

# Feature selection with acquisition cost for optimizing sensor system design

K. Iswandy and A. Koenig

Institute of Integrated Sensor Systems, University of Kaiserslautern, Erwin-Schroedinger-Str., 67663 Kaiserslautern, Germany

**Abstract.** Selection of variables from large sets of measurements is a common problem of data analysis and signal processing in many disciplines. In engineering and sensor technology the design of recognition systems can be optimized by judicious choice of subsets of relevant features. In particular, the effort required for signal processing and sensor registration can be considerably reduced by efficient feature selection. However, the current approaches in majority only consider the contribution of features or measurements to the classification ability of the system. The associated cost in terms of computation effort, the required electronics, and power dissipation is not explicitly in consideration. This paper proposes a multi-objective extension of feature selection including acquisition cost and employing and comparing two evolutionary optimization methods. The genetic and particle swarm algorithms and the results achieved with selected data sets will be presented. The results show, that particle swarm algorithm can select best features with lower cost and achieve more competitive results with regard to convergence time and classification accuracy than genetic algorithm.

---

## 1 Introduction

In many real-world applications such as pattern recognition and machine learning, the tasks of object detection often deal with large number of measurements or high dimensional data sets. In particular, in engineering and sensor technology, these large number of measurements or variables (here called features) usually contain some irrelevant and/or redundant features (Langley, 1996), which can degrade the reliability of system performance with regard to the accuracy of the classification. Therefore, the unnecessary features must be eliminated by using methods of dimensionality reduction. One

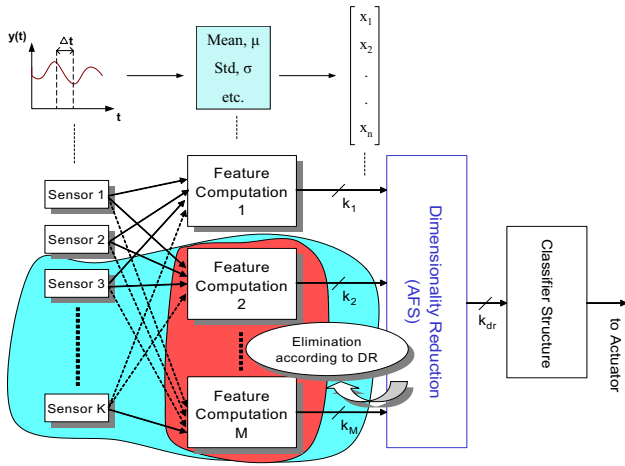
of the dimensionality reduction methods is feature selection, which is focused in this paper for optimizing sensor system design, applying to select a feature subset with the best performance (Jain and Zongker, 1997). Furthermore, the acquisition cost is an important issue (Paclík et al., 2002) in some applications, for example, medical diagnosis (time and price), electronics (power dissipation), texture segmentation (computational effort), etc. The current approaches in majority only consider the contribution of features or measurements to the classification ability of the system. The intention of associated cost is to preserve the quality with regard to classification with least expensive features. The aims of this paper are to propose a multi-objective extension of feature selection and acquisition cost employing two evolutionary computation methods, i.e. genetic algorithm (GA) and particle swarm optimization (PSO), in an optimization scheme and to compare the results between GA and PSO with regard to the reliability (classification accuracy) and the speed (computation and convergency). Figure 1 shows the process of optimization of sensor system design, where the applications of recognition system often use multiple sensors for object measurement. The sensors produce signals as measurement results and these signals, which are sampled in the certain time interval, are represented as array vectors. The array vectors will be processed by some feature computation methods (e.g. mean, standard deviation or variance, histogram, wave amplitude and duration, etc.) to obtain concatenated feature vectors. These feature vectors will be optimized with subset of relevant features associated with acquisition cost and the sensorial effort will be saved due to efficient selection. In the next section the feature selection is briefly presented.

## 2 Brief survey of feature selection

Feature selection is a method to find minimum feature subset giving optimum discrimination between two or more previ-

---

*Correspondence to:* K. Iswandy  
(kuncup@rhrk.uni-kl.de)



**Fig. 1.** The process of sensor system design for recognition system with optimizing structure.

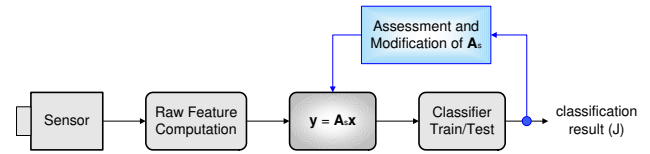
ously defined groups of objects. This feature selection is an iterative algorithm (also called automatic feature selection or AFS). The AFS can be divided into two groups (Kohavi and John, 1998):

- (1) wrapper approach that is performed dependently of the learning algorithm or classifier as shown in Fig. 2 and
- (2) filter approach that is performed independently of the classifier as shown in Fig. 3.

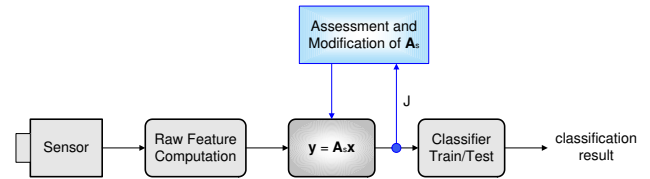
In general, the wrapper method provides features that lead to more accurate classification than that of the filter method. However, the filter method executes many times faster than wrapper method, and therefore stands a much better chance of scaling to datasets with a large number of features than the wrapper does (Mao, 2002).

Moreover, feature selection can be viewed as a heuristic search, where each state in the search space represents a particular subset of the available features. In all but the simplest cases, an exhaustive search of the state space is impractical for even moderate number of features (Jain et al., 2000), since it is also known as a combinatorial optimization problem. However, exhaustive search can be applied well for a small number of features. Several search approaches have been applied to this problem, ranging from simple greedy approaches such as sequential forward selection (SFS), sequential backward selection (SBS), and combination SFS and SBS called floating search (Siedlecki and Sklansky, 1988). The more powerful approach for optimization strategies have been proposed such as genetic algorithm (Siedlecki and Sklansky, 1989) and recently binary particle swarm (Agrafiotis and Cedeno, 2002).

The AFS is a special case of a more general technique known as feature weighting (Aha, 1998; Koenig, 2000) and a linear mapping based on the selection matrix  $A_S$  with



**Fig. 2.** The wrapper approach to automatic feature selection.



**Fig. 3.** The filter approach to automatic feature selection.

$Y = A_S X$ , where  $X$  is a set of  $d$  features.

$$A_S = \begin{pmatrix} c_1 & 0 & 0 & \dots & 0 \\ 0 & c_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & c_{d-1} & 0 \\ 0 & 0 & 0 & \dots & c_d \end{pmatrix}$$

In feature weighting, each feature is associated with weight value that indicates the contribution of that feature in the learning algorithm. According to the feature weighting, there are two categories: (1) feature weighting and (2) feature selection. For feature weighting, the weighting values are arbitrary real number and usually confined to the interval  $c_i \in [0, 1]$ . In feature selection, a weight can either be 1 or 0 to indicate whether the feature is used in the model or not (switch variables). In this paper, we consider only feature selection with using filter approach.

The optimum discrimination according to subset features is determined by an objective function  $J$ . In the filter model, there are several objective functions for feature assessment measures such as simple parametric overlap measure, nonparametric overlap measure, nonparametric compactness measure, nonparametric separability measure (Koenig, 2001; Koenig and Gratz, 2005), etc. In this paper, a supervised nonparametric overlap measure (NPOM) is applied as one criterion for feature space assessment with regard to class regions overlap. NPOM is inspired by nearest-neighbor concepts. This normalized measure gives values close to one for non-overlapping class regions and decreases towards zero proportional to increasingly overlapping of class regions. The NPOM, also denoted as  $Q_{ov}$  in the following, is expressed as

$$Q_{ov} = \frac{1}{L} \sum_{c=1}^L \frac{1}{N_c} \sum_{j=1}^{N_c} \frac{\sum_{i=1}^k q_{NNji} + \sum_{i=1}^k n_i}{2 \times \sum_{i=1}^k n_i} \quad (1)$$

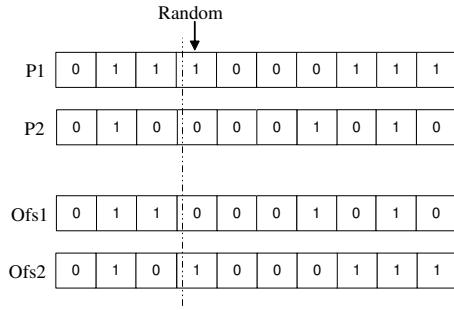


Fig. 4. One-point crossover.

with

$$n_i = 1 - \frac{d_{NN_{ji}}}{d_{NN_{jk}}}$$

and

$$q_{NN_{ji}} = \begin{cases} n_i & \text{if } \omega_j = \omega_i \\ -n_i & \text{if } \omega_j \neq \omega_i \end{cases}$$

where  $d_{NN_{ji}}$  is the Euclidean distance between the regarded pattern and one of its nearest neighbor and  $\omega$  denotes the class affiliation of the regarding pattern.

### 3 Feature selection with acquisition cost

In many applications, the measurement or computation process of object features is always associated with certain cost. The cost can be price of medical diagnosis, power dissipation in electronics, time or computation complexity, etc. The usage of feature selection method in majority only consider the contribution of features or measurements to the classification ability of the system. The accumulation of acquisition cost within automatic feature selection (AFSC) is still rarely investigated. With regard to including acquisition cost our work is similar to Paclík et al. (2002) and Yang and Honavar (1998). However, our approach is more general and employs advanced optimization techniques from evolutionary computation.

To optimize two or more objective functions, the multi-objectives optimization approaches will be used. In particular, the weighted aggregation method is applied in this paper, since this method is the simplest one (Zitzler and Thiele, 1999). According to this approach, all the objectives are summed to a weighted combination  $F = \sum_{i=1}^p \omega_i f_i(x)$ , where the weights are non-negative weights. It is usually assumed that  $\sum_{i=1}^p \omega_i = 1$ . In this paper, we used one feature assessment function (NPOM) and a cost function. The accumulative expression is

$$K = w_1 \times Q_{ov} + w_2 \times \left(1 - \frac{C_S}{C_T}\right) \quad (2)$$

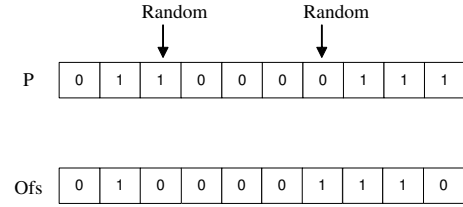


Fig. 5. Mutation.

where  $w_i$ : weight of objective functions,  $C_S$ : sum of selected features cost, and  $C_T$ : sum of total feature cost. The weight values used in all of our experiments were 0.6 and 0.4.

## 4 Optimization methods

### 4.1 Genetic algorithm

Genetic algorithm or GA is a class of optimization procedures inspired by the biology mechanisms of reproduction. GA operates iteratively on a population of structures, each one of which represents a candidate solution to the problem at hand, properly encoded as a string of symbols (e.g. binary). A random generated set of such strings forms the initial population from which the GA starts its search. Three basic genetic operators guide this search, i.e. selection, crossover, and mutation (Goldberg, 1989). The candidate solution or chromosome is a bit string whose length is determined by the number of features. The fitness function is used according to Eq. (1) for optimizing the AFS or Eq. (2) for optimizing the AFSC. The main algorithm is given below:

1. *Initialization*: Generate an initial population using a random mechanism. Denote the population as  $P_i = \{p_1, \dots, p_N\}$ , where  $N$  is the population size and  $p = \{f_1, \dots, f_n\}$ , where  $n$  is feature size.
2. *Selection*: Roulette wheel selection is used to select candidates into the mating pool and the selection probability of each individual is based on its proportional fitness value. Selected candidates represent an intermediate population,  $P_s$ , that has same size with current population.
3. *Crossover*: Here, one-point crossover takes place for search process and information exchange between two chromosomes from parent population. The parent chromosomes are split at a common point chosen randomly and the resulting sub-chromosomes are swapped (Fig. 4). This generates a next intermediate population  $P_r$  from  $P_s$ , where each two individuals take the crossover process and produce two offsprings. The crossover probability used in all of our experiments was 0.85.

4. *Mutation*: After the crossover process, the mutation generates the next intermediate population  $P_m$ . Each individual from  $P_r$  produces an offspring. Here, we use the traditional mutation operator which flips a specific bit with a very low probability (Fig. 5). The mutation probability used in all of our experiments was 0.01.
5. *Reproduction*: Offsprings from  $P_m$  will be selected into the next generation. The best 10% individuals of parent will be taken and competed with offsprings based on their fitness values. If the best parent is better than the offspring, then the best parent will be reproduced for next generation  $P_{i+1}$  and otherwise, they will be discarded.
6. *Termination condition*: The termination condition of this algorithm is referred to a pre-specified number of generation and criterion of fitness value. The algorithm will stop, if one of these two conditions are reached; otherwise go to step 2.

## 4.2 Binary particle swarm optimization

Particle swarm optimization (PSO) is a non-linear method which also is affiliated to evolutionary computation techniques. This method is a relatively new optimization paradigm introduced by Kennedy and Eberhart (1995). Particle swarms explore the search space through a population of particles, which adapt by returning to previously successful regions. The particles then fly over the state space, remembering the best solution encountered. The fitness function is determined by an application-specific objective function (here, Eq. (1) or (2)). During each iteration, the velocity of each particle is adjusted based on its momentum and influence of the best solutions encountered by itself and its neighbors. The particles then move to a new position, and the process is repeated for a prescribed number of iterations. In the original PSO implementation, the trajectory of each particle is governed by the equations:

$$\begin{aligned} \mathbf{v}_i(t+1) = \omega \mathbf{v}_i(t) + c_1 \cdot \text{rand}() \cdot (\mathbf{p}_i - \mathbf{x}_i(t)) \\ + c_2 \cdot \text{rand}() \cdot (\mathbf{p}_g - \mathbf{x}_i(t)) \end{aligned} \quad (3)$$

and

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (4)$$

where  $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  and  $\mathbf{v}_i$  are the current vector position and velocity of the  $i$ -th particle,  $\mathbf{p}_i$  is the position of the best state visited by the  $i$ -th particle,  $\mathbf{p}_g$  is the particle with the best fitness in the neighborhood of  $i$ , and  $t$  is the iteration number. The parameter  $c_1$  and  $c_2$  are called the cognitive and social learning rates. The parameter  $\omega$  is an inertia weight, which used to dampen the velocity during the course of the simulation, and allow the swarm to converge with greater precision. The parameter values of  $\omega$ ,  $c_1$ , and  $c_2$  used in all of our experiments were 1, 2, and 2, respectively.

The original PSO technique is designed for the real-value problems, whereas the feature selection only uses one or zero value to represent whether one feature is selected or not. Therefore, the algorithm now has been extended to tackle binary/discrete problems. Kennedy and Eberhart (1997) have proposed binary PSO (BPSO), where uses velocity as a probability to determine whether the components of  $\mathbf{x}_i$  (a bit) will be in one state or zero. They squashed  $v_i$  using a logistic function  $s(v) = 1/(1 + \exp(-v_i))$  while the velocity is calculated using the same equation in Eq. (3). If a randomly generated number within  $[0, 1]$  is less than  $s(v_{id})$ , then  $x_{id}$  is set to be 1, otherwise it set to be 0.

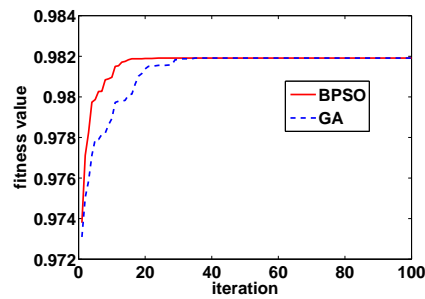
## 5 Experiments and results

In our experiments, we investigated our approach of the automatic feature selection with acquisition cost using two type of data sets: iris data and eye image data. Each dataset is split into two groups: training and test. The maximum iteration or generation is bounded as much as 100 for two optimization algorithms (genetic algorithm and binary particle swarm optimization). The parameters from each optimization described in Sect. 4 will be set same for both datasets during the experiments. For a population in this evolutionary optimization, we used 30 individuals for both GA and BPSO.

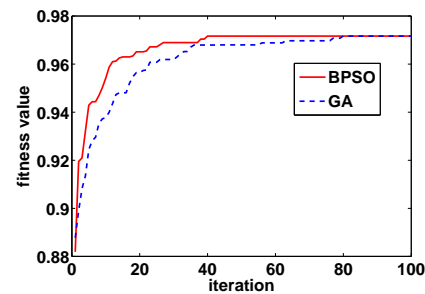
*Iris data*: This dataset is well-known and consists of 4 features and 150 patterns. The features are petal and sepal width of the iris flowers, and the three classes correspond to species setosa, virginia and versicolor. This iris data is separated in training and test dataset with 75 and 75 patterns. It has been known that the best feature subset of iris data consists of two features, i.e. the third and fourth feature. Since this data has small features, the whole iris data is repeated four times with different arbitrary cost assignment per feature. The purpose of repeating this data is to observe that our approach can find the same good features but least expensive cost. The following two sets of arbitrary cost assignments are employed:

1. cost I : {2, 3, 14, 16, 10, 8, 4, 15, 6, 5, 20, 3, 3, 12, 18, 17}.
2. cost II: {4, 1, 20, 18, 3, 4, 17, 20, 1, 2, 15, 15, 2, 3, 18, 22}.

*Eye image data*: Eye data originated from several pictures and scenes of persons in frontal view position (Koenig et al., 2002). From these grey value images, 17 by 17 pixel regions for eye and non-eye patterns were extracted. From these image blocks 12 dimensional Gabor jets, 13 dimensional the extension of the local autocorrelation (ELAC), and 33 dimensional local-orientation-coding (LOC) were computed. So eye data consist of 58 dimensional feature data with two classes for eye and non-eye patterns. This eye data is separated in training and test data set with 72 and 61 patterns. The Gabor jets were computed from four different orientations and three different frequencies of the Gabor filter. In



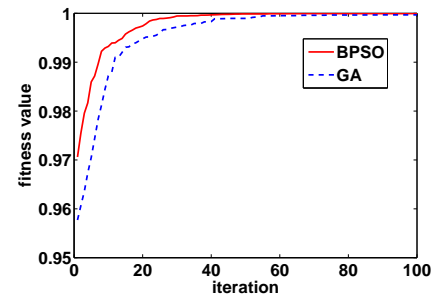
**Fig. 6.** The average fitness curves of AFS applying BPSO and GA for iris train dataset.



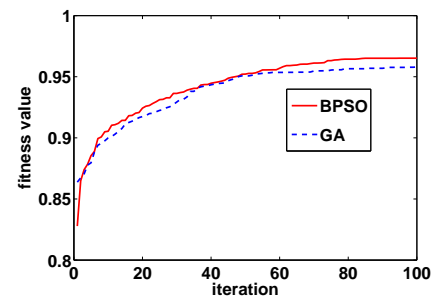
**Fig. 7.** The average fitness curves of AFSC applying BPSO and GA for iris train dataset with cost assignment I.

the ELAC, each pixel products with first order neighbors is computed according to masking information. For instance, left and right neighbors will be multiplied with the center pixel for a horizontal mask. From the number of possible masks only 13 significant masks have been chosen here. At each pixel position of the grey value image blocks, all masks will be applied and in a competitive scheme, the strongest response will determine the winning mask. A histogram of all masks keeps track of the frequency of occurrence of mask activation. So the bins of this histogram serve as the features of the ELAC operator. The LOC operator also work in computation with the first order neighborhood. In contrast to ELAC for investigated mask patterns, no multiplications take place, but pixel differences are computed to determine monotonicity. Overall 33 mask patterns are computed at each pixel position. Again, the histogram is computed for the whole region. The cost assignment for the three feature computation operators is determined with regard to the number of multiplication and addition operations. The assuming multiplication has the cost of 10 additions. Therefore, the cost of each feature for Gabor filter is 6358, for ELAC 3179 per feature, and 1445 per feature for LOC.

In our experiments, the training datasets are used to find the best feature subset using the AFS optimized by GA and BPSO. Then, the classification models are trained with regard to the selected features and tested by using the test



**Fig. 8.** The average fitness curves of AFS applying BPSO and GA for eye train dataset.



**Fig. 9.** The average fitness curves of AFSC applying BPSO and GA for eye train dataset.

datasets. Here, we used k-nearest neighbor (kNN), reduced nearest neighbor (RNN), and backpropagation neural networks (NN) for our classification models. The number of nearest neighbors in kNN is set to  $k = 9$  and the Euclidean distance is employed. More extensive experiments and simulations have shown, that for the given data the choice of  $k$  and the metric has impact on the results. The reduced-nearest neighbors (RNN) using Euclidean distance, and backpropagation neural networks (NN) with one hidden-layer in a  $x-4-3$  network topology for iris data and  $x-4-2$  network topology for eye data. The Levenberg-Marquardt method has been employed for learning.

In the first experiments, the AFS and AFSC optimized by BPSO and GA are investigated for the training enlarged iris dataset with applied 10 different initializations and the results are compared. Figures 6 and 7 show that the average curves of BPSO can converge faster than GA. Both BPSO and GA in optimizing AFS obtained equally the best feature subset. The features 3, 4, 8, 12, and 16 for the enlarged iris data are chosen by AFS. The features 7 and 12 for cost assignment I are found by the AFSC. For cost assignment II the features 2 and 12 are selected. The recognition rate results for three classification models are summarized in Table 1.

The last experiments, similar to the experiments with iris data, we extended the investigation of AFS and AFSC to more realistic eye image data. Figures 8 and 9 show that

**Table 1.** The best run results for iris data. Both BPSO and GA achieved same results.

Iris	cost I/II	selected features	k-NN (%)	RNN (%)	NN (%)
without AFS	156/165	16/16	94.67	93.33	93.33
AFS	65/95	5/16	94.67	96.00	94.67
AFSC (I)	7/-	2/16	94.67	97.33	97.33
AFSC (II)	-/16	2/16	94.67	93.33	94.67

BPSO can achieve better results and converge faster than GA. The best feature subset of 10 runs obtained by AFS using GA is five features from Gabor jets (2, 5, 8, 9, 11, 12), four features from ELAC (14, 17, 20, 21), and six features from LOC (48, 51, 53, 54, 56, 57). By applying BPSO, the selected features are three features from Gabor (10, 11, 12), three features from ELAC (17, 21, 23), and 12 features from LOC (29, 32, 38, 39, 47, 48, 49, 50, 51, 53, 54, 57). The AFSC employing both BPSO and GA can select low cost and more less number of features, i.e. only 6 of 58 features. The features 17, 21, 22, 35, 38 and 58 were obtained by using GA and the features 21, 27, 29, 43, 46, and 54 were selected by using BPSO, where the selected features are lower cost features than GA. The recognition rate results for three classification models are summarized in Table 2.

## 6 Conclusions

The design of sensor systems for recognition tasks can be optimized by eliminating several processing methods and/or sensors that are related to irrelevant features. The goal of this paper was to investigate and show the effectiveness of the automated feature selection with acquisition cost based on aggregation method, in particular employing and comparing two techniques from evolutionary computation, i.e. genetic algorithm and binary particle swarm optimization. The experiments, based on benchmark data (iris data) and more realistic eye image data, gave evidence that the achieved results of AFSC employing binary particle swarm are more robust and competitive with regard to speed, classification accuracy and low cost of selected features than GA to effectively optimize the design of recognition systems. In this paper, our approach only considered to optimize the recognition system according to the cost per feature.

In future work, the extension of the cost and feature grouping (Paclík et al., 2002) in the selection process and the inclusion of several additional assessment functions in a multi-objective optimization based on pareto approach (Zitzler and Thiele, 1999) with regard to increasing accuracy and reliability will be subject of investigation. In addition to our investigations of the automatic feature selection with acquisition cost using GA and BPSO in real applications, the op-

**Table 2.** The best run results for eye data.

Eye image	cost	selected features	k-NN (%)	RNN (%)	NN (%)
without AFS	165308	58/58	96.72	80.33	97.87
AFS GA	53176	16/58	98.36	95.08	96.81
AFSC GA	13872	6/58	96.72	95.08	96.81
AFS BPSO	45951	18/58	100	95.08	98.94
AFSC BPSO	10404	6/58	96.72	98.36	98.94

timization of multi sensor system in particular to gas sensing (Iswandy et al. , 2005) will also be considered.

## References

- Agrafiotis, D. K. and Cedeno, W. J.: Feature Selection for Structure - Activity Correlation Using Binary Particle Swarms, *J. Med. Chem.*, 45, 1098–1107, 2002
- Aha, D. W.: Feature Weighting for Lazy Learning Algorithms, edited by: Liu and Motoda, Kluwer Acad. Publ., 13–29, 1998.
- Goldberg, D. E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Comp. Inc., 1989.
- Iswandy, K., Koenig, A., Fricke, T., Baumbach, M., and Schuetze, A.: Towards Automated Configuration of Multi-Sensor Systems Using Evolutionary Computation - A Method and a Case Study, *J. Comput. and Theor. Nanoscience*, 2, 574–582, 2005.
- Jain, A. and Zongker, D.: Feature Selection: Evaluation, application, and small sample performance, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19, 153–158, 1997.
- Jain, A., Duin, R. and Mao, J.: Statistical Pattern Recognition: a Review, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22, 4–37, 2000.
- Kennedy, J. and Eberhart, R. C.: Particle Swarm Optimization, in *Proc. of IEEE Int. Conf. on Neural Networks (ICNN)*, 4, 1942–1948, 1995.
- Kennedy, J. and Eberhart, R. C.: A Discrete Binary Version of The Particle Swarm Algorithm, *Proc. of Conf. on System, Man, and Cybernetics*, 4104–4109, 1997.
- Koenig, A.: A Novel Supervised Dimensionality Reduction Technique by Feature Weighting for Improved Neural Network Classifier Learning and Generalization, in *Proc. of the 6th Int. Conf. on Soft-Computing and Information/Intelligent Systems IIZUKA2000*, Iizuka, Fukuoka, Japan, October, 746–753, 2000.
- Koenig, A.: Dimensionality Reduction Techniques for Interactive Visualization, Exploratory Data Analysis, and Classification, in: *Pattern Recognition in Soft Computing Paradigm*, World Scientific, FLSI Soft Comp. Series, edited by Pal, N. R., 2, 1–37, 2001.
- Koenig, A. and Gratz, A.: Advanced Methods for the Analysis of Semiconductor Manufacturing Process Data, in: *Advanced Techniques in Knowledge Discovery and Data Mining*, edited by Pal, N. R. and Jain, L. C., Springer Verlag, 27–74, 2005.
- Koenig, A., Mayr, C., Bormann, T. and Klug, C.: Dedicated Implementation of Embedded Vision Systems Employing Low-Power Massively Parallel Feature Computation, in: *Proc. of the 3rd*

- VIVA-Workshop on Low-Power Information Processing, Chemnitz, Germany, 1–8, 2002.
- Kohavi, R. and John, G. H.: The Wrapper Approach, in: Feature Extraction, Construction and Selection, edited by Liu and Motoda, Kluwer Academic Publishers, 33–50, 1998.
- Langley, P.: Elements of Machine Learning, Morgan Kaufmann Publishers, 1996.
- Mao, K. Z.: Fast Orthogonal Forward Selection, IEEE Trans. on Neural Networks, 13, 1218–1224, 2002.
- Paclík, P., Duin, R. P. W., van Kempen, G. M. P., and Kohlus, R.: On Feature Selection with Measurement Cost and Grouped Features, edited by Caelli T. et al.: SSPR and SPR 2002, LNCS 2396, Springer-Verlag, 461–469, 2002.
- Siedlecki, W. and Sklansky, J.: On Automatic Feature Selection, Int. J. Pattern Recognition, 2, 197–220, 1988.
- Siedlecki, W. and Sklansky, J.: A Note on Genetic Algorithm for Large Scale Feature Selection, Pattern Recognition Lett., 10, 335–347, 1989.
- Yang, J. and Honavar, V.: Feature subset selection using a genetic algorithm, in: Feature Extraction, Construction and Selection, edited by Liu and Motoda, Kluwer Academic Publishers, 117–136, 1998.
- Zitzler, E. and Thiele, L.: Multiobjective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach, IEEE Trans. on Evolutionary Computation, 3(4), 257–271, 1999.