ALGORITHM 83

OPTIMAL CLASSIFICATION OF OBJECTS

BRIAN H. MAYOH

Digital Computer Laboratory, University of Illinois, Urbana, Ill.

```
procedure OPTIMUM COVERING FINDER (Pattern, population, set number, set prices, chosen sets, bounds, overflow);
  Boolean array Pattern, chosen sets; integer population, set number, bounds; array set prices; label overflow;
begin comment The number of objects in some given set is given by population. The procedure is given a classification of these objects by a collection of overlapping subsets. A cost is assigned to each subset. Then OPTIMUM COVERING FINDER selects the cheapest subcollection such that every object is contained in at least one of the subsets of the subcollection. set prices[i] carries the cost of subset i. Pattern is an array of size [1:set number,1:population] such that Pattern[a,b] ≡ does subset a include object b. chosen sets[i] finally carries the answer to the question: Is set i in the cheapest subcollection? The programmer must restrict the amount of space available to the procedure by setting bounds. From experience bounds = set number ↑ 2 suffices to avoid most alarm exits to overflow.;
  Boolean array C[1:population], D[1:bounds, 1:population], R, S[1:bounds,1:set number];
  integer a, b, d, r, s;
  Boolean procedure HAVE WE A COVERING;
  begin procedure ADD to (Q,q,f); integer q;
        real f; Boolean array Q;
    begin if q=bounds then go to overflow else q := q+1;
      for a := 1 step 1 until set number do Q[q,a] := f
    end; for a := 1 step 1 until population do
          C[a] := false;
    for a := 1 step 1 until set number do
    begin if chosen sets[a] then
      for b := 1 step 1 until population do
      C[b] := C[b] ∨ Pattern[a,b]
    end; for a := 1 step 1 until population do
    begin if ¬ C[a] then go to E end;
    go to found;
E:  for d := 1 step 1 until s do
    begin for b := 1 step 1 until population do
      begin if C[b] ∧ ¬ D[d,b] then go to try another end;
      ADD to (R, r, chosen sets[a]);
      for b := 1 step 1 until set number do
      begin if chosen sets[b] ∧ ¬ S[d,b] then
        ADD to (R, r, S[d,a] ∨ a=b)
      end; go to F;
    try another:
    end of for statement labelled E;
    ADD to (S, s, chosen sets[a]);
    for a := 1 step 1 until population do D[s,a] := C[a];
F:  HAVE WE A COVERING := false
  end; r := s := 0;
  ECONOMISER 2 (HAVE WE A COVERING, set prices, set number, r, R, chosen sets);
found: end
```