This is to certify that the

thesis entitled

PROJECTION AND CLUSTERING BY SIMULATED ANNEALING

presented by

Raymond W. Klein

has been accepted towards fulfillment
of the requirements for

Master of Science degree in Computer Science

Richard G. Schultz

Major professor

Date Oct. 21, 1987

PROJECTION AND CLUSTERING BY SIMULATED ANNEALING

By

Raymond W. Klein

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Computer Science

1987

ABSTRACT

PROJECTION AND CLUSTERING BY SIMULATED ANNEALING

By

Raymond W. Klein

Simulated annealing is a stochastic relaxation algorithm which has been used successfully to optimize functions of many variables. This thesis analyzes the simulated annealing algorithm when applied to the minimization of functions from two common problems encountered in exploratory pattern analysis, projection and clustering. The projection is a nonlinear mapping of patterns in high dimension to two dimensions. The simulated annealing mapping is compared to eigenvector projection as well as gradient descent minimization of the same objective function. The simulated annealing clustering is compared to a k-means algorithm.

Formal experiments are performed using analysis of variance to determine the effects of simulated annealing and data parameters on independent measures of mapping and cluster validity. The structure of clustered data sets is analyzed in the mapping problem using measures developed in the thesis. Standard cluster validity measures are used for the clustering problem.

Empirical results show that simulated annealing can produce results as good as those obtained by conventional methods, but are impractical for small data sets because of the high computational cost. Simulated annealing does, in the case of the mapping problem, yield a better optimization and better retained structure for large data sets containing tight gaussian clusters.

Acknowledgements

I would like to thank my advisor, Professor Richard Dubes, for all his help, insight, time, and good humor. He spurred my interest in this field, taught me the value of scientific research, and made it possible for me to finish. I also thank my committee, Dr. Anil Jain, Dr. George Stockman, and Dr. Roy Erickson. Thanks also go to all the folks in the PRIP lab and Computer Science Department for their interest and concern, especially my good friends Bruce McMillin and Ywhyng Lee. I also wish to thank my parents, Philip and Barbara, for their continued moral support and for providing the best possible educational foundation. Of course, my deepest gratitude is for my fiance Atsuko for her patience and encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis examines two problems from exploratory data analysis requiring optimization of functions of many variables. One problem is the mapping of data to a two-dimensional space by a nonlinear method and the other is the clustering of data. The goal of this thesis is to determine whether simulated annealing provides a practical solution to these problems. Simulated annealing has provided good solutions to several other problems involving the optimization of multi-variate functions as discussed in Chapter 2. In the first two sections of this chapter the nonlinear projection and clustering problems are presented and some of the difficulties discussed. The next section contains a statement of the goals of the thesis, and the final section outlines the remainder of the thesis.

1.1. Motivation - Nonlinear Projection

We assume the subjects of study are measured along several features that characterize them. The data collected for one subject along these features comprise one pattern. Patterns can be viewed as vectors in a high dimensional space, where each dimension corresponds to one feature. One type of exploratory data analysis maps a set of patterns onto a two-dimensional space to allow visual inspection of the projected patterns for structure. The mapping, however, is useful only if the structure present in the high dimensional space is preserved in the two-dimensional representation.

Many mapping algorithms have been developed, all of which may be categorized as being either linear or nonlinear. Linear algorithms such as eigenvector projection [BIS 81] and projection pursuit [FRI 74] vary in complexity, but are generally simpler to implement than nonlinear algorithms. However, linear algorithms may be unable to preserve complex structures present in the data [BIS 81]. Sammon [SAM 69] proposed a popular nonlinear mapping algorithm that preserves structure in the data by trying to maintain the interpoint distances of the high dimensional data in the plane. This algorithm has been shown to be a special case of multidimensional scaling [KRU 71]. This thesis studies the advantages and disadvantages of using simulated annealing to define the mapping.

The Sammon algorithm is as follows:

Begin with N patterns $\{x_1, x_2, ..., x_N\}$ in a feature space of L dimensions.

Generate an initial random configuration of N points $\{y_1, y_2, \cdots, y_N\}$ in an l-dimensional space, $l<L$. (We consider the case l=2.)

Define $d_{ij}^{*} = ||x_i - x_j||$ to be the distance between patterns x_i and x_j in the L-space and $d_{ij} = ||y_i - y_j||$ to be the distance between the corresponding points in the l-space.

Define an error (herein called stress) in the l-dimensional configuration as a function of $(y_1, y_2, \cdots, y_N)$.

Reconfigure the l-space points and use a steepest descent algorithm to search for a minimum stress.

Stress is minimized as a function of $l \cdot N$ coordinates, $\{y_{ij}\}, i=1,...,N$ and $j=1,...,l$. These coordinates are not restricted, so the solution space is continuous. The value of an acceleration coefficient (called the Magic Factor) and an upper bound on the number of iterations to be performed by the steepest descent algorithm must be supplied. The algorithm terminates when no significant decrease in stress is obtained over the course of several iterations. Computation times using Sammon's algorithm may become large when N is large [BIS 81]. Chang and Lee [CHA 73] propose a modification to

Sammon's algorithm whereby fewer points are moved to obtain a new configuration.

Here, the following stress function is used to perform the mapping.

$$E(y_1,\ldots,y_N) = \frac{1}{\sqrt{\sum_{i<j} d_{ij}^*}} \sum_{i<j} \frac{|d_{ij}^* - d_{ij}|}{d_{ij}^*} \tag{1}$$

Minimizing E tends to preserve interpoint distances between dimensions. Sammon's stress replaces the absolute difference in equation (1) with a square. Simulated annealing can be used with any type of stress function whereas gradient descent is limited to "smooth" functions whose gradients can be estimated. We feel that using absolute difference is somewhat more natural than using squared differences.

One drawback to Sammon's method has to do with the specifics of the steepest descent algorithm. The algorithm terminates when a minimum stress is found, but there is no way to know whether this is a local or global minimum. It is therefore necessary to run the algorithm several times, with different initial configurations. The run returning the smallest value for the stress is chosen as the best mapping. It is important to remember that some data, especially unstructured data, cannot be truly represented in l-dimensions. An ill-fitting l-dimensional configuration is difficult to detect.

1.2. Motivation - Square Error Clustering

Cluster analysis is one of the most prevalent tools in exploratory data analysis [AND 73] [DUB 76] [GOR 81]. We again begin with N patterns in L dimensions, but now wish to partition the patterns into g groups or clusters. Many different algorithms for performing clustering have been proposed, but algorithms that minimize a square error criterion have many theoretical and practical advantages. Gordon and Henderson [GOR 77] formalize a square error clustering algorithm as follows:

$X = \left[x_{ij} \right]$ is an $N \times L$ pattern matrix.

$Y = \left[y_{ik} \right]$ is an $N \times g$ cluster membership matrix where $y_{ik} \in \{0,1\}$; $y_{ik}=1$ if pattern i is in cluster k and $y_{ik}=0$ otherwise.

$Z = \left[z_{kj} \right]$ is a $g \times L$ matrix of cluster centers where $z_{kj} = \sum_{i=1}^{N} y_{ik} x_{ij} \; / \; \sum_{i=1}^{N} y_{ik}$

$$S(y_{11},\ldots,y_{Ng}) = \sum_{i=1}^{N} \sum_{k=1}^{g} y_{ik} \sum_{j=1}^{L} \left[x_{ij} - z_{kj} \right]^2 \text{ is the square error.} \qquad (2)$$

The problem is to minimize S with respect to the entries of Y under the constraints:

$\sum_{k=1}^{g} y_{ik} = 1$, $y_{ik} \geq 0$, and for all k, $y_{ik}=1$ for at least one value of i.

The constraints ensure that every pattern belongs to exactly one cluster and that no cluster is empty. Gordon and Henderson [GOR 77] reformulate the problem as an unconstrained optimization problem so that a gradient descent approach can be used to locate the minimum for S. The same computational difficulties described in Section 1.1 are encountered here. However, the optimization problem has a discrete nature because Y is known to be a binary matrix. The number of possible Y matrices is a stirling number of the second kind [AND 73].

$$S_N^{(g)} = \frac{1}{m!} \sum_{k=0}^{m} (-1)^{g-k} \left[\begin{matrix} g \\ k \end{matrix} \right] k^N \qquad (3)$$

Although finite, this number is much too large to permit an exhaustive search. For even the small problem of sorting 25 patterns into 5 groups the number of possible clusterings is

$$S_{25}^{(5)} = 2{,}436{,}684{,}974{,}110{,}751.$$

Fuzzy c-means algorithms [BEZ 81] also minimize square error criteria, but permit y_{ik} to be any value on the unit interval. This class of algorithms will not be examined in this thesis, but could be attacked by simulated annealing.

1.3. Statement of Thesis Problem

In this thesis simulated annealing will be used to minimize the objective functions E and S. These two problems provide good tests of whether simulated annealing is a practical solution procedure; the projection problem has a continuous solution space and the clustering problem, a discrete one. The simulated annealing solution to the nonlinear projection problem will be compared to the gradient descent solution, and the simulated annealing solution to the square error clustering problem will be compared to the Forgy algorithm for K-means clustering [DUB 76]. The focus will be on the selection of parameters in the simulated annealing algorithm. These parameters are discussed in Chapter 2, but only a few of them will be examined in detail.

The specific goal of this thesis is to study the effects of changes in problem parameters and simulated annealing parameters on solutions and to compare solutions using simulated annealing to those using standard methods. Only a few parameters and a few kinds of data are examined. Results are studied empirically and an analysis of variance is used to determine the significance of the effects observed.

1.4. Outline of Thesis

Chapter 2 discusses the details of the simulated annealing algorithm and defines all parameters. Chapter 3 defines how comparisons between algorithms are made in this thesis and how validity is measured. Chapters 4 and 5 report the experiments performed using the simulated annealing versions of the mapping and clustering algorithms. Analyses of the experiments are included with the description of the experiments. Finally, Chapter 6 summarizes the thesis and suggests future work.

Chapter 2

Simulated Annealing

Simulated annealing is a method of function optimization that tries to avoid the pitfalls inherent in other optimization methods, such as the steepest descent approach; i.e., it seeks the global or near global minimum of a function without getting trapped in a local minimum. Simulated annealing is one algorithm in the class of stochastic relaxation algorithms designed to optimize functions of several hundred variables or more, and is especially attractive when the functions are not smooth [ROM 85]. Based on a method for simulating physical systems with large numbers of particles [MET 53], simulated annealing has been used in solving circuit routing problems [KIR 83], image processing [CAR 85] [GEM 84] [SMI 83], traveling salesman problems [AAR 85] [VAN 86] [RAN 86] [ROS 86] [REE 87], and in ergonomic design [OTT 84]. Simulated annealing is referred to as statistical cooling in some of the recent literature [AAR 86B]. This chapter discusses the algorithm, defines all parameters, and explains the effects of each parameter.

2.1. Background

Simulated annealing derives its name from analagous processes in both glassblowing and metallurgy. When working a glass or metal object, local areas are heated until the material is pliable enough to bend into the desired shape. The object, however, will be extremely fragile due to the internal stresses produced by the bending. The object is brought to a stable configuration by annealing, which heats the object to a temperature

just below its melting point, and then cools it very slowly to allow the molecules to align themselves and crystallize.

Simulated annealing was proposed independently by Kirkpatrick, et al. [KIR 83] and by Cerny [CER 85] as a method for minimizing functions of many variables, including NP-hard problems. The idea was derived from an algorithm proposed by Metropolis, et al. [MET 53] who simulated molecular processes. The annealing algorithm models the minimization of a function of many variables as a Markov chain [KIR 85] of many states, where the state corresponding to minimum energy, a stable state, is sought. The Markov chain is simulated and allowed to run until it reaches steady state. Sampling a state then produces an optimal or near optimal solution.

In the mapping problem, one state of the Markov chain corresponds to one arrangement of the points in the l-dimensional space. In the clustering problem, one state corresponds to one Y matrix, or labelling of the patterns being clustered. The state space is the set of all possible states, so the mapping problem presents us with an infinite number of states, while the clustering problem exhibits a finite but extremely large number of states.

The optimization problem begins with a cost function of many variables, C, such as stress or square error which has known analytical form but is otherwise unrestricted. The minimum corresponds to the most stable state of the underlying system of variables. Only state changes corresponding to decreases in cost are accepted using a standard algorithm, such as steepest descent. Simulated annealing also accepts such state changes but, in addition, accepts states which increase cost with a probability determined by a new parameter called temperature. The simulated annealing algorithm is stated below as a generalization of the Metropolis [MET 53] algorithm.

Set $k = 0$

Choose initial temperature c_k.

Choose initial random configuration of points in l-space; (for mapping).

Choose initial Y matrix as a random configuration of clusters; (for clustering).

repeat

  repeat

    perturb configuration i with cost C_i to configuration j with cost C_j;

    if $(\Delta C_{ij}=C_j-C_i) \leq 0$ then

      accept configuration j

    else

      accept configuration j with probability $\exp(-\Delta C_{ij}/c_k)$

  until quasi-equilibrium at c_k is reached;

  $c_{k+1} = f(c_k)$, where f is monotone decreasing.

until $c_{k+1} \leq c_f$ indicating the system is frozen.

The control parameter c is analagous to temperature in physical annealing and C is the cost function, such as stress. The steps of updating the control parameter c through function f, determining quasi-equilibrium at each temperature, and determining stopping criteria, are collectively referred to as the cooling schedule. The crucial parts of the algorithm are the cooling schedule and the definition of "move", or the way in which the configuration is perturbed. The choice of cooling schedule is a very difficult problem and one for which there is no "best" choice for all problems. The most definitive work to date on the subject is presented by van Laarhoven and Aarts [VAN 86]. The definition of move depends on the data and corresponds to movement in the solution space. Thus, a discrete space, as in the clustering problem, requires a different definition of move than a continuous space, as in the mapping problem.

Convergence of the annealing algorithm with various cooling schedules has been proven [VAN 86] [GID 85] [LUN 86] [MAF 86] [MIT 86] [ROM 84] [ROM 85]. Convergence at each temperature, or control parameter, means that a Markov chain has

reached steady state [HAM 64]. Convergence of the entire algorithm means that an optimum state, or cost value, has been reached. An exponentially long time may be required for convergence, depending on the size of the problem. Various heuristics in addition to the physical analogy must therefore be used in determining the cooling schedule. A polynomial time algorithm is presented in [AAR 85]. Because of the relationship of configurations of the points to physical states, van Laarhoven and Aarts [VAN 86] define a Markov chain to describe the set of configuration states reached at one temperature. The relationship of Markov chains to simulated annealing is also discussed in [VAN 86] [GID 85] and [KIR 85]. Background material on Markov chains and statistical mechanics, as well as a proof of convergence at one temperature are available in [HAM 64].

The simulated annealing algorithm as presented is quite simple to implement. Careful implementation, however, is important. The implementation used in this thesis is not optimal, but simple time saving measures are employed. Computation of the cost functions is expensive. Computation of the change in cost after one move, however, is much cheaper if properly implemented.

2.2. Simulated Annealing Parameters

The two main parts of simulated annealing are the cooling schedule and the definition of perturbation, or move. The parameters of the cooling schedule are starting temperature, the rule for decreasing temperature, the run length at each temperature, and stopping temperature. This thesis concentrates on all but the choice of starting temperature. This section describes all the parameters and explains our choice of parameters for study.

2.2.1. Moves

The first simulated annealing parameter considered here is the definition of "move", or perturbation of the data. When deciding how to perturb a configuration, it is helpful to remember the analogy to physical annealing. A move should select a next state that is close to the current state in the state space. Most of the research in simulated annealing

has involved problems that have discrete states, as in the clustering problem. A move in the clustering problem will move a randomly chosen pattern from its present cluster to a different, randomly chosen cluster. This corresponds to changing the state of a randomly selected entry in Y, subject to the constraints on Y. For the clustering problem it is hard to visualize what a "large" or "small" move is.

The mapping problem has continuous variables and therefore an uncountable number of states. Vanderbilt and Louie [VAN 84] suggest that, for continuous problems, it is necessary to make relatively large moves through the state space at high temperatures to avoid wasting time searching a small area, but to make smaller moves through the state space at low temperature to avoid missing the global solution. A move in the continuous problem will relocate a randomly chosen point a random distance up to a certain fraction of the projection window, where the projection window is a region in the the l-dimensional space. The nature of moves in the mapping problem requires that the algorithm make smaller moves as the system cools. When an annealing is started at high temperature, essentially all moves, large and small, are accepted. As the system cools, fewer large moves and eventually only small moves are accepted [WHI 84].

The algorithm used in this thesis ties the maximum size of a move to the rate of accepted moves. Initially, moves are defined to be a random distance in the interval [$-fract$, $fract$] where $2 \cdot fract$ is a fraction (usually around 0.1) of the projection window in the dimension in which the window is largest. When the percentage of rejected moves exceeds 50% for any Markov chain, *fract* is adjusted to be 0.9 times the previous value of *fract*. This allows smaller moves to be made at lower temperatures. The cutoff of 50% and the interval change of 0.9 are arbitrary values and no attempt is made to optimize them.

2.2.2. Cooling Schedule

The cooling schedule is discussed in the remainder of this chapter. A summary of some effective cooling schedules is presented in [VAN 86], and individual schedules are available in [AAR 85] [BUR 84] [OTT 84] and [WHI 84]. The initial value of the control parameter c_0 should be large enough to allow the algorithm to accept any new

configuration regardless of the change in stress. The decrement rule $c_{i+1} = f(c_i)$ controls the outer loop and should cool the system slowly enough to provide an accurate solution without running an excessively long time. The value of the final parameter c_f determines when the system should halt processing, usually when very little improvement in the optimization has been made over the most recent Markov chains. The quantity *very little* must be specified by the user and must be determined empirically such that solutions are accurate and terminate in a reasonable amount of time.

The time required for individual Markov chains to reach quasi-equilibrium controls the inner loop and is another parameter of the cooling schedule. Again, the trade off is between time and accuracy. For convergence, every state must be visited infinitely often. The algorithm used here creates the opportunity for every state to be visited. The Markov chains should run long enough to provide a good solution, but not so long as to provide little extra information for the time spent. Theory [VAN 86] provides us with an upper bound on Markov chain length for which the solution will be acceptable, but the numerical bound depends on the number of possible states a configuration may reach — a prohibitively large number. Therefore, some heuristic must determine Markov chain length.

The cooling schedule used in this thesis follows van Laarhoven and Aarts [VAN 86] and was chosen for its relative simplicity and promise. The parameters are as follows:

Markov chain length - the number of accepted moves at one temperature

c_0 - the initial value of control parameter

$c_{k+1} = f(c_k)$ - the decrement rule

c_f - the final value of control parameter

δ - a number close to zero that controls the rate of cooling

ε - a number close to zero that controls the freezing point

The notation $C_i(c_k)$ refers to the cost (stress or square error) of configuration or state i when the control parameter is c_k. Note that cost depends on c_k algorithmically, not

functionally. The term $\bar{C}(c_k)$ approximates the statistical expectation of cost at temperature c_k and is the average cost over n accepted moves, achieved after the chain has reached equilibrium.

$$\bar{C}(c_k) = \frac{1}{n}\sum_{i=1}^{n} C_i(c_k) \tag{4}$$

Similarly, $\overline{C^2}(c_k)$ approximates the second moment of cost.

$$\overline{C^2}(c_k) = \frac{1}{n}\sum_{i=1}^{n} C_i^2(c_k). \tag{5}$$

The sample variance of cost is defined as

$$\sigma^2(c_k) = \overline{C^2}(c_k) - [\bar{C}(c_k)]^2. \tag{6}$$

The initial value of the control parameter is

$$c_0 = \sigma(\infty) \tag{7}$$

where $\overline{C^2}(\infty)$ and $\bar{C}(\infty)$ are computed from an initial Markov chain at a very high value of c. Only a rough approximation to c_0 is needed because c_0 need only be large enough to allow almost all moves to be accepted.

The decrement rule f is established by equation (8).

$$c_{k+1} = c_k \cdot \left[1 + \frac{ln(1+\delta)\cdot c_k}{3\sigma(c_k)}\right]^{-1} \tag{8}$$

The smaller δ, the slower the system cools. Also note the dependence on $\sigma(c_k)$; the larger the variance of costs for the chain, the slower the system cools. Parameter δ is one of the parameters studied in this thesis.

The final value of the control parameter, c_f, is taken to be the first value of the control parameter that satisfies

$$\frac{\sigma^2(c_f)}{c_f(\bar{C}(c_0) - \bar{C}(c_f))} < \varepsilon \tag{9}$$

for some small value of ε. The smaller ε, the longer the algorithm runs. We also

examine the choice of ε.

Note that both the decrement rule and stopping rule are affected by σ, which is in turn affected by the Markov chain length. Chains that are too short will have a large σ. The interdependence of Markov chain length and ε is examined.

No practical, theoretical method of determining Markov chain length is provided in any of the literature. Chain lengths for the mapping and clustering problems are based on the number of points in the data set. For the mapping problem, chain length is taken to be roughly twice the cardinality of the data set. In the clustering problem, the initial chain length is based on the cardinality of the data set, but chains are allowed to grow shorter as the annealing proceeds. As an annealing run approaches a solution, few moves are accepted. Allowing the minimum acceptable chain length to get smaller decreases the running time.

The choices for δ and ε are quite important and must be determined empirically. Some experimentation is necessary to strike a balance between speed and accuracy.

Chapter 3

Methodology

Stress, equation (1), and mean square error, equation (2), are the cost functions to be minimized by simulated annealing. Of interest are the effects of varying the simulated annealing parameters. For the projection problem in which stress is minimized, an emphasis is placed on analyzing how well the structure present in the data is preserved in the mapped configuration. For the clustering problem that minimizes square error, the validity of the clusterings obtained is examined. This chapter describes the methodology employed and the specifics of the experiments.

Although convergence of the annealing algorithm in finite time has been proven [GID 85] [MIT 86], convergence can be quite slow, as demonstrated in empirical studies [KIR 84] [VAN 86]. This thesis reports an empirical study on the effectiveness of simulated annealing using two different types of cost functions, one continuous and one discrete.

Data sets of varying sizes containing different structures are described in Section 3.2. Most of the work to date uses the final value obtained for the cost function as the measure of the quality of the minimization. In this thesis, the measures of validity are independent of the cost function. These measures are discussed in Section 3.3. The description of the analysis of variance is in Section 3.4 and the individual experiments are described in Section 3.5.

3.1. Problems and Parameters to Study

The cooling schedule defined in Section 2.2.2 contains many parameters, all of which influence the solution. Equations (8) and (9) in Section 2.2.2 show that the cooling schedule parameters are interrelated. This thesis is concerned with analyzing the effects of a few of these parameters on the quality of the mappings and clusterings obtained on particular data sets. Because of the large number of possible combinations of parameters to study, only a subset will be considered.

The effects of the cooling speed on the final value of the cost function have been studied [RAN 86] [REE 87] [WHI 84] and it is known that cooling too quickly results in a sub-optimal solution. In this thesis, cooling speed is controlled by the parameter δ. Experiments with the mapping algorithm were performed in order to study how the value of δ interacts with parameters of the data.

The effects of Markov chain length on final cost have also been studied [AAR 85]. Markov chains should be long enough to allow the system to reach quasi-equilibrium at each temperature. The importance of choosing good values of c_0 and of c_f has been demonstrated [WHI 84]. If c_0 is too low or if c_f is too high, then the annealing will terminate in a sub-optimal solution. The effects of Markov chain length and of ε, which controls c_f, are examined in an experiment using the clustering algorithm. The parameter c_0 is not examined in any experiment in this thesis, but slightly different calculations are required for the mapping algorithm than for the clustering algorithm. The formula given in Section 2.2.2 is used by the mapping algorithm and two times the formula is used by the clustering algorithm.

Previous analyses of annealing parameters have not included any kind of Monte Carlo study. In this thesis each experiment is conducted as a formal study using an analysis of variance to determine the significance of the effects observed. In order to perform an in-depth study, analysis of some parameters must necessarily be neglected. The experiments reported here study one annealing parameter and one data parameter. The annealing parameter δ is chosen because it is easier to select conservative values of ε and Markov chain length and still have a run terminate in a reasonable period of time. The δ,

ε, and Markov chain length are, however, all related. The relationship is shown in equations (8) and (9) if it is understood that differences in Markov chain length affect $\sigma(c_k)$. In order not to neglect other parameters one of the experiments examines ε and Markov chain length.

Data parameters not formally studied in this thesis are cluster overlap, the number of true clusters in the data, the number of elements per cluster, and relative cluster size. The effects of dimensionality are not examined, but sample runs are performed using data from different dimensions. The size of the data set is not examined either, but the only real effect is that Markov chains must be longer to allow the system to reach quasi-equilibrium. Consequently, the run times increase with the size of the data set.

Although continuous problems have been studied [VAN 84], specific applications with continuous parameters, such as the mapping problem, have not. By including results from a discrete problem such as the clustering problem, it is expected that more will be learned about the application dependencies of simulated annealing. Few theoretical claims can be made in the mapping and clustering problems. The shapes of the cost functions and the distributions of the independent measures discussed in Section 3.3 are unknown. What is important are the actual solutions obtained and whether or not they are reliable.

3.2. Data

In the experiments that follow, several different types of data are used. All four mapping experiments use clustered data; the experiments differ in the size of the data sets and in the kind of local structure present in the clusters. There are two clustering experiments, one using clustered data, the other, random data. The random data sets consist of points in a unit plane generated from a uniform distribution.

For the experiments using clustered data, the data parameter examined is the cluster spread, σ, or the standard deviation of the clusters. This σ should not be confused with the σ in Chapter 2. The clusters with small spread do not overlap, and the cluster centers are generated at random. Data sets of three different cluster shapes are used. *Cluster*

shape refers to the distinction between different types of local structure. A *gaussian cluster* consists of points generated from a multidimensional normal distribution with covariance matrix $\sigma^2 I$ where I is the identity matrix. A *simplex cluster* is a set of points equidistant from one another. Although high dimensional data are hard to visualize, we do have some idea of how a simplex mapped to two dimensions should look. The cluster spread in this case is the common interpoint distance. A *lattice cluster* is composed of points arranged on a regular 3×3×3 grid. The cluster spread in this case is twice the distance between two adjacent points on one edge of the cube outlined. Lattice clusters and simplex clusters are both examples of data sets that have very regular local structure. Lattice clusters are used in place of simplex clusters in the larger experiment that maps data with regular shape. Large simplex clusters can only occur in high dimensional spaces, but lattice clusters fit the dimensionality within the scope of this thesis.

3.3. Measures of Comparison

In order to objectively examine the results of the annealing algorithms it is necessary to use some independent measures of performance. Many measures currently are available to evaluate cluster validity, or measure the goodness of the partition. For the experiment involving clustered data, the true class labels are known so an external measure is needed. The modified Rand coefficient [HUB 85] [DUB 86] is used in this thesis. This coefficient compares the way pairs of patterns are treated by two partitions, one partition being defined by true class labels and the other being assigned by the algorithm. A value of 1 means the cluster labels match the prior labels and values near 0 correspond to a random labelling of the patterns. The formulation of this statistic is in Appendix B.

In the experiment that clusters random data, an internal measure of validity is needed since there are no true labels for the data. The modified Hubert index [DUB 86] is used here (Appendix B). This index is a correlation between the matrix of dissimilarities for the patterns and a model matrix, and is based on a Mantel statistic [MAN 67] [HUB 76].

The experiments on the mapping algorithm use the measures of local and global structure presented in Appendix A. The phrase *local structure* refers to the relationship among points in individual clusters. The statistic *alocal* measures the similarity between two clustered data sets by concentrating on how well small interpoint distances are reproduced. *Global structure* refers to the relative placement of clusters and is measured by the statistic *adist*.

3.4. Analysis of Variance

Analysis of variance is concerned with measuring the effects of various experimental factors. The measures of comparison discussed in Section 3.3 are used in the analysis. Statistical tests of hypothesis assess the effect of the parameters being examined on these measures. A two-way fixed effects analysis [AFI 72] is performed for each experiment. Each parameter in an analysis is called a factor. In the two-way analysis the factors are called A and B. In general, there are I levels of factor A, J levels of factor B and K observations or replications of the experiment for each of the IJ cells. Table 3-1 shows the layout of a sample, generic experiment in which factor A has 4 levels and factor B has 5 levels. The number in each cell is the average of the validity measure over the K replications. Pooling of the sum of squares [AFI 72] is not done.

Table 3-1 Sample ANOVA Table

Levels of Factor A	Levels of Factor B				
	1	2	3	4	5
1	0.0129	0.0313	0.0257	0.0021	0.0021
2	0.0339	0.0403	0.0129	0.0084	0.0082
3	0.0620	0.0765	0.0932	0.0722	0.0718
4	0.0620	0.0765	0.0932	0.0722	0.0718

Three null hypotheses are tested:

1. No main effects due to factor A are present.
2. No main effects due to factor B are present.

3. No interaction effects of factors *A* and *B* are present.

Values of the Fisher *F* statistic are calculated as in [AFI 72] and evaluated on the scale of an *F* distribution. In Table 3-2 we can see that factor *A* exhibits significant variability at the 90% level, factor *B* exhibits no variability, and no significant interaction between factors is present.

Table 3-2 Sample *F* test table

F-test	*Validity Measure*	*F* 90%
$F_{A\ (3,20)}$	2.836	2.38
$F_{B\ (4,20)}$	0.105	2.25
$F_{AB\ (12,20)}$	0.165	1.89

3.5. Experiments

Six experiments were performed to evaluate simulated annealing algorithms. Many factors may affect the outcomes of the mappings and clusterings, but because simulated annealing is slow, only a few of these factors can be examined. The mapping experiments (M1 through M4) use data whose structure is known so the success of the algorithm can be assessed. The analysis concentrates on the recovery of structure in data sets of varying size and of different cluster shapes. Two types of local structure are examined, random and regular, and experiments are performed using both large and small data sets. We do not expect cluster shapes to affect the quality of the mapping. All mapping experiments examine the same two factors in the analysis of variance, the cooling parameter, δ (Factor *B*), and cluster spread, σ (Factor *A*). Results are in Chapter 4.

A clustering algorithm should recover the clusters of naturally clustered data, but it is also of interest to see how an algorithm performs when there are no true clusters. Two clustering experiments (C1 and C2) are therefore performed with simulated annealing, one in which the clusters are known and one with random data in which the true clusters are not known. The clustering experiments are performed for two-dimensional data so

that the actual patterns can be plotted. Results are in Chapter 5.

The simulated annealing mapping algorithm is compared to a gradient descent optimization of the same stress function, and the clustering by simulated annealing is compared to a standard K-means algorithm. The objective here is to assess the practicality of simulated annealing algorithms from a computational viewpoint. The best of 100 runs is compared to one simulated annealing run in experiments C1, and C2. The best of 100 gradient descent mapping runs is compared to one simulated annealing run in experiment M1 and, following Sammon [SAM 69], the best of 20 runs is compared to one simulated annealing run in experiments M2, M3, and M4. Additionally, the mapping experiments include a comparison to mappings calculated by eigenvector projection, a non-iterative method. The benefits of the nonlinear mapping over eigenvector projection have been shown visually [SAM 69], but this thesis uses independent measures of validity.

Experiment M1 — Mapping of Small Gaussian Clusters

Each data set contains 15 patterns in five dimensions with three gaussian clusters of five points each. There are 3 levels of δ and of cluster spread ($I = J = 3$) and 3 replications per cell ($K = 3$). An effect due to δ is expected. However, this experiment is so small that experimental factors may overwhelm the results.

Experiment M2 — Mapping of Small Simplex Clusters

As in experiment M1, the data sets each contain three clusters of five patterns each. The difference here is that each cluster is a simplex, whose local structure is well defined and different from that of a gaussian cluster. Both factors are examined at 3 levels ($I = J = 3$) and $K = 3$. Since all simplexes of N dimension are the same, replications are made by running the mapping with a different random number seed.

Experiment M3 — Mapping of Large Gaussian Clusters

This is a larger experiment than M1, but is similar to M1. Each data set contains five gaussian clusters of 15 patterns each and $I = J = K = 3$. The cost function of the larger sets will have more local minima than the small sets and it is expected that more of

an effect do to δ will be observed here than in M1.

Experiment M4 — Mapping of Large Lattice Clusters

Each data set contains five clusters of 27 patterns each in four dimensions and $I = J = K = 3$. The clusters in this experiment are lattices. The local structure is similar to that from M2, but this experiment is larger and therefore the results should be more reliable than those in M2. This experiment differs from M3 primarily in the shape of the data being mapped, and may identify mapping algorithm dependencies on cluster shape.

Experiment C1 — Clustering of Gaussian Data

The effects of the number of clusters requested from the algorithm (factor A) and cluster spread (Factor B) are examined. Each data set contains four gaussian clusters of 13 points each in two dimensions. The number of clusters in the partition requested from the algorithm is varied between two and eight, so $J = 7$. Four levels of cluster spread are examined ($I = 4$) and $K = 10$. It is expected that the clusterings will deteriorate as the cluster spreads increase. We know the number of true clusters for the data and varying the number of clusters lets us assess the effects of selecting an incorrect number of clusters.

Experiment C2 — Clustering of Random Data

The data in this experiment are random, so there are no data parameters to choose as factors in the analysis of variance. Two annealing parameters, ε (factor A) and Markov chain length (factor B) are studied. Runs are performed at 3 levels for each factor ($I = J = 3$) with five replications ($K = 5$). Each data set contains 50 random points in two dimensions. Since there is no true partitioning of the data, this is a difficult clustering problem and should "work" the algorithm harder than in experiment C1. It is expected that these two factors interact a great deal.

Chapter 4

Mapping Experiments

The results of experiments using the simulated annealing mapping algorithm are presented in this chapter. These experiments involve analysis of the cooling parameter, δ, and cluster spread, σ. Objectives are discussed in Chapter 3. The experimental parameters for each experiment are defined, some graphical and numerical results are shown, and explanations of the results are offered. Results are summarized at the end of the chapter.

4.1. Experiment M1 - Small Gaussian Clusters

In this first experiment, several small data sets were generated and the quality of the mapping was measured while varying cooling parameter δ, defined in Section 2.2.2, and cluster spread, σ, defined in Section 3.2. The statistics *alocal* and *adist*, defined in Chapter 3, measure the quality of the mapping by assessing local and global structure.

Factor *A* in the analysis of variance was cluster spread σ with levels {0.01, 0.07, 0.20} while factor *B* was cooling parameter δ with levels {0.02, 0.1, 0.5}. Recall that δ controls how much the temperature of the system is lowered at each new Markov chain; the smaller δ, the slower the cooling. All other parameters are held constant with ε fixed at 10^{-10} and Markov chain length fixed at 30.

The data sets all consist of three 5-dimensional gaussian clusters of 5 points per cluster. Three data sets were generated for each level of σ. Gradient descent was

performed for two different values of the Magic Factor, *MF*. The average *alocal* and *adist* values are in Table 4-1 and Table 4-2. The Fisher *F* values from the fixed effects analysis of variance are in Table 4-3. Results for the eigenvector projection are also included in the column labelled 'Eigen'. Each cell is the average of $K = 3$ replications.

Table 4-1 Experiment M1, Average Local Structure

σ (A)	δ (B)			*MF*		Eigen
	0.02	0.1	0.5	0.3	0.6	
0.01	0.0129	0.0313	0.0257	0.0021	0.0021	0.0312
0.07	0.0339	0.0403	0.0129	0.0084	0.0082	0.1720
0.20	0.0620	0.0765	0.0932	0.0722	0.0718	0.4418

Table 4-2 Experiment M1, Average Global Structure

σ (A)	δ (B)			*MF*		Eigen
	0.02	0.1	0.5	0.3	0.6	
0.01	0.0018	0.0442	0.1541	3.47×10^{-4}	1.48×10^{-4}	1.06×10^{-5}
0.07	0.0027	0.0274	0.0180	0.0046	0.0032	1.63×10^{-4}
0.20	0.0076	0.3022	0.0422	0.0426	0.0415	0.0277

Table 4-3 Experiment M1, Analysis of Variance

F-test	*alocal*	*adist*	*F* 90%
$F_{A\ (2,18)}$	2.136	0.810	2.62
$F_{B\ (2,18)}$	0.105	1.153	2.62
$F_{AB\ (4,18)}$	0.165	1.122	2.29

These *F* values indicate there is no significant effect due to either cluster spread σ or δ, and no interaction between the two factors for either the *alocal* or the *adist* statistic. Figure 4-1 shows the final stresses for every run of this experiment and Figure 4-2 shows

the values of *adist* for every run. The stress does, however, increase with increasing δ for tightly clustered data. This is plausible since it is known that cooling too quickly can trap the annealing in a local minimum and, intuitively, the tighter the clusters, the deeper the local minima of the stress function. That is, many more moves that increase stress must be accepted to move a poorly placed, small cluster than to move a poorly placed, large cluster.

The eigenvector projections for the tightly clustered data retained over 99% of the variance. This is not unusual since three tight clusters occupy little more than two dimensions. The variance retained with the larger cluster spreads was about 96% and 80%.

Table 4-1 indicates that the annealing reproduces local structure better than gradient descent, relatively, as cluster spread increases. This is most likely an effect of the statistics themselves. The definition of *alocal* in Appendix A indicates that small changes in within cluster spread, *S*, will result in relatively large changes in *alocal*. Since the clusters are very small it is likely that small differences in within cluster spreads are not visually noticeable. Consequently, the differences between the observed values of the statistics for simulated annealing and for gradient descent are probably artifacts of cluster size. The values of *adist* with tight clusters are much better for gradient descent than they are for annealing. This too is probably an effect of the construction of the statistics.

Figure 4-3 shows various features of one annealing run with $\delta = 0.02$ in addition to the resultant mapping. Figure 4-4 provides the stress curves and final mappings obtained for δ values of 0.1 and 0.5. The final stresses of the tightly clustered data are correlated with the corresponding values of the *adist* statistic. Figures 4-3a, 4-4a, and 4-4c show the mappings obtained for δ values 0.02, 0.1, and 0.5, respectively, taken from one of the tightly clustered data sets. Note that the scales of these figures are not all the same. Although the cluster spreads appear about the same in every instance, the relative locations differ with the differing δ values. This confirms what is known about δ and indicates that *adist* is useful as an independent measure of the quality of these mappings.

Figure 4-3b plots the value of the control parameter c versus consecutive Markov chain number and Figure 4-3c shows the size of a move as a fraction of the projection window versus Markov chain number. Both the control parameter and move size decrease logarithmically. This is true for all runs of the mapping algorithm in this thesis. Figure 4-3d shows the maximum and minimum stresses reached at each Markov chain. The two curves overlap quite a lot because there are so many chains for this run. Notice that in every case the separation between stress curves slowly decreases until the run stops.

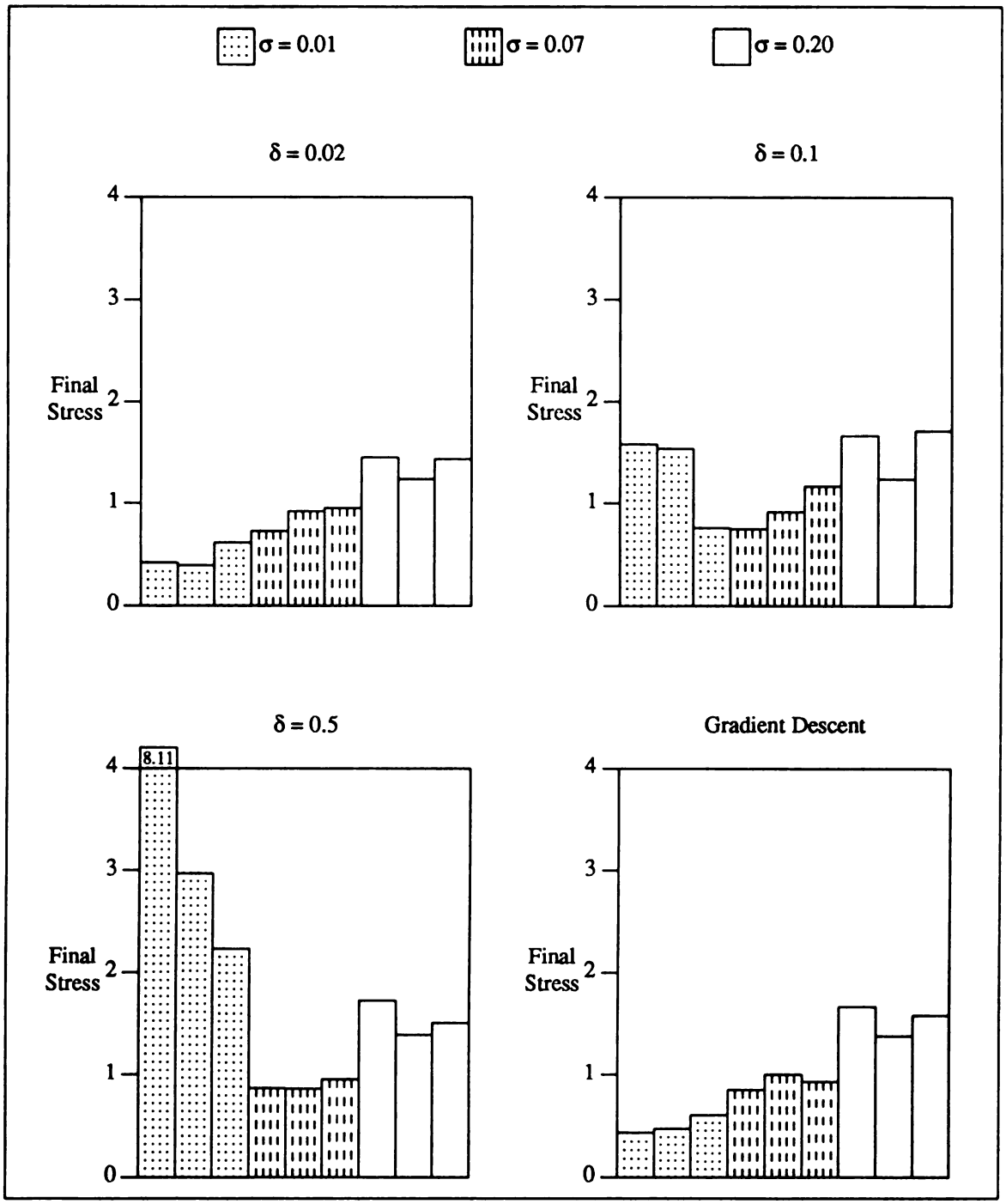

Figure 4-1 - Experiment M1. Plots showing final stress for all runs.

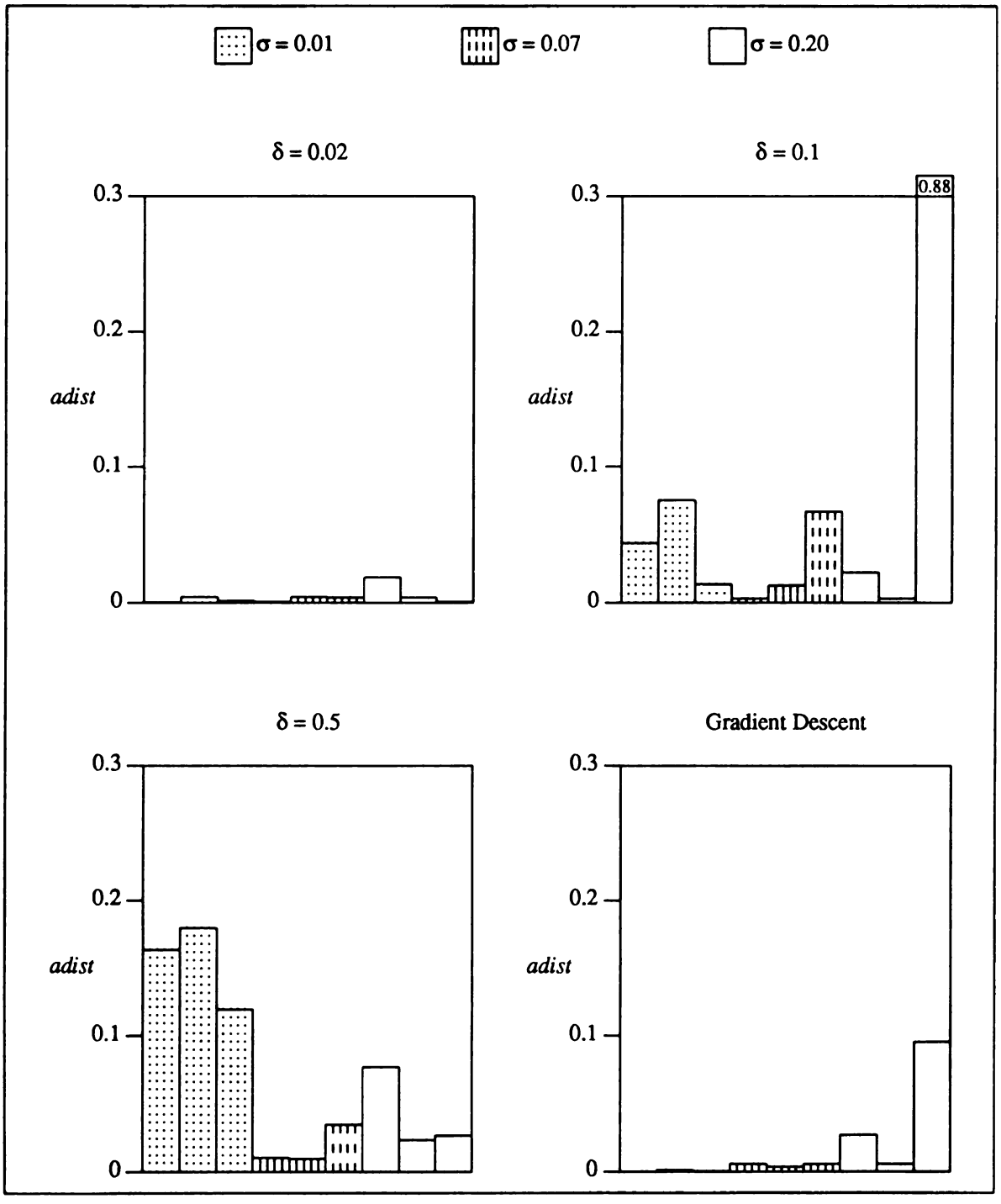

Figure 4-2 - Experiment M1. Plots showing *adist* for all runs.

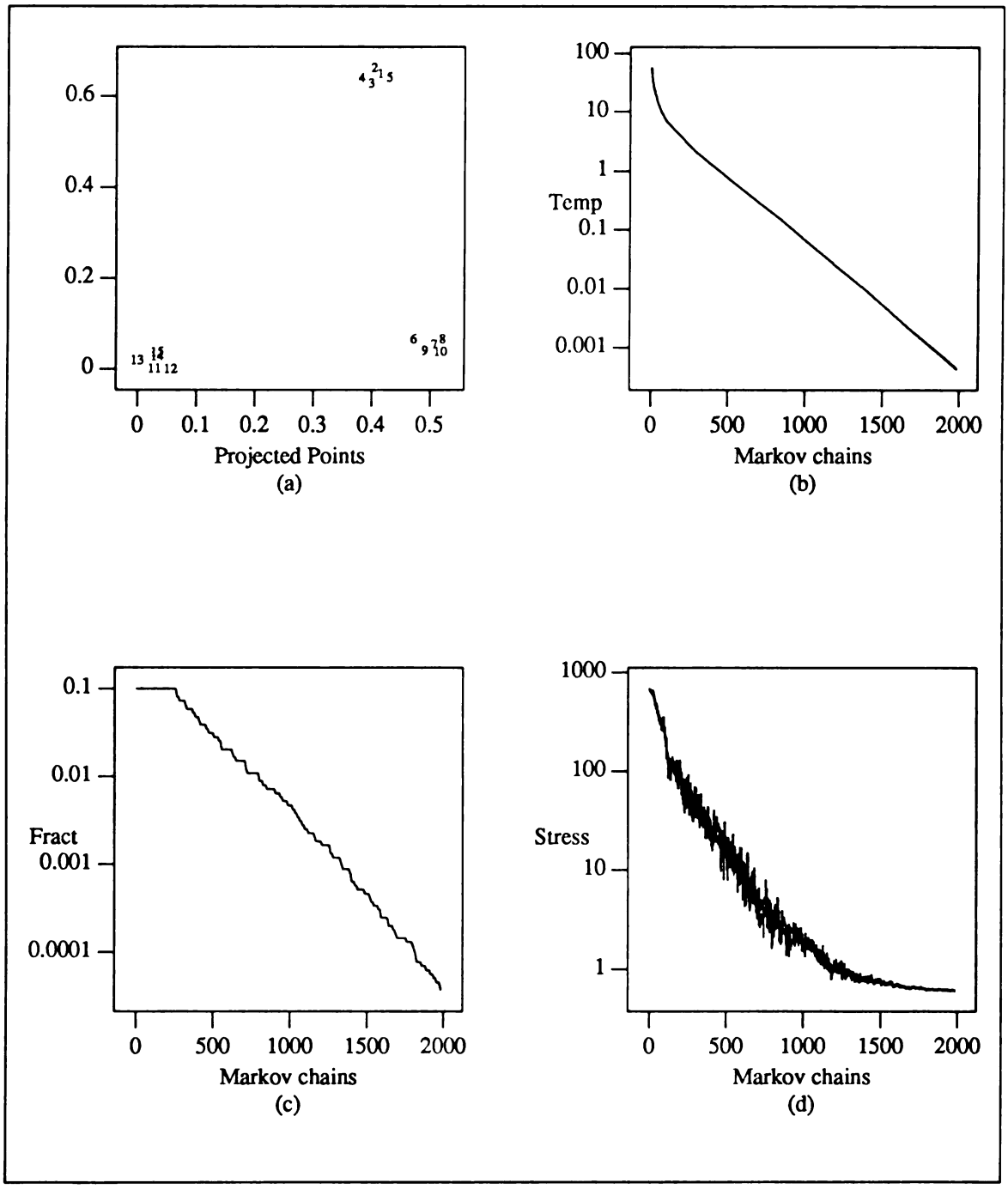

Figure 4-3 - Annealing curves for tightly clustered data.

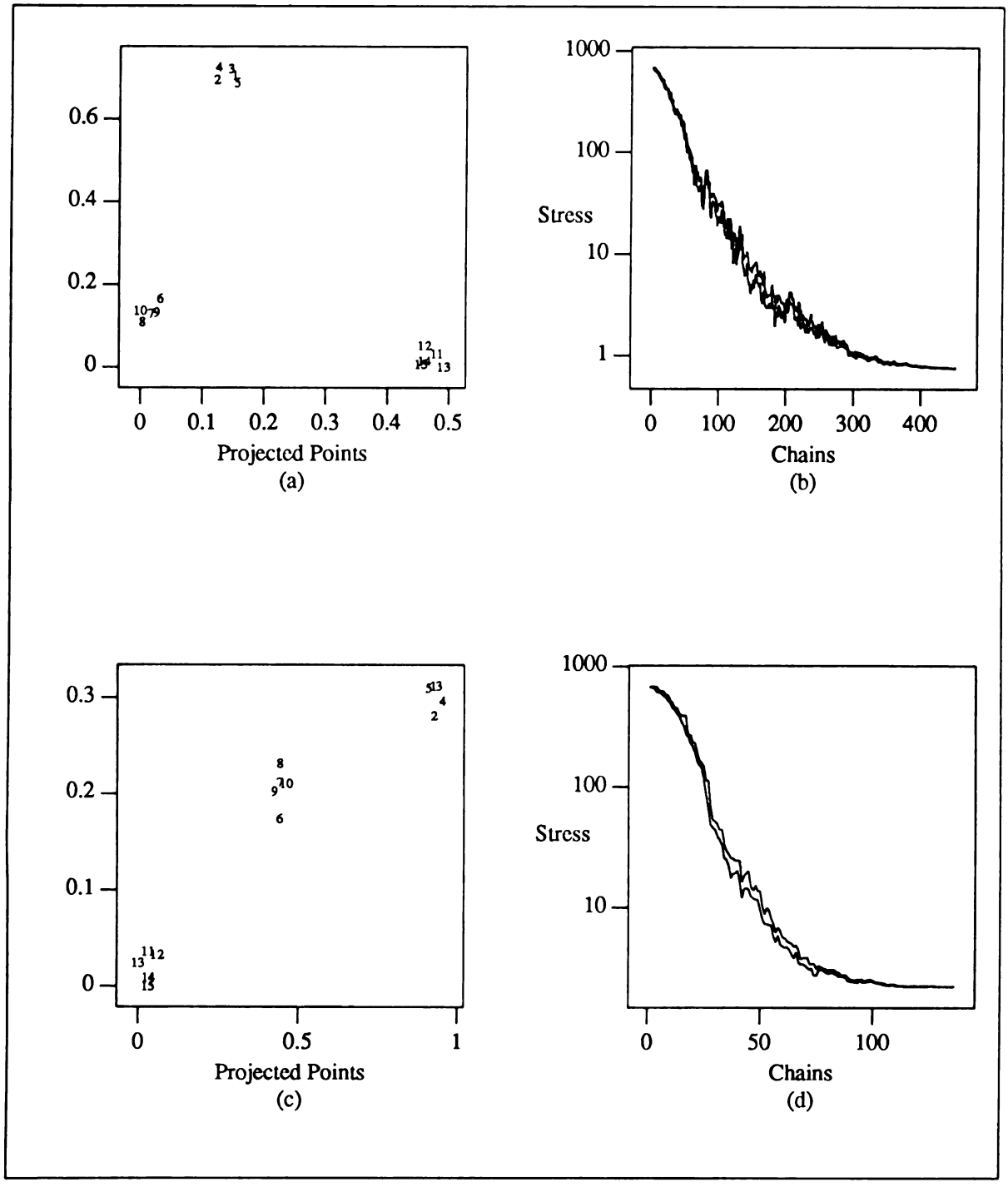

Figure 4-4 - Annealing curves for tightly clustered data.

Even though the slowest cooling produces a good mapping of the data, it is no better than that produced by the gradient descent mapping. One significant difference, however, is the run time. The slowest cooling requires almost 2000 Markov chains (temperatures) to converge and takes approximately 1100 seconds of CPU time. Even the fastest

cooling takes approximately 90 seconds, while the 100 gradient descent runs take 400 seconds for the tightly clustered data. The loosely clustered data requires approximately half as long to map under gradient descent as the tightly clustered data. The annealing is a little faster when mapping loosely clustered data than with tightly clustered data, but is still very slow. Eigenvector projection, which is not an iterative procedure, is the fastest of all, although the solutions obtained do not retain local structure well for the loosely clustered data. Values of *adist* for the eigenvector projection are lower than for other methods, but these differences are probably not significant for reasons previously discussed. A summary of approximate run times is in Table 4-4. The simulated annealing solution is as good as that obtained by gradient descent, but the run time is much longer, making simulated annealing impractical for such small data sets. The run times are also a function of the type of cooling schedule employed and of the specific implementation, which is a test program rather than a production program. Larger problems are discussed in Sections 4.3 and 4.4.

Table 4-4 Experiment M1, Run times

Projection	Approx. Run time (seconds)
$\delta = 0.02$	1100
$\delta = 0.1$	270
$\delta = 0.5$	90
Gradient Descent	400
Eigenvector Proj	0.5

4.2. Experiment M2 - Small Simplex Clusters

This experiment again looks at the quality of the mappings obtained for several small data sets while varying the annealing parameter, δ, and the data parameter, cluster spread. Factor *A* in the analysis of variance was cluster spread σ with levels {0.10, 0.20, 0.30}; factor *B* was cooling parameter δ with levels {0.02, 0.1, 0.5}. The value of ε was fixed at 10^{-9} and Markov chain length was fixed at 30. The data sets all consist of three

simplex clusters of 5 points per cluster in five dimensions.

The values of statistics *alocal* and *adist* averaged over the $K = 3$ replications are shown in Table 4-5 and Table 4-6. Replications are made by running the annealing with different random number seeds. Results for gradient descent and eigenvector projections are shown for comparison. Table 4-7 contains the Fisher F values for the analysis of variance.

Table 4-5 Experiment M2, Average Local Structure

σ (A)	δ (B)			*MF*		Eigen
	0.02	0.1	0.5	0.3	0.6	
0.10	2.68×10^{-2}	1.37×10^{-3}	2.16×10^{-2}	1.06×10^{-3}	1.37×10^{-3}	1.12×10^{-7}
0.20	6.49×10^{-4}	8.39×10^{-3}	2.54×10^{-3}	4.03×10^{-3}	4.70×10^{-3}	0
0.30	9.64×10^{-4}	4.74×10^{-3}	5.07×10^{-3}	8.89×10^{-3}	8.86×10^{-3}	0

Table 4-6 Experiment M2, Average Global Structure

σ (A)	δ (B)			*MF*		Eigen
	0.02	0.1	0.5	0.3	0.6	
0.10	8.98×10^{-4}	1.26×10^{-3}	2.93×10^{-3}	4.16×10^{-4}	6.56×10^{-5}	9.5×10^{-7}
0.20	8.90×10^{-4}	9.74×10^{-4}	1.79×10^{-3}	1.28×10^{-5}	6.58×10^{-5}	1.0×10^{-5}
0.30	7.66×10^{-4}	1.97×10^{-3}	1.77×10^{-3}	7.69×10^{-4}	7.64×10^{-4}	6.5×10^{-5}

Table 4-7 Experiment M2, Analysis of Variance

F-test	*alocal*	*adist*	*F* 90%	*F* 99%
$F_{A\ (2,18)}$	6.028	0.708	2.62	6.01
$F_{B\ (2,18)}$	0.829	5.254	2.62	6.01
$F_{AB\ (4,18)}$	3.257	1.094	2.29	4.58

Table 4-7 indicates that there is a significant effect on local structure due to σ and significant interaction between σ and δ. A significant effect on global structure occurs due to δ. This effect, however, is not visually detectable. Although not pictured, the apparent local and global structure looks about the same in mappings run with different δ.

The final stress values for each run are shown in Figure 4-5. The final stress does not vary much with δ. The δ value of 0.5 probably provided a slow enough annealing for this data. Larger values should result in higher final stress and smaller values, in longer run times. Figure 4-5 also shows that there is almost no difference in final stress between replicated runs for one value of σ, so annealing is relatively stable with respect to random number seed.

Figure 4-6 shows the values of *adist* for each run of the experiment. There is less correspondence between δ and the final stress values than in experiment M1 (Figure 4-2). All of the *adist* values are, however, very small and all of the mappings produced look like fairly good representations of the original data. The difference in *alocal* values observed between different methods used in this experiment is probably not significant even though the analysis of variance indicates that there are significant effects. As explained in Section 4.1, this is probably due to the small size of the data set and the way *alocal* is constructed.

The final stress values obtained by gradient descent are slightly higher than those obtained by simulated annealing, although the values of *adist* are about the same or lower. Values of *alocal* are roughly the same as those obtained by simulated annealing except when σ is smallest, in which case gradient descent has smaller values of *alocal*. The final mappings again look like fair representations of the data, although the gradient descent mapping looks a little better because the shape of the clusters is more regular. This result is curious and indicates a fault in either the local structure statistic or of the choice of stress as the criterion function. A similar result occurs in experiment M4.

The variance retained in the eigenvector projection of the tightly clustered data is 98%. As mentioned in experiment M1, we should expect three tight clusters to fit well in

two dimensions. The larger clusters have somewhat less of the variance retained, 94% and 89%. The values of *alocal* and *adist* for the mappings made by eigenvector projection are zero or almost zero. The mappings, however, are not very good representations of the data, since many of the points in the mappings project to the same point. This happens because the clusters are simplexes. Because each cluster of the mapping is exactly the same, *alocal* is zero, even though the mapping is not a good representation.

Figure 4-7 shows mappings obtained by all three methods for a σ value of 0.10. Since the original data contains clusters in which the points are all equidistant from one another, the mapped clusters should look roughly like pentagons. The gradient descent and the simulated annealing mappings look like good representations of the data, although one of the simulated annealing mappings (Figure 4-7b) contains two closely spaced points (14 and 15). This condition is not detected by *alocal*.

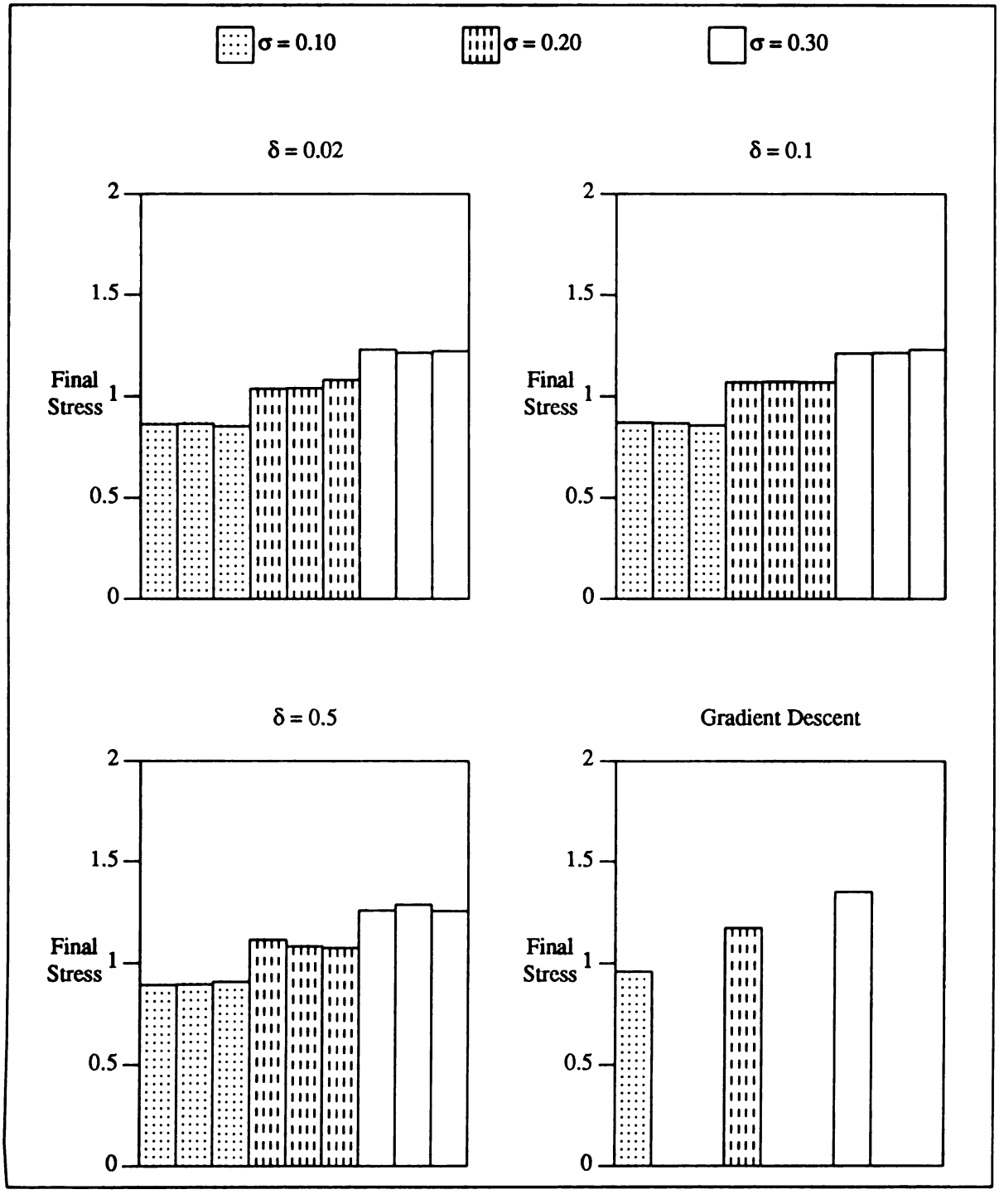

Figure 4-5 - Experiment M2. Plots showing final stress for all runs.

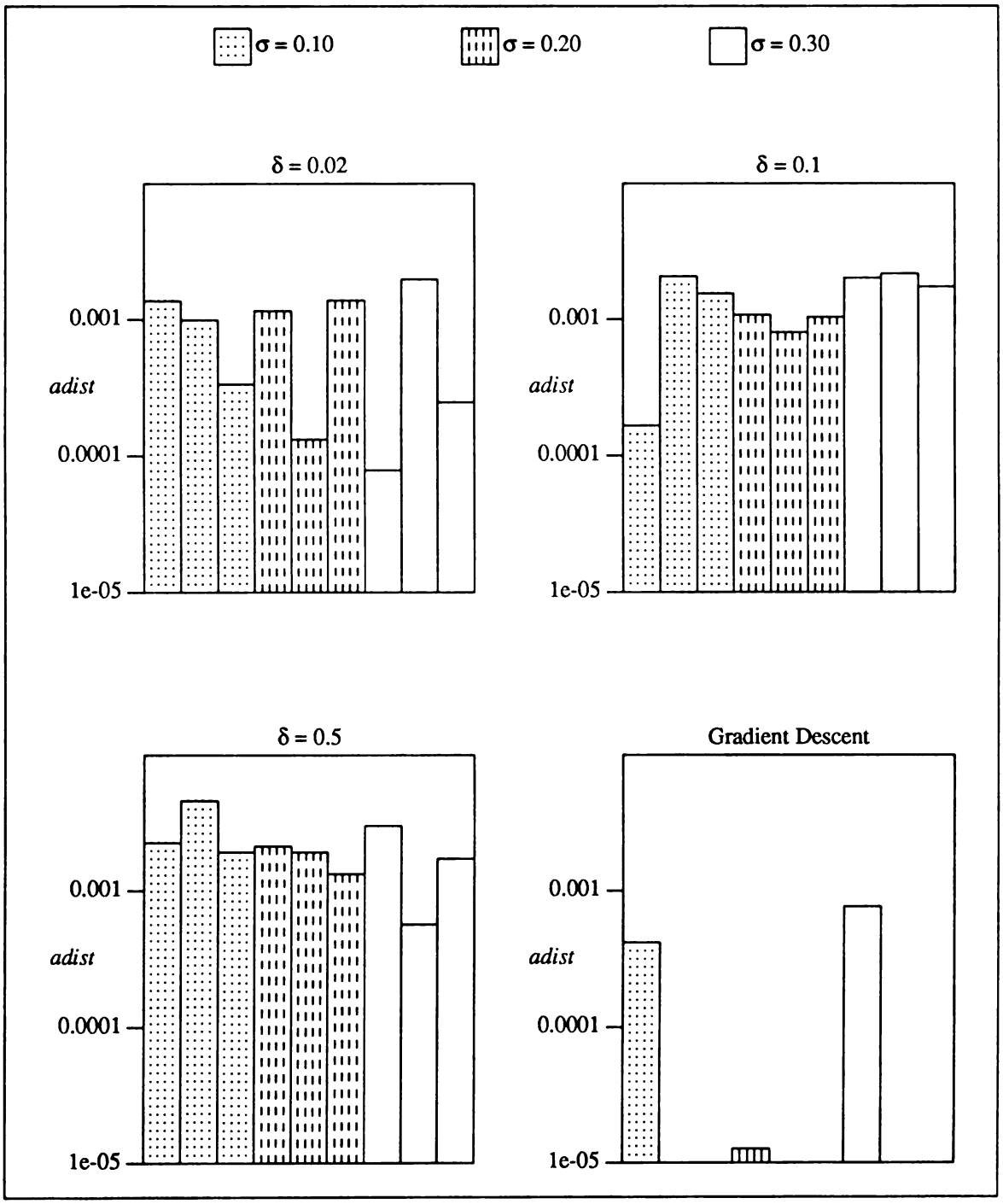

Figure 4-6 - Experiment M2. Plots showing *adist* for all runs.

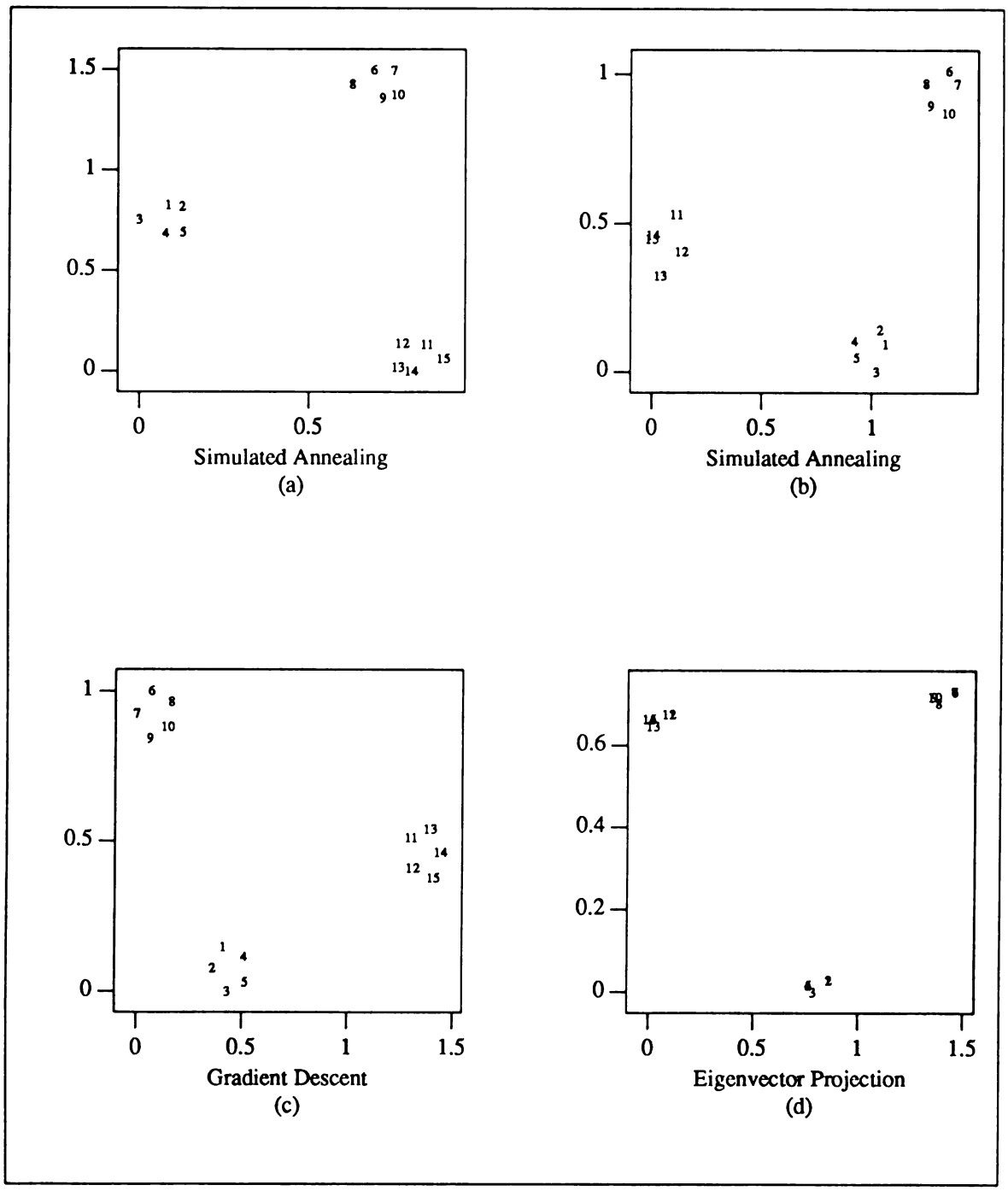

Figure 4-7 - Experiment M2, Assorted Mappings.

The annealing curves for the mappings in this experiment are similar in character to those from experiment M1 (Figure 4-3). The approximate run times are shown in Table 4-8. It is interesting to note that, at a fixed value of δ, the run times tend to increase as σ decreases. Because simulated annealing is so slow and the results are no better than

those obtained by gradient descent, it is not practical to use simulated annealing for such small data sets.

Table 4-8 Experiment M2, Run times

Projection	Approx. Run time (seconds)
$\delta = 0.02$	850
$\delta = 0.1$	230
$\delta = 0.5$	80
Gradient Descent (20 runs)	25
Eigenvector Projection	0.5

4.3. Experiment M3 - Large Gaussian Clusters

This experiment looks at the quality of the mappings obtained for several large data sets. Factor *A* in the analysis of variance was cluster spread σ with levels {0.01, 0.07, 0.21} and factor *B* was cooling parameter δ with levels {0.1, 0.5, 2.5}. The value of ε was fixed at 10^{-9} and Markov chain length was fixed at 150. The data sets all consist of five gaussian clusters of 15 points per cluster in six dimensions. There are three data sets for each level of σ.

The values of statistics *alocal* and *adist* are shown in Table 4-9 and Table 4-10, averaged over $K = 3$ replications per cell. Results for gradient descent and eigenvector projection are shown for comparison. Table 4-11 contains the Fisher *F* values for the analysis of variance.

Table 4-9 Experiment M3, Average Local Structure

σ (A)	δ (B)			*MF*		Eigen
	0.1	0.5	2.5	0.3	0.6	
0.01	0.0080	0.0305	22.6086	0.3876	0.1384	0.0692
0.07	0.0114	0.0139	0.0107	0.0134	0.0135	0.0597
0.21	0.0432	0.0301	0.0586	0.0393	0.0395	0.0191

Table 4-10 Experiment M3, Average Global Structure

σ (A)	δ (B)			*MF*		Eigen
	0.1	0.5	2.5	0.3	0.6	
0.01	0.1162	0.6080	0.4957	0.0690	0.0780	0.4038
0.07	0.0375	0.0401	0.0510	0.0496	0.0496	0.1696
0.21	0.0690	0.0708	0.0790	0.0531	0.0528	0.1799

Table 4-11 Experiment M3, Analysis of Variance

F-test	*alocal*	*adist*	*F* 90%	*F* 99%
$F_{A\ (2,18)}$	0.999	20.167	2.62	6.01
$F_{B\ (2,18)}$	1.003	3.824	2.62	6.01
$F_{AB\ (4,18)}$	1.000	3.571	2.29	4.58

Table 4-11 indicates very significant effects on global structure due to σ and significant effects due to δ and factor interaction, but no significant effects on local structure. The effect due to δ confirms results of experiment M1. The cell to cell variation is particularly evident in Tables 4-9 and 4-10 for tightly clustered data. When δ is too large the relative distances between clusters are not reproduced well in the mapping. However, it is not necessarily the case that the larger δ, the worse the results. A mapping made with too large a value of δ may occasionally produce adequate results.

The effect on global structure due to cluster spread can be traced to the stress function. Intuition dictates that stress functions corresponding to data sets with tight clusters have deeper local minima than those for more loosely clustered data. More annealing moves that increase stress are necessary to relocate an entire cluster to another minimum in the stress function for tightly clustered data than for loosely clustered data. Figure 4-8 shows the final stress values and Figure 4-9 shows the values of *adist* for each run of this experiment. The final stress values and *adist* values seem to correlate for the tightly clustered data as they did in experiment M1.

The results of the gradient descent mappings are, for the most part, comparable to the results from annealing, but in a few instances, the results from annealing are actually better. The values of *alocal* for each run in the experiment are shown in Figure 4-10. Although no significant effects on *alocal* were observed from the analysis of variance, the values of *alocal* for the mappings with small δ of the tightly clustered data are lower than for the corresponding gradient descent mappings. Additionally, the final stress obtained for the tightly clustered data was lower than that obtained by gradient descent, and the values of *adist* for the tightly clustered data were comparable to gradient descent in two of the three runs. The apparently poor performance of simulated annealing for δ of 0.1 and σ of 0.01 compared to gradient descent is caused by one run with a particularly large *adist*. This can be seen in Figure 4-9.

It is not surprising that simulated annealing performs relatively better for this large experiment than for the small data sets of experiments M1 and M2. The stress functions of the larger data sets are functions of more variables, and therefore have more complex shapes, probably containing more local minima.

The mappings made by eigenvector projection were not as good as those made by minimizing stress. The variances retained for the tightly clustered data, loosely clustered, and very loosely clustered data were approximately 75%, 82%, and 67%. The *alocal* and *adist* values were also larger in almost every case. Figure 4-11 shows final mappings of the tightly clustered data produced by each of the three methods. The simulated annealing clusters appear to have better local structure than those by gradient

descent. Remember that the clusters are very tight and that each cluster contains fifteen points.

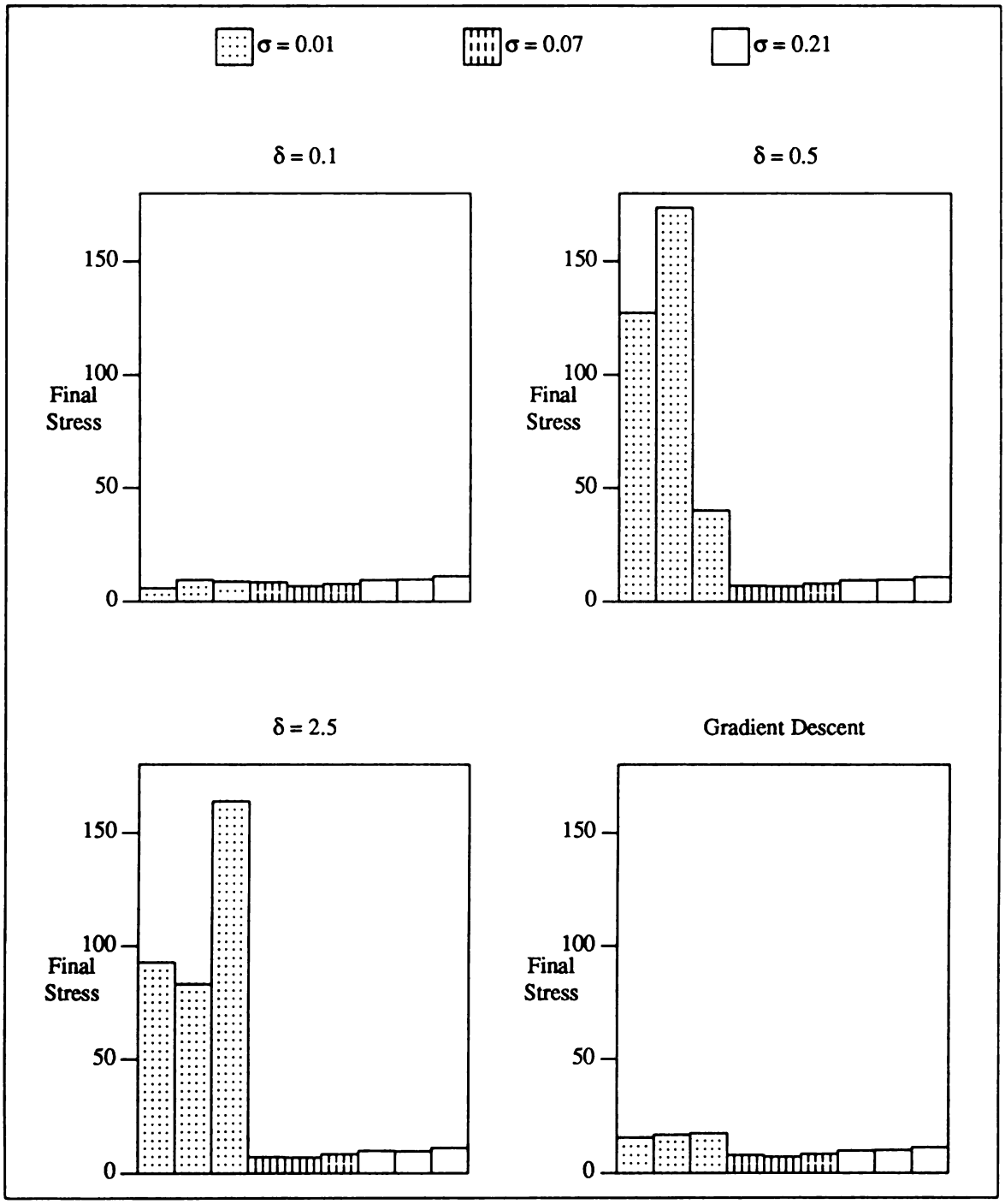

Figure 4-8 - Experiment M3. Plots showing final stress for all runs.

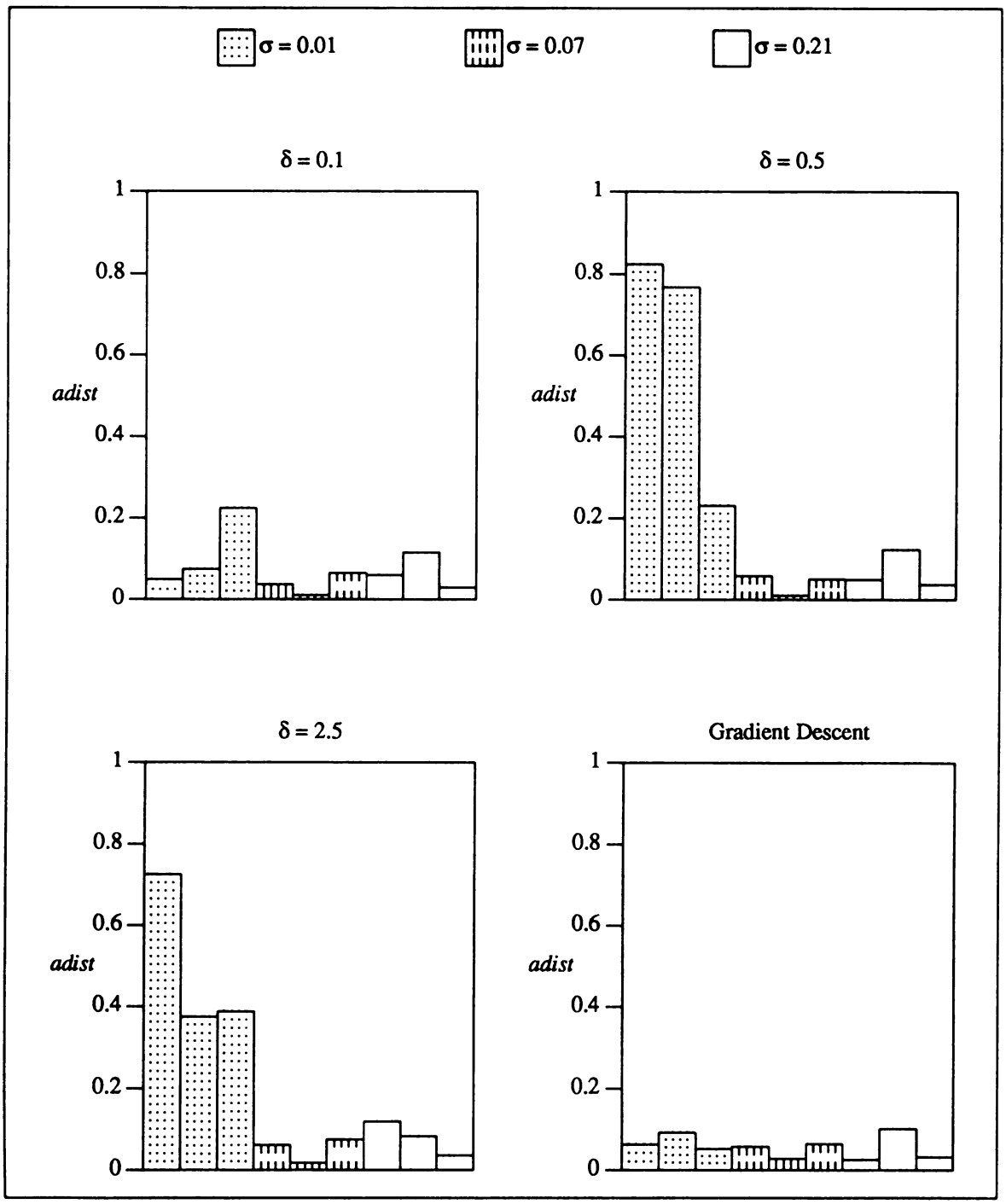

Figure 4-9 - Experiment M3. Plots showing *adist* for all runs.

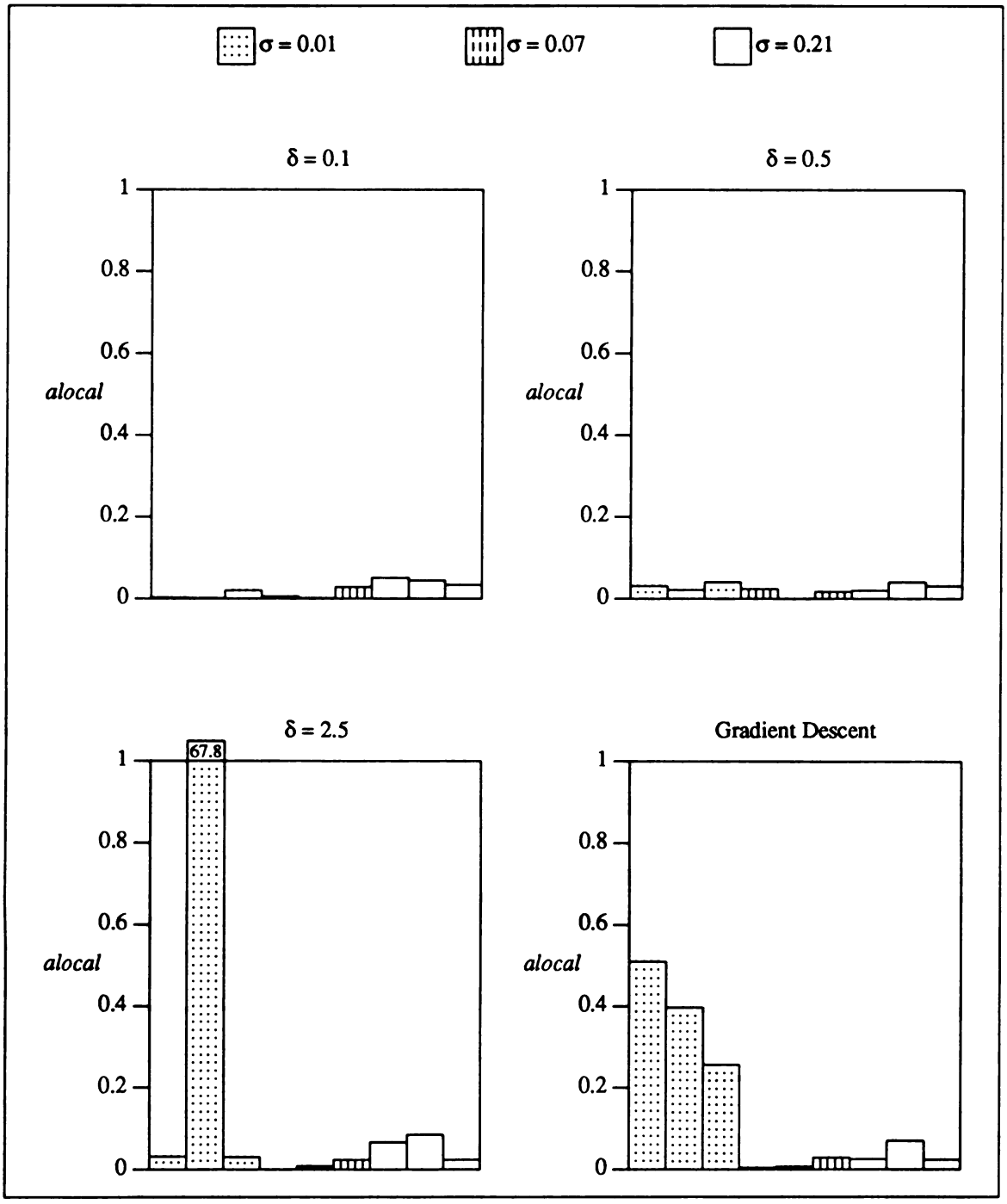

Figure 4-10 - Experiment M3. Plots showing *alocal* for all runs.

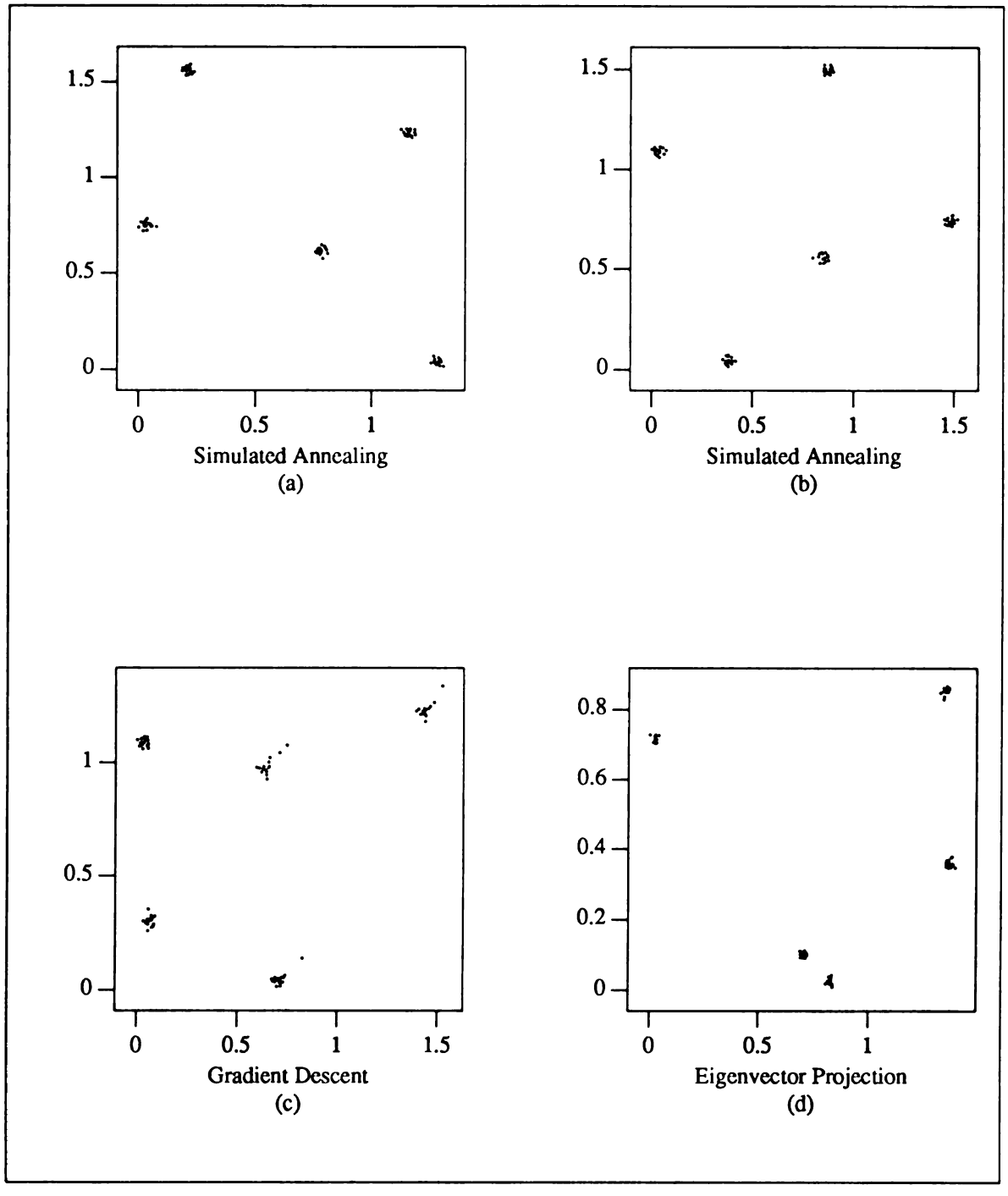

Figure 4-11 - Experiment M3, Assorted Mappings.

The approximate run times for this experiment are shown in Table 4-12. Both the simulated annealing and gradient descent take longer when mapping the tightly clustered data than when mapping other, more loosely clustered data, although the annealing can take up to 50% longer to map tight clusters than to map loose clusters. Simulated

annealing is still significantly slower than gradient descent, but takes only five times as long as gradient descent. In experiment M2, simulated annealing was about 35 times slower than gradient descent. There seems to be some evidence that simulated annealing may be practical for large problems.

Table 4-12 Experiment M3, Run times

Projection	Approx. Run time (seconds)	
	$\sigma = 0.01$	$\sigma > 0.01$
$\delta = 0.1$	11000	7200
$\delta = 0.5$	2700	2500
$\delta = 2.5$	1500	
Gradient Descent (20 runs)	2200	1500
Eigenvector Projection	0.5	

4.4. Experiment M4 - Large Lattice Clusters

This experiment looks at the quality of the mappings obtained for three large data sets. Factor *A* in the analysis of variance was cluster spread σ with levels {0.2, 0.4, 0.6} and factor *B* was cooling parameter δ with levels {0.1, 0.5, 2.5}. The value of ε was fixed at 10^{-9} and Markov chain length was fixed at 200. The data sets all consist of five lattice clusters of 27 points per cluster in four dimensions. Replications were made by running the annealing on a single data set while varying the random number seed.

The values of statistics *alocal* and *adist* averaged over the $K = 3$ replications are shown in Table 4-13 and Table 4-14. Results for gradient descent and eigenvector projection are shown for comparison. Table 4-15 contains the Fisher *F* values for the analysis of variance.

Table 4-13 Experiment M4, Average Local Structure

σ (A)	δ (B)			*MF*		Eigen
	0.1	0.5	2.5	0.3	0.6	
0.20	0.0032	0.0026	1.9488	0.0022	0.0022	0.0077
0.40	0.0025	0.0041	1.2555	0.0020	0.0021	0.0075
0.60	0.0118	0.0128	0.5412	0.0063	0.0063	0.0065

Table 4-14 Experiment M4, Average Global Structure

σ (A)	δ (B)			*MF*		Eigen
	0.1	0.5	2.5	0.3	0.6	
0.20	0.0332	0.0332	0.0359	0.0491	0.0493	0.3454
0.40	0.0377	0.0413	0.0373	0.0493	0.0491	0.7018
0.60	0.0672	0.1363	0.1143	0.0931	0.0925	2.4519

Table 4-15 Experiment M4, Analysis of Variance

F-test	*alocal*	*adist*	*F* 90%	*F* 99%
$F_{A\ (2,18)}$	0.6707	5.8786	2.62	6.01
$F_{B\ (2,18)}$	6.4399	0.5593	2.62	6.01
$F_{AB\ (4,18)}$	0.6982	0.4822	2.29	4.58

Significant effects occur on global structure due to σ at the 90% level and on local structure due to δ at the 99% level. A significant effect on global structure due to cluster spread was also observed in experiment M3, and is most likely related to the shape of the stress function. That is, the stress function of the more tightly clustered data set has deeper local minima than that for the loosely clustered data. The effect on local structure due to δ has to do with the type of sub-optimal mapping obtained by cooling too quickly. In experiments M2 and M3, cooling too quickly produced mappings in which the points were correctly mapped into clusters, but inter-cluster distances were not reproduced

correctly. Several of the sub-optimal mappings in this experiment contain clusters with outliers. The presence of an outlier significantly changes the local structure of a cluster.

The final stress for each run of this experiment is in Figure 4-12. Values of *adist* are shown in Figure 4-13. The minimization is as usual most reliable for the smallest δ. There is also correspondence between the final stress and the value of *adist*. The worst values of *alocal* were obtained with a δ of 2.5, due to outliers in the mappings. The high values of *alocal* in Figure 4-14 correspond to the higher values of stress.

Figure 4-15 shows mappings of the tightly clustered data obtained by all three mapping methods. The visual quality of the mappings obtained by simulated annealing is about the same as that of those obtained by gradient descent, but the local structure is not quite as nice. The final stress and *adist* values are slightly lower for simulated annealing, but the *alocal* values are slightly lower for gradient descent. The lower stress indicates that simulated annealing is the better optimization, although the local structure is not preserved as well as with gradient descent. As mentioned in Section 4.2, this is either a fault of the local structure statistic or of the choice of the stress criterion function.

The results of the eigenvector projection were significantly worse than in either of the other mappings. The retained variances for the increasing cluster spreads are 83%, 79%, and 75%. These data sets of five clusters in four dimensions can not be very well represented in two dimensions by using only the principal components.

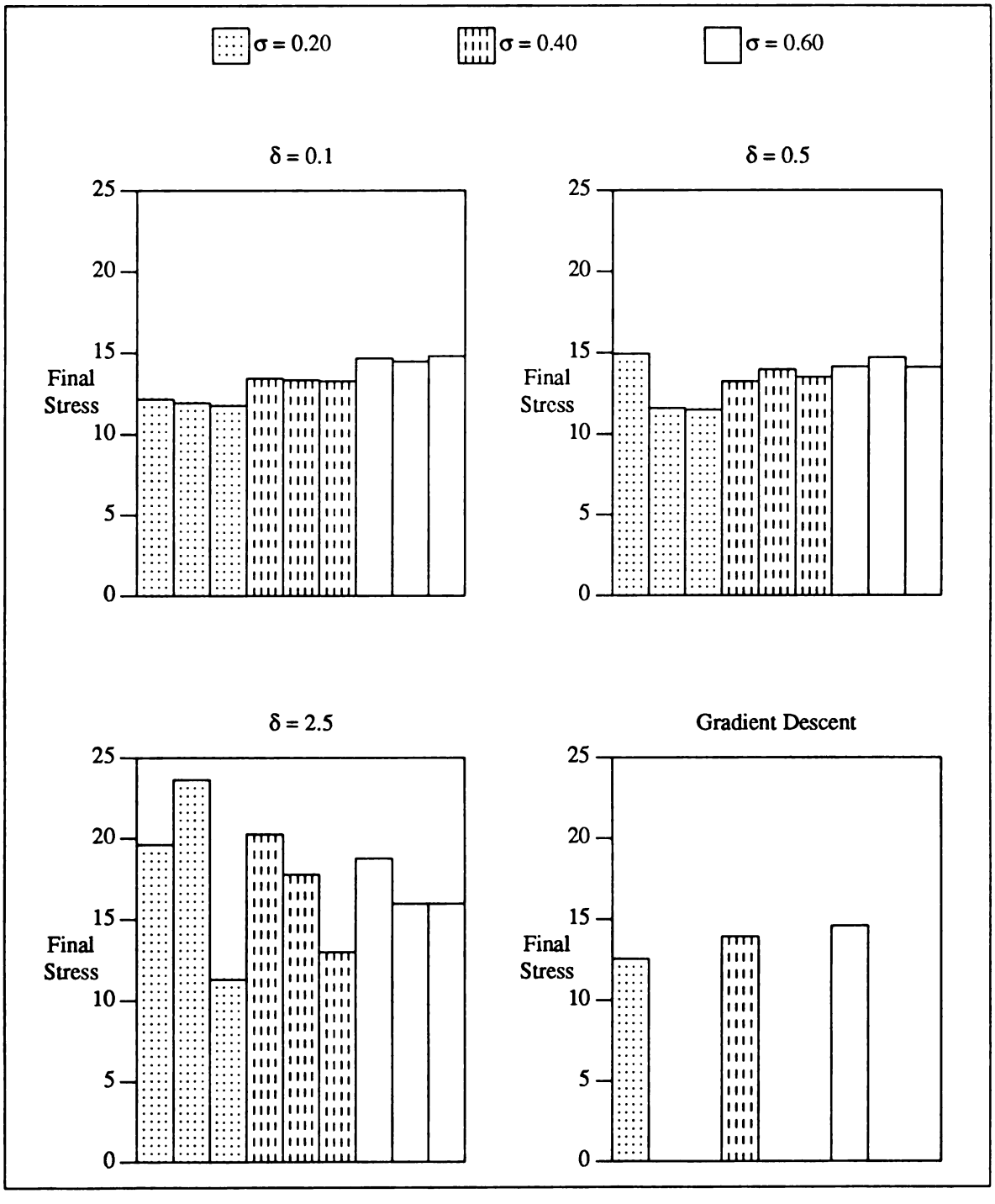

Figure 4-12 - Experiment M4. Plots showing final stress for all runs.

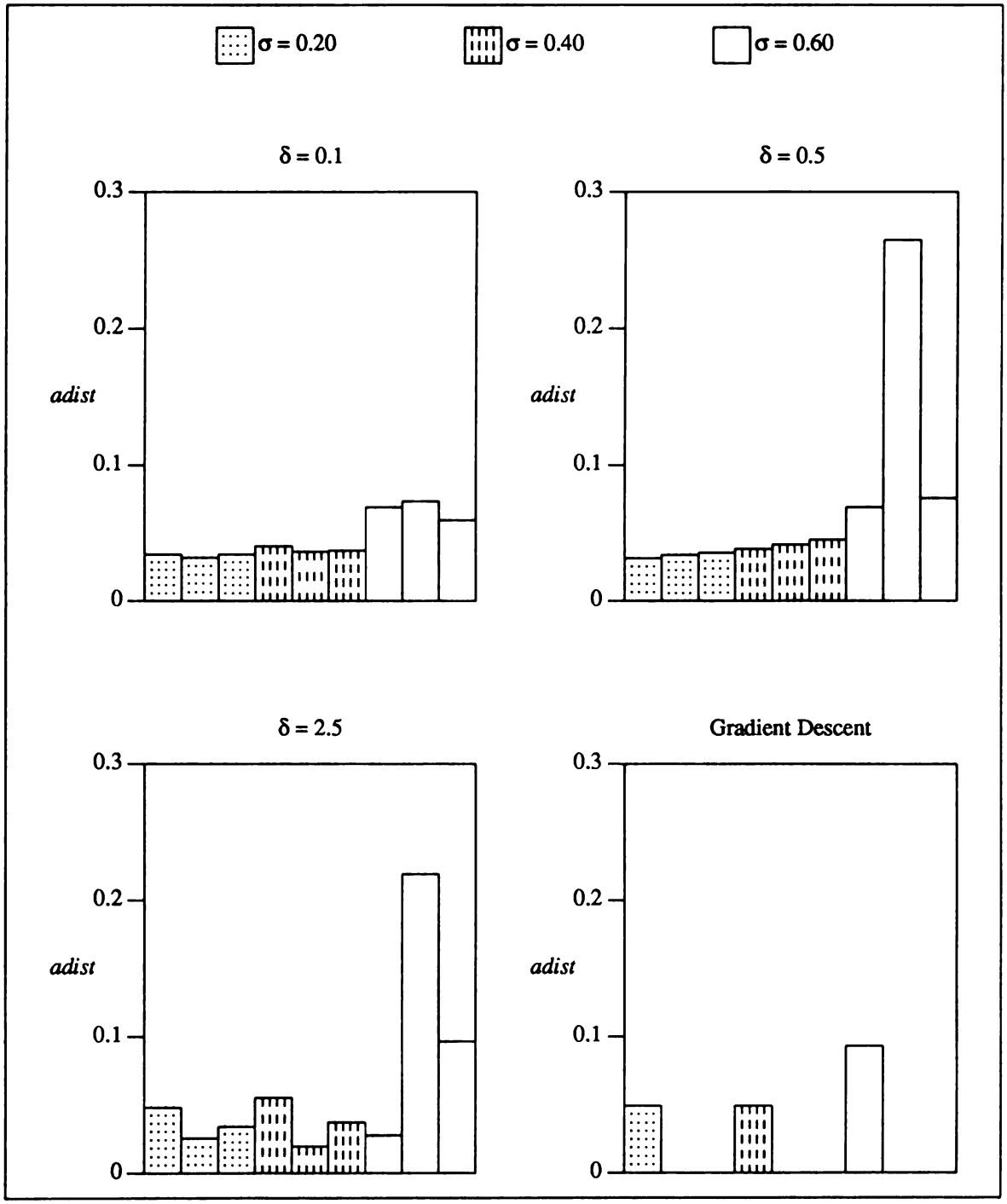

Figure 4-13 - Experiment M4. Plots showing *adist* for all runs.

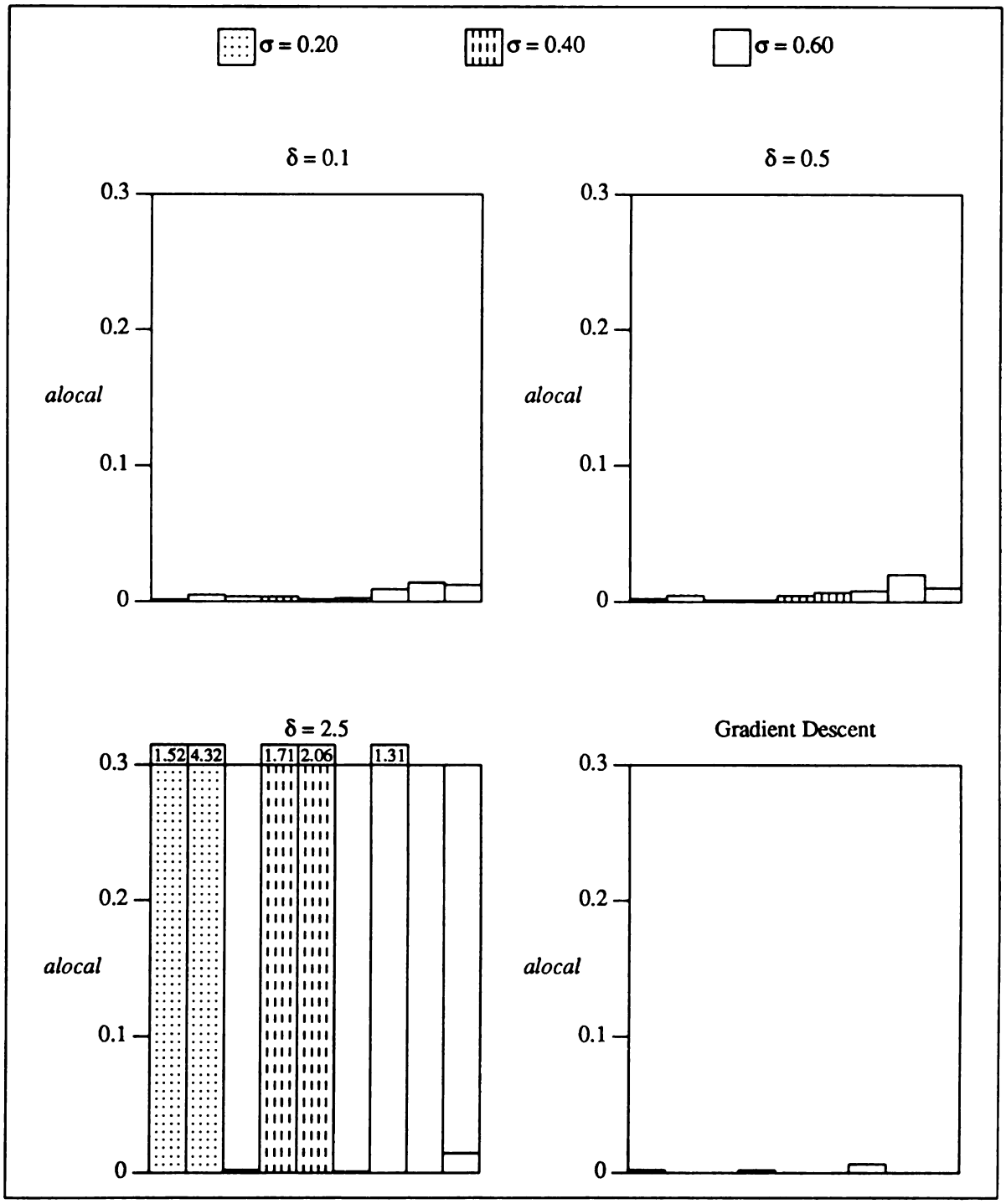

Figure 4-14 - Experiment M4. Plots showing *alocal* for all runs.

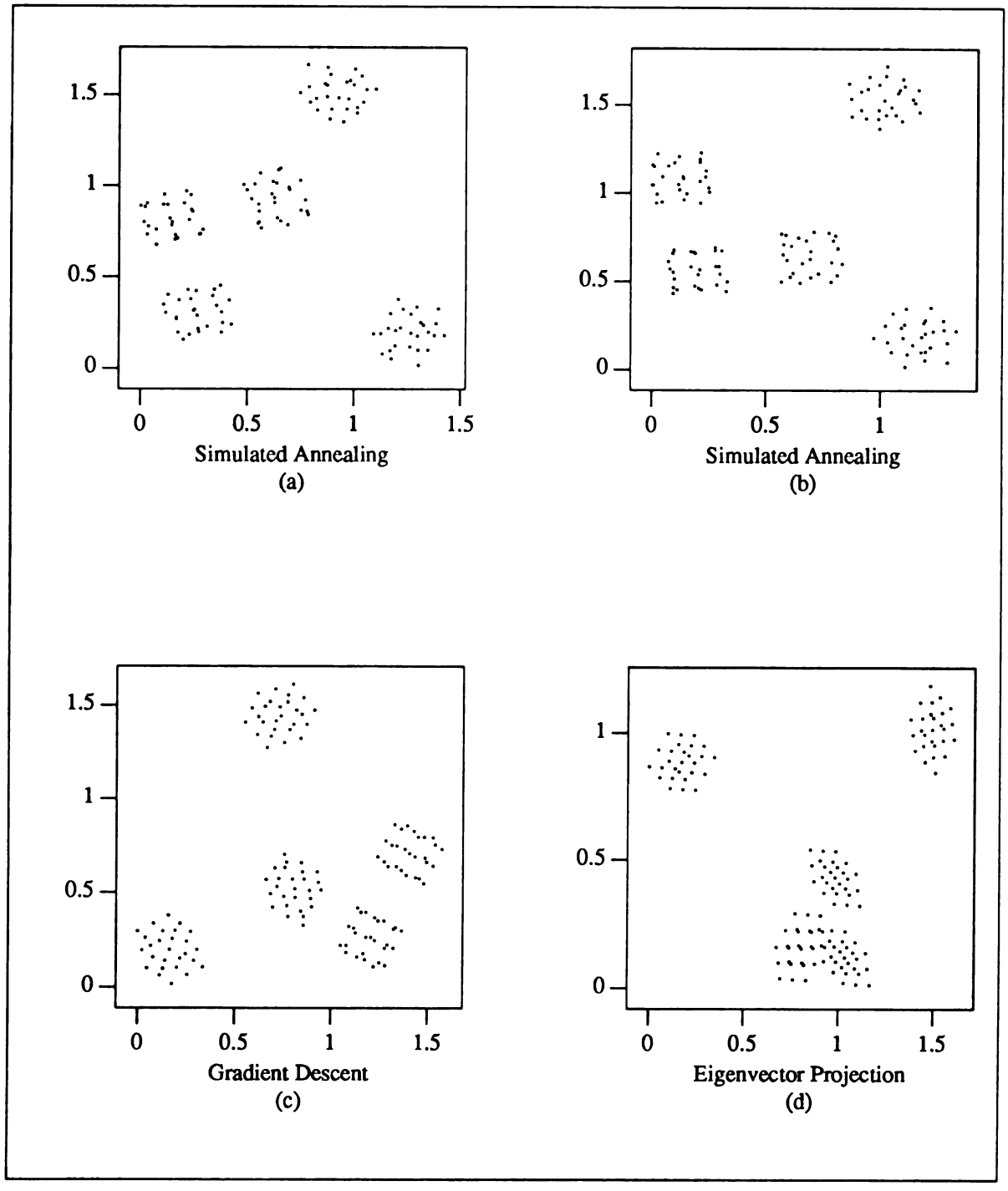

Figure 4-15 - Annealing curves for run pts-a1.

Approximate run times for the various mapping methods are summarized in Table 4-16. The results of the simulated annealing and gradient descent mappings are comparable, but have slightly different character. Both can be considered to produce reasonably good quality mappings. However, the simulated annealing takes approximately five

times longer than the gradient descent approach.

Table 4-16 Experiment M4, Run times

Projection	Approx. Run time (seconds)
$\delta = 0.1$	31000
$\delta = 0.5$	10000
$\delta = 2.5$	6400
Gradient Descent (20 runs)	6000
Eigenvector Projection	0.5

4.5. Mapping Experiment Summary

The experiments of this chapter have confirmed the effect of cooling speed on the outcome of the minimization that has been described in the literature [WHI 84]; slower cooling produces a more reliable result, particularly for tightly clustered data.

Experiments M1 and M2 showed some interesting effects, but served primarily as a guide to the larger experiments, M3 and M4. The most interesting artifact of sample size is the run time. In experiment M1 simulated annealing took about three times longer than gradient descent, but about five times more runs than suggested by Sammon [SAM 69] were used. Section 4.3 shows that the simulated annealing provides a better optimization than gradient descent, but gradient descent is still faster.

Simulated annealing produced mappings in experiments M3 and M4 with slightly better global structure than did gradient descent or eigenvector projections. The global structure is rather closely tied to the actual minimization problem taking place since the distances between patterns in different clusters comprise the largest component of the stress function. The local structure reproduced in the mappings depends on the type of local structure present in the data. For the highly structured lattice clusters, gradient descent produced mappings with better local structure than did simulated annealing. For the gaussian clusters, simulated annealing produced mappings with better local structure than

did gradient descent. This behavior may be an artifact of the stress criterion function or of the local structure statistic.

One of the biggest problems with simulated annealing for the mapping problem is the long run time. Simulated annealing takes five times as long as gradient descent for large data sets, and the mapping is not significantly better, considering the structure statistics obtained and visually inspecting the mappings produced. Eigenvector projection takes almost immeasurably less time, and the mappings produced, while not as good as those obtained by minimization, are adequate in some circumstances. An argument may be made in some applications that the results obtained for large data sets using the simulated annealing mapping algorithm are worth the extra computational cost.

Chapter 5

Clustering Experiments

Two experiments using the simulated annealing clustering algorithm are discussed in this chapter. Of particular interest is the behavior of the algorithm on clustered data of varying spread and on random data. The experimental parameters are discussed, some graphical and numerical results are shown, and then an explanation of the results is offered. Results for the clustering experiments are summarized at the end of the chapter.

5.1. Experiment C1 - Clustered Data

This experiment examines clustered data having four values of cluster spread. All data sets consist of four gaussian clusters in two dimensions with thirteen points per cluster. Factor A in the analysis of variance is the number of clusters requested with levels {2, 3, 4, 5, 6, 7, 8}. Factor B is the cluster spread with levels {0.05, 0.10, 0.15, 0.20}. The value of δ is fixed at 0.05, ε is fixed at 0.001, starting Markov chain length is fixed at 100 and minimum Markov chain length is fixed at 10.

Table 5-1 shows the values of the modified Rand statistic averaged over $K = 10$ replications per cell using simulated annealing. The additional column in Table 5-1 labelled 'Slow' show results with conservative choices of annealing parameters. The column is incomplete because of the long run times required. The 'slow' annealing is made with $\delta = 0.02$, $\varepsilon = 0.0001$, starting Markov chain length = 500, and minimum Markov chain length = 20. Table 5-2 shows the modified Rand when Forgy's algorithm is

used on the same data sets. Each cell in the table is the average of ten replications. Table 5-3 shows the *F* statistics from the analysis of variance.

Table 5-1 Experiment C1, Simulated Annealing

Number of Clusters (A)	Cluster Spread (B)				
	Fast Annealing				Slow
	0.05	0.10	0.15	0.20	0.05
2	0.372	0.386	0.338	0.338	
3	0.654	0.603	0.560	0.486	
4	0.727	0.786	0.609	0.500	0.956
5	0.777	0.722	0.575	0.451	0.879
6	0.679	0.645	0.474	0.412	
7	0.604	0.541	0.461	0.364	
8	0.553	0.500	0.407	0.348	

Table 5-2 Experiment C1, Forgy

Number of Clusters (A)	Cluster Spread (B)			
	0.05	0.10	0.15	0.20
2	0.386	0.395	0.349	0.346
3	0.699	0.636	0.571	0.452
4	0.995	0.888	0.655	0.509
5	0.898	0.812	0.575	0.457
6	0.811	0.746	0.530	0.412
7	0.742	0.676	0.496	0.372
8	0.672	0.570	0.435	0.356

Table 5-3 Experiment C1, Analysis of Variance

F-test	Annealing	Forgy	F 99%	F 99.9%
$F_{A\ (6,252)}$	30.99	65.07	2.80	3.74
$F_{B\ (3,252)}$	46.98	159.51	3.78	5.42
$F_{AB\ (18,252)}$	1.57	4.96	2.04	2.51

Table 5-3 indicates highly significant effects due to both the cluster spread and number of partitions for both clustering algorithms. The Forgy algorithm produced better results in almost every cell in Tables 5-1 and 5-2. If more conservative values of the annealing parameters δ and ε were used, as in the rightmost column of Table 5-1, the results would probably exceed those obtained for the Forgy clustering at the cost of significantly longer run times.

All the values in the Tables 5-1 and 5-2 produce significantly better clusterings than random labellings, since the modified Rand values are 15 to 40 standard deviations greater than the mean of zero under randomness [HUB 85]. Within each column, the values of the modified Rand statistic are best for partitions of four clusters, the true value, with one exception. For a fixed clustering the Rand values get better as the cluster spread decreases. These results agree with intuition. The exception is in Table 5-1, where the tightly clustered data groups into five clusters better than into four clusters. This appears to be a result of poor annealing parameters. The results obtained from the 'slow' cooling are quite significant. If we were to look at the Rand statistic for the tightly clustered data ($\sigma = 0.05$) to try to determine the true number of clusters, we would make a correct decision based on the numbers obtained from the slow cooling but an incorrect decision from the faster cooling.

Figure 5-1 shows an example of how clusterings differ with annealing schedules for the tightly clustered data. Figure 5-1 shows scatter plots of the patterns and exhibits the maximum and minimum square error values from each Markov chain. The numerals indicate the labels assigned by the algorithm. The clustering in Figure 5-1a has a modified Rand value of 0.413. It is one of the worst of the ten replications for

partitioning the tightly clustered data into four clusters. Figure 5-1b contains the plot of the same data set when clustered by more conservative cooling parameters. It corresponds to a Rand value of 1.

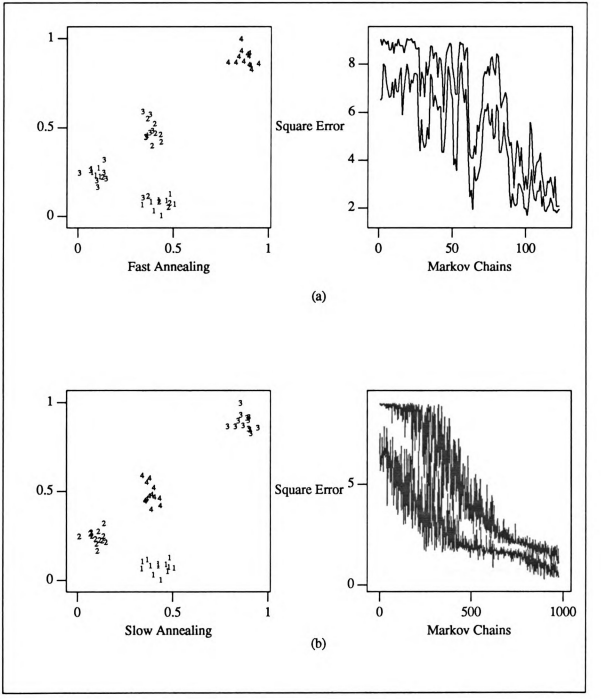

Figure 5-1 - Experiment C1, Assorted Plots.

These results again demonstrate the importance of choosing good values of the annealing parameters. Unfortunately, the problem with choosing adequate parameters values is that the run times may become so long that a run will not finish between machine downtimes. Approximate run times for the clustering algorithms used in this experiment are summarized in Table 5-4. The run times tend to increase as the cluster spread of the original data gets larger. Clearly, the simulated annealing algorithm is not practical for clustering these small data sets.

Table 5-4 Experiment C1, Run times

Problem	Approx. Run time (seconds)
2 clusters (fast)	160
3 clusters (fast)	320
4 clusters (fast)	450
5 clusters (fast)	500
6 clusters (fast)	550
7 clusters (fast)	550
8 clusters (fast)	600
4 clusters (slow)	8200
5 clusters (slow)	10500
Forgy (100 runs)	20

5.2. Experiment C2 - Random Data

This experiment examines the effects of simulated annealing parameters ε and Markov chain length with random data. Factor *A* in the analysis of variance is stopping parameter ε with levels {0.0001, 0.001, 0.01}. Factor *B* is Markov chain length with levels {50, 100, 200}. Parameter δ is fixed at 0.05. Five different random data sets are generated; each contains 50 points in two dimensions. The data are always clustered into seven partitions, which is a reasonable number for 50 points.

Table 5-5 contains the average (modified Hubert) gamma statistic over the $K = 5$ replications per cell. The final square errors are listed in Table 5-6. Slightly different results are obtained from square error than from the gamma statistic. Table 5-7 contains the F statistics from the analysis of variance.

Table 5-5 Experiment C2, Gamma Statistics

ε (A)	Markov Chain Length (B)			Forgy
	50	100	200	
0.0001	0.841	0.837	0.749	0.853
0.001	0.738	0.763	0.749	
0.01	0.526	0.634	0.527	

Table 5-6 Experiment C2, Square Error

ε (A)	Markov Chain Length (B)			Forgy
	50	100	200	
0.0001	0.934	0.909	0.912	0.905
0.001	1.399	1.073	0.912	
0.01	2.408	1.632	1.353	

Table 5-7 Experiment C2, Analysis of Variance

F-test	Gamma	Square Error	F 99.9%
$F_{A\ (2,36)}$	22.07	64.16	8.50
$F_{B\ (2,36)}$	1.64	21.99	8.50
$F_{AB\ (4,36)}$	0.66	7.27	5.90

Table 5-7 shows highly significant effects on gamma and square error due to ε. There are also effects on square error due to Markov chain length, and significant factor interaction with square error. The effects on square error are not surprising, since square

error is the objective function being minimized, but the presence of these effects confirms our knowledge of how Markov chain length and termination of the annealing affect the minimization of the criterion function.

The gamma statistics in Table 5-5 are significant since they are all 8 to 14 standard deviations above the mean of zero. The gamma statistic seems to be affected by only ε with $\delta = 0.05$. It therefore makes little sense to use the larger values of Markov chain length, since larger values require more run time without enhancing results. The reason Markov chain length has a significant effect on square error and not on gamma may be that labelling a few patterns 'incorrectly' produces larger changes in square error than in gamma. Gamma and square error cannot be compared directly because they do not have the same scale.

Figures 5-2, 5-3, and 5-4 show scatter plots and annealing curves for some of the data clustered in this experiment. Figure 5-2 shows the resultant clustering and annealing curves for one of the worst clusterings obtained with $\varepsilon = 0.01$ and Markov chain length = 50. Figure 5-3a shows a better clustering of the same data set with $\varepsilon = 0.0001$ and Markov chain length = 50, and Figure 5-3b shows the best clustering of the same data set obtained with $\varepsilon = 0.0001$ and Markov chain length = 200. Figure 5-4 shows the above mentioned plots along with the result from Forgy's algorithm for comparison. Notice that Figure 5-4a and 5-4d are the same except for cluster numbering. Visually, the clusterings obtained with the smallest ε and largest Markov chain length are the best, although some of the others are not bad. The clusterings obtained with large ε and small chain length are the worst ones of all.

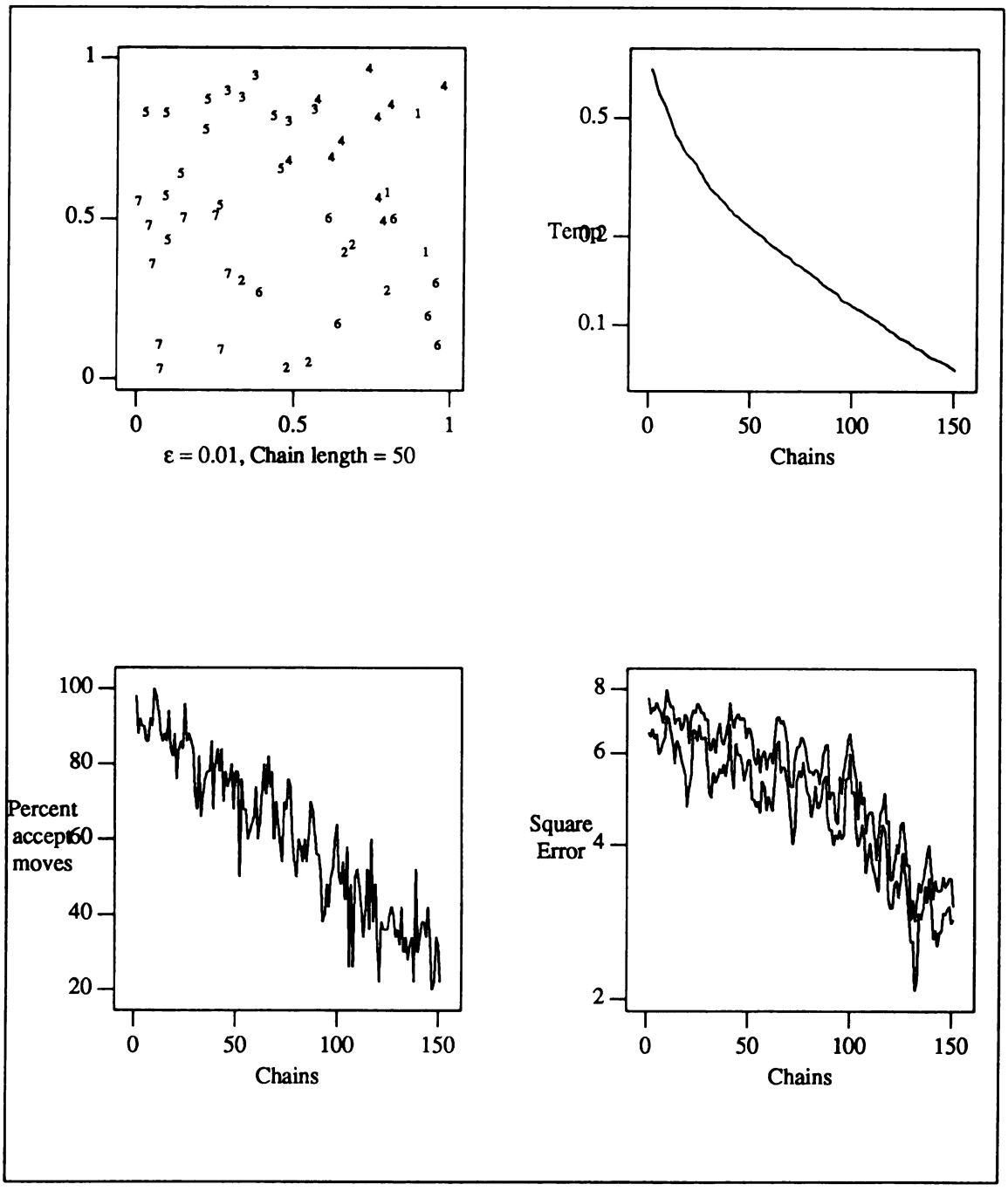

Figure 5-2 - Experiment C2, Annealing curves.

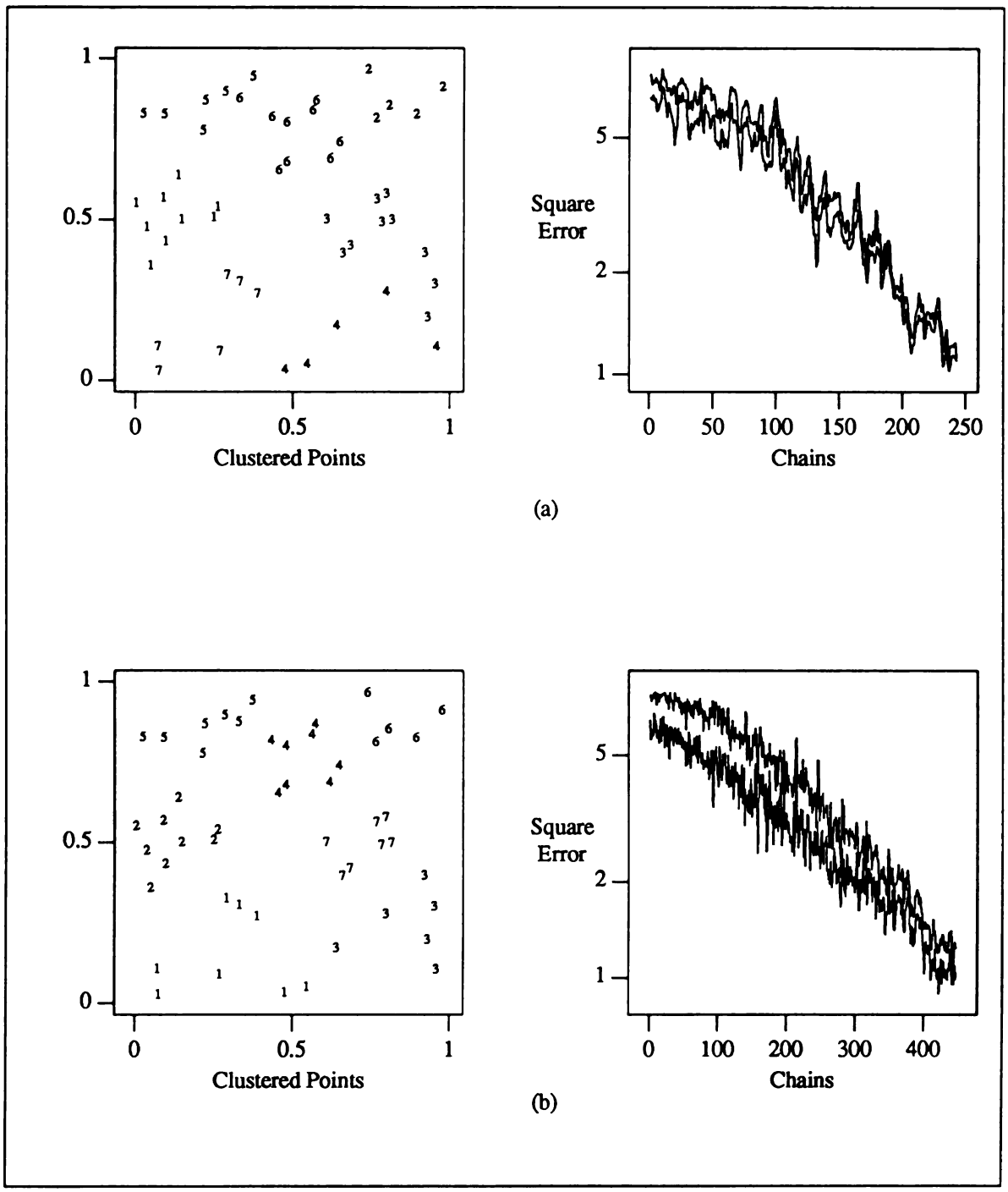

Figure 5-3 - Experiment C2, Annealing curves.

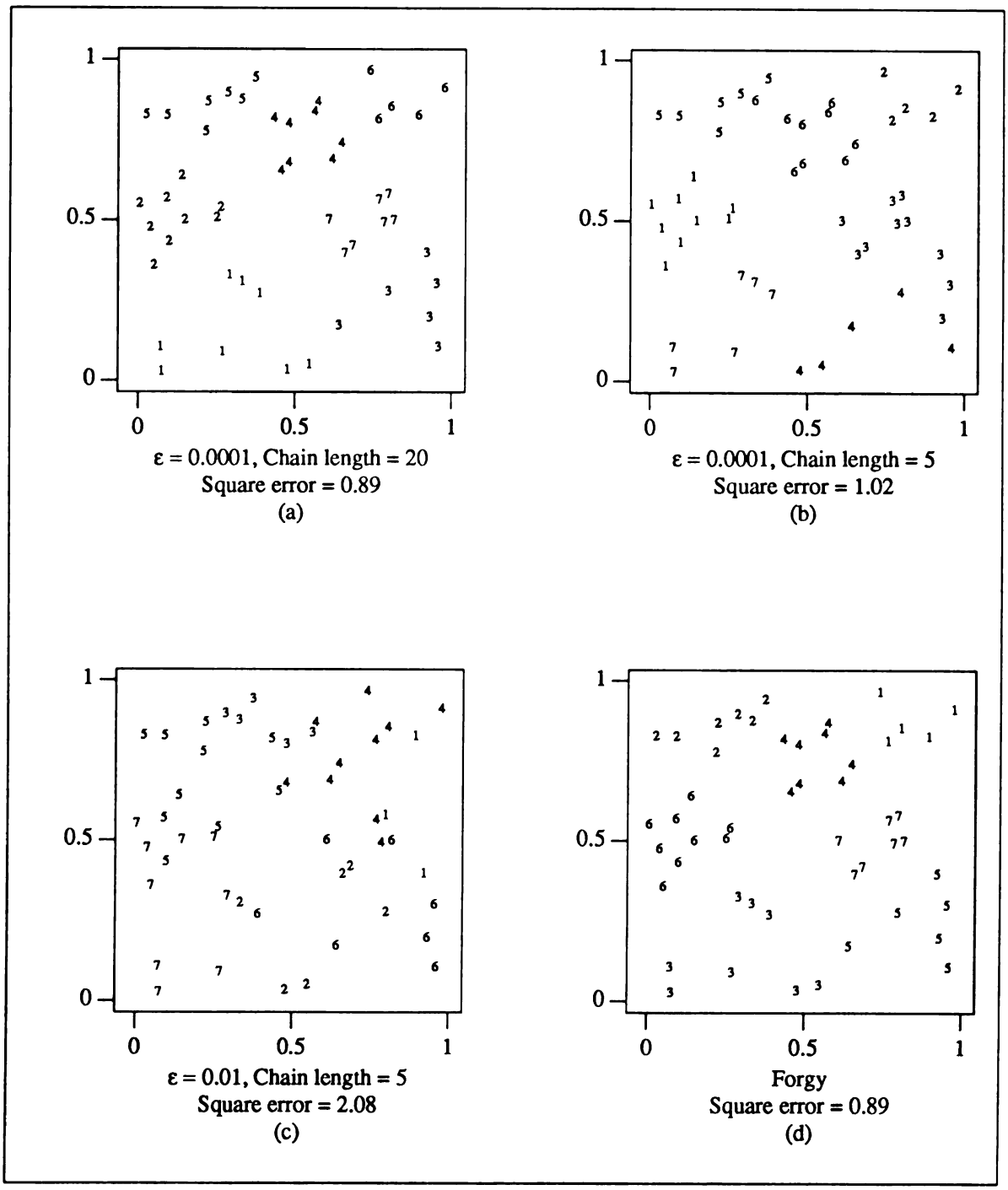

Figure 5-4 - Experiment C2, Comparison of Clusterings

The importance of choosing good annealing parameters is again demonstrated in this experiment. If ε is very small, run times may be very large. A reason for this is discussed in the next section.

The simulated annealing run times in this experiment ranged from about 150 seconds to about 5000 seconds. In general, the faster times are associated with large ε and short Markov chain length, and the longer times are associated with small ε and long Markov chain length, but occasionally a run will require an exceptionally long time to finish when ε is small and Markov chain length is large. It is not possible to determine in advance when this will happen. Because the variation in the run times is so large, they can not be easily represented in a table. The results from this experiment are not very promising. The simulated annealing clustering algorithm did not perform very well in a reasonable amount of time.

5.3. Clustering Experiment Summary

The simulated annealing clustering algorithm performed at least as well as the Forgy clustering algorithm when conservative parameter values are used. The simulated annealing, however, requires exceptionally long run time, whereas Forgy takes only about 20 seconds to get the best of 100 groupings of these small data sets.

One problem that occurs with clustering by annealing that did not occur with the mapping algorithm has to do with the termination parameter, ε. Decreasing ε will usually decrease final cost, but run time will increase. Parameter ε usually has a predictable effect on the run time with mapping, but not with clustering. Large values of ε produced predictable run times, but small values made the run time marginally longer or very much longer. This is probably due to the discrete nature of the clustering problem. The labelling gets to a point where few moves will be accepted because the points are almost partitioned into a minimum square error configuration and the temperature is low, but the stopping criterion, based on ε, has not yet been reached.

These studies show that even though simulated annealing works, it is so slow that simulated annealing can not be considered practical for this particular problem. Existing algorithms are significantly faster.

Chapter 6

Conclusions

Simulated annealing has been applied to two representative optimization problems from exploratory data analysis, nonlinear mapping and square error clustering. Empirical results were presented in Chapters 4 and 5. This chapter draws conclusions about the use of simulated annealing for these two problems and suggests future work.

6.1. Summary and Conclusions

The simulated annealing mapping algorithm has been examined in four experiments, focussing on the effects of cooling speed, sample size, and data shape. It is not surprising that slow cooling is more reliable than fast cooling, but much more computationally expensive. Both large and small data sets can be adequately mapped, but mapping small data sets takes much more time, relatively, than mapping large data sets, when compared to the gradient descent algorithm.

The effects of data shape were the most interesting. The choice of cooling speed parameter was more important for tightly clustered data than for loosely clustered data. When mapping large sets containing gaussian clusters (experiment M3), simulated annealing reproduced local structure better than gradient descent, but when mapping large sets containing lattice clusters (experiment M4), gradient descent reproduced local structure better than simulated annealing. This result may be a problem with simulated annealing when mapping highly structured data or may be an artifact of the local

structure statistic.

Using final stress as the performance criterion, simulated annealing performed better than gradient descent in experiments M3 and M4 indicating that simulated annealing is more practical for mapping large data sets than for small ones. The run time of the simulated annealing algorithm increased slower with sample size than did gradient descent when mapping large data sets.

The annealing parameters ε and Markov chain length of the clustering algorithm were examined using both clustered and random data. In general, annealing can perform as well as Forgy's algorithm, but the run times are very long. Of particular interest is the effect that ε can have on run time. If ε is chosen too conservatively, some runs never finish because few or no moves are accepted at low temperatures. When this happens in the mapping problem the algorithm makes the moves smaller as suggested by [VAN 84]. There is no analagous technique for the clustering problem.

The run times in this thesis suggest that simulated annealing holds more promise in optimizing the stress function than in optimizing the square error function. This is most likely due to the difference in the number of accepted moves between the continuous and the discrete problem. It is important to remember that this is only one of many cooling schedules and there may be one with termination criteria that are better suited to the clustering problem. As implemented the simulated annealing mapping algorithm shows promise for projecting large data sets containing gaussian clusters.

De Soete, et al. [DES 87] have studied a problem similar to the mapping problem and report that although simulated annealing works, run times are too long to make simulated annealing worthwhile. There are, however, many problems reported where simulated annealing has shown promising results [KIR 83] [CAR 85] [VAN 86]. The choice of cooling schedule and parameter values is certainly very important as is the definition of move. Efficient implementation is also important, although not a primary concern in this thesis.

The major contribution of this thesis lies in the formal examination of the performance of the simulated annealing mapping and clustering algorithms. The algorithms

were judged not by the minimizations, but by independent measures of validity. This methodology has not previously been used in simulated annealing research. In order to examine the mapping algorithm objectively, it was necessary to develop measures of local and global structure. This is a minor contribution. The new results from this formal study indicate that simulated annealing is not appropriate for small data sets. The mapping algorithm, however, performs well on large data sets containing tight gaussian clusters.

6.2. Future Work

The simulated annealing algorithm in this thesis uses one of many possible cooling schedules and the data sets contain just a few of the many possible parameter variations. Another variation on the idea of simulated annealing has been presented by Bohachevsky, et al. [BOH 86]. Their algorithm minimizes a continuous function and differs in the interpretation of the cooling schedule and should be considered for the mapping problem.

One of the problems with simulated annealing is the time spent on calculating the cost for moves that are eventually rejected, particularly with small values of the control parameter. Some work has been done to design an algorithm with fewer rejected moves [GRE 84]. This thesis handled the problem of too many rejected moves in the mapping problem by decreasing the size of the move made at lower values of the control parameter. The solution to the clustering problem contains no such time saving measures, but there are several ideas which may be worth investigating.

Most of the patterns have been assigned to correct partitions toward the end of a simulated annealing clustering run, yet the annealing continues for a long time because patterns to be moved are picked at random and very few patterns remaining need to be reassigned to other clusters. One way to make the annealing algorithm more efficient might be to move only those patterns that are in the wrong partition. It is reasonable to expect that the misclassified patterns are those nearest the border between partitions. The actual implementation used in one set of test runs consisted of limiting the possible candidate patterns when the number of rejected moves became too large. Only those

patterns whose nearest neighbor currently has a different partition label were considered. Preliminary trials using this technique were not successful for small data sets. Another possibility worth trying is to use some k-nearest neighbor choice of candidates.

Research in developing parallel implementations of the simulated annealing algorithm has been reported [AAR 86B]. The main problem here is that simulated annealing is inherently sequential, although the objective functions used in this thesis lend themselves to parallel operations. A near linear speedup using parallel processors could not be expected, but these new algorithms may allow large problems to terminate that previously could not do so.

Certainly, simulated annealing is not practical for all optimizations, but from the limited number of experiments made here it is fairly safe to say that simulated annealing is worth trying on problems for which there are currently no other good minimization techniques. Cooling schedules other than the one used here also bear investigation and may yield better results. A nice feature of simulated annealing is that the choice of criterion function to optimize is unlimited. The definition of move may cause some difficulty. Production algorithms may require 'clever' implementations and should use efficient data structures.

Appendix A

Structure Statistics

This appendix defines the statistics used to measure the structure present in clustered data and to judge the performance of projection algorithms.

A.1. Local Structure Statistic

The statistic $alocal_d$ defined below was designed to measure the local structure in a clustered data set in d dimensions. The statistic *alocal* is defined to be $|alocal_L - alocal_l|$ and determines how well a mapping from L to l dimensions preserves the local structure of clustered data. Values close to zero imply that local structure is preserved well.

$$alocal_d = \frac{1}{nclu\ (nclu-1)}\left[\sum_{i=1}^{nclu}\sum_{j\neq i}\frac{S_i^{(d)}}{S_j^{(d)}}\right]$$

Here, $S_i^{(d)}$ is the within cluster spread of cluster i, d is the dimensionality of the data, and *nclu* is the number of clusters.

$$S_i^{(d)} = \frac{1}{d}\sum_{k=1}^{d}\left[\frac{1}{n_i}\sum_{l=1}^{n_i}x_{lki}^2 - \left[\frac{1}{n_i}\sum_{l=1}^{n_i}x_{lki}\right]^2\right]$$

where x_{lki} is the kth feature of the lth pattern of the ith cluster, and n_i is the number of patterns in cluster i. Note that the clusters are specified by the generating process so the notation is different from that in Section 1.2.

The statistic is composed of the sum of all possible combinations of ratios of within cluster spread. This number distinguishes between data sets whose clusters have the same spread and those whose clusters vary in their spread. The normalization constant of $alocal_d$ is the reciprocal of the number of terms in the summation. This makes it possible to interpret $alocal_d$ independent of the number of clusters in the data. The closer the value is to one, the more similar are the clusters of the data set.

The statistic $alocal_d$ is now shown to be invariant to scale changes. Scale all patterns by factor K to obtain

$$\frac{1}{nclu\ (nclu-1)} \sum_{i=1}^{nclu} \sum_{j \neq i} \frac{\sum_{k=1}^{d} \left[\frac{1}{n_i} \sum_{l=1}^{n_i} (Kx_{lki})^2 - \left[\frac{1}{n_i} \sum_{l=1}^{n_i} Kx_{lki} \right]^2 \right]}{\sum_{k=1}^{d} \left[\frac{1}{n_j} \sum_{l=1}^{n_j} (Kx_{lkj})^2 - \left[\frac{1}{n_j} \sum_{l=1}^{n_j} Kx_{lkj} \right]^2 \right]} = alocal_d$$

Clearly a K^2 term can be factored from both numerator and denominator.

The statistic $alocal_d$ is also invariant to translation since $S_i^{(d)}$ is invariant to translation. Suppose all patterns are translated by c_k.

$$\frac{1}{d} \sum_{k=1}^{d} \left[\frac{1}{n} \sum_{l=1}^{n} \left(x_{lk}+c_k \right)^2 - \left[\frac{1}{n} \sum_{l=1}^{n} \left(x_{lk}+c_k \right) \right]^2 \right]$$

$$= \frac{1}{d} \sum_{k=1}^{d} \left[\frac{1}{n} \sum_{l=1}^{n} \left(x_{lk}^2+2c_k x_{lk}+c_k^2 \right) - \left[\frac{1}{n} \sum_{l=1}^{n} \left(x_{lk}+c_k \right) \right] \left[\frac{1}{n} \sum_{l=1}^{n} \left(x_{lk}+c_k \right) \right] \right]$$

$$= \frac{1}{d} \sum_{k=1}^{d} \left[\frac{1}{n} \sum_{l=1}^{n} x_{lk}^2+\frac{2}{n} \sum_{l=1}^{n} c_k x_{lk}+\frac{1}{n} \sum_{l=1}^{n} c_k^2 - \left[\frac{1}{n} \sum_{l=1}^{n} x_{lk}+\frac{1}{n} \sum_{l=1}^{n} c_k \right] \left[\frac{1}{n} \sum_{l=1}^{n} x_{lk}+\frac{1}{n} \sum_{l=1}^{n} c_k \right] \right]$$

$$= \frac{1}{d} \sum_{k=1}^{d} \left[\frac{1}{n} \sum_{l=1}^{n} x_{lk}^2+\frac{2}{n} \sum_{l=1}^{n} c_k x_{lk}+\frac{1}{n} \sum_{l=1}^{n} c_k^2 - \left[\frac{1}{n} \sum_{l=1}^{n} x_{lk} \right]^2 - \frac{2}{n} c_k \sum_{l=1}^{n} x_{lk} - c_k^2 \right]$$

$$= \frac{1}{d} \sum_{k=1}^{d} \left[\frac{1}{n} \sum_{l=1}^{n} x_{lk}^2 - \left[\frac{1}{n} \sum_{l=1}^{n} x_{lk} \right]^2 \right] = S^{(d)}$$

It can also be argued that $alocal_d$ does not vary greatly with dimensionality since the summation over dimension is present in both numerator and denominator.

A.2. Global Structure Statistic

The statistic *adist* was designed to measure the global structure in a clustered data set of gaussian clusters in d dimensions. The phrase "global structure" refers to the relative placement of the clusters. The statistic *adist* is defined to be $|adist_L - adist_l|$ and measures the degree to which global structure is preserved in a mapping from L to l dimensions. As with *alocal*, values close to zero imply that global structure is preserved well.

The statistic $adist_d$ is composed of the sum of all possible combinations of ratios of distances between pairs of clusters. Values close to one imply that the clusters are equally spaced.

The statistic $adist_d$ is defined as follows:

$$adist_d = \frac{4}{(nclu+1)\,nclu\,(nclu-1)\,(nclu-2)} \left[\sum_{i=1}^{nclu} \sum_{j>i} \sum_{k \neq j} \sum_{\substack{l>k \\ or \\ l \neq i}} \frac{D_{ij}}{D_{kl}} \right]$$

where D_{ij} is the distance between cluster centers z_i and z_j and is given by the following formula:

$$D_{ij} = \sum_{q=1}^{d} \left[\frac{1}{n_i} \sum_{p=1}^{n_i} x_{pqi} - \frac{1}{n_j} \sum_{p=1}^{n_j} x_{pqj} \right]^2$$

The statistic $adist_d$ can be shown to be invariant to scale and translation, and argued not to vary greatly with dimensionality as was $alocal_d$.

Appendix B

Cluster Validity Measures

This appendix contains the formulation of the cluster validity statistics that are used to judge the performance of clustering algorithms.

B.1. External Measure - The modified Rand statistic

For the experiment involving clustered data, the true class labels are known so an external measure is needed. The modified Rand coefficient [HUB 85] [DUB 86] is used.

Let $\{L(i)\}$ be the set of n cluster labels assigned to the n patterns by the clustering algorithm and $\{T(i)\}$ be the apriori cluster labels. The Rand coefficient is defined to be

$$\frac{a}{a+b}$$

where

$$a = |\{(i,j): i>j,\ [L(i)=L(j),\ T(i)=T(j)] \vee [L(i)\neq L(j),\ T(i)\neq T(j)]\}|$$

$$b = |\{(i,j): i>j,\ [L(i)=L(j),\ T(i)\neq T(j)] \vee [L(i)\neq L(j),\ T(i)=T(j)]\}|$$

and $a + b = \binom{n}{2}$. The statistic a is the number of pairs of patterns which are treated the same by both labellings. That is, the number of pairs which either have the same label in both labellings or have different labels in both partitions; b is the number of pairs of patterns which have the same label in one partition and different labels in the other partition.

The modified Rand statistic [HUB 85] contains a correction factor that accounts for random labellings of the data. The modified Rand is defined

$$\frac{Rand - Expected\ Rand}{Maximum\ Rand - Expected\ Rand}$$

where Maximum Rand is taken to be one, and the expected Rand is computed from

$$Expected\ Rand = 1 + \frac{2 \cdot \sum_i \binom{n_i}{2} \sum_i \binom{m_i}{2}}{\binom{n}{2}^2} - \frac{\sum_i \left[\binom{n_i}{2} + \binom{m_i}{2} \right]}{\binom{n}{2}}$$

where n_i is the number of patterns in group i of the clustering, and m_i is the number of patterns in group i of the true labelling. This statistic has a value of 1 when $b = 0$ and should be around 0 when the cluster labels are assigned randomly. Some special cases of the statistic for the 52 point data sets used in experiment C1 are 0.948 when any one pattern is classified in the wrong cluster, and 0.975 when one pattern is placed in a singleton cluster.

B.2. Internal Measure - Modified Hubert's Gamma

For the experiment involving random data, the true class labels are not known so an internal measure of cluster validity is needed. The modified Hubert gamma statistic [DUB 86] is based on the Mantel statistic [MAN 67] and Hubert's gamma statistic [HUB 76]. The statistic is the point serial correlation coefficient between proximity matrix for the n patterns, $\{x_i,\ 1 \le i \le n\}$ and a "model" matrix. The cluster centers from the clustering are considered to be the "true" locations of the patterns, so the model matrix is the proximity matrix in which proximity between patterns is indicated by distance between centers of clusters containing these patterns. Let L denote the label function that maps the set of patterns to the set of cluster labels.

$$L_i = k \text{ if } y_{ik} = 1$$

Let $\{z_i,\ 1 \le i \le g\}$ be the cluster centers and M be a shorthand notation for $\frac{n(n-1)}{2}$. Denote the euclidean distance between vectors a and b by

$$\delta(a,b) = \left[(a-b)^T (a-b) \right]^{\frac{1}{2}}$$

Note that this function δ is not related to the simulated annealing cooling parameter. The modified Hubert statistic, *MH*, will be a function of the following components.

$$r = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta(x_i, x_j)\, \delta(z_{L_i}, z_{L_j})$$

$$M_p = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta(x_i, x_j)$$

$$M_c = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta(z_{L_i}, z_{L_j})$$

$$\sigma_p^2 = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta^2(x_i, x_j) - M_p^2$$

$$\sigma_c^2 = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta^2(z_{L_i}, z_{L_j}) - M_c^2$$

The statistic *MH* is defined:

$$MH = \frac{r - M_p M_c}{\sigma_p \sigma_c}$$

Since this statistic is a correlation, values close to 0 indicate a poor clustering and values close to 1 indicate a good clustering.

Bibliography

[AAR 85] Aarts, E.H.L. and P.J.M. van Laarhoven; "Statistical Cooling: A General Approach to Combinatorial Optimization Problems", *Philips Journal of Research*, V40; 1985; pp. 193-226.

[AAR 86A] Aarts, E.H.L. and P.J.M. van Laarhoven; "Simulated Annealing: A Pedestrian Review of the Theory and Some Applications", *NATO Advanced Study Institute on Pattern Recognition: Theory and Applications*; Spa, Belgium; June 1986.

[AAR 86B] Aarts, E.H.L., F.M.J. de Bont, E.H.A. Habers, and P.J.M. van Laarhoven; "Parallel Implementations of the Statistical Cooling Algorithm", *Integration, the VLSI Journal*, 1986, pp. 209-238.

[AFI 72] Afifi, A.A. and S.P. Azen; *Statistical Analysis*; Academic Press, 1972.

[AND 73] Anderberg, Michael R.; *Cluster Analysis for Applications*, Academic Press, New York, 1973.

[BEZ 81] Bezdek, James C.; *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenum Press, 1981, pp. 65-86.

[BIS 81] Biswas, G., A.K. Jain, and R.C. Dubes; "Evaluation of Projection Algorithms", *IEEE Transactions on PAMI*, Vol. PAMI-3, No. 6, November 1981; pp. 701-708.

[BOH 86] Bohachevsky, Ihor O., Mark E. Johnson, and Myron L. Stein; "Generalized Simulated Annealing for Function Optimization", *Technometrics*, Vol. 28, No. 3, August 1986, pp. 209-217.

[BUR 84] Burkard, R.E. and F. Rendl; "A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems", *European Journal of Operational Research*, V17; 1984; pp. 169-174.

[CER 85] Cerny, V.; "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm", *Journal of Optimization Theory and Applications*, V45, No. 1; January 1985; pp. 41-51.

[CAR 85] Carnevalli, P., L. Coletti, and S. Patarnello; "Image Processing by Simulated Annealing", *IBM Journal of Research Development*, V29, No. 6; November 1985, pp. 569-579.

[CHA 73] Chang, C.L. and R.C.T. Lee; "A Heuristic Relaxation Method for Nonlinear Mapping in Cluster Analysis", *IEEE Transactions on Systems, Man, and Cybernetics*, March 1973, pp. 197-200.

[DES 87] De Soete, Geert, Lawrence Hubert, and Phipps Arabie; "The Comparative Performance of Simulated Annealing on Two Problems of Combinatorial Data Analysis", April 1987, *preprint.*

[DUB 76] Dubes, R.C. and A.K. Jain; "Clustering Techniques: The User's Dilemma", *Pattern Recognition*, Vol. 8, 1976, pp. 247-260.

[DUB 86] Dubes, R.C.; "Experiments in Estimating the Number of Clusters", *Technical Report*, MSU-ENGR-86-019, Department of Computer Science, Michigan State University, 1986.

[FRI 74] Friedman, J.H. and J.W. Tukey; "A Projection Pursuit Algorithm for Exploratory Data Analysis", *IEEE Transactions on Computers*, V C-23, No. 9; September 1974, pp. 881-889.

[GEM 84] Geman, S. and D. Geman; "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images", *IEEE Transactions on PAMI*, V PAMI-6, 1984, pp. 721-741.

[GID 85] Gidas, B.; "Nonstationary Markov Chains and Convergence of the Annealing Algorithm", *Journal of Statistical Physics*, V 39, Nos. 1/2; 1985; pp. 73-131.

[GOR 77] Gordon, A.D., and J.T. Henderson; "An Algorithm for Euclidean Sum of Squares Classification", *Biometrics*, V 33, June 1977, pp. 355-362.

[GOR 81] Gordon, A.D.; *Classification*; Chapman and Hall Ltd., 1981.

[GRE 84] Greene, J.W. and K.J. Supowit; "Simulated Annealing without Rejected Moves", *Proceedings of the IEEE International Conference on Computer Design*, Port Chester, November 1984, pp. 658-663.

[HAM 64] Hammersley and Handscomb; *Monte Carlo Methods*; John Wiley and Sons; 1964; Chapter 9, pp. 113-126.

[HUB 76] Hubert, Lawrence and James Schultz; "Quadratic Analysis as a General Data Analysis Strategy", *Br. J. math. statist. Psychol.*, V 29, 1976, pp. 190-241.

[HUB 85] Hubert, L., and P. Arabie; "Comparing Partitions", *Journal of Classification*, V 2, 1985; pp. 193-218.

[KIR 83] Kirkpatrick, S., C.D. Gelatt Jr., and M.P. Vecchi; "Optimization by Simulated Annealing", *Science*, V 220, No. 4598; May 13, 1983; pp. 671-680.

[KIR 84] Kirkpatrick, S.; "Optimization by Simulated Annealing: Quantitative Studies", *Journal of Statistical Physics*, Vol. 34, Nos. 5/6; 1984; pp. 975-986.

[KIR 85] Kirkpatrick, S. and R.H. Swendson; "Statistical Mechanics and Disordered Systems", *Communications of the ACM*, V 28, No. 4; April 1985; pp. 363-373.

[KRU 71] Kruskal, Joseph B.; "Comments on 'A Nonlinear Mapping for Data Structure Analysis'", *IEEE Transactions on Computers*, V C-20, December 1971, p. 1614.

[LUN 86] Lundy, M. and A. Mees; "Convergence of an Annealing Algorithm", *Mathematical Programming*, V 34; 1986; pp. 111-124.

[MAF 86] Maffioli, F.; "Randomized Algorithms in Combinatorial Optimization: A Survey", *Discrete Applied Mathematics*, V 14; 1986, pp. 157-170.

[MAN 67] Mantel, Nathan; "The Detection of Disease Clustering and a Generalized Regression Approach", *Cancer Research*, V 27 Part I, February 1967, pp. 209-220.

[MET 53] Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller; "Equations of State Calculations by Fast Computing Machines", *The Journal of Chemical Physics*, V 21, No. 6; June 1953; pp. 1087-1092.

[MIT 86] Mitra, Debasis, Fabio Romeo, and Alberto Sangiovanni-Vincentelli; "Convergence and Finite-Time Behavior of Simulated Annealing", *Advances in Applied Probability*, V 18, 1986, pp. 747-771.

[OTT 84] Otten, R.H.J.M. and L.P.P.P. van Ginneken; "Floorplan Design Using Simulated Annealing", *IEEE International Conference on Computer Aided Design*, (ICCAD-84); November 1984; Santa Clara; pp. 96-98.

[RAN 86] Randelman, R.E. and G.S. Grest; "*N*-City Traveling Salesman Problem: Optimization by Simulated Annealings", *Journal of Statistical Physics*, Vol. 45, Nos. 5/6, 1986, pp. 885-890.

[REE 87] Rees, Stephen, and Robin C Ball; "Criteria for an optimum simulated annealing schedule for problems of the travelling salesman type", *Journal of Physics A*, V 20, April 1987, pp. 1239-1249.

[ROM 84] Romeo, F., A. Sangiovanni-Vincentelli, and C. Sechen; "Research on Simulated Annealing at Berkeley", *Proceedings of the IEEE International Conference on Computer Design*, Port Chester, November 1984, pp. 652-657.

[ROM 85] Romeo, F., and A. Sangiovanni-Vincentelli; "Probabilistic Hill Climbing Algorithms: Properties and Applications", *Proceedings of the 1985 Chapel Hill Conference on VLSI*; pp. 393-417.

[ROS 86] Rossier, Y., M. Troyon, and Th. M. Liebling; "Probabilistic Exchange Algorithms and Euclidean Traveling Salesman Problems", *OR Spektrum*, V 8, 1986, pp. 151-164.

[SAM 69] Sammon Jr., J.W.; "A Nonlinear Mapping for Data Structure Analysis", *IEEE Transactions on Computers*, V C-18, No. 5; May 1969; pp. 401-409.

[SMI 83] Smith, W.E., H.H. Barrett, and R.G. Paxman; "Reconstruction of Objects From Coded Images by Simulated Annealing", *Optics Letters*, V 8, No. 4; April 1983; pp. 199-201.

[VAN 86] van Laarhoven, P.J.M. and E.H.L. Aarts; "Simulated Annealing: A Review of the Theory and Applications", c/o Philips Reasearch Labs.

[VAN 84] Vanderbilt, D., and S.G. Louie; "A Monte Carlo Approach to Optimization over Continuous Variables", *Journal of Computational Physics*, V 56, 1984, pp. 259-271.

[WHI 84] White, S.R.; "Concepts of Scale in Simulated Annealing", *Proceedings of IEEE International Conference on Computer Design*, Port Chester, November 1984, pp. 646-651.