Computational Economics **9:** 229–239, 1996.

Clustering Problems in Optimization Models

SANTOSH KABADI
Business School, University of New Brunswick, Fredericton, New Brunswick, Canada E3A 1K8
KATTA G. MURTY
Dep. of IOE, Univ. of Michigan, Ann Arbor, MI 48109-2117, USA, and Hong Kong, University of Science and Technology
COSIMO SPERA*
Dep. of Quantitative Methods, Univesity of Siena, p.zza San Francesco, 53100 Siena, Italy

(Accepted 8 September 1995)

Key words: Combinatorial optimization, clustering, classifying objects, classification criteria, modeling

Abstract. We discuss a variety of clustering problems arising in combinatorial applications and in classifying objects into homogenous groups. For each problem we discuss solution strategies that work well in practice. We also discuss the importance of careful modelling in clustering problems.

1. Introduction

The input for a clustering problem is a set of objects, each object usually comprising data measuring r (greater than or equal to one) of its relevant characteristics. The desired output is a partition of the set of objects into disjoint clusters (also called classes or groups) satisfying certain constraints on their cardinalities that either minimize an objective function (for clustering in optimization problems), or to make each cluster as homogeneous as possible (for clustering in classification problems).

Clustering is an extremely important part of quantitative methods in many applied sciences. Indeed we show here that clustering is the main component of many combinatorial optimization problems. We then discuss some important clustering problems and algorithms that solve them with practical efficiency. Finally we show a clustering problem that yields strange results to help demonstrate the importance of careful modeling for getting results that make sense.

2. Clustering problems in combinatorial optimization models

Many combinatorial optimization problems involve finding a partition of a set into nonempty subsets satisfying certain conditions. Such problems can usually be interpreted as clustering problems. We illustrate with several examples.

* Part of this work was carried out while the author was visiting the IOE Department, University of Michigan on a CNR fellowship. Author for correspondence

2.1. A TASK ALLOCATION PROBLEM

A task allocation problem arises in determining a minimum cost design for the microcomputer architecture in an automobile design of the future [1, 5]. For this vehicle, many tasks, such as the monitoring of the integrated chasis and the active suspension, will be performed by microcomputers linked by high- and/or low-speed communication lines. The system cost is the sum of the costs of the processors (microcomputers) and the data links that provide the inter-processor communication bandwidth. Each task processes data coming from sensors, actuators, and signal processors, digital filters and has a throughput requirement in KOS (kilo operations/second). Several types of processors are available. For each, we are given its cost, the maximum number of tasks it can handle, and its throughput capacity in terms of KOS. All the tasks must be processed within one time cycle, and the load on any processor cannot exceed its capacity.

The tasks are inter-dependent. To complete one task, we may need data from another. So tasks allocated to different processors may need communication links, while tasks executing in the same processor do not need this communication overhead. The problem is to partition the set of tasks into groups (or clusters), with each group assigned to a processor, so as to satisfy all the constraints with minimum system cost. This problem is a clustering problem.

The number of tasks, n, typically varies between 50 and 100. The number of processors, m (the number of clusters to be formed), is around 6. Integer programming formulations of this problem involve $m\,(n^2 + 1)$ binary variables [1, 5] and are difficult to solve with currently available software. Problems having $n = 20$ tasks and $m = 7$ processors can run for a week or more on today's fastest workstations and need not lead to satisfactory solutions. On the other hand the specially designed genetic algorithm in [1, 5] produces a very satisfactory solution to this problem in a reasonable time.

2.2. TRAINING CENTER LOCATION PROBLEMS

Telephone companies and national and multi-national food chains (e.g. McDonald's) have a steady demand for training new employees for different locations. Suppose a company has offices in n locations in need of trained employees and that it has decided to set up at most p training centers. Once the training centers are established, new employees from each location will be sent to one for training. In typical applications n is usually ≥ 500, and p is of the order of 3 or 4. The problem is to partition the set of n locations into p groups or clusters, each cluster to be handled by a single training center, and to find the site where the training center for each cluster should be set up. The objective function typically is to minimize the sum of the annual operating costs of the training centers and the total annual travel costs for the trainers. Such problems can also be interpreted as clustering problems.

The zero-one integer programming formulation of this problem leads to a P-median type location model [3]. In practice, large scale models of this type can be solved quite easy using commercial integer programming software. Local search methods, such as exchange or interchange heuristics, can also produce excellent solutions for such problems [3].

2.3. RELATED PROBLEM IN DRILLING FOR OIL OFFSHORE

Clustering problems are also relevant to the development of offshore oil fields. Here, exploratory wells are drilled to help discover new fields. Then step-out-wells, drilled from mobile drilling rigs are used to determine the size and other characteristics of the field. The data obtained from this activity are used to decide the location of production wells, or targets. A typical 3 × 3 mile offshore field would have between 25 and 300 production wells. The drilling of production wells is carried out from fixed platforms that are placed on the ocean floor. The cost of drilling a production well depends on the length and the angle of the hole drilled from the fixed platform to the target. Exact data on these costs are very hard to obtain; they have to be approximated using information from past drilling jobs.

Setting up fixed platforms is very expensive. The cost of a fixed platform depends on the water depth and bottom conditions at its location and on its size, which is measured by the number of production wells to be drilled from it. Size can vary from 6 to 25. The problem is to partition the set of production wells into clusters or groups of size 6 to 25, each group to be drilled from a single fixed platform to be set up. Also, the best location for each such platform has to be determined. The objective is to minimize the sum of the costs for setting up the fixed platforms for drilling the production wells. This is again a clustering problem.

This important practical problem involves large sums of money, but the cost data can only be estimated. It can be viewed as a two stage problem: in stage 1 the locations for the fixed platforms are selected; in stage 2 the production wells are allocated to fixed platforms where locations have been determined. Solution approaches for this problem usually iterate between stages, using integer programming and network flow models, until a good plan is obtained.

2.4. THE ARMY'S M-CCTT LOCATION AND ROUTING PROBLEM

This next problem involves clustering in two hierarchical stages. It arises training the Reserve Component of the US Army (RC) on Combat Vehicle simulators, called Mobile-Close Combat Tactical Trainers (M-CCTT). RC units are widely distributed in many villages, towns, and cities throughout the U.S. The RC hold regular jobs outside the army. As part of their Army Reserve Commitment, they agree to several weekends training each year to keep their skills sharpened.

To prevent occupying too much of the weekend in travel, the Army provides training to each RC unit at a training station that is to be no farther than a Maximum

Travel Distance (MTD), (Currently 100 miles) from where they live. The word *armory* refers to a place (village, town, or city) where RC units to be trained are residing. There are about $n = 400$ armories geographically dispersed across the U.S. The word *site* refers to a place suitable for being a home base (HB) for an M-CCTT. The Army has identified about 30 such M-CCTT sites but plans to set up at most $p = 20$ of them. Each M-CCTT costs several million dollars to establish, so it is important to find the minimum number to do the job.

At the top level, the problem is to partition the n armories into at most p clusters, each cluster to be trained at an M-CCTT. And for each cluster, one must find the best site to be the HB for the M-CCTT that will train the RC residing at an armory in the cluster. Once the clusters are formed and the HB to service each cluster is determined, all the RC at armories within each cluster which are within the MTD of their HB go to it for training. However, a cluster may contain armories that are farther than the MTD from their HB.

At the second level, it is necessary to form the armories into subclusters and to select a Secondary Training Site (STS) within each subcluster to train that subcluster. Each armory in a subcluster should be within the MTD of its STS. The M-CCTT stationed at the HB for the cluster will then travel to each of the STS to give training to RC units within its subcluster. As an example, we show below in Figure 1 the HB for an M-CCTT by a star (it is Camp Bowie in Texas), three STS that this HB travels to by square nodes, and armories that are serviced by one of these four places by circle nodes, from the solution developed in [6] to this problem.

There are several objective functions. The one of highest importance is to minimize the number of M-CCTT to be stationed. The next most important objective

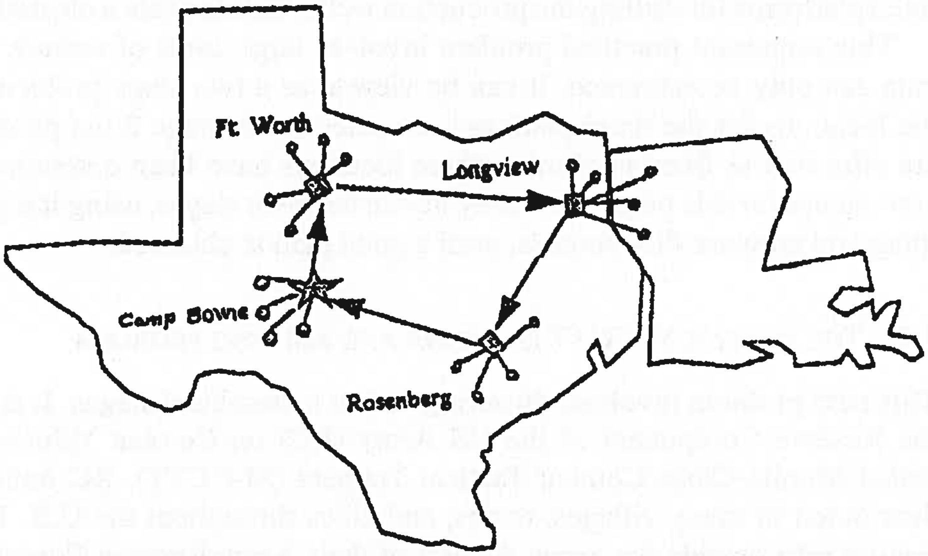

Figure 1. Army's M-CCTT location and routing problem. The location of a HB for an M-CCTT is shown using the star-node. This M-CCTT travels to three different STS indicated by square nodes. RC-armories who get training at these various places are indicated by small circle nodes.

function is to minimize the total mileage of the M-CCTT fleet in traveling to their STS's, since the cost per mile of moving a M-CCTT is very high. Finally, the third objective is to minimize the total bus mileage of all the RC units to get to their assigned training places. The whole problem was solved in [6], using P-median, set covering, and multi-depot vehicle routing models. It leads to a nice solution saving the Army several millions of dollars.

3. Clustering problems in classification

The problem of classifying objects into homogeneous clusters appears commonly in all sciences. In such studies, one also often needs simple but reliable criteria for classifying objects that may arrive in future. One technique for developing such criteria is presented in section 3.2.

3.1. BREAST CANCER DIAGNOSIS USING BREAST CYTOLOGY

A commonly used technique for checking for breast cancer analyzes breast masses from fine needle aspirates (FNA). For each breast FNA, specific features are measured for each nucleus. These are: size (area, radius, perimeter), symmetry, number and sizes of cavities, fractal dimension of the boundary, smoothness (local variation of radial segments), texture (variance of gray levels inside the boundaries). In using these data, it is required to classify breast FNA into three classes:

- malignant;
- suspicious (need to be tested again in six months);
- non malignant.

The percent of misclassification should be as small as possible.

Mangasarian and his associates [4] develop criteria for this classification based on a piecewise linear function developed using a neural network approach. This technique is being used successfully for breast cancer diagnosis at the General Hospital, University of Wisconsin at Madison.

3.2. DEVELOPING A SIMPLE CLASSIFICATION RULE BASED ON A LINEAR FUNCTION

Suppose we have data for k different characteristics measured on n objects reliably known to belong to a specific cluster or group or population. Let $x^j = (x^j_1, \ldots, x^j_k)^T$ be the vector of measurements of characteristics $1, \ldots, k$ on objects j, for $j = 1$ to n.

Normally people use classification criteria only if they are simple. For this reason, suppose it is desired to find a linear function of the k characteristics that best characterizes this group to use for on-line classification of objects appearing in the future. This requires finding two parallel hyperplanes H_1 and H_2 in $\mathcal{R}^k$, separated by the smallest distance possible, that together contain all the n points $x^1, \ldots, x^n$. See Figure 2 for an illustrative example involving measurement on

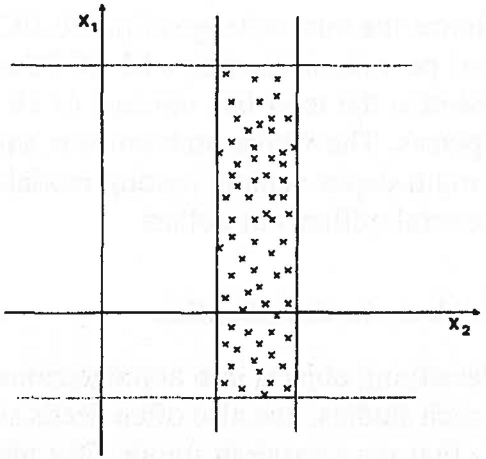

Figure 2. Pairs of hyperplanes containing the set of data. Two pairs of hyperplanes (the dashed pair, the solid pair) containing all the points representing a population between them. The solid pair has the smallest distance separating them, so best characterizes the population.

$k = 2$ characteristics on a group of objects represented by points plotted on the two-dimensional Cartesian plane.

Suppose such an optimal pair of hyperplanes in $\mathcal{R}^k$ has been found, and these hyperplanes are represented by the linear equations $a_1x_1 + \cdots + a_kx_k = d_1$ and $a_1x_1 + \cdots + a_kx_k = d_2$, respectively, where, without loss of generality, we assume $d_2 \geq d_1$. The distance between these hyperplanes is $\Delta = (d_2 - d_1)/\sqrt{a_1^2 + \cdots + a_k^2}$. If Δ is reasonably small, we can use the following criteria to classify whether an object with measurements $(x_1, \ldots x_k)^T$ belongs to the group: thus $(x_1, \ldots, x_k)^T$

belongs to the group if $\rightarrow d_1 \leq a_1x_1 + \cdots + a_kx_k \leq d_2$
does not belong to the group $\rightarrow$ otherwise.

If Δ is large, it would not seem reliable to characterize this group using simple bounds on a linear function of the k characteristic measurements.

The problem of finding an optimal pair of hyperplanes leads to the following nonlinear program: find $a_1, \ldots, a_k, d_1, d_2$ to

$$\begin{aligned} \text{minimize} \quad & f = (d_2 - d_1)/\sqrt{a_1^2 + \cdots + a_k^2} & \\ \text{subject to} \quad & d_2 \geq d_1, & (1) \\ & d_1 \leq (a_1, \ldots, a_k)x^j \leq d_2, \qquad j = 1 \text{ to } n & (2) \\ & (a_1, \ldots, a_k) \neq 0. & (3) \end{aligned}$$

This is a nonconvex, nonlinear program; in particular the constraint (3) makes it difficult.

One approach for solving this problem eliminates constraint (3) and solves k separate problems. For $t = 1$ to k, the t-th problem is to minimize f subject to *only* (1) and (2) fixing $a_t = 1$, but leaving the other variables free. We take the best of the solutions to these k problems. While each of these k problems is a linearly constrained nonconvex, nonlinear program, we find that very satisfactory results can be obtained using commercial nonlinear programming software.

In practical applications, one may have data on n objects, but may not be absolutely sure that they all belong to the group, i.e., there may be some outliers. In this case one can select a target fraction $\lambda(\lambda = .95$ or $.99$ is typical) and find the optimal parallel hyperplane pair that includes at least the fraction λ of data points between them. To model this problem, we introduce the binary variables, for $j = 1$ to n,

$$y_i = \begin{cases} 1 & \text{if the } j\text{-th object is in between the two parallel hyperplanes} \\ 0 & \text{otherwise.} \end{cases}$$

Then for $t = 1$ to k, the t-th problem discussed above, modified to ensure that at least a fraction λ of data points lies between the parallel hyperplanes is: find $a_1, \dots, a_k, d_1, d_2$ to

$$\begin{array}{ll} \text{minimize} & f = (d_2 - d_1)/\sqrt{a_1^2 + \cdots + a_k^2} \\ \text{subject to} & d_2 \geq d_1, \\ & d_1 y_j - \alpha(1-y_j) \leq (a_1, \dots a_k)x^j \leq d_2 + \alpha(1-y_j), \quad j = 1, \dots, n \\ & a_t = 1, \\ & y_1 + \cdots + y_n \geq \lambda n, \\ & y_j = 0, 1, \qquad \text{for } j = 1 \text{ to } n, \end{array}$$

where α is a large positive number. This is a nonlinear integer program. Even though software for solving nonlinear integer programs is not available commercially, we found that satisfactory solutions can be obtained using research software if n is not large. Of course we take the best solution from among those obtained for the k problems. Examining solutions based on different values for the fraction λ may lead to useful information.

4. Importance of a careful modeling

For developing criteria for classifying objects into homogeneous clusters, one often uses models that try to minimize measures of within-cluster variation. These measures and associated constraints must be chosen carefully if the results are to make sense. We illustrate this with a model that leads to strange results.

4.1. A CLUSTERING MODEL BASED ON MEASUREMENTS OF A SINGLE CHARACTERISTIC

Suppose we have data on the measurements of a single important characteristic of n objects. From practical considerations, suppose we know that these objects belong to k distinct groups (here $n \gg k$). To develop classification criteria for these groups based on the data, one is faced with the problem of partitioning the set of n objects into k nonempty disjoint clusters that are as homogeneous as possible.

Let $a_1, \dots, a_n$ be the measurements on the n objects arranged in increasing order (i.e., $a_1 \leq a_2 \leq \cdots \leq a_n$). Given a subset S of these objects, a commonly used measure of nonhomogeneity is

$$f(S) = \sum (|a_i - a_j| : \text{ over pairs of objects } i, j \in S).$$

So, if $S_1, \dots, S_k$ is a partition of objects $1, \dots, n$ into k nonempty disjoint clusters, we could use $f(S_1) + \cdots + f(S_k)$ as a measure of nonhomogeneity of the set of clusters.

Now consider the problem of forming objects $1, \dots, n$ into k nonempty disjoint clusters to minimize this measure. This leads to a 0–1 integer programming model with the following decision variables:

$$\text{for } i = 1 \text{ to } n, t = 1 \text{ to } S, y_{it} = \begin{cases} 1 & \text{if object } i \text{ is in cluster } t \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{for } 1 \leq i \leq j \leq n, x_{ij} = \begin{cases} 1 & \text{if object } i, j \text{ in some cluster} \\ 0 & \text{otherwise.} \end{cases}$$

Then the model is:

$$\begin{array}{lll} \text{minimize} & \sum_{i=1}^{n} \sum_{j=i+1}^{n} (a_j - a_i) x_{ij} & \\ \text{subject to} & \sum_{t=1}^{k} y_{it} = 1, & \text{for } i = 1 \text{ to } n \\ & \sum_{i=1}^{n} y_{it} \geq 1, & \text{for } t = 1 \text{ to } k \\ & y_{it} + y_{jt} - xij \leq 1, & \text{for } t = 1 \text{ to } k, 1 \leq i \leq j \leq n \\ & \text{all variables are 0 or 1.} & \end{array}$$

We found that this model is quite easy to solve, even for large values of n, with available integer programming software. We discuss the results with two examples.

4.2. Partitioned optimal cluster

Example 1. Suppose $k = 2, n = 8$, and the measurements are 98, 100, 103, 105, 150, 151, 155, 160. The optimal clusters are {98, 100, 103, 105} and {150, 151, 155, 160} shown in Figure 3. The intervals spanned by the clusters are disjoint. This agrees with out intuitive judgement of what the clusters should be. As a classification rule for other objects that arrive in future, we can define a criterion

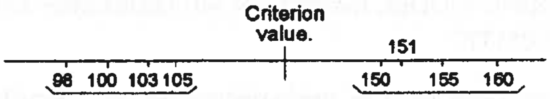

Figure 3. The optimal clusters.

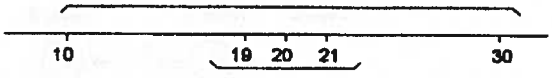

Figure 4. The optimal clusters.

value of 127.5 and put objects having measurement $\leq$ 127.5 in the lower group and those having measurement > 127.5 in the higher group.

4.3. NOT PARTITIONED OPTIMAL CLUSTER

Example 2. Suppose $k = 2, n = 5$, and the measurements are 10, 19, 20, 21, 30. The optimal clusters are {10, 30} and {19, 20, 21}, see Figure 4. The intervals spanned by the clusters are not disjoint, but they are nested (i.e., one is a subset of the other). This is very counterintuitive, and does not lead to a rule for classifying future objects.

Indeed Boros and Hammer [2] prove that for an optimal clustering in this problem, the intervals spanned by the clusters are nested (i.e., one is a subset of the other). They also prove that if the intervals spanned by the cluster are not nested, then they are disjointed. When the intervals spanned by the clusters are nested, the optimal clustering obtained does not lead to a simple rule for classifying future objects.

4.4. A MODIFICATION OF THE CLUSTERING MODEL DISCUSSED IN SECTION 4.3

We have seen that when the clustering model discussed in section 4.1 produces clusters with intervals that are not disjoint, the results are not useful for developing rules for classifying future objects, see the example in section 4.3. A way around this difficulty is to constrain the intervals spanned by the constraints to be disjoint. This leads to the following problem:

Input: objects $\{1, \ldots, n\}$ with measurements $a_1, \ldots, a_n$ in increasing order; $k =$ number of clusterd desired, $k < n$.

Output needed: form the n objects into k nonempty clusters, $S_1, \ldots, S_k$, minimizing $f(S_1) + \cdots + f(S_k)$ while satisfying the property that the intervals spanned by them are disjoint.

This model leads to an optimum set of clusters useful for developing rules for classifying future objects. We show that this problem can be posed as a shortest chain problem and solved efficiently.

4.5. THE SHORTEST CHAIN FORMULATION

To derive the shortest chain formulation of the cluster problem in section 4.2, we construct the relative network $\mathcal{G}(V, E)$, in the following way:

a) Define the set V.

V is partitioned into $k+2$ disjointed sets $V = V_0 \cup V_1 \cup \cdots \cup V_{k+1}$, each representing a layer. V_0 and V_{k+1}, being, respectively, the first and the last layer, have only one

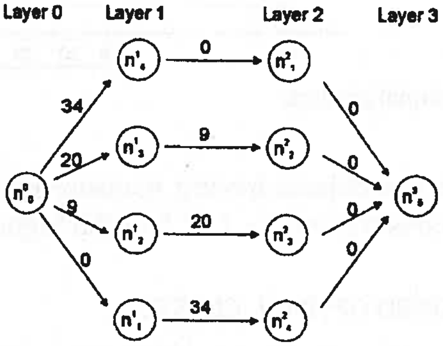

Figure 5. Network constructed on the example 2.

node. The other layers may have at most n nodes, being n the number of objects. $n_0^0 \in V_0$ is the source node; $n_0^{k+1} \in V_{k+1}$ is the destination node. The notation for the nodes has a superscript and a subscript. The first indicates to which layer it belongs, the second the number of elements in the partitioned cluster.

b) Define the set E.

E is the set of the arcs in the network. Arcs are directed and defined as $E : V_j \times V_{j+1} \rightarrow \mathcal{R}^+, j : 0, \ldots, k$. The network is $(k+2)$-partite, and the weights of the arcs represent the contribution given to the objective function by a possible partition. To compute these weights, indicated as $w(P_{|j|}), j = 1, \ldots, n$, we may use the following recursive formula:

$$
\begin{aligned}
&w(P_{|j|}) = (j-1)(\max_{a_j \in P_{|j|}} a_j - \min_{a_j \in P_{|j|}} a_j) + w(P_{|j-2|})\\
&w(P_{|1|}) = 0\\
&w(P_{|2|}) = a_i - a_j, \qquad a_i > a_j
\end{aligned}
$$

where $P_{|j|}$ is the partition having $|j|$ consecutive elements.

The network based on the data of the example 2 in section 4.3 is drawn in Figure 5. Arcs from layer 2 to layer 3 (generally from layer k to layer $(k+1)$) have weights equal to zero. Arcs from layer 0 to layer 1 represent the cost of the first partition having, respectively, (from the top to the bottom of Figure 5) 4, 3, 2, 1 elements; arcs from layer 1 to layer 2 represent the cost of the second partition having, respectively, 1, 2, 3, 4 elements.

No further arcs occur for the network in Figure 5 because no other partitions are allowed. In fact, if we add an arc from node n_3^1 to n_3^2, and if the optimal shortest chain solution includes this arc, we exceed the number of elements in our data set. If we add an arc from node n_3^1 to n_1^2 and, if the optimal shortest chain solution includes this arc, an element is left out of the two clusters. Therefore it is clear that the shortest chain from the source node to the destination node determines the optimal partition. Again we point out that this may not be the optimal cluster. Being the network acyclic from the operation research literature, see [7], we know that the optimal shortest chain can be computed with time complexity $O(|E|)$ by a specialised algorithm. The algorithm takes the name of *"reaching"*.

5. Summary

We discuss a variety of clustering models that arise in applications of combinatorial optimization and in classifying objects into homogeneous groups. Most of these models lead to NP-hard problems. We also discuss solution strategies that work well in practice, based on integer programming software, local search heuristics such as the interchange heuristic, or specially designed genetic-algorithm or neural-network methods. We emphatize the importance of a careful modeling.

References

1. Ben Hadj-Alouane, A., J.C. Bean and K.G. Murty, "A Hybrid Genetic/Optimization Algorithm for a Task Allocation Problem", IOE Department, University of Michigan, Ann Arbor, 1994.
2. Boros, A. and P. Hammer, "Optimal Clustering", Discrete Mathematics, special issue in honor of 70th birthday of P. Erdos, 1989.
3. Klincewicz, J.G., "Locating Training Facilities to Minimize Travel Costs", Bell Labs Technical Report, Holmdel, NJ, 1980.
4. Mangasarian, O.L., R. Setiono and W.H. Wolberg, "Pattern Recognition via Linear programming: Theory and Application to Medical Diagnosis". In T.F. Coleman and Y. Li (eds), *Large-Scale Numerical Optimization*, pp. 22–31, Philadelphia, Pennsylvania, 1990. SIAM. *Proceedings of the Workshop on Large-Scale Numerical Optimization*, Cornell University, Ithaca, New York, October 19–20, 1989.
5. Murty, K.G., *Operation Research: Deterministic Optimization Models,*, Prentice Hall, Englewood Cliffs, NJ, 1995.
6. Murty, K.G., P.A. Djang and B.B. Scott, "US Army's M-CCTT Location and Routing Problem", White Sands Missile Range, NM, 1995.
7. Murty, K.G., *Network Programming*, Prentice Hall, Englewood Cliffs, NJ, 1992.