J Glob Optim (2009) 45:111–129
DOI 10.1007/s10898-008-9393-8

Mathematical modeling and efficient optimization methods for the distance-dependent rearrangement clustering problem

Scott R. McAllister · Peter A. DiMaggio Jr. · Christodoulos A. Floudas

Received: 9 December 2008 / Accepted: 17 December 2008 / Published online: 6 January 2009

Abstract In this article we present a computational study for solving the distance-dependent rearrangement clustering problem using mixed-integer linear programming (MILP). To address sparse data sets, we present an objective function for evaluating the pair-wise interactions between two elements as a function of the distance between them in the final ordering. The physical permutations of the rows and columns of the data matrix can be modeled using mixed-integer linear programming and we present three models based on (1) the relative ordering of elements, (2) the assignment of elements to a final position, and (3) the assignment of a distance between a pair of elements. These models can be augmented with the use of cutting planes and heuristic methods to increase computational efficiency. The performance of the models is compared for three distinct re-ordering problems corresponding to glass transition temperature data for polymers and two drug inhibition data matrices. The results of the comparative study suggest that the assignment model is the most effective for identifying the optimal re-ordering of rows and columns of sparse data matrices.

Keywords Clustering · Mixed-integer linear programming · Sparse data sets

1 Introduction

Problems of data organization and data clustering are prevalent across a number of different disciplines. These areas include pattern recognition [1], image processing [2], information retrieval [3], microarray gene expression [4], and protein structure prediction [5,6] to name a few. The goal of data clustering, regardless of the application, is to organize data in such a way that the similar data points group together. Once a similarity measure between two data points has been defined, there are a number of techniques that have been proposed for clustering.

S. R. McAllister · P. A. DiMaggio Jr. · C. A. Floudas (✉)
Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5263, USA
e-mail: floudas@titan.princeton.edu

The first major category of clustering techniques is hierarchical clustering [4]. These approaches yield a series of nested clusters, graphically illustrating the distance between them. Partitioning clustering techniques, in contrast, separate data points into clusters using a fixed number of partitions. The k-means algorithm is the most common choice for partitioning clustering algorithms due to its ease of implementation and the linear dependence of the runtime on the number of data points [7]. Many other data clustering approaches have been introduced, including model-based clustering [8,9], neural networks [10], simulated annealing [11], and even genetic algorithms [12,13]. Recently, novel clustering methods based on global optimum search and decomposition principles were introduced and applied to microDNA array data for yeast [14–16]. A comprehensive review of clustering methods can be found elsewhere [17].

Another technique that has been proposed to deal with this problem is rearrangement clustering. Given a matrix of disordered data, the basic goal of rearrangement clustering has been to minimize the sum of the pairwise distances between rows by reordering them. The bond energy algorithm (BEA) was originally proposed to deal with the problem of rearrangement clustering [18]. It has been shown that the rearrangement clustering problem can be formulated as a traveling salesman problem (TSP) and solved to optimality [19,20]. Alpert and Kahng proposed a restricted partitioning approach that solved the rearrangement clustering problem through a traveling salesman approach and subsequently determined cluster boundaries given a pre-specified number of clusters [21]. An algorithm to solve for the cluster partitions during the traveling salesman problem, TSP+k, has been developed to simultaneously address the two problems [22]. Recent work has utilized optimal re-ordering in an iterative framework for the biclustering of data matrices in systems biology [23,24]. A key condition in the existing rearrangement clustering methods is that the data matrix needs to be dense with few missing elements.

Despite the multitude of clustering techniques that have been proposed, they rely upon a similarity metric that is defined only between two adjacent points. This reliance becomes a limitation for problems that have copious amounts of missing data. One approach to this problem is to ignore or marginalize the missing data values. By simply computing the pairwise distances without including the contributions of missing data, problems with few missing data values can be addressed.

A second approach is to use interpolation or data imputation methods to replace the missing data and to utilize the previously mentioned techniques. Singular value decomposition, K Nearest Neighbors and simple row averaging have been applied to missing data problems in DNA microarrays with 1–20% missing data [25].

The quadratic assignment problem (QAP), like the traveling salesman problem, is an area of active study in the field of combinatorial optimization. The generic quadratic assignment problem can be formulated as shown in Eqs. 1–3 [26]. The binary variables x_{ij} represent the assignment of object i to position j and the parameter c_{ijkl} is the score associated with the assignment of a pair of objects to a specific pair of positions. For further analysis of the quadratic assignment problem the reader is directed to [27–30].

$$\min \ \frac{1}{2} \cdot \sum_{i;i\neq k} \sum_{j;j\neq l} \sum_{k} \sum_{l} c_{ijkl} \cdot x_{ij} \cdot x_{kl} \tag{1}$$

$$\sum_{j} x_{ij} = 1 \quad \forall i \tag{2}$$

$$\sum_{i} x_{ij} = 1 \quad \forall j \tag{3}$$

The models presented in this paper will address quadratic assignment problems that satisfy the relationship in Eq. 4, which defines a distance between i and k if object i is assigned to position j and object k is assigned to position l. This definition represents the distance between a pair of objects, where the objects are assigned to equally-spaced points on a line.

$$d_{ik} = \sum_{j} \sum_{l} |l - j| \cdot x_{ij} \cdot x_{kl} \quad \forall i, k : i \neq k \tag{4}$$

The proposed objective function assumes the form of Eq. 5, where the product of this distance and a weight for this distance, c_{ik}, is optimized. The formulation of this problem as a mixed-integer linear programming problem shares some similarities with the single-row facility layout problem, another specialized form of the quadratic assignment problem. The interested reader is directed to a recent review for more information on the formulation of and solution to this class of problems [31].

$$\sum_{k} \sum_{i;i \neq k} c_{ik} \cdot d_{ik} \tag{5}$$

In this work, we present an objective function to define a distance-dependent rearrangement clustering approach that is designed to handle large amounts of missing data. This approach has the ability to consider not only the similarity between neighboring rows or columns of data, but also the similarity of all pairs as a function of the distance between two rows or columns. Three mixed-integer linear programming formulations are presented to address the distance-dependent rearrangement clustering problem and the utility of each is discussed. Specifically, we present a model based on (1) the relative ordering of elements (relative ordering model), (2) the assignment of elements to a final position (assignment model), and (3) the assignment of a distance between a pair of elements (indexed-distance model). We also present cutting planes and heuristic methods to increase computational efficiency of these models for finding and proving globally optimal solutions to the distance-dependent rearrangement clustering problem. A comparative study for the methods is presented for glass transition temperature data [32] and two sparse data matrices provided by Pfizer Inc. corresponding to compound libraries for candidate drug discovery.

2 Mathematical modeling

In this section, we define the variables and parameters of the distance-dependent rearrangement problem and present the general form of the objective function used to evaluate the quality of the reordering. We then propose three distinct models for performing the physical re-ordering of the rows and columns of the data matrix: (1) A relative ordering based representation, (2) an assignment based representation, and (3) an indexed-distance based representation. For each model we highlight its attributes in terms of their strengths and weaknesses for solving the rearrangement problem and also present heuristics and cutting planes which can assist in the convergence to the globally optimum solution.

2.1 Parameters and variable definitions

The objective function for the distance-dependent rearrangement clustering problem requires the definition of the appropriate indices, parameters, and continuous variables. The index pair

(i, j) corresponds to a specific row i and column j of a matrix, where the value of this pair in the data matrix is denoted as $a_{i,j}$. The cardinality (or in this case, the dimension) of the rows and columns of the matrix will be represented as $|I|$ and $|J|$, respectively. For the sake of brevity in this section and the remainder of the article, we present the terminology and mathematical model only for the rows of the matrix, but an analogous representation follows for the columns since the problems can be solved independently. When computing the similarity between two rows, it is necessary to incorporate the distance between them in the final arrangement. We define the distance between two rows i and i' to be the continuous variable, $d_{i,i'}$, as presented below:

$$d_{i,i'} = \text{distance between row } i \text{ and row } i' \text{ in the final rearrangement}$$

The closest distance between any two rows is equal to 1 (i.e., they are adjacent) the maximum distance between any two rows is equal to $|I| - 1$ (i.e., when they are on opposite ends of the matrix), as shown by the bounds:

$$1 \leq d_{i,i'} \leq |I| - 1 \quad \forall i, i' > i \tag{6}$$

2.2 Objective function

The objective function for the distance-dependent rearrangement clustering problem is based on the product of the distance between a pair of rows in a data matrix and a similarity measure for this pair of rows. The major distinction between the original rearrangement clustering problem and the objective function presented here is the necessity to include the similarity values for non-neighboring elements as a function of the distance between them, $d_{i,i'}$. The general form for the objective function for this problem is presented in Eq. 7.

$$\min \sum_{i} \sum_{i'} \sum_{j} \theta(d_{i,i'}) \cdot \phi(a_{i,j}, a_{i',j}) \tag{7}$$

In this equation, the parameters $a_{i,j}$ and $a_{i',j}$ denote the data values of the jth column for rows i and i', respectively. One possible form for each of the component functions, $\theta(d_{i,i'})$ and $\phi(a_{i,j}, a_{i',j})$, is shown in Eq. 8.

$$\min \sum_{i} \sum_{i'} \sum_{j} \frac{|I| - d_{i,i'}}{|I| - 1} \cdot (a_{i,j} - a_{i',j})^2 \tag{8}$$

In this equation, $\theta(d_{i,i'})$ is linear with respect to $d_{i,i'}$, achieving a maximum value of 1 when $d_{i,i'} = 1$ and a minimum value of $1/(|I| - 1)$ when $d_{i,i'} = |I| - 1$. The term $\theta(d_{i,i'})$ can be thought of as a weighting factor between two rows that decreases with increasing distance between them in the final rearrangement. The parameter $\phi(a_{i,j}, a_{i',j})$ in Eq. 8 is the pairwise squared difference between the two rows. Other forms of $\theta(d_{i,i'})$ and $\phi(a_{i,j}, a_{i',j})$ could be introduced if problem-specific details suggest they would be more appropriate. The only restrictions on $\theta(d_{i,i'})$ is that it is a linear function in $d_{i,i'}$.

2.3 Model 1: relative ordering representation

The first mathematical model we present for the distance-dependent rearrangement clustering problem is based on the relative ordering of rows in the final arrangement. We define the binary variable, $y_{i,i'}$, to indicate whether row i is placed *before* row i' in the final ordering, as shown below:

$$y_{i,i'} = \begin{cases} 1, & \text{if row } i \text{ occurs before } i' \text{ in the final ordering} \\ 0, & \text{otherwise} \end{cases}$$

Note that this does not provide any information regarding the final positions of rows i and i'. To model this, we define positive variables, p_i, to denote the final position of row i in the data matrix.

$$p_i = \text{position of row } i \text{ in the final ordering}$$

Where the values for the final positions are bounded by:

$$1 \leq p_i \leq |I| \quad \forall i > 1 \tag{9}$$

From these final positions, we can also define the distance between any two rows i and i' in the final arrangement by the positive variable $d_{i,i'}$. The distance between two final positions i and i' is given by the *nonlinear* equation $d_{i,i'} = |p_i - p_{i'}|$. However, exact lower bounds on this distance can be represented by two linear inequality constraints, as shown in Eqs. 10 and 11.

$$d_{i,i'} \geq p_i - p_{i'} \quad \forall i < i' \tag{10}$$
$$d_{i,i'} \geq p_{i'} - p_i \quad \forall i < i' \tag{11}$$

We can define upper bounds on the distance variables by utilizing the information regarding the relative ordering of the rows i and i', represented by the binary variables $y_{i,i'}$, and big M constraints. Therefore, if row i' is placed above row i in the final arrangement, then its distance is exactly equal to the difference between positions, $p_i - p_{i'}$, and vice versa. This is represented mathematically by Eqs. 12 and 13.

$$d_{i,i'} \leq p_i - p_{i'} + M \cdot y_{i,i'} \quad \forall i < i' \tag{12}$$
$$d_{i,i'} \leq p_{i'} - p_i + M \cdot (1 - y_{i,i'}) \quad \forall i < i' \tag{13}$$

One can observe that either Eqs. 12 or 13 provides an exact upper bound on the distance between rows i and i', depending on which occurs first in the final ordering. The parameter M is selected so that it is large enough to relax the inequality that is not valid.

An exact equality between the final position of a row and the relative orderings of all other rows can be derived by simply counting the number of rows that are above row i in the final arrangement, as shown in Eq. 14, which is based on a network flow [33–37].

$$p_i - 1 = \sum_{i'>i} y_{i,i'} + \sum_{i'<i} (1 - y_{i',i}) \quad \forall i \tag{14}$$

We can derive tighter linear programming relaxations based on distributions of all final positions and distances. One such constraint utilizes the fact that the summation of all the final distances is equal to a known constant, $C_d(|I|)$, as shown in Eq. 15.

$$\sum_{i} \sum_{i'>i} d_{i,i'} = C_d(|I|) \tag{15}$$

For instance, if there are only four rows, then $C_d(4) = 3 \cdot 1 + 2 \cdot 2 + 1 \cdot 3 = 10$. Equivalently, we also know that the sum over all possible final positions must be equal to a known constant, $C_p(|I|)$, as shown in Eq. 16.

$$\sum_{i} p_i = C_p(|I|) \tag{16}$$

To alleviate some of the issues associated with problem symmetry, we can restrict that the final position of the first row lies in the first half of the matrix, as shown in Eq. 17.

$$1 \leq p_1 \leq floor(|I| + 1/2) \tag{17}$$

2.3.1 Additional cutting planes

The linear programming relaxations can be further tightened by using additional cutting plane constraints that are specific to this model. These constraints are only added if their corresponding conditions are violated at the current branch-and-bound node. For instance, since the distances are Euclidean we can impose the restriction that they satisfy the triangle inequality constraint in Eq. 18.

$$d_{i,i'} \leq d_{i,i''} + d_{i'',i} \quad \forall i, i', i'' \tag{18}$$

However, this results in O($|I|^3$) additional constraints and leads to memory issues for standard commercial solvers for even moderately sized problems. To circumvent this memory issue, we introduce these constraints dynamically during the program execution as *cuts*. In other words, we include only those triangle inequality constraints for which Eq. 18 is violated and this ensures that only the most necessary constraints are added to the problem.

We can also introduce cuts based on bounds on the distances of a single row. Let A_i contain an arbitrary set of indices i', such that $i' \neq i$. For instance, in a 4-row problem, the sum of the distances between row 1 and all other rows must be less than or equal to $3 + 2 + 1 = 6$ and greater than or equal to $1 + 1 + 2 = 4$. Similarly, the sum of any two distances between row 1 and two other rows must be less than or equal to $3 + 2 = 5$ and greater than or equal to $1 + 1 = 2$.

$$\sum_{i' \in A_i} d_{i,i'} \leq F^U(|A_i|) \quad \forall i, A_i \tag{19}$$

$$\sum_{i' \in A_i} d_{i,i'} \geq F^L(|A_i|) \quad \forall i, A_i \tag{20}$$

Similarly, we can introduce cuts based on bounds on the distances for *all* rows. Let B contain an arbitrary set of index pairs i, i', such that $i \neq i$. For instance, in a 4-row problem, the sum of any 4 pair-wise distances must be less than $3 + 2 + 2 + 1 = 8$ and greater than or equal to $1 + 1 + 1 + 2 = 5$.

$$\sum_{i,i' \in B} d_{i,i'} \leq G^U(|B|) \quad \forall B \tag{21}$$

$$\sum_{i,i' \in B} d_{i,i'} \geq G^L(|B|) \quad \forall B \tag{22}$$

The mixed-integer linear programming problem for the relative ordering based model is the minimization of the objective function in Eq. 23 over the variables $y_{i,i'}$, p_i, and $d_{i,i'}$, subject to the constraints in Eqs. 9 through 22. This model can be solved to global optimality using existing solvers such as CPLEX [38].

$$\min_{y_{i,i'}, p_i, d_{i,i'}} \sum_i \sum_{i'} \sum_j \frac{|I| - d_{i,i'}}{|I| - 1} \cdot (a_{i,j} - a_{i',j})^2 \tag{23}$$

2.4 Model 2: assignment representation

The second model we present for the distance-dependent rearrangement clustering problem is based on an assignment representation [39,40]. In this model, we define binary variables, $y_{i,k}$, to represent the assignment of a row i to some position k in the final ordering.

$$y_{i,k} = \begin{cases} 1, & \text{if row } i \text{ is assigned to position } k \text{ in the final ordering} \\ 0, & \text{otherwise} \end{cases}$$

In definition of this variable, the index k is in the set $k = 1 \ldots |K|$, where $|K| = |I|$. Analogous to the relative ordering model presented in Sect. 2.3, the positive variables p_i and $d_{i,i'}$ denote the final position of row i and the distance between rows i and i' in the final ordering, respectively. For the readers convenience, we present the constraints that are applicable to this assignment model from Sect. 2.3.

$$d_{i,i'} \geq p_i - p_{i'} \quad \forall i < i' \tag{24}$$

$$d_{i,i'} \geq p_{i'} - p_i \quad \forall i < i' \tag{25}$$

$$\sum_i \sum_{i'>i} d_{i,i'} = C_d(|I|) \tag{26}$$

$$1 \leq p_1 \leq floor(|I| + 1/2) \tag{27}$$

An intuitive constraint to impose on the row assignments is that a final position can contain only one row and a row can be assigned to at most one final position. This is given by Eqs. 28 and 29.

$$\sum_k y_{i,k} = 1 \quad \forall i \tag{28}$$

$$\sum_i y_{i,k} = 1 \quad \forall k \tag{29}$$

We can relate $y_{i,k}$ to p_i using a simple equality constraint as shown in Eq. 30.

$$p_i = \sum_k k \cdot y_{i,k} \quad \forall i \tag{30}$$

Another valid constraint on the distance variables can be derived from the fact that once row i has been assigned to some final position k, then the sum of the distances from position k to all other positions is known, which we represent as a general function $G(k)$. For instance, if a row i is assigned to final position 2 in a 4-row problem, then the sum of its distances to all other rows i' in the final arrangement is $(2-1)+(3-2)+(4-2) = 4 = G(2)$. The general form for this constraint is presented in Eq. 31.

$$\sum_{i'>i} d_{i,i'} + \sum_{i'<i} d_{i,'i} = \sum_k G(k) \cdot y_{i,k} \quad \forall i \tag{31}$$

Equation 32 introduces another constraint to provide a valid upper bound on the distance by incorporating information about the final positions of two row assignments. Let m be the midpoint of the valid position assignments (i.e., $m = (|I|+1)/2$). If row i is assigned to final position k and row i' is assigned to final position k', then the maximum distance between row i and any other row i' must be less than $|k-m| + |k'-m|$ (i.e., the sum of their distances to the midpoint).

$$d_{i,i'} \leq \sum_k |k-m| \cdot \left(y_{i,k} + y_{i',k}\right) \quad \forall i, i' > i \tag{32}$$

2.4.1 Additional cutting planes

As in Sect. 2.3, cutting planes based on triangle inequalities can be added for the distance variables, as defined in Eq. 18. Another set of constraints which bounds the distances with respect to a fixed point, $1 \leq p^* \leq |I|$, are introduced as cuts into the problem, where p^* is chosen during program execution. These cutting planes are a generalization of Eq. 32, where the midpoint m is replaced by p^*.

$$d_{i,i'} \leq \sum_k \left|k - p^*\right| \cdot \left(y_{i,k} + y_{i',k}\right) \quad \forall i, i' > i, p^* \tag{33}$$

2.4.2 Additional branching variables

The performance of the proposed mixed-integer linear programming assignment formulation can be enhanced by the definition of symmetry-breaking binary variables. Although this increases the number of binary variables needed to represent a system, branching on these new binary variables in a branch-and-bound framework is more informative (i.e., leads to tighter relaxations) both when branching up (i.e., $y = 1$) and down (i.e., $y = 0$). Equation 34 defines the variable $\hat{y}_i$, which is active if row i is assigned to a final position k greater than the midpoint m and inactive if row i is assigned to a final position k less than or equal to the midpoint m.

$$\hat{y}_i = \sum_{k>m} y_{ik} \quad \forall i \tag{34}$$

The mixed-integer linear programming problem for the assignment based model is the minimization of the objective function in Eq. 35 over the variables $y_{i,k}$, p_i, and $d_{i,i'}$, subject to the constraints in Eqs. 24 through 34. This model can be solved to global optimality using existing solvers such as CPLEX [38].

$$\min_{y_{i,k}, p_i, d_{i,i'}} \sum_i \sum_{i'} \sum_j \frac{|I| - d_{i,i'}}{|I| - 1} \cdot (a_{i,j} - a_{i',j})^2 \tag{35}$$

2.5 Model 3: indexed-distance representation

The final model we present for the distance-dependent rearrangement clustering problem is based on incorporating the distance variables as indices of a binary variable. That is, we define the binary variable, $z_{i,i',n}$ to indicate that the distance between two rows i and i' is n in the final ordering, as presented below:

$$z_{i,i',n} = \begin{cases} 1, & \text{if row } i \text{ and } i' \text{ are separated by a distance } n \text{ in the final ordering} \\ 0, & \text{otherwise} \end{cases}$$

In this variable definition, the index n is in the set $n = 1 \ldots |N|$, where $|N| = |I|$. It should be noted that the binary variables, $z_{i,i',n}$, are related to the distance variables, $d_{i,i'}$, defined in Sect. 2.1 by Eq. 36. This relationship can be used to rewrite the objective function in Eq. 7 in terms of $z_{i,i',n}$. Equation 36 does not have to be included as a constraint in the mixed-integer linear programming formulation, but is presented here for clarity.

$$\sum_{n<|N|} z_{i,i',n} \cdot (n) = d_{i,i'} \quad \forall (i, i' > i) \tag{36}$$

The distance between two rows i and i' in the final ordering must be a unique distance and this is enforced by Eq. 37.

$$\sum_{n<|N|} z_{i,i',n} = 1 \quad \forall (i, i' > i) \tag{37}$$

We can also infer constraints based on the distribution of the values for the distances. For instance, we know that there can only exist one distance of $|I| - 1$ (i.e., between the two rows on opposite ends of the matrix), two distances of $|I| - 2$, etc. This is generically represented by Eq. 38.

$$\sum_{i}\sum_{i'>i} z_{i,i',n} = |N| - n \quad \forall (n < |N|) \tag{38}$$

We can also state that for any distance $n \geq \frac{|N|+1}{2}$, there can only exist at most one point i' a distance n away from i. That is, any point i on one side of the midpoint of the matrix can have only one other point, i', of distance n away on the *other side* of the midpoint. This is presented in constraint Eq. 39.

$$\sum_{i'>i} z_{i,i',n} \leq 1 \quad \forall \left(i, \frac{|N|+1}{2} \leq n < |N| \right) \tag{39}$$

However, for all other distances $n < \frac{|N|+1}{2}$, it is possible to have at most two points a distance n away from i, as shown in Eq. 40.

$$\sum_{i'>i} z_{i,i',n} \leq 2 \quad \forall \left(i, n < \frac{|N|+1}{2} \right) \tag{40}$$

Furthermore, there should exist *at least* one point i' a distance n away from i when $n < \frac{|N|+1}{2}$, which is accomplished via Eq. 41.

$$\sum_{i'} (z_{i,i'>i,n} + z_{i',i>i',n}) \geq 1 \quad \forall \left(i, n < \frac{|N|+1}{2} \right) \tag{41}$$

We can also enforce an either/or type of assignment to the distances. In other words, we know that if row i is assigned to some position $n \leq \frac{|N|+1}{2} - 1$, then it has exactly two other rows a distance of n and $N - n$. This is generally written as Eq. 42.

$$\sum_{i'} \left(z_{i,i'>i,n} + z_{i',i>i',n} + z_{i,i'>i,|N|-n} + z_{i',i>i',|N|-n} \right)$$
$$= 2 \quad \forall \left(i, n < \frac{|N|+1}{2} - 1 \right) \tag{42}$$

2.5.1 Additional cutting planes

The triangle inequalities are also applied to the distance assignments as they are in Sects. 2.3 and 2.4, but here they are expressed in terms of the binary variables $z_{i,i',n}$. The triangle inequality constraints in this form are shown in Eq. 43. It should be noted that although the constraints of Eq. 43 are introduced as cutting planes, the indexed-distance model is not valid without the introduction of these triangle inequalities. Therefore, while the models presented in Sects. 2.3 and 2.4 can be solved without introducing their respective cutting plane constraints, the model presented in this section must have the following cutting plane constraints.

$$\sum_{n<|N|} z_{i,i',n} \cdot (n) \leq \sum_{n<|N|} z_{i,i''>i,n} \cdot (n) + \sum_{n<|N|} z_{i'',i'>i'',n} \cdot (n)$$
$$+ \sum_{n<|N|} z_{i'',i>i'',n} \cdot (n) + \sum_{n<|N|} z_{i',i''>i',n} \cdot (n) \quad \forall (i' > i, i \neq i'', i' \neq i'') \tag{43}$$

The mixed-integer linear programming problem for the index-distance based model is the minimization of the objective function in Eq. 44 over the variables $z_{i,i',n}$ subject to the constraints in Eqs. 37 through 43. This model can be solved to global optimality using existing solvers such as CPLEX [38].

$$\min_{z_{i,i',n}} \sum_{i} \sum_{i'>i} \sum_{j} \frac{|I| - \sum_{n<|N|} z_{i,i',n} \cdot (n)}{|I| - 1} \cdot (a_{i,j} - a_{i',j})^2 \tag{44}$$

2.5.2 *Alternate objective functions*

One advantage to the indexed-distance formulation is that the objective function can be quickly altered to evaluate only a subset of neighboring elements in the final arrangement and can also be extended to a nonlinear form. Equation 45 illustrates the transition from an arbitrary function of a distance ($\theta(d_{i,i'})$) to the product of a binary variable ($z_{i,i',n}$) and an arbitrary function of a constant index ($\theta(n)$).

$$\theta(d_{i,i'}) = \sum_{n<|N|} \theta(z_{i,i',n} \cdot (n)) = \sum_{n<|N|} z_{i,i',n} \cdot \theta(n) \quad \forall (i, i' > i) \tag{45}$$

2.6 Use of heuristics

The general rearrangement problem has a total of $\frac{N!}{2}$ possible orderings, which makes the problem difficult but also allows for heuristic methods to easily find integer feasible solutions. In this section, we discuss heuristic techniques based on row swapping for finding quick integer solutions to help close the integrality gap.

The simplest heuristic is the random swapping of rows. After each swap, we evaluate the objective function of the new ordering and the swap is accepted if it results in a lower objective function value. The acceptance criteria of this operation could be altered to allow for objective function increases based on a probability (a basic Monte Carlo approach), but the utilization of several initial points was found to be effective in finding good solutions. Given some ordering of the rows, this basic random swapping operation is a reasonable local minimization approach.

The initial ordering to minimize via these swapping operations can be generated using the linear programming relaxations from each node in the branch-and-bound tree. For instance, the initial ordering for the model based on the assignment representation can be determined by (1) identifying the two rows with the maximum distance, $d_{ii'}$, and fixing these rows to be the beginning and end of the ordering and (2) sorting the remaining rows based on their distances to these two fixed endpoints. Similar strategies can be used to establish an initial ordering for the relative ordering model and the indexed-distance model. This ordering is then subject to a local minimization via the aforementioned random swapping operations.

3 Computational studies

To benchmark the performance of the three mathematical models proposed in Sects. 2.3, 2.4, and 2.5, we applied them to three different sparse re-ordering problems and compared the computational requirements associated with each model. The data matrices examined in this section correspond to glass transition temperatures of a polymer library, $\log(\mathrm{IC}_{50})$ data and percent inhibition data for compound libraries in molecular discovery. Each of the proposed methods utilizes cutting planes and heuristics to identify the best possible ordering, but we also present the models without the use of cutting planes and heuristics when applicable so as to illustrate the importance of these components. The problems were selected in order to represent a broad distribution of sizes, where $|I| = 14, 28$, and 39 for the problems studied in this section. All of the problems were solved using CPLEX 9.1 on an Intel 3.2 Ghz processor.

We will adopt the following convention for the results presented below. The term "cutting planes" and "heuristics" corresponds to the constraints and methods presented in Sect. 2 unless otherwise noted and the term "LP" means linear programming. In the tables presented, the number of "Applied cuts" corresponds to the number of cutting planes from Sect. 2 that were applied, the "LP relax time" and "LP relax value" correspond to the time and value of the linear programming relaxation at the root node, "After cuts value" is the value of the LP relaxation after applying the cuts defined in Sect. 2 *and* the cuts applied by CPLEX (i.e., Gomory fractional, mixed-integer rounding, etc.), and "Best bound" corresponds to the best linear programming relaxation value for all nodes.

3.1 Glass transition temperature data

This first data set analyzed by the proposed models is a 14 by 8 data matrix corresponding to 14 diol and 8 diacid substituent sites of a polymer [32]. The glass transition temperatures for each polymer (i.e., for a specific diol and diacid) are the property values of this data matrix. It was previously discovered that there exists a reordering of this polymer library which exhibits a smooth and regular landscape [41]. We randomly sampled 35 out of the 112 possible compounds (i.e., 31.3% dense) to assess the ability of the proposed methods to handle sparse data and this matrix is presented in Table 1.

The relative ordering, assignment, and indexed-distance models were applied to the reordering of the rows (i.e., $|I| = 14$) of this sparse matrix and the computational results are presented in Table 2. For reference, we also present the results for a general quadratic assignment problem (QAP) model for the same problem. In Table 2 we see that the relative ordering model results in the smallest problem size with respect to both the number of variables and constraints and solves the rearrangement problem to global optimality in only 0.4 CPU seconds. However, its root node relaxation of 6,128 is the poorest among the three proposed methods. The formulation for the indexed-distance model results in the largest number of constraints and variables, but exhibits a good linear programming relaxation of 10,915 at the root node. Due to its size, it takes substantially longer than the other three proposed models to close the integrality gap. For this problem, the assignment model has the best linear programming relaxation at the root node and solves the problem to global optimality in the least amount of CPU time (0.16 s). It is interesting to note the improvements in the linear programming relaxations after the cutting planes are added. After cuts, all methods are within 1.6% of the optimal integer solution, with the assignment model completely closing the integrality gap at the root node.

Table 1 Sparse sampling of the glass transition temperature data matrix from [32], where "–" indicates a missing element

	1	2	3	4	5	6	7	8
1	–	–	–	55.0	46.0	–	–	–
2	78.0	42.0	76.0	66.0	–	–	47.0	–
3	–	–	–	–	–	19.0	–	17.0
4	–	–	–	–	61.0	63.0	–	–
5	–	–	–	21.0	–	–	–	18.0
6	–	–	45.0	–	–	33.0	–	–
7	91.0	47.0	–	–	–	–	–	63.0
8	–	–	40.0	–	–	30.0	21.0	–
9	–	33.0	–	–	–	–	42.0	–
10	82.0	44.0	–	68.0	–	–	–	58.0
11	–	36.0	–	–	–	–	46.0	–
12	–	–	–	–	65.0	–	54.0	–
13	–	–	–	42.0	32.0	–	–	–
14	–	–	–	–	28.0	–	22.0	–

Table 2 Comparison of solve stats for the proposed models for the rows of the glass transition temperature data matrix (size 14), using CPLEX 9.1

	Relative ordering (no cuts, no heur)	Assignment (no cuts, no heur)	Relative ordering	Assignment	Index-distance
Constraints	380	420	380	420	440
Binary variables	91	210	91	210	1183
Continuous variables	105	105	105	105	0
Applied cuts	–	–	309	252	262
LP relax time (s)	0.01	0.03	0.01	0.03	0.08
LP relax value	6128	13856.8	6128	13857	10915
After cuts value	7525	13901.7	14222	14453	14277
Nodes	78	18	2	0	470
Best integer	14453	14453	14453	14453	14453
Best bound	14453	14453	14453	14453	14453
CPU time (s)	4.0	1.1	0.4	0.16	173.6

To provide a benchmark of the performance of these proposed methods, we present the results for the relative ordering and assignment models without the use of cutting planes or heuristics in columns 1 and 2 of Table 2. We see that for the relative ordering model without cutting planes, the relaxation at the root node is only within 48% of the optimal integer solution so it takes 78 nodes of branching and 4 CPU seconds to prove optimality for this problem. Similar results are observed for the assignment model without cuts in Table 2, where the LP relaxation without cutting planes is within 4% of the optimal integer solution and 18 nodes of branching are required to close the optimality gap.

Table 3 Comparison of solve stats for the QAP approaches for the rows of the glass transition temperature data matrix (size 14)

	MILP linearization	Branch-and-bound	Gilmore-Lawler bound	Elimination bound
LP relax time (s)	38.1	–	–	–
LP relax value	14213	–	–	–
Best integer	14453	14453	–	–
Best bound	14453	14453	11420	11760
CPU time (s)	1264.3	0.11	0.00	0.00

As a basis for comparison with a general quadratic assignment problem model, we applied two QAP formulations and two rigorous lower bounding methods for the quadratic assignment problem to this data matrix. Specifically, we downloaded Fortran code from the Quadratic Assignment Problem Library (QABLIB) developed by Burkard et al. [42] for (1) a branch-and-bound algorithm that solves QAPs to optimality for problems less than size 33, (2) an algorithm that computes the Gilmore-Lawler bound [43], and (3) another algorithm that computes the elimination bound for quadratic assignment problems. We also implemented a MILP linearization for the quadratic assignment problem that was reviewed in [29]. The results for these methods are presented in Table 3. The branch-and-bound algorithm finds the optimal solution of 14,453 in only 0.11 CPU seconds whereas the MILP linearization takes 1264.3 CPU seconds to find this optimal re-ordering. Both the Gilmore-Lawler bound and the elimination bounding methods require almost no CPU time and provide bounds within 20.8% and 18.6% of the optimal solution, respectively.

3.2 IC50 inhibition data

The second data matrix studied contains 28 rows and 32 columns of log(IC_{50}) values for an unknown set of compounds. These values represent the concentrations of these compounds needed to yield 50% inhibition of an unknown target. In this library, the most desirable compounds are those that achieve the required inhibition at the lowest concentrations (i.e., these compounds will have the lowest log(IC_{50}) values). Of the possible 896 data values in this matrix, only 340 (38%) have been synthesized and IC_{50} measured experimentally.

The results for the proposed models for optimally re-ordering the rows of the IC_{50} data matrix (e.g., $|I| = 28$) are presented in Table 4. The relative ordering model results in the fewest number of binary variables and also requires the shortest time for solving the linear programming relaxation at the root node. However, the value of its LP relaxation is at least 33% worse than the other two proposed models. In 10 nodes and 29.91 CPU seconds, the relative ordering model is able to prove global optimality for this rearrangement clustering problem. The indexed-distance model results in the largest number of constraints and binary variables and as a result requires the longest LP relaxation time at the root node. However, its LP relaxation value at the root node is within 4% of the optimal integer solution and the use of cutting planes closes the optimality gap at the root node in 100 CPU seconds. The assignment model is also able to close the optimality gap at the root node after the addition of cutting planes, but it has an initial LP relaxation within 3% of the optimal integer solution and only takes 4 CPU seconds.

For comparison, we also presented the performance of the relative ordering and assignment models without the use of cutting planes or heuristic methods in the first and second

Table 4 Comparison of solve stats for the proposed models for the rows of the IC50 data matrix (size 28), using CPLEX 9.1

	Relative ordering (no cuts, no heur)	Assignment (no cuts, no heur)	Relative ordering	Assignment	Indexed-distance
Constraints	1542	1274	1542	1274	1861
Binary variables	378	812	378	812	10206
Continuous variables	406	406	406	406	0
Applied cuts	–	–	2313	1527	1417
LP relax time (s)	0.14	0.68	0.14	0.67	2.28
LP relax value	790.1	1190.5	790.1	1190.5	1179.2
After cuts value	813.2	1191.5	1225.2	1229.2	1229.2
Nodes	19026	15	10	0	0
Best integer	1229.2	1229.2	1229.2	1229.2	1229.2
Best bound	1083.0	1229.2	1229.2	1229.2	1229.2
CPU time (s)	7200+	23.77	29.91	4.17	100.44

Table 5 Comparison of solve stats for the QAP approaches for the rows of the IC50 data matrix (size 28)

	MILP linearization	Branch-and-bound	Gilmore-Lawler bound	Elimination bound
LP relax time (s)	–	–	–	–
LP relax value	–	–	–	–
Best integer	–	1232.3	–	–
Best bound	–	–	1160.6	1129.2
CPU time (s)	7200+	7200+	0.00	0.00

columns of Table 4. The value of the LP relaxation for the relative ordering model without cutting planes results in an optimality gap of 34% at the root node. When allowing this model to run for 7,200 CPU seconds, it finds the optimal integer solution but still has an optimality gap of 12% after examining 19,026 nodes. For the assignment model without cuts, the linear programming relaxation is within 3% of the optimal solution and it only takes 15 nodes of branching and 23.8 CPU seconds to prove global optimality for this problem.

In an attempt to compare to the quadratic assignment problem algorithms studied in the previous section, we applied the MILP linearization, branch-and-bound algorithm, Gilmore-Lawler bound, and elimination bound to the QAP representation of this problem. The MILP linearization model was too large and violated the internal memory requirements for CPLEX when attempting to initialize the model. The branch-and-bound algorithm was able to load the problem but did not return an optimal solution after 7,200 CPU seconds. We then modified this algorithm to report the best integer solution found after that time, which is 1232.3 as shown in Table 5 and is within 0.3% of optimality. However, we could not obtain a rigorous lower bound from this method to determine the corresponding optimality gap. The Gilmore-Lawler lower bound is within 5.6% and the elimination bound is within 7.6% of the optimal solution for this problem. We see from Table 5 that the MILP linearization and branch-and-bound quadratic assignment problem models are unable to address this problem (size $|I| = 28$) and thus will not be applied to any larger problems studied in the remainder of this article.

3.3 Percent inhibition data

The third data matrix analyzed contains 62 rows and 39 columns, where the columns and rows correspond to different functional groups that can be appended to two distinct substitution sites of a molecular scaffold. The data values in this matrix denote the percent inhibition data for a specific compound, where the selection of a particular row and column defines a new compound. The most desirable compounds are the strongest inhibitors of an unknown target, which correspond to the highest percentage inhibition values in this library. Of the possible 2,418 compounds, only 1,229 (51%) have been synthesized and measured experimentally.

The results for the proposed models for the optimal rearrangement of the columns of this data matrix (i.e., $|I| = 39$) are presented in Table 6. As with the previous matrices, the relative ordering model results in the fewest number of binary variables and requires the shortest time for solving the LP relaxation at the root node, which has an optimality gap of 49%. After cutting planes, the LP relaxation is improved to be within 5% of the optimal integer solution, however, the model is unable to prove optimality after 7,200 s and 675 nodes of branching. The indexed-distance problem has 28,158 binary variables, which is more than 12 times the number of binary variables for the other two methods combined. However, it has an LP relaxation value within 8% of the optimal integer solution which improves to only 0.2% after the addition of cutting planes. But due to its size it can only branch for 2 nodes in over 7,200 CPU seconds and cannot prove global optimality for this problem. The assignment model has the best linear programming relaxation of the three models, which is within 5.5% and 0.1% of global optimality before and after cutting planes, respectively. Since the LP relaxation time for the assignment model is only one-sixth of the time for the indexed-distance model, it is able to prove global optimality for this rearrangement problem in only 1 node and 96 CPU seconds.

Table 6 Comparison of solve stats for the proposed models for the columns of the percent inhibition data matrix (size 39), using CPLEX 9.1

	Relative ordering (no cuts, no heur)	Assignment (no cuts, no heur)	Relative ordering	Assignment	Indexed-distance
Constraints	3005	3120	3005	3120	3704
Binary variables	741	1560	741	1560	28,158
Continuous variables	780	780	780	780	0
Applied cuts	–	–	10524	5745	3693
LP relax time (s)	0.84	3.87	0.84	3.87	23
LP relax value	0.890E6	1.652E6	0.890E6	1.652E6	1.604E6
After cuts value	0.917E6	1.653E6	1.696E6	1.746E6	1.743E6
Nodes	10994	235	675	1	2
Best integer	2.119E6	1.747E6	1.747E6	1.747E6	1.747E6
Best bound	1.061E6	1.747E6	1.742E6	1.747E6	1.743E6
CPU time (s)	7200+	541	7200+	95.6	7200+

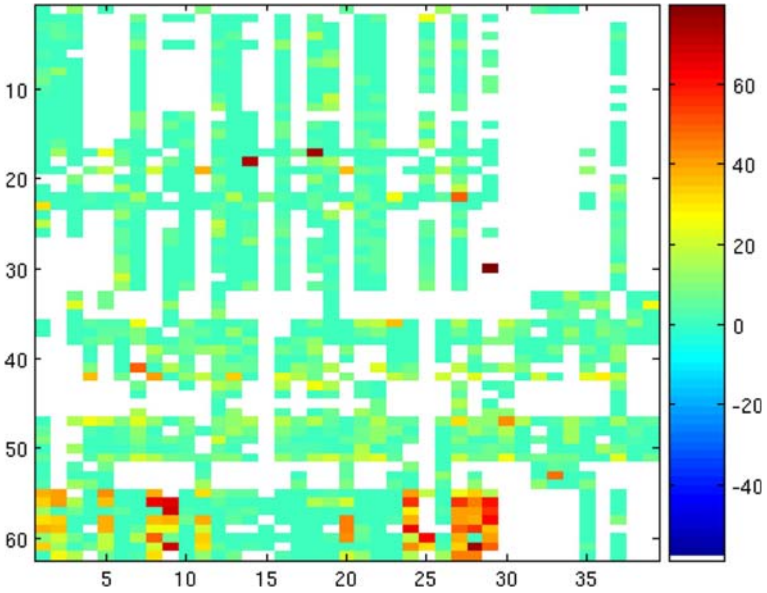

Fig. 1 Original ordering for percent inhibition data matrix. White spaces denote missing elements

We also present the results for the relative ordering and assignment model without the use of cutting planes and heuristics, as shown in Table 6. Since the relative ordering model was not able to prove optimality with the use of cutting planes and heuristics, it is no surprise that is unable to do so without. After 7,200 s of CPU time, it finds a best integer solution of 2.119E6 and a best relaxation value of 1.061E6, which corresponds to an integrality gap of 50%. For the assignment model without cutting planes and heuristics, it only requires 235 nodes and 541 CPU seconds to prove global optimality since the linear programming relaxation is within 5% of optimality at the root node.

To illustrate the utility of re-ordering sparsely sampled data matrices, we also present the original and optimally re-ordered percent inhibition data in Figs. 1 and 2, respectively. Note that the optimally re-ordered data matrix over both the rows and columns in Fig. 2 exhibits an excellent grouping of the high inhibition compounds (shown in orange and red) in the upper-left corner of the matrix. The optimal re-ordering over the rows (e.g., $|I| = 62$) was accomplished using the assignment model, which solved it to optimality in 3 nodes and 1,863 CPU seconds. The initial LP relaxation for this problem is 2.441E6 after cuts and the optimal integer solution is 2.444E6. These re-orderings could be useful for directing the synthesis of future compounds towards those unknown compounds that gather in this region.

4 Conclusions

In this article, we have demonstrated how the distance-dependent rearrangement clustering problem assumes the form of a quadratic assignment problem with special structure. This special structure was exploited in the development of three mixed-integer linear programming (MILP) formulations based on (1) the relative ordering of the rows, (2) the assignment

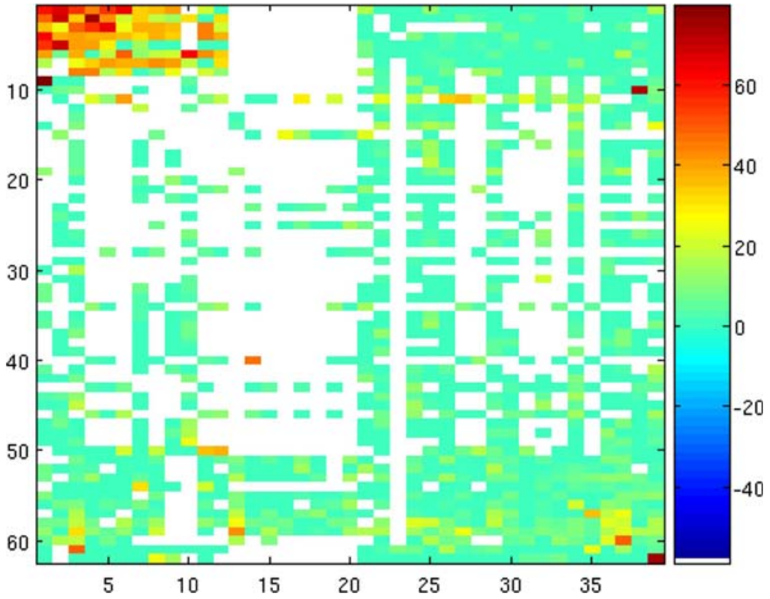

Fig. 2 Optimally re-ordered rows and columns for percent inhibition data matrix. White spaces denote missing elements

of the rows to a final position, and (3) the assignment of a unique distance to a pair of rows. When applying these models to the rearrangement of three distinct data matrices, it was shown that the assignment model was the most efficient for finding and proving the global optimum solution. It was also shown that the relative ordering model resulted in the minimum number of binary variables and that the indexed-distance model can be easily extended to more sophisticated forms of the objective function. We have also illustrated the utility of incorporating cutting planes and heuristic methods for generating integer feasible solutions and closing the optimality gap. These models can be utilized in a number of applications, such as molecular discovery, since the re-orderings group compounds with similar properties in the same regions of the data matrix. For instance, this re-ordering approach could be combined with local interpolation techniques in an iterative fashion to identify an effective synthesis strategy for the molecular discovery problem. It is noteworthy that the proposed techniques could be applied to clustering ensembles of conformers resulting from free energy calculations of oligopeptides [44–46] or proteins from multiple sequence alignment [47,48], de novo sequences generated in protein design with varied lengths [49,50], as well as design and scheduling of batch processes [51,52].

Acknowledgements The authors would like to acknowledge Dr. Xiao-Jiang Feng, Professor Joshua D. Rabinowitz, and Professor Herschel A. Rabitz for helpful discussions related to the molecular discovery problem, Dr. T.-C. Hua of Pfizer for providing the inhibition data analyzed in this article, and support from the US EPA (GAD R 832721-010). Although the research described in the article has been funded in part by the US Environmental Protection Agency's STAR program through grant (R 832721-010), it has not been subjected to any EPA review and therefore does not necessarily reflect the views of the Agency, and no official endorsement should be inferred.

References

1. Anderberg, M.R.: Cluster Analysis for Applications. Academic Press, New York (1973)
2. Jain, A.K., Flynn, P.J.: Image segmentation using clustering. In: Ahuja, N., Bowyer, K. (eds.) Advances in Image Understanding: A Festschrift for Azriel Rosenfeld, pp. 65–83. IEEE Press, Piscataway (1996)
3. Salton, G.: Developments in automatic text retrieval. Science **253**, 974–980 (1991)
4. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl Acad. Sci. **95**, 14863–14868 (1998)
5. Zhang, Y., Skolnick, J.: SPICKER: A clustering approach to identify near-native protein folds. J. Comput. Chem. **25**, 865–871 (2004)
6. Mönnigmann, M., Floudas, C.A.: Protein loop structure prediction with flexible stem geometries. Protein Struct. Funct. Bioinform. **61**, 748–762 (2005)
7. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: a K-means clustering algorithm. Appl. Stat. **28**, 100–108 (1979)
8. Edwards, A.W.F., Cavalli-Sforza, L.L.: A method for cluster analysis. Biometrics **21**, 362–375 (1965)
9. Wolfe, J.H.: Pattern clustering by multivariate mixture analysis. Multivariate Behav. Res. **5**, 329–350 (1970)
10. Jain, A.K., Mao, J.: Artificial neural networks: a tutorial. IEEE Comput. **29**, 31–44 (1996)
11. Klein, R.W., Dubes, R.C.: Experiments in projection and clustering by simulated annealing. Pattern Recognit. **22**, 213–220 (1989)
12. Raghavan, V.V., Birchand, K.: A clustering strategy based on a formalism of the reproductive process in a natural system. In: Proceedings of the Second International Conference on Information Storage and Retrieval, pp. 10–22. Dallas, Texas (1979)
13. Bhuyan, J.N., Raghavan, V.V., Venkatesh, K.E.: Genetic algorithm for clustering with an ordered representation. In: Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 408–415. San Mateo, California (1991)
14. Tan, M.P., Broach, J.R., Floudas, C.A.: A novel clustering approach and prediction of optimal number of clusters: global optimum search with enhanced positioning. J. Glob. Optim. **39**(3), 323–346 (2007)
15. Tan, M.P., Broach, J.R., Floudas, C.A.: Evaluation of normalization and pre-clustering issues in a novel clustering approach: global optimum search with enhanced positioning. J. Bioin. Comp. Bio. **5**(4), 895–913 (2007)
16. Tan, M.P., Smith, E., Broach, J.R., Floudas, C.A.: Microarray data mining: a novel optimization-based approach to uncover biologically coherent structures. BMC Bioinform. **9**, 268–283 (2008)
17. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**, 264–323 (1999)
18. McCormick, W.T., Schweitzer, P.J., White, T.W.: Problem decomposition and data reorganization by a clustering technique. Oper. Res. **20**, 993–1009 (1972)
19. Lenstra, J.K.: Clustering a data array and the traveling salesman problem. Oper. Res. **22**, 413–414 (1974)
20. Lenstra, J.K., Rinnooy Kan, A.H.G.: Some simple applications of the traveling salesman problem. Oper. Res. Q. **26**, 717–733 (1975)
21. Alpert, C.J., Kahng, A.B.: Splitting an ordering into a partition to minimize diameter. J. Classif. **14**, 51–74 (1997)
22. Climer, S., Zhang, W.: Rearrangement clustering: pitfalls, remedies, and applications. J. Mach. Learn. **7**, 919–943 (2006)
23. DiMaggio, P.A., McAllister, S.R., Floudas, C.A., Feng, X.J., Rabinowitz, J.D., Rabitz, H.A.: A network flow model for biclustering via optimal re-ordering of data matrices. J. Glob. Optim. (2009, in press)
24. DiMaggio, P.A., McAllister, S.R., Floudas, C.A., Feng, X.J., Rabinowitz, J.D., Rabitz, H.A.: Biclustering via optimal re-ordering of data matrices in systems biology: rigourous methods and comparative studies. BMC Bioinform. **9**, 458 (2008)
25. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., Altman, R.B.: Missing value estimation methods for DNA microarrays. Bioinformatics **17**, 520–525 (2001)
26. Koopmans, T.C., Beckmann, M.J.: Assignment problems and the location of economic activities. Econometrica **25**, 53–76 (1957)
27. Pardalos, P.M., Rendl, F., Wolkowicz, H.: The quadratic assignment problem: a survey. In: Pardalos, P.M., Wolkowicz, H. (eds.) Quadratic Assignment and Related Problems, vol. 16 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pp. 1–42. AMS, Rhode Island (1994)
28. Anstreicher, K., Brixius, N., Goux, J.P., Linderoth, J.: Solving large quadratic assignment problems on computational grids. Math. Progr. **91**(3), 563–588 (2002)
29. Loiola, E.M., de Abreu, N.M.M., Boaventura-Netto, P.O., Hahn, P., Querido, T.: A survey for the quadratic assignment problem. Eur. J. Oper. Res. **176**, 657–690 (2007)

30. Adams, W.P., Guignard, M., Hahn, P.M., Hightower, W.L.: A level-2 reformulation-linearization technique bound for the quadratic assignment problem. Eur. J. Oper. Res. **180**, 983–996 (2007)
31. Singh, S.P., Sharma, R.R.K.: A review of different approaches to the facility layout problems. Int. J. Adv. Manuf. Technol. **30**, 425–433 (2006)
32. Reynolds, C.H.: Designing diversed and focused combinatorial libraries of synthetic polymers. J. Comb. Chem. **1**(4), 297–306 (1999)
33. Floudas, C.A., Grossmann, I.E.: Synthesis of flexible heat exchanger networks with uncertain flowrates and temperatures. Comp. Chem. Eng. **11**(4), 319–336 (1987)
34. Ciric, A.R., Floudas, C.A.: A retrofit approach for heat-exchanger networks. Comp. Chem. Eng. **13**(6), 703–715 (1989)
35. Floudas, C.A., Anastasiadis, S.H.: Synthesis of distillation sequences with several multicomponent feed and product streams. Chem. Eng. Sci. **43**(9), 2407–2419 (1988)
36. Kokossis, A.C., Floudas, C.A.: Optimization of complex reactor networks-II: nonisothermal operation. Chem. Eng. Sci. **49**(7), 1037–1051 (1994)
37. Aggarwal, A., Floudas, C.A.: Synthesis of general separation sequences—nonsharp separations. Comp. Chem. Eng. **14**(6), 631–653 (1990)
38. CPLEX: ILOG CPLEX 9.1 User's Manual (2005)
39. McAllister, S.R., Feng, X.-J., DiMaggio, P.A. Jr., Floudas, C.A., Rabinowitz, J.D., Rabitz, H.: Descriptor-free molecular discovery in large libraries by adaptive substituent reordering. Bioorg. Med. Chem. Lett. **18**, 5967–5970 (2008)
40. DiMaggio, P.A., McAllister, S.R., Floudas, C.A., Feng, X.J., Rabinowitz, J.D., Rabitz, H.A.: Enhancing molecular discovery using descriptor-free rearrangement clustering techniques for sparse data sets (submitted for publication)
41. Shenvi, N., Geremia, J.M., Rabitz, H.: Substituent ordering and interpolation in molecular library optimization. J. Phys. Chem. **107**, 2066–2074 (2003)
42. Burkard, R.E., Karisch, S.E., Rendl, F.: QAPLIB—a quadratic assignment problem libary. J. Glob. Optim. **10**(4), 391–403 (1997)
43. Gilmore, P.C.: Optimal and suboptimal algorithms for the quadratic assignment problem. SIAM J. Appl. Math. **10**, 305–313 (1962)
44. Androulakis, I.P., Maranas, C.D., Floudas, C.A.: Prediction of oligopeptide conformations via deterministic global optimization. J. Glob. Optim. **11**, 1–34 (1997)
45. Klepeis, J.L., Floudas, C.A.: Free energy calculations for peptides via deterministic global optimization. J. Chem. Phys. **110**, 7491–7512 (1999)
46. Klepeis, J.L., Floudas, C.A., Morikis, D., Lambris, J.D.: Predicting peptide structures using NMR data and deterministic global optimization. J. Comp. Chem. **20**(13), 1354–1370 (1999)
47. Klepeis, J.L., Floudas, C.A.: Ab initio tertiary structure prediction of proteins. J. Glob. Optim. **25**, 113–140 (2003)
48. Klepeis, J.L., Floudas, C.A.: ASTRO-FOLD: a combinatorial and global optimization framework for ab initio prediction of three-dimensional structures of proteins from the amino acid sequence. Biophys. J. **85**, 2119–2146 (2003)
49. Klepeis, J.L., Floudas, C.A., Morikis, D., Tsokos, C.G., Argyropoulos, E., Spruce, L., Lambris, J.D.: Integrated computational and experimenal approach for lead optimization and design of compstatin variants with improved activity. J. Am. Chem. Soc. **125**(28), 8422–8423 (2003)
50. Fung, H.K., Floudas, C.A., Taylor, M.S., Zhang, L., Morikis, D.: Towards full sequence de novo protein design with flexible templates for human beta-defensin-2. Biophys. J. **94**, 584–599 (2008)
51. Lin, X., Floudas, C.A.: Design, synthesis and scheduling of multipurpose batch plants via an effective continuous-time formulation. Comp. Chem. Eng. **25**, 665–674 (2001)
52. Janak, S.L., Lin, X., Floudas, C.A.: Enhanced continuous-time unit-specific event based formulation for short-term scheduling of multipurpose batch processes: Resource constraints and mixed storage policies. Ind. Eng. Chem. Res. **43**, 2516–2533 (2004)