Selecting Small Audio Feature Sets in Music Classification by Means of Asymmetric Mutation

Bernd Bischl[2], Igor Vatolkin[1], and Mike Preuss[1]

[1] Chair of Algorithm Engineering, TU Dortmund, Germany
[2] Chair of Computational Statistics, TU Dortmund, Germany

Abstract. Classification of audio recordings is often based on audio-signal features. The number of available variables is usually very large. For successful categorization in e.g. genres, substyles or personal preferences small, but very predictive feature sets are sought. A further challenge is to solve this feature selection problem at least approximately with short run lengths to reduce the high computational load. We pursue this goal by applying asymmetric mutation operators in simple evolutionary strategies, which are further enhanced by mixing in greedy search operators. The resulting algorithm is reliably better than any of these approaches alone and in most cases clearly better than a deterministic greedy strategy.

Keywords: Evolutionary Strategies, Asymmetric Mutation, Feature Selection, Music Information Retrieval, Machine Learning.

1 Introduction: Motivation and Music Classification as Optimization Problem

Personal digital music collections have been growing rapidly during the recent years. Approaches for smart navigation through large audio libraries or recommendation techniques provide obviously needed remedies for music management. Supervised audio classification methods build models from labeled music examples and extracted features.

A large number of parameters have an impact on the classification results, e.g. for feature extraction, different feature source time frames can be selected. Larger frames allow more precise frequency resolution; however, if they are too large, several notes can be mixed and it will be harder to learn anything at all from the spectrum distribution. Feature processing optimization may select the optimal feature set or choose different preprocessing methods. On the other hand, hyperparameters or model classes of classification methods can be optimized over for achieving better performance.

Because of the nonlinear interactions between different parameters, search for optimal or sufficient solutions can profit from heuristics. Continuing our previous work [15], [16], we concentrate here on the experiments for feature selection by evolutionary strategies (ES). Since very different optimal feature sets may exist for different music categories (as shown e.g. in [11]), the search for the best features must be done for each category separately. Several facts motivate the additional search for rather small feature sets: Firstly, the storage of more features requires large indexing disc space. Secondly,

R. Schaefer et al. (Eds.): PPSN XI, Part I, LNCS 6238, pp. 314–323, 2010.

the algorithms for extraction, processing and classification need more computing time. Moreover, classification models built from larger feature sets increase the danger to overfit the currently used song sets, leading to a much weaker performance in general. Introducing a bias towards less features can be seen as a regularization method to produce more stable solutions. Also, smaller sets provide a better possibility to interpret the less complex classification model.

A simple (1+1)-ES [1] already obtains a good result, however it often selects relatively large feature sets, and is sometimes beaten by a simple greedy strategy. Obviously, we have a two-criteria problem which could be approached by a multi-criterial evolutionary algorithm (EA). We abstain from doing so because the reasonably available amount of function evaluations is very low (in the order of a few hundred). In this work, we suggest to encode the need for small datasets directly into the mutation operator via asymmetric mutation. Moreover, we show that importing parts of the greedy strategy into the ES further improves performance, leading to a reliable and well performing method that by design choses only small feature sets.

2 Music Genre Classification and Feature Selection

2.1 Genre Classification

In music classification, generally a set of raw features are extracted from segments of a song, which describe different characteristics like timbre, harmony, melody, rhythm or time and structural properties (for details see [16]). Using these covariates and some given labels, which specify the genre of the song, a model is built by using one of the many classification algorithms from machine learning. Models are evaluated by applying them to new data, which were not used during training and are assumed to be i.i.d. drawn from the same data generating process, and their predictions are measured by applying an appropriate loss function. Often this is zero-one loss for classification, but due to the segmentation of songs into multiple parts we use a mean squared error on song level

$$E^2 = \frac{1}{L} \sum_{i=1}^{L} \left(\hat{s_i} - s_i\right)^2 . \tag{1}$$

Here, L is the number of songs and the s_i are their true labels. As we only consider binary categories, these will always be from $\{0, 1\}$.

$$\hat{s_i} = \frac{1}{P} \sum_{j=1}^{P} \hat{g}(x_{ij}) \tag{2}$$

is a "voting score", obtained by applying the fitted learning machine $\hat{g}$ to all partitions of a song and averaging the predictions. The x_{ij} are the feature vectors corresponding to the P partitions of song s_i.

It is usually unknown which learning algorithm performs best for a task at hand, therefore extensive model selection (e.g. by cross-validation) among the different classes of inducers has to be performed. As we want to focus on the feature selection algorithms

and maintain comparability to our previous work, we will only consider decision trees though. Further, the CART methodology [3] is very fast, does not require tuning of many hyperparameters, and performs well in comparison to many other data mining methods.

2.2 Feature Selection

There are at least two (somewhat contradictory) viewpoints of feature selection in machine learning: On the one hand, it is well known that in the setting of many noisy, highly correlated and possibly irrelevant covariates the predictive power of a model fitted on the whole feature set will be suboptimal [4]. Thus, feature selection might be employed mainly as a method to improve generalization performance. On the other hand, one could select variables to construct smaller, and therefore more interpretable models, if one is mainly interested in understanding the data. In this case one might even accept a substantial loss in predictive performance to achieve a smaller model. We will follow the former perspective here.

One of the most popular class of feature selection algorithms - because of their general applicability and strength to build very predictive models - are wrappers [7]. These algorithms internally use a learning algorithm as a black-box and search for an optimal set of input features w.r.t. the learner by repeatedly adjusting the variable set, fitting a model and evaluating it. This basically reduces the feature selection problem to a discrete, binary optimization problem, with a possibly noisy target function, as the target value can vary because of either the stochastic nature of the resampled training and test sets or because of a stochastic model fitting scheme. For an overview of competing feature selection techniques,like filters and embedded methods, see [4].

2.3 Categories and Feature Sets

For the following music classification experiments we created an mp3 database with 120 commercial albums, whose songs are labeled as Classic, Pop/Rock, Rap, Electronic, R&B, ClubDance and Heavy Metal by AllMusicGuide[1]. The relations between songs and categories were manually created by the AMG music experts. As demonstrated in [11], different features might be relevant to identify each category, so we transformed the multi-class problem into seven binary ones by “one-vs-rest”.

For the training of classification models we employ two audio feature sets: An older one with 198 features (most of them described in [14]) and a newer one consisting of 572, which were extended by MIR Toolbox functions (see [8]) like tonal centroids, fluctuation patterns, etc. It is possible to incorporate other feature groups (playlists and tags from the web community, metadata etc.), however they are not always available or can be erroneous - only audio signal based features guarantee their availability for the automatic extraction for each music piece from any personal music collection.

In contrast to the experiments in [16], the dimensions of the multidimensional features, like twelve chroma vector values, are treated as independent variables, so that it is e.g. possible to use only the first and third chroma characteristics, discarding all

[1] www.allmusic.com, visited in April 2010.

others. This allows more flexible feature selection while increasing the complexity of the optimization problem.

3 Search Strategies

For large feature spaces the wrapper approach is computationally expensive and its performance obviously depends on the efficiency of the used search algorithm. It should be mentioned though, that more complex search methods as well as very long runs or even exhaustive searches do not necessarily equal superior results in generalization performance, as overfitting might occur on this level as it possibly does on the lower level of model fitting itself [9].

As we are mainly interested in the properties of the optimization techniques itself, we don't employ a more appropriate, but even more time-consuming setup like nested cross-validation. Instead use 20 songs in the training set (each with roughly 100 segments) as prototypes for the categories to be learned, and a rather large optimization set of 120 songs for feature set evaluation to avoid overfitting through extensive search. We also regularize the search by only allowing a rather low number of 500 function evaluations [10].

3.1 Monte Carlo Search (MC / Random)

As a baseline comparison we use a random search which simply draws bit vectors from a binomial distribution, evaluates the corresponding feature sets w.r.t. the optimization set and selects the one with the minimal E^2. Therefore, feature f_i occurs with probability 0.5 in every random feature set and on average the random feature sets are half as large as the full set.

3.2 Greedy Forward Search with Correlation Heuristic (GFS)

Because the commonly used sequential forward selection cannot be used due to the restricted number of function evaluations, we construct a variant which rapidly moves through the variables in a greedy, heuristically guided order: On the training set we rank the features by considering the absolute empirical correlation $|\rho(\mathbf{f}_i, \mathbf{y})|$ between the feature vector $\mathbf{f}_i$ and the binary label vector $\mathbf{y}$ of the song segments. Now, starting from the empty set, we successively try to add variables to the current set in the order of their ranking. If a variable improves the performance on the optimization set, it is accepted, otherwise not and the following variable in the ranking is tried.

This technique is sometimes called Rank-Search and variants of it are proposed in [5,12].

3.3 Evolutionary Strategy with Local Operators (ES-LO)

We consider the (1+1)-ES with local, hybrid operators to select more uncorrelated feature sets as introduced in [16]. To obtain smaller feature sets, we enhance it by an asymmetric mutation operator as in [6], using different mutation probabilities for set

and unset bits. Let the current feature set be a binary vector $\mathbf{m}$ with length N. The mutation is a bit flip for the ith feature with probability

$$p_m(i) = \frac{\gamma}{N}|m_i - p_{01}| \tag{3}$$

γ impacts the general mutation probability and can be interpreted as a step size. p_{01} controls the balance between the number of 1- and 0-entries in the bit feature vector [6]. If e.g. only five percent of ones are desired, p_{01} is set to 0.05. It can be seen as a probability to switch a bit on, if it is currently not in the solution, and $1 - p_{01}$ as a probability to switch it off, if it is.

After the successful mutations we apply a neighborhood search. Here the mean empirical correlation between a feature vector $\mathbf{f}$ and the M features currently in the set is calculated:

$$R(\mathbf{f}) = \frac{1}{M}\sum_{j=1}^{M}|\rho(\mathbf{f}, \mathbf{f}_j)| \,. \tag{4}$$

Briefly speaking, the local search operator LO^+ adds a new feature with the smallest $R(\mathbf{f})$, trying to extend the feature set with another covariate which is least correlated to the current. LO^- removes the feature with the highest $R(\mathbf{f})$, removing the covariate with the highest correlation with the current features.

3.4 Evolutionary Strategy (ES)

This is the same search algorithm as the previous ES-LO, simply without the local operators. We consider this algorithm in order to analyze whether the previously mentioned local operators really improve the efficiency of the search.

3.5 Evolutionary Strategy with Success Rule Adaptation (ES-SRA)

One of the early attempts to adapt the mutation strength of an ES was the 1/5 success rule, with increasing mutation strength for more than 1/5 successes, and decreasing mutation strength for less than 1/5 successes [18]. However, this method was envisioned for real-valued search spaces and as Schwefel has shown at the same time, does not work reliably for discretized search spaces [13]. The reason for this failure is that the problem must provide maximal success rates for minimal steps, which is not generally the case already for integer variables, let alone for bit vectors. We assume that it is also not the case for our problem and thus use a different rule that increases the mutation probability $p_m(i)$ by a factor of 1.2 if less than 1/10 successes are recorded and decrease it by the same factor if more than 3/10 successes are recorded. As the number of function evaluations for one run is very limited, we employ a small window size of 10 for measuring success rates. This method may be interpreted as 'controlled restart' to get out of an area where Hamming-1 distance steps are very rarely successful. Interestingly, our scheme has a predecessor in [19], where for minimal success rates the mutation strength is doubled.

3.6 Evolutionary Strategy with Greedy Heuristic (ES-GH)

This is the same search algorithm as the plain ES, but with a modified mutation operator. As section 4 will show, GFS performs very well on some data sets. Therefore we try to encode its greedy heuristic into the mutation operator of the ES:

$$p_m(i) = \frac{\gamma}{N} |m_i - p_{01}| \cdot |m_i - |\rho(\mathbf{f_i}, \mathbf{y})|| \tag{5}$$

The second factor in $p_m(i)$ forms a correlation-based probability to switch bits on with higher probability, which are highly correlated with the target, and to switch bits off which are not. We simply multiply the two probabilities from the mutation operator of the normal ES and the correlation-based heuristic to form a combined probability and incorporate both advantages. Section 4 also shows, that SRA is essential for the success of the optimization, therefore we use it here as well to adapt the step size γ.

4 Experimental Assessment of Search Strategies

We are most interested in assessing the performance of the local operators (LO), the success-rule adaptation (SRA) and the greedy-heuristic (GH) enhanced ES variants against the simple Monte Carlo search and the Greedy Forward Search (GFS). This is tested in two steps, first LO and SRA, and the most successful variant is then tried with and without GH enhancement. Possible parameter variations and representation issues are either used throughout the following experiments (asymmetry of mutations) or dealt with in the pre-experimental planning phase (mutation strength and base feature set). The overall goal of our experiments is to compose the best achievable ES variant and test if it is reliably better than MC and GFS.

For our experiments we used the statistical programming language *R* [17] and the package *mlr* [2], which allows to select from a wide range of machine learning, variable selection and hyperparameter tuning methods.

Experiment: Do local operators and/or success rule adaptation of mutation strenghts improve ES performance?

Pre-experimental planning. First experimentation showed that for ES, ES-LO and ES-SRA the mutation probability parameter γ can be set to 16 for the old and new feature sets. Comparing the MSE values attained for these two representations leads to the conclusion that the new, larger feature sets most often allow finding better classifiers. In the following, we therefore fix γ at 16 and employ only the new feature sets. Moreover, the run lengths are fixed to 500 function evaluations as a good compromise between achievable performance and consumed real time (around 2h for one run on a modern PC). For the same reason, we perform only 5 repeats.

Task. A method is recognized as better than another, if its average final MSE is better in at least 4 of the 7 categories.

Setup. We test 3 ES variants against each other, ES-NLO-SRA (ES without local operators but with success rule adaptation), ES-LO-NSRA (ES-LO without SRA) and ES-LO-SRA (both switched on), each time with asymmetric bit flip probability set to

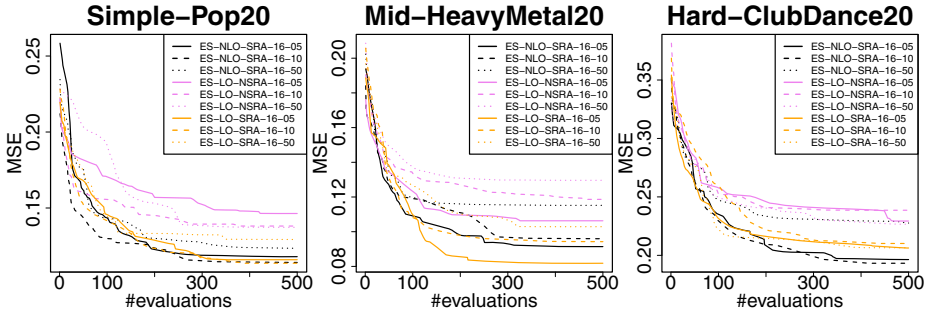

Fig. 1. Comparison of different ES variants. Curves are averaged over 5 runs.

$p_{01} \in \{0.05; 0.10; 0.50\}$. All runs are done over all 7 categories. Note that $p_{01} = 0.5$ means symmetric mutation.

Results/Visualization. The average performance of all variants is plotted in fig. 1 for 3 of the 7 categories[2]. The best recorded variant of our comparison is ES-NLO-SRA-16-05 (no local operators, success rule adaptation, $p_{01} = 0.05$, by winning 4 categories and being placed second, otherwise.

Observations. In general (over all categories), we can state that the local operators do not increase performance but sometimes decrease it. However, the success rule adaptation is obviously necessary to reach good MSE values. Concerning the mutation probabilities, the results get the better, the stronger the deviation from symmetry is.

Discussion. It is no surprise that the asymmetric mutation improves performance as sparsely selecting alternative algorithms (e.g. greedy feature selection) also cope well with the treated problems, meaning that there must be good small feature sets.

The failure of the local operators can be explained by considering the extreme case of completely irrelevant, random features. This will neither be removed by LO^-, as they are completely uncorrelated to all features currently in the set, and they will constantly be proposed to be added by LO^+ for exactly the same reason. We also verified this undesirable effect on synthetic data sets, were the truly relevant features are known.

However, it is more difficult to explain why the success rate adaptation does so well. We therefore visualize the current feature set and its Hamming-1 distance throughout a typical run of ES-NLO-SRA-16-05 in fig. 2. It gets clear that in higher generations, most 1-bit flips are neutral, and there are only very few 1-bit mutations left that lead to an improvement. In such situations, it makes sense to increase the mutation probability to flip several bits at once.

Experiment: Is a greedy-heuristic enhanced ES reliably better than GFS/MC?

Pre-experimental planning. In initial tests, we try several ES-GH parameterizations and set the p_{01} again to 0.05 and $\gamma = 32$ as the flip probabilities are reduced by the multiplication in (5).

[2] Complete results and plots are available as supplementary material from `http://www.statistik.uni-dortmund.de/~bischl/ppsn2010_suppl`

Task. As before, an algorithm is termed better than another one if it is better on average in at least 4 categories.

Setup. We run the best method of experiment 1 (ES-NLO-SRA-16-05) against the full featured classifier, random search, greedy forward search and ES-GH (with SRA), again over all 7 categories.

Results/Visualization. Figure 3 reports the (averaged where applicable) performance of all algorithms on 3 of 7 categories.

Observations. The greedy-enhanced ES-GH performs consistently better than ES-NLO-SRA-16-05, and is able to achieve better MSE values than GFS in 5 of 7 cases. MC and the full featured classifier are much worse in all cases. Over all categories, the ES-GH seems to be much more robust than GFS which sometimes fails dramatically. Even for the 2 lost categories, ES-GH achieves acceptable results.

Discussion. Importing a greedy mechanism into the previously best ES variant obviously leads to the envisioned effect: We can combine the best of both methods by obtaining a reliable algorithm that always performs well and is in most cases better than all others we tried.

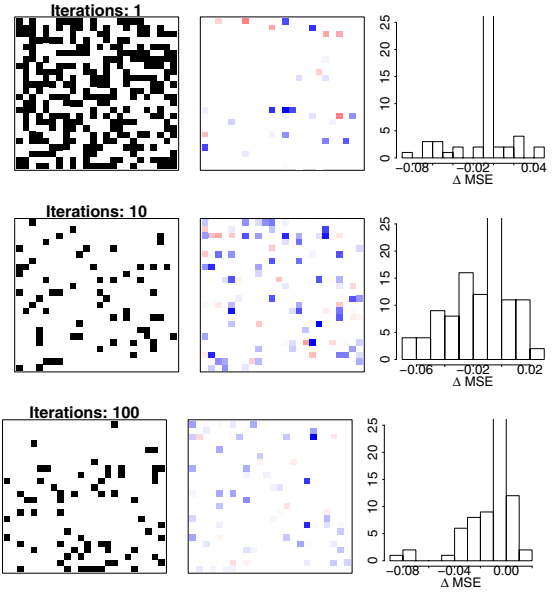

Fig. 2. Analysis of Hamming-1-neighborhood for a run of ES-NLO-SRA-16-05, logged at different numbers of function evaluation during the run. The left pattern shows the variables currently in the feature set, a black square means the bit is switched on. The middle plot is produced by flipping each bit separately and measuring the difference in MSE to the current solution. A positive value means improvement. A color gradient from white to red is used for improvement, and from white to blue if flipping the bit worsens the solution. The right plot simply shows a histogram of the differences in MSE from the middle plot, capped at 25.

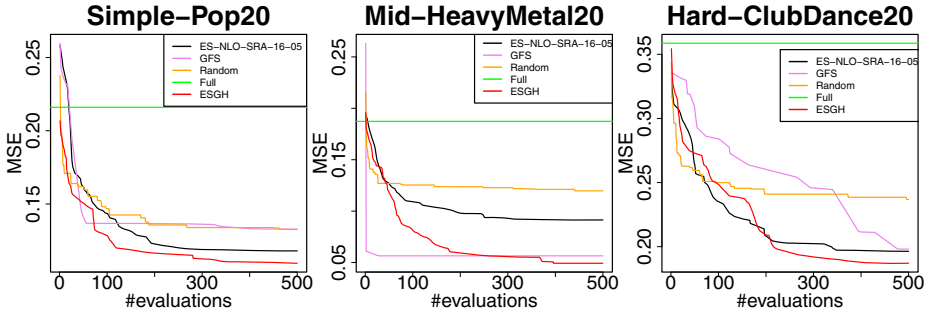

Fig. 3. Comparison of best ES variant of experiment 1 with baseline methods and improved ES. Curves are averaged over 5 runs.

5 Conclusions and Outlook

In this work we continued the design of ES for feature selection in music classification. We have seen that enforcing sparse selection of features by means of an asymmetric mutation operator produces competitive results after the optimization. This leads to a reduced number of features which must be computed and stored and improves the generalization performance.

We also demonstrate by experiment, that previously proposed local operators do not lead to a better algorithm, but instead might deteriorate the performance, and we also provide theoretical aspects to explain this result. A rather simple greedy Rank-Search is presented, which sometimes achieves quite impressive performance results by selecting extremely small feature sets. But this is not a reliable behavior across all data sets, so we show how to combine both advantages of the ES-LO with success rate adaptation and the GFS into a reliably well performing method.

It seems quite clear, that the full potential to include heuristics for feature selection into the stochastic optimization is yet to be explored. If one can allow for only a low or moderate number of function evaluations, guiding the search into the relevant areas of the search space quickly - even by crude measures - will be crucial for success.

Acknowledgments

We thank the Klaus Tschira Foundation for the financial support. Due to the computationally intensive nature of this case study, the majority of the calculations were performed on the LiDO HPC cluster at the TU Dortmund.

References

1. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies – A comprehensive introduction. Natural Computing: an International Journal 1, 3–52 (2002)
2. Bischl, B.: The mlr Package: Machine Learning in R (2010), `http://mlr.r-forge.r-project.org`

3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
4. Guyon, I.: An Introduction to Variable and Feature Selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
5. Hall, M.A., Holmes, G.: Benchmarking attribute selection techniques for discrete class data mining. IEEE Transactions on Knowledge and Data Engineering 15(6), 1437–1447 (2003)
6. Jelasity, M., Preuß, M., Eiben, A.E.: Operator Learning for a Problem Class in a Distributed Peer-to-peer Environment. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 172–183. Springer, Heidelberg (2002)
7. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
8. Lartillot, O., Toiviainen, P.: MIR in Matlab (II): A Toolbox for Musical Feature Extraction From Audio. In: Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR), pp. 127–130 (2007)
9. Loughrey, J., Cunningham, P.: Overfitting in Wrapper-Based Feature Subset Selection: The Harder You Try the Worse it Gets. In: Research and Development in Intelligent Systems XXI, pp. 33–43 (2005)
10. Loughrey, J., Cunningham, P.: Early-Stopping to Avoid Overfitting in Wrapper-Based Feature Selection Employing Stochastic Search. Computer Science Technical Report, Trinity College Dublin, TCD-CS-2005-37 (2005)
11. Pohle, T., Pampalk, E., Widmer, G.: Evaluation of Frequently Used Audio Features for Classification of Music into Perceptual Categories. In: Fourth International Workshop on Content-Based Multimedia Indexing (2005)
12. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recognition 39(12), 2383–2392 (2006)
13. Schwefel, H.-P.: Cybernetic Evolution as Strategy for Experimental Research in Fluid Mechanics. Diploma Thesis, Hermann Föttinger-Institute for Fluid Mechanics, Technical University of Berlin (1965) (in German)
14. Theimer, W., Vatolkin, I., Eronen, A.: Definitions of Audio Features for Music Content Description, Algorithm Engineering Report TR08-2-001, Technical University Dortmund (2008)
15. Vatolkin, I., Theimer, W.: Optimization of Feature Processing Chain in Music Classification by Evolution Strategies. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1150–1159. Springer, Heidelberg (2008)
16. Vatolkin, I., Theimer, W., Rudolph, G.: Design and Comparison of Different Evolution Strategies for Feature Selection and Consolidation in Music Classification. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation (CEC 2009). IEEE Press, Piscataway (2009)
17. R Development Core Team, R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2010), ISBN 3-900051-07-0, http://www.R-project.org
18. Rechenberg, I.: Cybernetic Solution Path of an Experimental Problem. In: Fogel, D.B. (ed.) Evolutionary Computation - The Fossil Record. IEEE Press, Los Alamitos (1998)
19. Greenwood, G., Zhu, Q.: Convergence in Evolutionary Programs with Self-Adaptation. Evolutionary Computation 9(2), 147–158 (2001)