;
        k2:=j2-i2;
        k1:=j2+i1;

```

```

for k:=j2+1 step 1 until k1 do
  exch(k,k-k2);
k1:=i2+i;
for k:=i+1 step 1 until j1 do
  begin
    k1:=k1+k-2;
    exch(k1,j2+k)
  end k;
k2:=j-i;
k1:=j2+j;
i2:=p-2;
for k:=j1 step 1 until i2 do
  begin
    k1:=k1+k;
    exch(k1,k1-k2)
  end k
end transpose;
if init
then
begin
for i:=1 step 1 until p do
  begin
    h:=a[i];
    if h>0
    then
      begin
        a[i]:=-h;
        if i≠h
        then
          begin

```

```

s:=i;
et:   if s<h
      then transpose(s,h)
      else transpose(h,s);
      s:=h;
      h:=a[s];
      a[s]:=-h;
      if h≠i
      then go to et
      end i≠h
      end h>0
      end i;
      for i:=1 step 1 until p do
        a[i]:=-a[i]
      end init
      else
      for i:=1 step 1 until p do
        a[i]:=i;
iter:
      minq:=.0;
      for t:=2 step 1 until p do
        begin
          t1:=t-1;
          t2:=(t-2)*t1÷2;
          h1:=t2+t1;
          for s:=1 step 1 until t1 do
            begin
              ts:=t+s;
              s1:=s-1;
              s2:=(s-2)*s1÷2;

```

```

qr:=.0;
for v:=1 step 1 until s1 do
  qr:=qr+d[t2+v]-d[s2+v];
h:=h1;
for v:=t+1 step 1 until p do
  begin
    qr:=qr+d[h+s]-d[h+t];
    h:=h+v-1
  end v;
qr:=qr*(t-s);
h:=s2+s1;
for v:=s+1 step 1 until t1 do
  begin
    qr:=qr+(v+v-ts)*(d[h+s]-d[t2+v]);
    h:=h+v-1
  end v;
if qr<minq
  then
    begin
      minq:=qr;
      i:=s;
      j:=t
    end qr<minq
  end s
end t;
if minq<.0
  then
    begin
      transpose(i,j);
      t:=a[i];

```

```

    a[i]:=a[j];
    a[j]:=t;
    go to iter
end minq<.0;
q:=.0;
h:=1;
i:=p-1;
for v:=1 step 1 until i do
  begin
    qr:=.0;
    t:=h;
    for s:=v+1 step 1 until p do
      begin
        qr:=qr+d[t];
        t:=t+s
      end s;
    q:=q+v*qr;
    h:=h+v
  end v
end qoptord

```

**3. Certification.** The procedure *qoptord* has been successfully run on the Odra 1204 computer using the ALGOL 1204 MT compiler. The calculation times depend heavily on the problem size and on the initial ordering of objects, as may be seen in Table 1.

TABLE 1. Running times of *qoptord* on test problems

Problem size		5	10	15	20	25	30	35	40	45	50
Running time (in secs.)	minimum	0	2	3	11	19	34	55	82	117	161
	maximum	2	12	66	215	487	1018	2224	3670	6256	9102
No. of test runs		8	10	10	7	5	4	4	5	4	3

## Reference

- [1] F. A. Szczotka, *On a method of ordering and clustering of objects*, Zastosow. Matem. 13 (1972), p. 23-34.

INSTITUTE OF INFORMATICS  
UNIVERSITY OF WROCLAW  
50-384 WROCLAW

Received on 17. 11. 1977

ALGORYTM 72

J. KUCHARCZYK (Wrocław)

QUASI-OPTYMALNE PORZĄDKOWANIE OBIEKTÓW

STRESZCZENIE

Procedura *goptord* wyznacza quasi-optymalne uporządkowanie obiektów według metody zaproponowanej w [1].

Dane:

- $n$  — liczba obiektów;  
 $d[1 : p \times (p-1) \div 2]$  — dolny trójkąt macierzy odległości między obiektami;  
*init* — zmienna boolowska, która powinna mieć wartość **true**, jeśli na wejściu podano początkowe uporządkowanie obiektów ( $a_1, a_2, \dots, a_p$ ); w przeciwnym razie przyjmuje się uporządkowanie początkowe (1, 2, ...,  $p$ );  
 $a[1 : p]$  — tablica całkowita, zawierająca początkowe uporządkowanie obiektów (ważne tylko wtedy, gdy *init* ma wartość **true** na wejściu).

Wyniki:

- $d[1 : p \times (p-1) \div 2]$  — dolny trójkąt macierzy odległości, przestawiony w taki sposób, że w wierszu  $i$  podane są odległości od obiektu  $a[i]$ ;  
 $a[1 : p]$  — tablica całkowita, zawierająca numery obiektów w uporządkowaniu quasi-optymalnym;  
 $q$  — wskaźnik dobroci uporządkowania (por. wzór (2) w [1]).

Metodę, podaną w [1], zaprogramowano dla odległości (a nie podobieństw) między obiektami, posługując się przy tym zmodyfikowaną postacią wzoru (3) z [1], wygodniejszą w obliczeniach. Kolejne przestawienia obiektów pociągają za sobą także przestawienie odpowiednich odległości w macierzy odległości. Czasy obliczeń dla przykładów kontrolnych podano w tabelicy 1. Do obliczeń użyto maszyny cyfrowej Odra 1204 i translatora ALGOL 1204 MT.