Finding the Optimal Gene Order in Displaying Microarray Data

Seung-Kyu Lee[1], Yong-Hyuk Kim[2], and Byung-Ro Moon[2]

[1] NHN Corp., 7th floor, Startower
737 Yoksam-dong, Kangnam-gu, Seoul, Korea
spin30@soar.snu.ac.kr
[2] School of Computer Science & Engineering, Seoul National University
Shilim-dong, Kwanak-gu, Seoul, 151-742 Korea
{yhdfly, moon}@soar.snu.ac.kr

Abstract. The rapid advances of genome-scale sequencing have brought out the necessity of developing new data processing techniques for enormous genomic data. Microarrays, for example, can generate such a large number of gene expression data that we usually analyze them with some clustering algorithms. However, the clustering algorithms have been ineffective for visualization in that they are not concerned about the order of genes in each cluster. In this paper, a hybrid genetic algorithm for finding the optimal order of microarray data, or gene expression profiles, is proposed. We formulate our problem as a new type of traveling salesman problem and apply a hybrid genetic algorithm to the problem. To use the 2D natural crossover, we apply the Sammon's mapping to the microarray data. Experimental results showed that our algorithm found improved gene orders for visualizing the gene expression profiles.

1 Introduction

The recent marvelous advances of genome-scale sequencing have provided us with a huge amount of genomic data. Microarrays [36,37], for example, have been used for revealing gene expression profiles for more than thousands of genes. In general, microarray data can be represented by a real-valued matrix; each row represents a gene and each column represents a condition, or experiment. If we let the matrix be X, the element X_{ij} represents the expression level of gene i for a given condition j. The microarray data are usually preprocessed with a clustering algorithm. The clustered microarray data, then, can be analyzed by biologists.

A number of algorithms for clustering gene expression profiles were proposed. Eisen *et al.* [10] applied hierarchical clustering [38] which has been a widely used tool [1,22,24,35]. It also has some variants [2,17]. Self-organizing maps (SOMs) [42,44] and k-means clustering [43] were also used for the same purpose. Ben-Dor *et al.* [3] developed an algorithm, cluster affinity search technique (CAST), which has a good theoretical basis. Merz and Zell [28] proposed a memetic algorithm for the problem formulated as finding the minimum sum-of-squares clustering

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 2215–2226, 2003.

[48,9]. However, all the proposed algorithms were ineffective in visualizing the microarray data since they were not concerned about aligning genes within each cluster in a meaningful way. This raises the problem of finding the optimal order of genes for visualization.

Although there is no standard optimal criterion for evaluating which order is better than the others for visualization, placing genes with similar or the same expression profiles next to each other is considered to be natural and intuitive. Since finding the optimal order of microarray data is known to be NP-hard [5], evolutionary approaches such as genetic algorithms [18,13], memetic algorithms [29] are considered to be well suited for solving the problem.

Recently Tsai *et al.* [46] formulated the problem as the traveling salesman problem (TSP) and applied family competition genetic algorithm (FCGA). In the FCGA, the edge assembly crossover [30] was combined with the family competition concept [45] and neighbor-join mutation [47]. Using this consolidation, they showed that their formulation was effective in finding attractive gene orders for visualizing microarray data. However, they implicitly tried to minimize the distance between distant genes as well, which is less important for visualization.

In this paper, we propose a hybrid genetic algorithm for finding the optimal gene order of microarray data. We suggest a new variation of TSP formulation for this purpose. We use the 2D natural crossover [20,21], which is one of the state-of-the-art crossovers in the TSP literature. To use the 2D natural crossover, we need a 2D mapping of the data, which are virtually real-valued vectors, from a high-dimensional space into a two-dimensional Euclidean space. This mapping is necessary since the crossover exploits two-dimensional geographical information. We choose the Sammon's mapping [34] among several candidates.

Another important contribution of this paper is that we used a new formulation of TSP for the problem. In this model of TSP, relatively long edges in a tour are ignored for fitness evaluation. This is because reducing the length of long edges, which represents distant genes in relation, is not very meaningful for visualizing microarray data. We tested this idea on a spectrum of different rates of excluded edges.

The remainder of the paper is organized as follows. In Section 2, we summarize the traveling salesman problem and the Sammon's mapping. In Section 3, we describe our variation of TSP formulation for finding the optimal gene order in displaying microarray data. In Section 4, we explain our hybrid genetic algorithm in detail and present the experimental results in Section 5. Finally, we make our conclusions in Section 6.

2 Preliminaries

2.1 Traveling Salesman Problem

Given n cities and a distance matrix $D = [d_{ij}]$ where d_{ij} is the distance between city i and city j, the traveling salesman problem (TSP) is the problem of finding a permutation π that minimizes $\sum_{i=1}^{n-1} d_{\pi_i,\pi_{i+1}} + d_{\pi_n,\pi_1}$. In metric TSP the cities lie in a metric space (i.e., the distances satisfy the triangle inequality). In Euclidean

TSP, the cities lie in $\Re^d$ for some d; the most popular version is 2D Euclidean TSP where the cities lie in $\Re^2$. Euclidean TSP is a sub-case of metric TSP.

2.2 Sammon's Mapping

Sammon's mapping [34] is a mapping technique for transforming a dataset from a high-dimensional (say, m-dimensional) input space onto a low-dimensional (say, d-dimensional) output space (with $d < m$). The basic idea is to arrange all the data points on a d-dimensional output space in such a way that minimizes the distortion of the relationships among data points.

Sammon's mapping tries to preserve distances. This is achieved by minimizing an error criterion which penalizes the differences of distances between the points in the input space and the output space. Consider a dataset of n objects. If we denote the distance between two points x_i and x_j in the input space by δ_{ij} and the distances between x'_i and x'_j in the output space by δ'_{ij}, then Sammon's stress measure E is defined as follows:

$$E = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{(\delta_{ij} - \delta'_{ij})^2}{\delta_{ij}}.$$

The stress range is [0,1] with 0 indicating a lossless mapping. This stress measure can be minimized using any minimization technique. Sammon [34] proposed a technique called pseudo-Newton minimization, a steepest-descent method. The complexity of Sammon's mapping is $O(n^2 m)$. There were many studies about Sammon's mapping [8,33,31].

3 A New TSP Formulation

To visualize the microarray data, or gene expression profiles, in a meaningful way, it is natural and intuitive to align genes with similar expression profiles, or within the same group, close together. For genes with similar expression profiles to be aligned next to each other, it is useful to formulate the problem as the TSP.

For the TSP formulation, a distance measure is needed to quantify the similarity between gene expression profiles, which then defines the similarity between the genes themselves. Several distance measures were proposed to define the distance. They include Pearson correlation [1], absolute correlation [2], Spearman rank correlation [39], Kendall rank correlation [23], and Euclidean distance. In this paper, we choose the Pearson correlation as the distance measure.

Let $X = x_1, x_2, \ldots, x_k$ and $Y = y_1, y_2, \ldots, y_k$ be the expression levels of two genes X and Y, which were observed over a series of k conditions. The Pearson

[1] Karl Pearson (1857-1936) is considered to be the first to call the quantity a correlation efficient in 1896 [7]. It first appeared as a published form by Harris [16]. It is also referred to as Pearson product-moment correlation coefficient.

[2] absolute value of correlation

(a) with the naive TSP formulation (b) with our new TSP formulation

Fig. 1. A comparison of good tours between the naive and our new TSP formulation.

correlation of the two genes X and Y is

$$s_{X,Y} = \frac{1}{k}\sum_{i=1}^{k}(\frac{x_i - \overline{X}}{\sigma_X})(\frac{y_i - \overline{Y}}{\sigma_Y})$$

where $\overline{X}$ and σ_X are the mean and the standard deviation of the expression levels, respectively. Then we define the distance between the genes X and Y by

$$D(X,Y) = 1 - s_{X,Y}$$

where $s_{X,Y}$ is the Pearson correlation.

Once the distance measure is defined, it is possible to formulate finding the optimal gene order for visualization with the TSP model. Each gene corresponds to a city in TSP, and the distance between two genes corresponds to the length of the edge between the two cities. Tsai *et al.* [46] reduced the problem of finding the optimal order of genes to the problem of finding the shortest tour of the corresponding TSP.

In the model, the fitness function is naturally defined to be

$$\sum_{i=1}^{n} D(g_{\pi_i}, g_{\pi_{i+1}})$$

where $g_{\pi_{n+1}} = g_{\pi_1}$, g_i denotes a gene, π denotes a gene order, n is the number of genes, and $D(g_i, g_j)$ is the distance between two genes g_i and g_j.

The above TSP formulation [46] aims at aligning genes with similar profiles close together. However, it also tries to minimize the distances between pairs of genes with not-very-similar profiles. When two genes are adjacent in a TSP tour, they are placed next to each other. If they have distant profiles, replacing a gene by a third one with a less distant profile has little meaning, as long as they are not considerably similar. Tsai *et al.*'s TSP formulation implicitly tries to reduce this type of edges as well.

To alleviate the problem, we propose a new fitness function. Our key idea is that we can improve the visualization results by excluding less meaningful edges in a tour from the fitness function. Figure 1 illustrates the motivation of our new TSP formulation. The length of the tour in Fig. 1(a) is shorter than that of the tour in Fig. 1(b), thus it is preferred in the naive TSP formulation. However, the

tour in Fig. 1(b) can be a better tour in that it reflects the natural grouping, denoted by ellipses. Since an edge between two genes means that the two genes are placed next to each other in visualization, we can make the naturally-grouped genes placed next to each other if a meaningful tour like Fig. 1(b), possibly including long edges, is preferred. For such tours to be preferred, relatively long edges are excluded from the fitness function. The dotted edges in Fig. 1(b) represent the relatively long edges, thus they are not counted in our new TSP formulation. By excluding them, meaningful tours like Fig. 1(b) can be favored.

More formally, our variation of the TSP formulation defines the fitness function by

$$\sum_{i=1}^{n} D(g_{\pi_i}, g_{\pi_{i+1}})\delta(g_{\pi_i}, g_{\pi_{i+1}})$$

where

$$\delta(i,j) = \begin{cases} 0 & \text{if } (i,j) \in L \\ 1 & \text{otherwise} \end{cases}$$

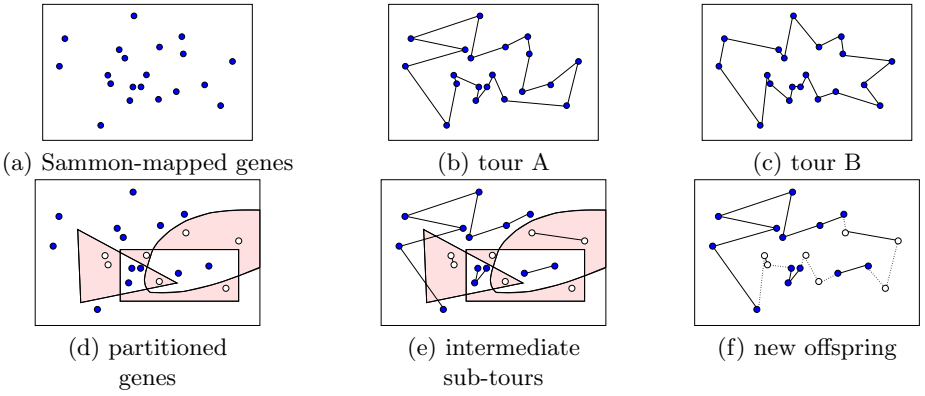
in which (i, j) is the edge connecting gene i and gene j, and L is the set of excluded edges.

We use the new fitness function only in selection and replacement. In other words, the other stages of GA except them still use the common TSP formulation which considers all the edges in a tour. This setting was settled down after some experiments.

4 A Hybrid Genetic Algorithm

A genetic algorithm hybridized with local optimizations is called a hybrid GA. A great many studies about hybridization of GAs were proposed [49,32].

- Sammon's Mapping
  Since the microarray data are virtually real-valued vectors in a high-dimensional space, we map them into the two-dimensional space in order to use the 2D natural crossover, which operates on chromosomes encoded by 2D *graphic images*. We chose the Sammon's mapping described in Section 2.2. Figure 2(a) shows a Sammon-mapped image from a small subset of real-field microarray data.
- Encoding
  Using the Sammon's mapping, we obtain a 2D Euclidean TSP instance using the distance information. We use the *graphic image itself* of a tour as a chromosome. This encoding was used in [20,21] and showed successful results on most TSP benchmarks.
- Initialization
  All the chromosomes are created at random. We set the population size to be 50 in our algorithm.
- Selection
  We use the tournament selection [14]. The tournament size is 2.



(a) Sammon-mapped genes (b) tour A (c) tour B

(d) partitioned genes (e) intermediate sub-tours (f) new offspring

Fig. 2. A Sammon-mapped image and an example crossover on it

– Crossover
  We use the natural crossover [20,21]. The natural crossover draws *free curves* on the 2D space where genes are located. The curves divide the chromosomal positions into two disjoint partitions. Then we copy the genes in one partition from one parent to the offspring and those in the other partition from the other parent. Figures 2(b) through (f) show an example operation of the natural crossover on a Sammon-mapped chromosome.
– Mutation
  The double-bridge kick move, which is known to be effective from the literature [19,27], is used.
– Local Optimization
  We use the Lin-Kernighan (LK) algorithm [26], which is one of the most effective heuristics for TSP. The LK used here is an advanced version incorporating the techniques of *don't-look bit* [4] and *segment tree* [12] which cause dramatic speed-up.
– Replacement
  The replacement scheme proposed in [6] is used. The offspring tries to first replace the more similar parent, measured by Hamming distance [15], if it fails, then it tries to replace the other parent (replacement is done only when the offspring is better than one of the parents). If the offspring is worse than both parents, we replace the worst member of the population (GENITOR-style replacement [50]).
– Stopping Criterion
  The GA stops when one of the three conditions is satisfied: i) 80% of the population is occupied by solutions with the same quality, whose chromosomes are not necessarily the same, ii) the number of consecutive fails to replace the best solution reaches 200, or iii) the number of generations reaches 2000.

Table 1. Data Set

Data Set Name	Number of Genes	Number of Experiments
Cell cycle cdc15	782	24
Cell cycle	803	59
Yeast complexes	979	79

5 Experimental Results

5.1 Test Beds and Test Environment

We tested the proposed algorithm on three data sets, Cell Cycle cdc15, Cell Cycle, and Yeast Complexes. The first two data sets consist of about 800 genes each, which are cell cycle regulated in *saccharomyces cerevisia* with different numbers of experiments [40]. They are classified into five groups termed G1, S, S/G2, G2/M, and M/G1 by Spellman *et al.* [40]. Although it is controversial whether the group assignment does reflect the real grouping, it is known to be meaningful to some degree [46]. The final data set, Yeast Complexes, is from MIPS yeast complexes database [10]. All these three data sets can be found in [2] and downloaded at a web site.[3] Table 1 shows a brief description of each data set.

All programs were written in C++ language and run on Pentium III 866 MHz with Linux 2.2.14. They were compiled using GNU's *g++* compiler. We performed 100 runs for each experiment.

5.2 Performance

We denote by NNGA our proposed hybrid GA using the natural crossover and the new TSP formulation. The performances of the visualization results are evaluated by a score described in [46], which is defined by

$$Score = \sum_{i=1}^{n} G(g_{\pi_i}, g_{\pi_{i+1}})$$

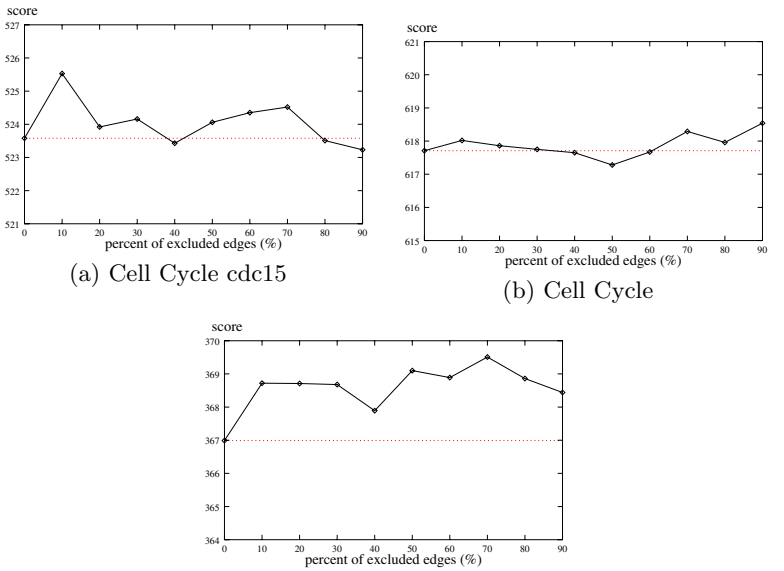
where $g_{\pi_{n+1}} = g_{\pi_1}$, and

$$G(g_{\pi_i}, g_{\pi_j}) = \begin{cases} 1, \text{ if } g_{\pi_i} \text{ and } g_{\pi_j} \text{ are in the same group} \\ 0, \text{ if } g_{\pi_i} \text{ and } g_{\pi_j} \text{ are not in the same group} \end{cases} .$$

It is clear that a solution gets a higher score, under this scoring system, when more genes with the same group are aligned next to each other.

Figure 3 shows the scores found by NNGA when the percent of the excluded edges varies from 0% to 90% at intervals of 10%. The NNGA improved the

[3] http://www.psrg.lcs.mit.edu/clustering/ismb01/optimal.html



(a) Cell Cycle cdc15

(b) Cell Cycle

(c) Yeast Complexes

Fig. 3. Scores on a spectrum of different rates of excluded edges

results for most of the tested percents. In particular, for the Yeast Complexes data set, it showed improvement for all of the tested percents.

It is interesting that two peaks, not necessarily the highest ones, were observed at around 10% and 70% for all data sets. They imply that it is more favorable to exclude either a small or a large number of edges than do an intermediate number of edges in our experimental settings.

Table 2 compares the performance of our NNGA with state-of-the art algorithms for clustering gene expression profiles in terms of the score. The Single-, Complete-, and Average-linkage represent different versions of hierarchical clustering [10] and SOM [42] is a self-organizing map. We used the CLUSTER package [4] for the three versions of hierarchical clustering and the SOM. The NNGA with the new TSP formulation dominated the others.

It is more intuitive to inspect the visualized results than to just compare the scores between the algorithms for clustering gene expression profiles. To visualize them, we should assign a color to each expression level. We follow the typical red/green coloring scheme [41,10], while other schemes using different colors are available [41]. The red/green coloring scheme is as follows:

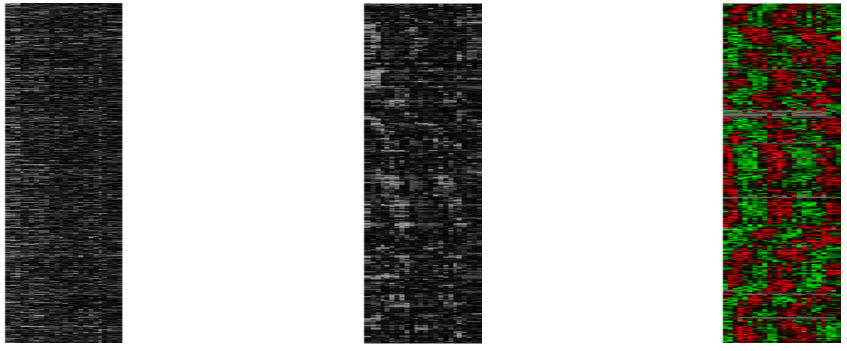
- Expression levels of zero are colored black, increasingly positive levels with reds of increasing intensity, and increasingly negative levels with greens of increasing intensity.
- Missing expression levels are usually colored gray.

[4] http://genome-www.stanford.edu/clustering

Table 2. Comparisons of NNGA with other algorithms in terms of best score

	Cell cycle cdc15	Cell cycle	Yeast Complexes
NNGA	**539**	**634**	**384**
FCGA	521	627	N.A.
Single-linkage	251	336	300
Complete-linkage	498	598	340
Average-linkage	500	581	331
SOM	461	578	306

N.A. : Not Available



(a) Random (b) FCGA (c) NNGA

Fig. 4. Visualization results for Cell cycle cdc15

Figure 4 shows the visualization results for Cell cycle cdc15. In particular, Fig. 4(a) shows a random order, which is the original order and Fig. 4(b) and 4(c) show the best orders found by FCGA and NNGA, respectively. The NNGA shows a notable feature that clusters gene expression profiles with many missing data. One can find the clustered gray rows in Fig. 4(c).

6 Conclusions

We proposed a hybrid genetic algorithm for finding the optimal gene order in displaying the microarray data. To use the natural crossover, which exploits two-dimensional geographical information, we applied the Sammon's mapping to the data.

Furthermore, we improved the visualization results using our new TSP formulation. Our key idea is that reducing relatively long edges in a tour is less meaningful for visualization and it is thus advantageous to exclude the long edges from the fitness function. Experimental results showed that our idea improved the visualization results. Using the new fitness function, we could align

more genes with the same group next to each other compared to state-of-the-art algorithms.

However, there is still a lot of work to give insight into the visualization of the microarray data. Since there have been no official criterion for evaluating visualization results, it is controversial to claim which one is better than the others in terms of a measure. We think that it is because most biologists analyze the results based on their limited visual intuition. We believe that examining what visualization is more meaningful to biologists is one of the most fundamental and demanding study.

The distance measure itself is also an interesting issue. While the Pearson correlation has been extensively used in the literature, it is not clear that the Pearson correlation is the best measure for defining the similarity between gene expression profiles. More elaborate distance measures such as an information-theoretic measure [11,25] are left for future studies.

Acknowledgments. The authors would like to thank Soonchul Jung and Huai-Kuang Tsai for invaluable discussions on this paper. This work was partly supported by Optus Inc. and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

References

1. A. A. Alizadeh, M. B. Eisen, and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.
2. Z. Bar-Joseph, D. K. Gifford, and T. S. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17:22–29, 2001.
3. A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expresssion patterns. *Journal of Computational Biology*, 6:281–297, 1999.
4. J. L. Bentley. Experiments on traveling salesman problem. In *1st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '90)*, pages 129–133, 1990.
5. T. Biedl, B. Brejova, and et al. Optimal arrangement of leaves in the tree representing hierarchical clustering of gene expression data. Technical Report Technical Report CS-2001-14, Dept. of Computer Science, University of Waterloo, 2001.
6. T. N. Bui and B. R. Moon. Graph partitioning and genetic algorithms. *IEEE Transactions on Computers*, 45:841–855, 1996.
7. H. David. First (?) occurrence of common terms in mathematical statistics. *The American Statistician*, 49:121–133, 1995.
8. W. Dzwinel. How to make Sammon mapping useful for multidimensional data structures analysis. *Pattern Recognition*, 27(7):949–959, 1994.
9. A. Edwards and L. Cavalli-sforza. A method for cluster analysis. *Biometrics*, 21:362–375, 1965.
10. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of the National Academy of Sciences*, pages 14863–14867, 1998.
11. A. M. Fraser. Reconstructing attractors from scalar time series: a comparison of singular system and redundancy criteria. *Physica D*, 34:391–404, 1989.

12. M. L. Fredman, D. S. Johnson, L. A. McGeoch, and G. Ostheimer. Data structures for traveling salesman. In *4th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '93)*, pages 145–154, 1993.
13. D. E. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison-Wesley, Reading, MA, 1989.
14. D. E. Goldberg, K. Deb, and B. Korb. Do not worry, be messy. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 24–30, 1991.
15. R. Hamming. Error detecting and error correcting codes. *Bell systems Technical Journal*, 29(2):147–160, 1950.
16. J. Harris. The arithmetic of the product moment of calculating the coefficient of correlation. *American Nature*, 44:693–699, 1910.
17. J. Herrero, A. Valencia, and J. Dopazo. A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics*, 17:126–136, 2001.
18. J. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, 1975.
19. D. S. Johnson. Local optimization and the traveling salesman problem. In *17th Colloquium on Automata, Languages, and Programming*, pages 446–461, 1990.
20. S. Jung and B. R. Moon. The natural crossover for the 2D Euclidean TSP. In *Genetic and Evolutionary Computation Conference*, pages 1003–1010, 2000.
21. S. Jung and B. R. Moon. Toward minimal restriction of genetic encoding and crossovers for the 2D Euclidean TSP. *IEEE Transactions on Evolutionary Computation*, 6(6):557–565, 2002.
22. S. Kawasaki, C. Borchert, and et al. Gene expression profiles during the initial phase of salt stress in rice. *Plant Cell*, 13(4):889–906, 2001.
23. M. Kendall. A new measure of rank correlation. *Biomerika*, 30:81–93, 1938.
24. A. B. Khodursky, B. J. Peter, and et al. DNA microarray analysis of gene expression in reponse to physiological and genetic changes that affect tryptophan metabolism in escherichia coli. In *Proceedings of the National Academy of Sciences*, pages 12170–12175, 2000.
25. W. Li. Mutual information functions versus correlation functions. *Journal of Statistical Physics*, 60:823–837, 1990.
26. S. Lin and B. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21(4598):498–516, 1973.
27. O. Martin, S. Otto, and E. Felten. Large-step Markov chains for the traveling salesman problem. *Complex Systems*, 5:299–236, 1991.
28. P. Merz and A. Zell. Clustering gene expression profiles with memetic algorithms. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, pages 811–820, 2002.
29. P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report Technical Report C3P Report 826, Concurrent Computation Program, California Institute of Technology, 1989.
30. Y. Nagata and S. Kobayashi. Edge assembly crossover: A high-power genetic algorithm for the traveling saleman problem. In *7th International Conference on Genetic Algorithms*, pages 450–457, 1997.
31. E. Pekalska, D. De Ridder, R. P. W. Duin, and M. A. Kraaijveld. A new method of generalizing Sammon mapping with application to algorithm speed-up. In *Fifth Annual Conference of the Advanced School for Computing and Imaging*, pages 221–228, 1999.
32. J. M. Renders and H. Bersini. Hybridizing genetic algorithms with hill-climbing methods for global optimization: Two possible ways. In *Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 312–317, 1994.

33. D. De Ridder and R. P. W. Duin. Sammon's mapping using neural networks: a comparision. *Pattern Recognition Letters*, 18(11–13):1307–1316, 1997.
34. J. W. Sammon, Jr. A non-linear mapping for data structure analysis. *IEEE Transactions on Computers*, 18:401–409, 1969.
35. R. Schaffer, J. Landgraf, and et al. Microarray analysis of diurnal and circadian-regulated genes in arabidopsis. *Plant Cell*, 13(1):113–123, 2001.
36. M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expresssion patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, 1995.
37. D. Shalon, S. J. Smith, and P. O. Brown. A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. *Genome Research*, 6(7):639–645, 1996.
38. R. R. Sokal and C. D. Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.
39. C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:72–101, 1904.
40. T. S. Spellman, G. Sherlock, and et al. Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisia* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998.
41. A. Sturn. Cluster analysis for large scale gene expression studies. Master's thesis, Graz University of Technology, Graz, Austria, 2001.
42. P. Tamayo, D. Slonim, and et al. Interpreting patterns of gene expresssion with self-organizing maps: Methods and application to hematopoietic differentiation. In *Proceedings of the National Academy of Sciences*, pages 2907–2912, 1999.
43. S. Tavazoie, J. D. Hughes, and et al. Systematic determination of genetic network architecture. *Nature Genetics*, 22:281–285, 1999.
44. P. Toronen, M. Kolehmainen, G. Wong, and E. Castren. Analysis of gene expression data using self-organizing maps. *FEBS Letters*, 451:142–146, 1999.
45. H. K. Tsai, J. M. Yang, and C. Y. Kao. A genetic algorithm for traveling salesman problems. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001)*, pages 687–693, 2001.
46. H. K. Tsai, J. M. Yang, and C. Y. Kao. Applying genetic algorithms to finding the optimal order in displaying the microarray data. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002)*, pages 610–617, 2002.
47. H. K. Tsai, J. M. Yang, and C. Y. Kao. Solving traveling salesman problems by combining global and local search mechanisms. In *Proceedings of the Congress on Evolutionary Computation (CEC 2002)*, pages 1290–1295, 2002.
48. J. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.
49. D. Whitley, V. Gordon, and K. Mathias. Larmarckian evolution, the baldwin effect and function optimization. In *International Conference on Evolutionary Computation*, Oct. 1994. *Lecture Notes in Computer Science*, 866:6–15, Springer-Verlag.
50. D. Whitley and J. Kauth. GENITOR: A different genetic algorithm. In *Proceedings of Rocky Mountain Conference on Artificial Intelligence*, pages 118–130, 1988.