Solving capacitated clustering problems *

John M. MULVEY and Michael P. BECK
School of Engineering and Applied Science, Department of Civil Engineering, Princeton University, Princeton, NJ 08544, U.S.A.

Abstract. This paper presents two effective algorithms for clustering n entities into p mutually exclusive and exhaustive groups where the 'size' of each group is restricted. As its objective, the clustering model minimizes the sum of distances between each entity and a designated group median. Empirical results using both a primal heuristic and a hybrid heuristic-subgradient method for problems having $n \leqslant 100$ (i.e. 10100 binary variables) show that the algorithms locate close to optimal solutions without resorting to tree enumeration. The capacitated clustering model is applied to the problem of sales force territorial design.

Keywords: Integer programming, multivariate statistics, cluster analysis

1. Introduction

It has become commonplace to state that integer programming (IP) problems are hard to solve. Not only are these problems NP complete, exhibiting exponential worst-case performance for all known exact algorithms, but also the performance of existing algorithms on 'average' problems is often disappointing. Many realistic-size applications lie outside the scope of practical solution. Less well-known are the tremendous advances that have occurred in solution strategies for various classes of IP's—those possessing special structure. Since 1970, the most successful concepts for solving IP's with special structure (aside from networks) have been based upon heuristics and Lagrangian relaxation. Fisher [6] and Shapiro [29] present numerous applications in which Lagrangian relaxation strategies have proved effective.

This paper describes two algorithms for solving another integer programming model with special structure—capacitated clustering problems. Cluster analysis involves the grouping of data entities (points) in a way that maximizes the homogeneity of points within a group and, at the same time, the heterogeneity of the points between groups. Numerous heuristics and exact procedures with accompanying objective functions have been proposed for the classical uncapacitated clustering problems; for example, see [1,11,22]. Herein we examine algorithms for solving a more general problem whereby upper bounds are imposed on the 'sizes' of the cluster groups.

The capacitated clustering problem is stated as follows [CCP]:

$$v(\bar{x}) \equiv \text{minimize} \sum_{i\in I}\sum_{j\in J} d_{ij}x_{ij}, \tag{1.0}$$

subject to

$$\sum_{j\in J} y_j = p, \tag{1.1}$$

$$\sum_{j\in J} x_{ij} = 1, \qquad i\in I, \tag{1.2}$$

$$x_{ij} \leqslant y_j, \qquad i\in I, j\in J, \tag{1.3}$$

$$x_{ij}, y_j = \{0, 1\}, \qquad i\in I, j\in J, \tag{1.4}$$

$$\sum_{i\in I} w_i x_{ij} \leqslant W_j, \qquad j\in J, \tag{1.5}$$

where

$$d_{ij} = \left[\sum_{k=1}^{l} (a_{ik} - a_{jk})^S\right]^{1/S}, \quad i\in I, j\in J,$$

$$\|I\| \equiv n, \qquad \|J\| \equiv m,$$

and

$\bar{x}', \bar{y}'$: solution to [CCP].

The population of points to be clustered is desig-

* This research was sponsored in part by the Office of Tax Analysis, U.S. Treasury Department

Received August 1983; revised November 1983

0377-2217/84/$3.00 © 1984, Elsevier Science Publishers B.V. (North-Holland)

nated as set $\{I\}$ in which each data point $i \in I$ is defined across l attributes: $(a_{il}, a_{i2}, \ldots, a_{ik})$. Model (1.0)–(1.4) depicts the uncapacitated clustering problem as discussed in Mulvey and Crowder [22]. The capacity constraints (1.5) ensure that the 'size' of cluster $j \in J$, i.e., its cumulative weight, is less than the exogenous coefficient W_j. [CCP]'s goal is constructing p clusters that are as homogeneous as possible, while not violating the capacity constraints (1.5).

In [CCP], the binary x_{ij} variable indicates whether or not point i is assigned to cluster j. If so, the variable equals 1. Otherwise the variable equals 0. The condition $y_j = 1$ specifies point j as the median for cluster $j \in J$. Constraints (1.2) ensures that all points are assigned. Constraints (1.3) prevents assignments to points that have not been chosen as medians.

As an example, one may be interested in clustering a population of $n = 20$ hospitals in city xyz into $p = 6$ clusters. While there are numerous hospital characteristics available, the chosen attributes—$(a_{i1}, a_{i2}, \ldots, a_{il})$—must relate to the ultimate objective of the cluster analysis. For instance, interest in physical proximity might dictate that hospital attributes be grid locations (a_{i1} = latitude, a_{i2} = longitude) and that pairwise distance between hospitals d_{ij} be defined as Euclidean. In many cases, of course, the use of Euclidean distance is inappropriate, especially when the attributes are correlated. Before selecting a distance criteria, the analyst should consult a standard cluster analysis text (e.g., [1,11]), a multivariate statistical text (e.g. [3,20]), or one of the references appearing therein.

Model [CCP] is a significant special case of the facility location problem [1] [9,25,27] and thereby closely related to the generalized assignment [5,17] and p-median [16,26] problems. As a consequence it can be shown to be NP complete [7]. Standard IP algorithms for solving [CCP] are available though they are ineffective for large examples. Note that the number of decision variables is a function of n^2. Heuristics and approximation procedures are, therefore, the only practical techniques for handling larger problems.

Model [CCP] has a variety of practical applications. Some recent examples are listed below:

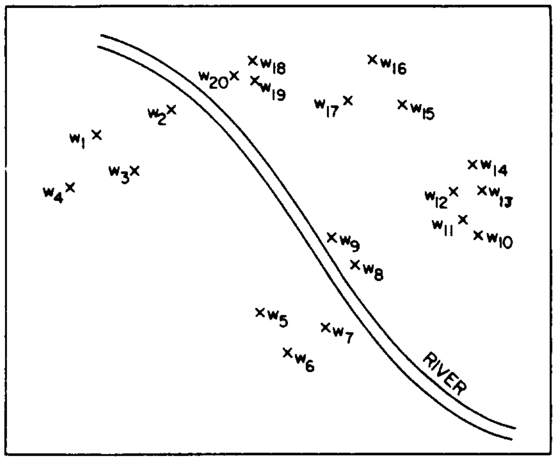

Fig. 1. Twenty customer (hospital) locations in city xyz.

– grouping of customers into vehicle routes such that a cluster's total customer requirements does not exceed the delivery capacity of the vehicle(s) assigned to a route [6],
– constructing optimal index funds where upper limits are imposed on individual stock investment levels [2],
– partitioning of nodes in a distributed computing network where each cluster of nodes should possess roughly the same computing power [23],
– determining strata boundaries for use in stratified sampling [21].

An illustration of the approach is shown in Figs. 1 and 2. Here, we have aggregated twenty indicated customers in one city into six compact territories. The customers represent hospitals dis-

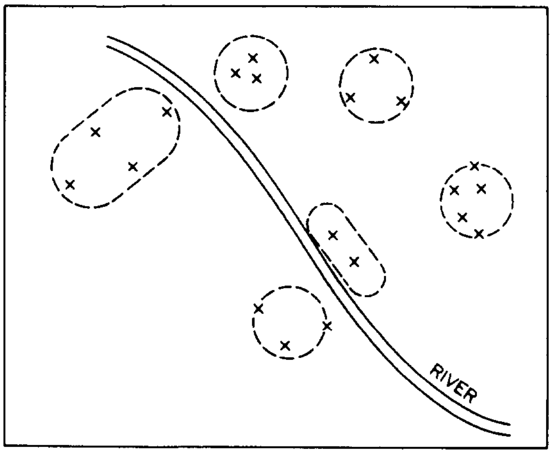

Fig. 2. Aggregating the twenty customers into six regions.

[1] Most capacitated plant location models do not place integer restrictions on the x_{ij} variables.

[2] This model was recently developed for a large international drug company by one of the authors.

persed across a geographic region; the corresponding weight (w_i) refers to the estimated annual sales for the ith hospital [2]. Each of the resulting clusters is designated as a sales territory and is assigned a single salesman according to the solution of [CCP]. Capacity constraints are required so that a salesman can provide adequate service for the assigned customers. Otherwise he would be too busy to respond to customer needs. Note that the most likely location for the salesman's home office is the cluster median. (This scheme was employed by the aforementioned drug company.) Also, it is a simple matter to enforce barriers, e.g., the river, in [CCP] by adjusting the distance function —d_{ij}—thereby penalizing (i, j) combinations in which two hospitals lie on opposite sides of the barrier.

A somewhat similar approach has been proposed by Hess [15] and Hess and Samuels [14]. However, their solution strategies are heuristics without lower bounds. See Shanker et al. [28], for a procedure based on the set partitioning model.

2. Primal heuristic

Many clustering and location heuristics depend either upon local improvements or upon greedy algorithms. The following heuristic for solving [CCP] is no exception to this rule:

First, a random set of p starting medians is chosen. Next we try assignments of data entities to their nearest median, ...and so on, until we locate the closest median assignment that will not exceed a cluster's capacity (constraint (1.5)). Entities are assigned in decreasing order of regret where regret is defined as the absolute value of the difference in distance between an entity's first and second nearest medians. The assignment of entities having the largest regret values is undertaken first so that the large penalties associated with assigning an entity to a very 'distant' second or third or ... pth nearest median can be minimized.

After all assignments are complete, the median for each cluster is recomputed by locating that entity which minimizes the sum of distances between it and all other cluster members. This point, in some cases, will correspond to the original median. If one or more new medians is found and the objective function decreases by a prespecified minimum value, ϵ, the process is repeated beginning with a reassignment of entities to the newly computed set of medians. The procedure continues until no change in the median set occurs from one iteration to the next, or until an iteration limit is reached.

In the next stage, improvements to the solution are attempted vial local switches. This is accomplished by examining all pairwise interchanges of entities between clusters. If an interchange will improve the current solution and at the same time not violate capacity constraints the swap is made. After performing all feasible interchanges the solution is recorded. The entire procedure is then repeated for a predetermined number of steps, called major iterations, by selecting a new random set of starting medians and proceeding as above. That assignment having the minimal value of all solutions obtained is designated as the incumbent.

This heuristic resembles the p-means algorithm of MacQueen [18]. There are, however, some notable extensions including the use of capacity constraints, a series of random starts, a regret vector to order assignments, and the interchange procedure. A more formal definition of the algorithm in pidgin ALGOL is shown in the Appendix. The results of solving [CCP] with the heuristic are presented in Section 5.

3. Subgradient algorithm

Several strategies are available for evaluating the performance of heuristics:

(1) worst-case analysis (computational complexity),

(2) statistical and probabilistic indicators for average performance, and

(3) problem-specific measures.

Lagrangian relaxation methods fall under the last category. The idea is straightforward. By solving a relaxed version of [CCP] we generate a lower bound to the optimal objective function value, $v(\bar{x}')$. The gap between the Lagrangian lower bound and the best primal solution—the incumbent—is reduced by employing a subgradient algorithm to compute improved Lagrange multipliers.

A Lagrangian relaxation of [CCP] may be formulated by dualizing the equality constraints (1.2) and introducing a set of Lagrange multipliers $\bar{u} =$

$(u_1, u_2, \ldots, u_n)$. The resulting relaxed model is the following [LCCP]:

$$L(\bar{u}) = \underset{\bar{x}, \bar{y}}{\text{minimize}} \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} + \sum_{i \in I} u_i \cdot \left(1 - \sum_{j \in J} x_{ij}\right),$$

subject to

(1.1) and (1.3)–(1.5).

The objective function for [LCCP] may be equivalently expressed as

$$L(\bar{u}) \equiv \min \sum_{i \in I} u_i + \sum_{i \in I} \sum_{j \in J} (d_{ij} - u_i) x_{ij}.$$

It is easy to see that $L(\bar{u})$ provides a lower bound to [CCP]. Observe that

$$L(\bar{u}) \leqslant v(\bar{x}') + \sum_{i \in I} u_i \cdot \left(1 - \sum_{j \in J} x'_{ij}\right) = v(\bar{x}').$$

This inequality follows from the definition of $L(\bar{u})$ and from the facts that

$$v(\bar{x}') = \sum_{i \in I} \sum_{j \in J} d_{ij} x'_{ij} \quad \text{and} \quad \left(1 - \sum_{j \in J} x'_{ij}\right) = 0.$$

To compute $L(\bar{u})$ for a given set of Lagrange multipliers, we proceed as follows:

Let

$$g_{ij} \equiv \begin{cases} (d_{ij} - u_i) & \text{if } d_{ij} - u_i < 0, \\ 0 & \text{otherwise.} \end{cases}$$

Considering each entity $j \in J$ in turn, a series of m *integer*-knapsack problems of the following form are solved [KNAP$_j$]:

$$G^j \equiv \min_{\bar{x}} \sum_{i \in I} g_{ij} x_{ij},$$

s.t.

$$\sum_{i \in I} w_i x_{ij} \leqslant W_j,$$

$$x_{ij} = (0, 1)$$

where index $j \in J$ is fixed for each of the m subproblems.

Letting $\bar{G} = [G^1, G^2, \ldots, G^m]$ be a vector of values obtained from the m solutions of [KNAP$_j$], we define the indicator set $J1 \subseteq J$ for the p smallest values in $\bar{G}$:

$$J1 \equiv \{ j \in J \mid j = \arg\min_k \bar{G} \text{ and } k = 1, \ldots, p \}$$

where

$\min_k \bar{G}$: kth smallest value of $\bar{G}$,

$\arg\min_k \bar{G}$: argument of $\min_k \bar{G}$

and

$$J2 \equiv \{J\}/\{J1\}.$$

The Lagrangian function can now be computed as:

$$L(\bar{u}) = \sum_{i \in I} u_i + \sum_{j \in J1} G^j.$$

The binary variables $y_{j \in J1}$ are set equal to 1, and the remaining $y_{j \in J2} = 0$. The x_{ij} variables for $j \in J1$ are equal to the solution of the jth knapsack problem [KNAP$_j$].

In the interest of obtaining an improved lower bound to $v(\bar{x}')$ a subgradient algorithm is used. The subgradient problem has the form [SGO]:

$$\max_{\bar{u}} \{ \min(z^f + \bar{u} \bar{v}^f) : f = 1, \ldots, F \}$$

where

$$z^f = \sum_{i \in I} \sum_{j \in J} d_{ij} x^f_{ij}, \qquad v^f_i = \left(1 - \sum_{j \in J} x^f_{ij}\right)$$

and where x^f_{ij} is the value of x_{ij} in the fth basic feasible solution. Solution strategies for [SGO] and computational experiences are discussed in [12,13,22]. These and other studies have shown convincingly that the subgradient method often provides close lower bounds on the optimal solution and hence is effective as a problem-specific performance measure for heuristics.

We employ a simple fathoming test during the subgradient algorithm. This procedure identifies those y_j that can be fixed at values 0 or 1 during the progress of the algorithm and is based on the following two theorems.

Theorem 1. *Variable $y_j = 0$ for the remainder of the subgradient steps if $\Delta^j_1 \geqslant \Delta_g$, where $\Delta_g \equiv v(\bar{x}_h) - L(\bar{u}_c)$, $v(\bar{x}_h)$ represents the incumbent solution value, $L(\bar{u}_c)$ represents the solution to the current Lagrangian problem* [LCCP], *and $\Delta^j_1 \equiv G^j - G^q$ where $q = \arg\min_p \bar{G}$.*

Proof. Suppose one appends the constraint $y_j = 1$ to the problem [LCCP], creating a candidate subproblem. The solution to this subproblem equals:

$$L'(\bar{u}) = \sum_{i \in I} u_i + \sum_{k \in J1'} G^k + G^j,$$

$$J1' \equiv \begin{cases} \{J1\} - \{q\} & \text{if } j \notin J1, \\ \{J1\} - \{j\} & \text{otherwise,} \end{cases}$$

where $q = arg \min_p \bar{G}$.
Note that this solution provides a lower bound for the restricted subproblem. If $\Delta_1^j \geqslant \Delta_g$ then setting $y_j = 1$ implies that the candidate subproblem is fathomed by Geoffrion and Marsten's criterion #2 (see [10]), i.e., the lower bound for the candidate subproblem exceeds the value of the incumbent. Hence $y_j = 0$ for the remainder of the subgradient steps.

An analogous theorem is proposed for fixing variable $y_j = 1$:

Theorem 2. *Suppose $j \in J1$, then $y_j = 1$ for the remainder of the subgradient steps if $\Delta_2^j \geqslant \Delta_g$ where Δ_g is defined as before and $\Delta_2^j = G^{q1} - G^j$, where* $q1 = \arg\min_{p+1} \bar{G}$.

Proof. Letting $y_j = 0$ in the restricted subproblem, this proof parallels the one given for Theorem 1.

Besides being an efficient mechanism for pegging variables and improving the bounds generated by the subgradient method, these two theorems motivate the usage of the primal algorithm in conjunction with the subgradient method. This topic is taken up next.

4. Combining primal heuristic and Lagrangian relaxation

With minor modifications, the primal heuristic described in Section 2 can be embedded within the [SGO] algorithm. This is accomplished by eliminating at each iteration the starting set of random points and substituting the entities targeted by the subgradient method, i.e., $\{J1\}$. Since Theorems 1 and 2 suggest that points in $\{J1\}$ are more likely to be medians in the optimal solution than those in $\{J2\}$, it seems appropriate to commence the primal heuristic with $\{J1\}$. This master-slave relationship between a primal heuristic and an optimization procedure has been proposed in a different context by Magnanti [19].

Given $\{J1\}$ as a starting solution, the heuristic attempts to locate a feasible assignment of points satisfying constraints (1.1)–(1.5). It then improves this solution through the usual iterative algorithm. The subgradient algorithm continues in this manner until the difference between the lower and upper bounds is less than some acceptable epsilon, say 1%, or until a predetermined iteration count is reached. If the final solution is deemed unacceptable the hybrid procedure can be easily placed within a tree enumeration (branch and bound) algorithm. Our experiences indicate that such a step is generally unnecessary.

5. Computational results

A series of computational tests was conducted to evaluate the performance of the primal heuristic and the combined algorithms. All test problems were derived from a random problem generator with the following characteristics. Given a problem with n data entities to be clustered into p distinct and mutually exclusive groups, an average number of entities per cluster was calculated from (n/p). A set of (n/p) data entities was then selected for each cluster. Each entity was described by two attributes drawn from the normal distributions $N(u_{1k}, s)$ and $N(u_{2k}, s)$ where

$$u_{1k} = 35\,000 + k'(500),$$

$$u_{2k} = 95\,000 + k'(500),$$

and

$$s = 250 \quad \text{for } k = 1, \ldots, p.$$

Capacity size constraints, W_k, were assumed to be equal to

$$1.2 \cdot \Big(\sum_{i \in I} w_i / p \Big) \quad \text{for clusters } k = 1, \ldots, p,$$

where weights, w_i, were drawn from the uniform distribution U(0, 1). Note that these capacity constraints are 20% in excess of the expected cluster size for a problem having data entities of uniform weight and spacing, thus resulting in difficult clustering problems. A Euclidean distance metric with two attributes was used for all test problems.

To assess the quality of generated solutions the following formula was employed:

$$\%\text{gap} = (V_H - V_S)/V_S$$

where V_H and V_S are the best solution values of the heuristic and subgradient procedures, respectively. In all cases, the subgradient algorithm was terminated when either a percentage gap of 0.5% was obtained or 150 iterations was reached. Integer knapsack subproblems were solved using a highly efficient program developed by Robert Nauss [24].

Table 1 presents typical results. Here, eighteen randomly generated problems ranging in size from 10 to 100 entities are solved. Included are execution times and solution values produced by the heuristic and subgradient procedures along with the percentage gap between the two solution values. Corresponding statistics for the combined method are also reported.

As seen in Table 1, the difference between the subgradient lower bound and the heuristic solution value (i.e., the % gap) is found to be within 1.1% for problems with $n < 50$. For solutions of problems having 50 and 100 entities the gap is found to range from a low of 0.43% to a high of 5.4%. These results show that the heuristic locates solutions of good quality for these difficult clustering problems. We have experienced similar behavior with real world clustering problems. Execution times for the heuristic procedure range from 0.27 seconds for a 10 entity problem to 35.5 seconds for a 100 entity problem.

The combined algorithm also identifies solutions of good quality for these problems. For fifteen of the eighteen problems tested, the gap is found to be within 0.5% while for the remaining three problems the gap is found to range between 1.1% and 3.6%. Execution times for the combined algorithm are relatively small, considering the computational complexity of the problems. CPU times range from a low of 0.10 seconds for a problem of size 10 to a high of 121.1 seconds for a 100 data entity problem.

These experiments show that both the heuristic and the combined algorithm yield almost identical and close to optimal solutions for model [CCP].

6. Summary and directions for future research

The capacitated clustering problem [CCP] has relevance to a number of application areas, includ-

Table 1
Function values and percent gap [a]

#	n	p	Subgradient			Primal heuristic procedure			Heuristic-subgradient algorithm				
			L.B.	No. iter.	CPU sec.	Solution value	% gap	CPU sec.	L.B.	U.B.	% gap	No. iter.	CPU sec.
1	10	2	21973	11	0.03	22082	0.49	0.27	21973	22082	0.49	11	0.10
2	10	2	24626	12	0.03	24732	0.42	0.30	24626	24732	0.42	12	0.12
3	10	2	23902	13	0.03	23995	0.38	0.28	23902	23995	0.38	13	0.11
4	15	3	29650	13	0.06	29761	0.37	0.60	29651	29761	0.37	13	0.20
5	15	3	36764	29	0.17	36945	0.49	0.52	36764	36945	0.49	29	0.58
6	15	3	32657	13	0.06	32780	0.37	0.54	32657	32780	0.37	13	0.19
7	20	4	44787	15	0.14	44993	0.46	0.95	44811	44993	0.40	17	0.45
8	20	4	64777	150	1.5	65515	1.1	0.86	64777	65515	1.1	150	3.7
9	20	4	53401	16	0.13	53614	0.39	0.95	53401	53614	0.37	16	0.47
10	25	5	59687	69	1.0	59971	0.47	1.28	59687	59971	0.47	68	2.15
11	25	5	72886	62	0.87	73244	0.49	1.39	72884	73244	0.49	60	2.65
12	25	5	61321	53	0.72	61614	0.46	1.44	61329	61614	0.46	53	2.25
13	50	10	107001	28	1.2	107427	0.37	6.67	107030	107427	0.37	28	5.3
14	50	10	114335	46	6.8	114851	0.45	7.23	114332	114851	0.45	46	7.9
15	50	10	126052	150	6.8	132893	5.4	6.65	126139	130571	3.5	150	27.3
16	100	20	235731	150	24.2	237781	0.87	33.5	237335	245870	3.6	150	121.5
17	100	20	254818	30	24.2	255980	0.43	35.3	254871	255980	0.43	30	44.8
18	100	20	237494	59	33.8	238651	0.48	33.3	237997	238651	0.48	59	50.0

[a] Fortran H Compiler, 1 Meg High Speed memory, IBM 3033 Computer at Princeton University.

ing salesforce allocation, vehicle routing, and stratified sampling. We have presented two effective algorithms for solving model [CCP]. Computational testing has indicated that the heuristic embedded within a subgradient algorithm, in most cases, produces solutions whose objective values is within one percent of the optimum. The heuristic as a stand-alone solution procedure also appears to produce close to optimal solutions.

Additional research in the development and testing of solution strategies for problem [CCP] is required. One attractive option is the dual ascent method. Erlenkotter [4] and Fisher et al. [5] have achieved considerable success with the use of this method for solving uncapacitated facility location and generalized assignment problems, respectively. An interesting research question is whether the dual ascent method can be successfully applied to [CCP]. On another front, Gavish [8] has proposed a linear programming formulation for the dual of [CCP] whose solution gives optimal multipliers for relaxed binary integer programming problems. Integration of Gavish's 'best' multiplier technique with the hybrid heuristic-subgradient algorithm represents another research topic.

For the solution of larger [CCP] problems (i.e., $n \geqslant 500$), enhancements to the heuristic-subgradient algorithm are required. Creating a 'working' database which contains a small subset of the total problem data is one strategy. This approach would limit searches to the k closest entities in $\{J\}$ rather than all m data entities. Periodically, data in the active set would be revised; some entities would be added to the active set while others would be deleted. Application of this 'large scale restriction' strategy to large uncapacitated clustering problems has yielded good results. Its applicability to [CCP] has not yet been fully explored.

Appendix

Algorithm CAPCLUST

```
input:          $\epsilon$ = small number for minimum objective function improvement
                $I$ = set of $n$ data entities
                $p$ = desired number of median clusters
                $D$ = distance or similarity matrix having elements $d_{ij}$
                $w_i$ = weight of the $i$th data entity
                $W_j$ = cluster capacity for $j$th cluster
          max_limit = no. of major iterations
output:    $X\_Best$ = assignment matrix: $x_{ij} = 1$ if data entity $i$ is assigned to median $j$, $x_{ij} = 0$ otherwise.
begin
set $max\_count = 1$;
set $best\_soln = \infty$
while ($max\_count < max\_limit$) do
    select set of $p$ random median $\{J_p\}$ from $\{J\}$
    $flag = 0$
    while ($flag = 0$) do
        apply $order(\{J_p\}, \{J_{ip}\})$
        for $i \in I$ do
            set $regret_i = |J_{i1} - J_{i2}|$;
        end;
        (comment: assign data entities to medians in decreasing order of regret)
        set $curr\_cap_p = 0$, $p \in \{J_p\}$
        for decreasing $i \in \{regret_i\}$ do
            set $l = 1$;
            while $i$ not assigned do
                if $curr\_cap_{J_{il}} + w_i \leqslant W_{J_{il}}$
                then set $x_{iJ_{il}} = 1$
                    set $curr\_cap_{J_{il}} = curr\_cap_{J_{il}} + w_i$
          **else** set $l = l + 1$;
        **end**;
      **end**;
      (**comment**: compute improved set of medians)
    set $J_{Temp} = \emptyset$;
    set $total = 0$;
    set $total\_last = 0$;
    **for** $p \in \{J_p\}$ **do**
      set $best = \infty$;
      **for** i $\in \{x_{ip} = 1\}$ **do**
        set $d\_new = \sum_{k \in I} d_{kp} x_{kp}$;
        **if** $(d\_new < best)$
        **then** set $best = d\_new$
            set $best\_i = \{i\}$
      **end**;
        **for** $i \in \{x_{ip} = 1\}$ **do**
          *set* $x_{ip} = 0$;
          *set* $x_{i,best\_i} = 1$
          *set* $total = d\_new + total$;
        **end**;
        set $J_{\text{Temp}} = \{J_{\text{Temp}}\} \cup \{best - i\}$;
      **end**;
      **if** $(\{J_p\} = \{J_{Temp}\}) \wedge (total\_last - total > \epsilon)$
      **then** set $total\_last = total$
        set $flag = 1$
        apply $switch(\{J_p\}\ curr\_cap,\ X,\ W_j,\ total)$
      **else** *set* $\{J_p\} = \{J_{Temp}\}$;
      **end**;
      set $max\_count = max\_count + 1$;
      **if** $(total < best\_soln)$
      **then** set $best\_soln = total$;
        set $X\_best = X$;
    **end**;
    **end procedure;**
```

```
**Procedure** $order(\{J_p\}, \{J_{ip}\})$
(**comment**: order the closeness of medians to each data entity)
*input*: $\{J_p\}$ = current set of medians
*output*: $\{J_{ip}\}$ = a matrix of values representing the *p*th closest median for data entity $i$
**begin**
  **for** $i \in \{I\}$ **do**
    $J_{ip} = p \in \{J_p\}$ sorted is decreasing order of closeness
  **end**;
**return**;
```

```
**Procedure** $switch(\{J_p\}, curr\_cap, X, W, total)$
(**comment**: attempt improved solution via switching of data entities)
*input* $\{J_p\}$ = set of medians
  $curr\_cap$ = current capacities for $p \in \{J_p\}$ clusters
          $X$ = current assignments
         $W_j$ = cluster capacity restrictions
       $total$ = current objective function value
```

```
output: curr_cap = revised $j \in \{J\}$ cluster capacities
               X = revised assignments
           total = revised objective function value
begin
for $p \in \{J_p\}$ do
  for $i \in (x_{ip} = 1)$ do
    for $k \in (\{J_p\} - \{p\})$ do
      for $j \in (x_{jk} = 1)$ do
        if $(d_{ik} + d_{jp}) < (d_{ip} + d_{jk}) \wedge$
          $(curr\_cap_k + w_i - w_j < W_k) \wedge$
          $(curr\_cap_p + w_j - w_i < W_p)$
        then set $x_{ik}, x_{jp} = 1$;
             set $x_{ip}, x_{jk} = 0$;
             set $curr\_cap_k = curr\_cap_k + w_i - w_j$;
             set $curr\_cap_p = curr\_cap_p + w_j - w_i$;
             set $total = -d_{ip} - d_{jk} + d_{ik} + d_{jp} + total$;
      end;
    end;
  end
end;
return;
```

References

[1] Anderberg, M.R., *Cluster Analysis for Applications*, Academic Press, New York, 1973.

[2] Beck, M.P., and Mulvey, J.M., "Constructing optimal index funds", Princeton University Report No. EES-82-1, 1982.

[3] Chatfield, C., and Collins, A.J., *Introduction to Multivariate Analysis*, Chapman and Hall, London, 1980.

[4] Erlenkotter, D., "A dual-based procedure for uncapacitated facility location", *Operations Research* 26 (6) (1978) 992–1008.

[5] Fisher, M.L., Jaikumar, R., and Van Wassenhove, L.N., "A multiplier adjustment method for the generalized assignment problem", Decision Sciences Working Paper, University of Pennsylvania, 1981.

[6] Fisher, M.L., "The Lagrangian relaxation method for solving integer programming problems", *Management Science* 27 (1) (1981) 1–17.

[7] Garey, M.R., and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[8] Gavish, B., "On obtaining the 'best' multipliers for a Lagrangian relaxation for integer programming", *Computers and Operations Research* 5 (1978) 55–71.

[9] Geoffrion, A.M., and McBride, R.C., "Lagrangian relaxation applied to capacitated facility location problems", *AIIE Transactions* 10 (1978) 40–47.

[10] Geoffrion, A.M., and Marsten, R.E., "Integer programming algorithms: A framework and state-of-the-art survey", *Management Science* 18 (7) (1972) 465–491.

[11] Hartigan, J.A., *Clustering Algorithms*, Wiley, New York, 1975.

[12] Held, M., and Karp, R.M., "The traveling salesman problem and minimum spanning trees: part II", *Mathematical Programming* 1 (1) (1971) 6–26.

[13] Held, M., Wolfe, P., and Crowder, H.P., "Validation of subgradient optimization", *Mathematical Programming* 6 (1974) 62–88.

[14] Hess, S.W., and Samuels, S.A., "Experiences with a sales districting model: Criteria and implementation", *Management Science* 18 (4) (1971) 41–54.

[15] Hess, S.W., "Realigning districts—By computer", *Wharton Quarterly* (1969) 25–30.

[16] Jarvinen, P., Rajala, J., and Sinerva, H., "A branch-and-bound algorithm for seeking the p-median", *Operations Research* 20 (1972) 173–178.

[17] Klastorin, T.D., "An effective subgradient algorithm for the generalized assignment problem", Working Paper, University of Washington, 1977.

[18] MacQueen, J.B., "Some methods for classification and analysis of multivariate observations", Proc. *5th Berkeley Symposium on Mathematical Statistics and Probability (Vol. I)*, University of California Press, 1967.

[19] Magnanti, T.L., "Combinatorial optimization and vehicle fleet planning: Perspectives and prospects", *Networks* 11 (2) (1981) 179–213.

[20] Morrison, D.F., *Multivariate Statistical Methods*, McGraw-Hill, New York, 2nd ed., 1976.

[21] Mulvey, J.M., "Multivariate stratified sampling by optimization", *Management Science* 29 (6) (1983) 715–724.

[22] Mulvey, J.M., and Crowder, H.P., "Cluster analysis: An application of Lagrangian relaxation", *Management Science* 25 (4) (1979) 392–340.

[23] Murphy, C.M., and Ignizio, J.P., "A model and heuristic solution for multi-criteria network partitioning", Pennsylvania State University Working Paper, 1981.

[24] Nauss, R.M., "An efficient algorithm for the 0–1 knapsack problem", *Management Science* 23 (1) (1976) 27–31.

[25] Nauss, R.M., "An improved algorithm for the capacitated facility location problem", *Journal of the Operational Research Society* 29 (12) (1978) 1195–1202.

[26] Neebe, A.W., "A branch and bound algorithm for the *p*-median transportation problem", *Journal of the Operational Research Society* 29 (10) (1978) 989–995.

[27] Sa, G., "Branch-and-bound and approximate solutions to the capacitated plant-location problem", *Operations Research* 17 (1969) 1005–1016.

[28] Shanker, R.J., Turner, R.E., and Zoltners, A.A., "Sales territory design: An integrated approach", *Management Science* 22 (3) (1975) 309–320.

[29] Shapiro, J.F., "A survey of Lagrangian techniques for discrete optimization, in: P. Hammer, E. Johnson and B. Korte (eds.), *Discrete Optimization*, Annals of Discrete Mathematics 5, North-Holland, Amsterdam, 1979, 113–138.