AKTION Österreich – Tschechische Republik

AKTION 23p20

Comparison of statistical and evolutionary approaches to feature selection in pattern recognition

aktion-002-2000 June 14, 2000

Conventional and Evolutionary Feature Selection of SAR Data Using a Filter Approach

The 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2000)

July 23–26, 2000, Orlando, Florida

Helmut A. Mayer and Petr Somol

helmut@cosy.sbg.ac.at, somol@utia.cas.cz

Department of Computer Science
University of Salzburg
Austria

Department of Pattern Recognition
Czech Academy of Sciences
Czech Republic

Correspondence to:

Universität Salzburg
Institut für Computerwissenschaften
Jakob–Haringer–Strasse 2
A–5020 Salzburg
Austria

Conventional and Evolutionary Feature Selection of SAR Data Using a Filter Approach

Helmut A. Mayer
Department of Computer Science, University of Salzburg
A–5020 Salzburg, Austria

and

Petr Somol
Department of Pattern Recognition, Czech Academy of Sciences
CZ–18208 Prague 8, Czech Republic

ABSTRACT

The steady increase of the amount of information gathered by remote sensing systems makes the development of highly efficient algorithms to reduce the data volume even more important. Ideally, however, such methods should achieve this goal without or only slight loss of information contained in the data. Generally, feature selection algorithms offer the possibility to reduce noise, to eliminate irrelevant features in a data set, and to increase the performance of subsequent processing steps. We compare recently developed and highly effective sequential feature selection algorithms with approaches based on evolutionary algorithms enabling parallel feature subset selection. We introduce the oscillating search method, employ permutation encoding offering some advantages over the more traditional bitmap encoding for the evolutionary search, and compare these algorithms to the often studied and well–performing sequential forward floating search. The investigated SAR (synthetic aperture radar) data set is divided into 7 land–cover classes by a total of 57 features including SAR amplitude and coherence, and first and second order statistics. We assess the quality of feature subsets by means of the average Bhattacharyya distance measure.

1. INTRODUCTION

The problem of selecting a "good" subset of features from the total of available features arises in a great variety of problem domains in science, engineering, and economy. As the number of possible subsets increases exponentially with the number of total features, a variety of deterministic and nondeterministic *Feature Selection* (FS) algorithms have been developed in order to escape the *Curse of Dimensionality*. In this work we would like to compare some very efficient sequential FS algorithms with parallel FS techniques based on *Evolutionary Algorithms* (EAs). Basically, a sequential FS algorithm adds a feature to or leaves out a feature from the subset to be constructed in an iterative manner. Hence, the feature subset generation depends on initial and intermediate subsets. A parallel FS algorithm constructs a complete feature subset at once. The latter can be achieved by EAs which are generally believed to be well–suited for nonlinear, high–dimensional problems of exponential complexity [1] [2].

The two basic categories of approaches to FS are the *Filter* and the *Wrapper* approach [3]. The crucial point in discriminating these methods is the absence or presence, respectively, of a classifier to assess the quality of a feature subset. Either, a statistical measure independent of a classifier is employed (filter approach), or the error rate of a classifier determines the usefulness of a feature subset (wrapper approach). In this work we adopt the filter approach, as our main goal is to compare the performance of conventional to evolutionary FS algorithms. The additional use of specific classifiers (and training algorithms) would increase the already complex FS process interactions, and make it even more difficult to isolate the effectiveness of the FS algorithms to be compared.

For experimental comparisons we employ the Bhattacharyya (B–) distance measure [4], and investigate algorithm performance on a land–cover classification problem with the total number of features ranging from the single best feature to the (almost) full feature data set. Feature extraction for classification of SAR images is commonly based on combined point and texture classification [5]. Hence, we extracted 57 features from first and second order statistic.

Feature Selection Problem

Following the statistical approach to pattern recogni-

*This work has been supported by *AKTION Österreich – Tschechische Republik* under grant AKTION 23p20: "Comparison of statistical and evolutionary approaches to feature selection in pattern recognition"

tion, we assume that a pattern or object described by a real D-dimensional vector $\mathbf{x} = (x_1, x_2, \cdots, x_D)^{\mathrm{T}} \in \mathcal{X} \subset \mathcal{R}^D$ is to be classified into one of a finite set of C different classes $\Omega = \{\omega_1, \omega_2, \cdots, \omega_C\}$. The patterns are supposed to occur randomly according to some true class conditional probability density functions (pdfs) $p^\star(\mathbf{x}|\omega)$ and the respective *a priori* probabilities $P^\star(\omega)$. Since the class conditional pdfs and the *a priori* class probabilities are rarely known in practice, it is necessary to estimate these probability functions from the training sets of samples with known classification.

If the pdfs are a priori known to be unimodal, probabilistic distance measures, e.g., Mahalanobis or Bhattacharyya distance, may be appropriate to evaluate the quality of a feature subset. As pointed out by Siedlicki and Sklansky (1988) the error rate with respect to the chosen measurement criterion $J(\cdot)$ is even better (computational feasibility provided) [6].

Bhattacharyya Distance

In the following formulation B–distance measures the separability of normal distributions for two classes indexed by i and k [4]:

$$B_{ik} = \frac{1}{8}(\mu_i - \mu_k)^t \Sigma^{-1} (\mu_i - \mu_k) + \frac{1}{2} \ln \left(\frac{|\Sigma|}{|\Sigma_i|^{\frac{1}{2}} |\Sigma_k|^{\frac{1}{2}}} \right) \quad (1)$$

$$\Sigma = \frac{\Sigma_i + \Sigma_k}{2} \quad (2)$$

where μ_i, μ_k are the feature mean vectors and Σ_i and Σ_k denote class covariance matrices for classes i and k, respectively. We point out, that a more general distance measure, such as the Chernoff distance, is in general closer to the error rate than B–distance, on the other hand such a measure is not easy to obtain [7].

2. FEATURE SELECTION ALGORITHMS

Assuming that a suitable criterion function $J(\cdot)$ has been chosen to evaluate the effectiveness of feature subsets, FS is reduced to a search for a (sub)optimal feature subset based on the selected measure.

Although an exhaustive search is a sufficient procedure to guarantee the optimality of a solution, in many realistic problems it is computationally prohibitive. Therefore, in practice one has to rely on computationally feasible procedures to avoid exhaustive search for the price of suboptimal results. A comprehensive list of suboptimal procedures and the corresponding formulas can be found in [8]. An excellent taxonomy of currently available FS methods in pattern recognition is presented in [9].

The strategies used to find a useful subset of features range from the simple but popular sequential forward (SFS) and sequential backward selection (SBS), to more sophisticated but computationally more expensive algorithms. In the following section we will describe three of the latter which will then be compared on three benchmark problems.

Sequential Forward Floating Search

The sequential forward floating search (SFFS) [10] can be viewed as SFS with backtracking. SFS is a simple greedy algorithm starting with an empty feature subset, then iteratively adding the feature which maximizes the evaluation criterion of the feature subset up to a specified target size t. Obviously, this procedure does not account for nonlinear interactions of features, hence SFFS offers the following improvements:

After adding a feature x_i to a subset of size k with $J(k)$ in SFS manner (inclusion), SFFS tries to find a feature x_j with $j \neq i$ which can be excluded so that the new $J'(k) > J(k)$ (conditional exclusion). This step is continued as long as a feature can be excluded under the above condition which demands that the best values of $J(\cdot)$ are recorded for each subset size (continuation of conditional exclusion). If an exclusion step is not successful, inclusion continues until the algorithm has "floated" through all possible subset sizes or has reached a certain target size t.

While SFFS starts with an empty feature subset dominantly searching in forward direction, the sequential backward floating search (SBFS) "floats" in backward direction and analogously can be viewed as SBS with backtracking [10].

Oscillating Search

A very recent development is the oscillating search (OS) algorithm [11]. Most of the known suboptimal FS methods are based on step–wise adding of features to an initially empty feature set, or on step–wise removing features from an initial set of all features. A single search direction – forward or backward – is usually preferred. It is apparent that all these algorithms spend a lot of time testing feature subsets having cardinalities far distant from the required cardinality t. Unlike other methods, OS is based on repeated modification of the current subset X_t of t features. This is achieved by alternating the *down–* and *up–swings*. The down–swing removes o "worst" features from the current set X_t to obtain a new set X_{t-o} at first, then adds o "best" ones to X_{t-o} to obtain a new current set X_t. The *up-swing* adds o "good" features to the current set X_t to obtain a new set X_{t+o} at first, then removes o "bad" ones from X_{t+o} to obtain a new current set X_t again. The intitial subset X_t is generated by SFS (or SBS), and an up– and down–swing is achieved by SFS and SBS, respectively.

Let us denote two successive opposite swings as an *oscillation cycle*. Then, the oscillating search consists of repeating oscillation cycles. The parameter o is termed *oscillation cycle depth* and should initially be set to 1. If the last oscillation cycle did not find a better subset X_t of t features, the algorithm increases the

oscillation cycle depth by setting $o = o+1$. Whenever any swing finds a better subset X_t of t features, the depth value o is restored to 1. The algorithm terminates, when the value of o exceeds the user–specified limit Δ. Besides the basic OS algorithm described here, variants, e.g., random generation of initial subset X_t, can be found in [11].

Evolutionary Algorithms for Feature Selection
When employing EAs for FS, complete subsets are generated in parallel potentially eliminating the problems inherent to sequential FS methods. Basically, a start population of EA chromosomes (feature subsets) is generated randomly, and each individual receives a fitness according to the evaluation criterion $J(\cdot)$. The fitness determines the probability of an individual to be selected for mating, where (usually) two individuals exchange their genetic information. The offspring then undergo a mutation operation and form the next generation. This evolutionary cycle is repeated for a user–defined number of generations, and the best individual (feature subset) of all generations represents the final solution.
In the basic approach to feature selection using EAs a feature subset is encoded as a binary vector (*Bitmap Encoding*) $\vec{a} = (a_1, \ldots, a_d)$, where $a_i = 1$ indicates the presence of the $i-$th feature in the subset, while the absence of the $i-$th feature is expressed by $a_i = 0$. The bitmap encoding is well suited for an evolutionary feature selection technique based on a wrapper approach [12].
In [13] the bitmap encoding was generalized to a weighted encoding, where features were assigned different weights resulting in a modification (warping) of the feature space for a k–NN classifier. In [14] bitmap encoding has been employed for the evolutionary search of feature subsets for an ANN classifier. A mixture of filter and wrapper approach is presented in [15], where an EA with a fitness function comprising k–NN classifier accuracy and multiple correlation coefficients is employed for FS. Although all these works could report on improvements in accuracy or subset size, none of them compared evolutionary to conventional FS methods.
With the filter approach and the inherent monotony of the B–distance measure we are using in this work, bitmap encoding together with a fitness function only assessing B–distance would simply result in convergence to the full feature set. A penalty or cost term could be introduced in the fitness function $f(\vec{a}) = J(\vec{a}) + p(\vec{a})$, where the function $p(\vec{a})$ depends on the number of features present in $\vec{a}$. The penalty term $p(\vec{a})$ can then be used to favor feature subsets of a given cardinality.
Intuitively, for the experimental framework in this work the use of a penalty term unnecessarily complicates the evolutionary search, as the EA has to find a subset size which is known from the very beginning. Moreover, as soon as the EA arrives at the correct size of the feature subset it will likely stay with this solution, because only very specific interactions of genetic operators will allow the transition to a different solution of the given target size. Experimental results confirmed these assumptions, hence we introduce a permutation encoding method.

3. PERMUTATION ENCODING

As outlined in Section 2, instead of the more traditional bitmap encoding for the generation of subsets, we experimented with variants of permutation encoding which is primarily used for order problems such as the *Traveling Salesperson Problem* (TSP). The bases [1] of the EA chromosome are integers building a permutation as shown in Figure 1.

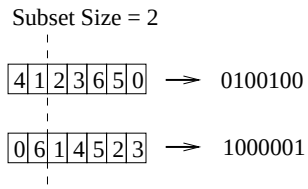

Figure 1: Permutation encoding for the generation of subsets.

For the generation of a subset of a specific target size t the chromsome is scanned from left to right, and the first t bases representing a feature index are used to construct the subset. The permutation encoding ensures that each feature taken into account is different, and always yields a subset of the given target size t.
In order to preserve the permutation property of the chromosomes, specific genetic operators have been devised. The mutation operator simply exchanges two random bases on the chromosome with a given mutation rate p_m (usually in the range of $0.001 - 0.01$). One of the most prominent crossover operators for permutation encoding is the *Partially Matched Crossover* (PMX) proposed in [16]. Its basic mechanisms are presented in Figure 2.

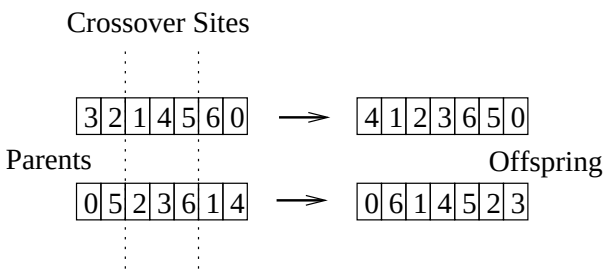

Figure 2: Partially Matched Crossover (PMX).

Generally, for crossover two parent chromosomes are selected, and crossover is performed according to a

[1] Many researchers use the term *gene*, but a wild type gene is a much more complex structure, and the term *base* is equally a synonym for an atomic information unit in biology.

user–defined crossover rate p_c (usually in the range of $0.6 - 1.0$). If no crossover occurs, the two parents are simply copied to two offspring chromosomes. In the crossover phase two crossover sites are selected randomly (sites are the same for both parents). Then, the bases in between the crossover sites are exchanged. Up to this point we have exactly described the very common *2–point Crossover*, but if the bases were only exchanged, the permutation property would be lost. Thus, each base to be copied to the other parent is searched in that parent and swapped with the base currently at the locus, where the exchange takes place (just like a single mutation). In doing so the partial order between the crossover sites can be exchanged between the parents without corrupting the permutation property.
As can be seen in Figure 1 parts of the chromosome are not expressed. If both crossover sites fall into this region, it might appear that crossover does not change the expressed feature subset. But assuming an exemplary target subset size $t = 2$ in Figure 2, it can be observed that with PMX the features in the subset can be altered even under this condition. Though, a mutation in an unexpressed region of the chromsome does not effect the encoded subset, it can indirectly take an influence by means of a subsequent PMX crossover.

4. SAR IMAGE

The SAR data set used for FS experiments is a geocoded (2.5m grid) and georeferenced image from the area of Solothurn, Switzerland. The initial features are X–Band SAR amplitude, P–Band SAR amplitude, and X-Band (InSAR Interferometric SAR) coherence. From each of these three features eight first order and ten second order statistical features are extracted yielding a total number of 57 features. The first order features within a 7x7 window are: mean, coefficient of variation, skewness, kurtosis, contrast, homogeneity, median, and range. The second order features derived from the co–occurrence matrix of an 11x11 window are: energy, entropy, maximum, angular second moment, inverse difference moment, correlation, cluster shade, cluster prominence, information correlation I, and information correlation II. The seven land–cover classes are low/medium/high vegetation, low/high forest, and residential/industrial area. The data sample for calculation of B–distances comprises 1721 examples. For a subset of t features B–distance is computed for each combination of two classes and averaged on the number of combinations. The rather high number of features of this land–cover classification problem challenges the search capabilities of the FS algorithms to be compared.

Experimental Setup
Experiments have been run using the three FS algorithms SFFS, OS, and EA to be compared. All subset sizes from $t = 1$ to $t = n - 1$ (with n being the total number of features) have been investigated. The following parameters have been used with all the experiments in this paper:
SFFS Parameters: Initial subset size = 2 (generated by SFS), Runs = 1 (deterministic behavior).
OS Parameters: Initial subset size = t (generated by SFS), $\Delta = 10$ (number of features), Runs = 1 (deterministic behavior).
EA Parameters: Population Size = 50, Generations = 100, Crossover Probability $p_c = 0.6$, Mutation Probability $p_m = 0.01$, Crossover = PMX, Selection Method = Binary Tournament, Runs = 20 (the EA parameters are fairly standard and are not based on extensive experiments).

5. EXPERIMENTAL RESULTS

Figure 3 presents the best B–distances for all feature subset sizes ranging from $t = 1$ to $t = n - 1$.

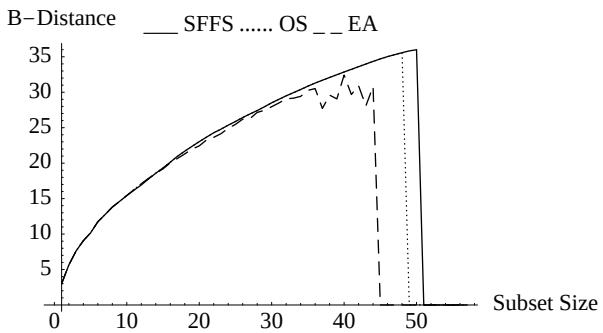

Figure 3: Best B–distances found by Sequential Floating Forward Search (SFFS), Oscillating Search (OS), and Evolutionary Algorithm (EA) (averaged on 20 runs).

In this problem SFFS and OS exhibit nearly identical performance. Only for subset sizes $t = 11$ and $t = 12$ is OS able to improve the result of SFFS. Up to a subset size of $t \sim 17$ EA is very close to OS and for some subset sizes even slightly excels the results of the deterministic oscillations (e.g., $t = 10$ or $t = 13$). However, for increasing subset sizes performance of the EA decreases.
The main reason for this behavior is the fact that three of the 57 features are faulty, i.e., the values of the second order XSAR features maximum, angular second moment, and inverse difference moment are 0.0 for some classes. While the deterministic algorithms explicitely consider the quality of features to be included or excluded, the EA performs a "blind search" making it more difficult to dismiss the faulty features. Specifically, when no valid solution is present in the population it is impossible for the EA to determine any difference in the quality of solutions, as technically, all (invalid) solutions receive a fitness of 0.0. Hence, in some runs the EA detects no valid solution, while in others it finds some feasible solution

by chance which becomes more and more unlikely with increasing subset size (Figure 4). This effect accounts for the decreasing average B–distance, although this measure is monotonic (provided the best features are selected).

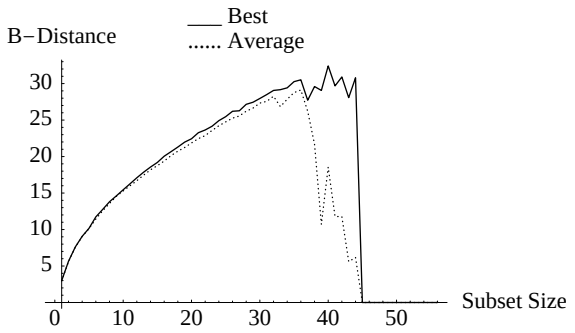

Figure 4: Best and average B–Distances found by the Evolutionary Algorithm (EA) (averaged on 20 runs).

The deterministic algorithms also face this problem, but it unravels only at subset sizes close to $t = 54$ being the number of consistent features. At a subset size $t = 52$ none of the compared algorithms finds a valid solution.
In terms of computational cost (Figure 5) OS is the most expensive, but also delivers the best results. However, increase of the number of generations of the EA would further improve its solutions.

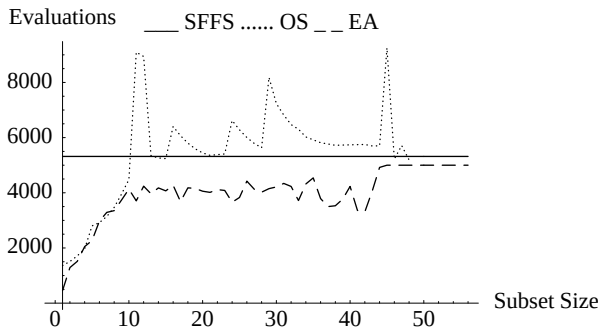

Figure 5: Mean number of evaluations to find the best B–Distance of Sequential Floating Forward Search (SFFS), Oscillating Search (OS), and Evolutionary Algorithm (EA) (averaged on 20 runs).

Considering computational cost together with algorithm performance confirms the well established fact that EAs find good solutions in a short time, but are less effective in honing these solutions.
Table 1 shows a ranking of the features most frequently selected into all subsets generated by EA (X is for features derived from X–Band SAR, XI for X–Band InSAR).
The most noticeable property of the ranking is that no feature extracted from P–Band SAR is among the best (6 X, 3 XI). Interestingly, the three most often selected features are X second order statistics. However, the first order statistics contrast and homogeneity are the only top features selected for both, X and XI.

Feature	Frequency
X-Correlation (14)	0.9888
X-Entropy (10)	0.9650
X-Cluster Shade (15)	0.9400
XI-Contrast (43)	0.8875
X-Contrast (5)	0.8612
X-Homogeneity (6)	0.8438
X-Mean (1)	0.8413
XI-Homogeneity (44)	0.7375
XI-Cluster Prominence (54)	0.7000

Table 1: Ranking of features according to their relative frequency in all feature subsets generated by EA (relative frequency ≥ 0.7, feature number in parentheses).

If we look at the specific results of SFFS, OS, and EA for a subset size $t = 9$, we find that all algorithms have found the same subset containing all the features listed in Table 1.

6. OUTLOOK

The results presented in this paper are an indicator for the very good performance of OS and EA considering that SFFS has been evaluated as one of the best available FS algorithms in [9]. In rare cases OS could even improve the result of SFFS, while in turn EA could even improve OS and SFFS results for some small subset sizes, but it suffered from faulty features in the data set in the case of larger subset sizes. On average EA took a smaller number of evaluations to reach a performance level comparable to the deterministic algorithms. Clearly, these results have to be confirmed with a number of additional data sets and the algorithms should also be investigated in the environment of a wrapper approach employing a number of different classifiers. A very promising future research direction could be the hybridization of EA and OS combining the speed of an EA to find a good solution with the ability of the OS to improve an existing feature subset.

7. ACKNOWLEDGMENTS

We would like to thank the executives of the Aktion board for their very friendly and motivating support of this inter–European research initiative. We appreciate the generosity of Reinhold Huber to provide the SAR image data set and to assist with valuable comments on various issues of remote sensing, specifically, feature extraction from SAR data. Many thanks to our project colleagues Pavel Pudil, Jiri Grim and Roland Schwaiger for their support. We also thank

Markus Amersdorfer, Martin Angerer, Sandor Herramhof, and Harald Schweiger for development of a Java prototype for evolutionary feature selection in their software engineering course.

References

[1] H.-P. Schwefel. *Evolution and Optimum Seeking.* Sixth–Generation Computer Technology Series. Wiley, New York, 1995.

[2] Melanie Mitchell. *An Introduction to Genetic Algorithms.* Complex Adaptive Systems. MIT Press, Cambridge, MA, 1996.

[3] G. John, R. Kohavi, and K. Pfleger. Irrelevant Features and the Subset Selection Problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, San Mateo, CA, 1994. Morgan Kaufmann.

[4] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[5] Anne H. Schistad Solberg and Anil K. Jain. Texture Fusion and Feature Selection Applied to SAR Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 35(2):475–479, March 1997.

[6] W. Siedlecki and J. Sklansky. On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(2):197–220, June 1988.

[7] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE Transactions on Communications Technology*, 15(1):52–60, 1967.

[8] P. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach.* Prentice, 1982.

[9] A. Jain and D. Zongker. Feature Selection: Evaluation, Application and Small Sample Performance. *IEEE Transactions on PAMI*, 19(2):153–158, 1997.

[10] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125, 1994.

[11] Petr Somol and Pavel Pudil. Oscillating Search Algorithms for Feature Selection. In *Submission to the 15th International Conference on Pattern Recognition, Barcelona*, 2000.

[12] Wojciech Siedlecki and Jack Sklansky. A Note on Genetic Algorithms for Large–Scale Feature Selection. *Pattern Recognition Letters*, 10:335–347, 1989.

[13] W. F. Punch, E. D. Goodman, Min Pei, Lai Chia-Shun, P. Hovland, and R. Enbody. Further Research on Feature Selection and Classification Using Genetic Algorithms. In Stephanie Forrest, editor, *Fifth International Conference on Genetic Algorithms*, pages 557–564, San Mateo, CA, July 1993. Morgan Kaufmann.

[14] J. Yang and V. Honavar. Feature Subset Selection Using a Genetic Algorithm. In *Genetic Programming*, pages 380–385, 1997.

[15] Nidapan Chaikla and Yulu Qi. Genetic Algorithms in Feature Selection. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages V – 538–540. IEEE, October 1999.

[16] David E. Goldberg and Robert Lingle. Alleles, Loci, and the Traveling Salesman Problem. In John J. Grefenstette, editor, *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, pages 154–159. Texas Instruments, Inc. and Naval Research Laboratory, Lawrence Erlbaum Associates, 1985.