Variable-Dimensional Optimization With Evolutionary Algorithms Using Fixed-Length Representations

Joachim Sprave and Susanne Rolf

Dept. of Computer Science
University of Dortmund
D–44221 Dortmund
Germany

Abstract. This paper discusses a simple representation of variable-dimensional optimization problems for evolutionary algorithms. Although it was successfully applied to the optimization of multi-layer optical coatings, it is shown that it introduces a unintentional bias into the search process with respect to the probability of a dimension being generated by mutation and recombination. In order to examine the impact of the bias, the representation was applied to another variable-dimensional problem, the simultaneous estimation of model orders and model parameters of instances of autoregressive moving average processes (ARMA). The results of the parameter study show that quality of the estimation can be improved by removing the bias.

1 Motivation

Most applications of evolutionary algorithms (EAs) cannot be easily described by a homogeneous set of real or integer numbers. Instead, they depend on problem-specific non-standard representations, such as graphs or matrices. Suitable non-standard representations are not only hard to find, but, even worse, they are difficult to analyze. This paper focuses on a special type of non-standard representations, namely variable dimensional representations.

2 Variable Dimensional Problems

In the usual parameter optimization task a fixed number of variables corresponds to the number of degrees of freedom of the system modeled by the objective function, i.e., the model is fixed with respect to the structure of the problem. As soon as structural optimization is required, the number of parameters becomes a variable itself.

While there are already some approaches to non-standard representations which are at least experimentally verified (see [5] for an overview), there are only a few approaches to variable dimensional encodings and operators. Most of them are discussed in [8].

The most obvious way to optimize variable dimensional problems with evolutionary algorithms is to use variable length chromosomes for the EA, e.g. [3, 4]. This approach requires the definition of new operators to increase or decrease the length of the internal representation. Following the paradigm of natural evolution, these operators are commonly modeled as *gene deletion* and *gene duplication*. On the other hand, new operators also introduce new problems and questions, e.g. if and how chromosomes of different lengths can be recombined, which values are appropriate for newly inserted variables, and so on. An alternative approach was introduced in [9]. Instead of a variable length representation, a fixed-length representation was chosen for a real-valued variable-dimensional problem. The representation consists of a fixed-length vector of problem variables and a binary vector containing information to reduce the number of problem variables.

In the following sections, two structural optimization problems are discussed which were successfully solved by EAs with fixed-length genomes. For the second application, a statistical parameter estimation problem, we present a detailed analysis of the influence of two different internal representations with respect to the quality of the optimization results.

2.1 Coding of the Multi-layer Optical Coatings

The design problem of multi-layer optical coatings (MOC) is a mixed-integer variable-dimensional optimization problem. The tasks of MOC design is to find a sequence of layers of a small set of materials, such that a given reflection profile of the resulting coating is achieved. For example, in order to construct an infrared filter, one has to assure that there is minimum reflection in the infrared range while there is maximum reflection all over the rest of the spectrum.

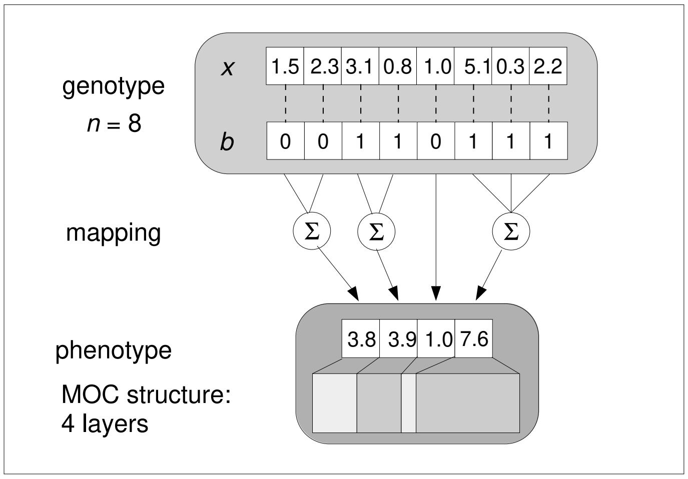

Fig. 1. Mapping from a fixed dimensional internal representation to a variable dimension MOC structure.

A first attempt to a solve the MOC problem by means of EAs is documented in [7] and [1]. The idea was to use gene deletion and duplication together with a self-adapted mutation rate for dimensional mutations.

An alternative representation was presented in [9], which works as follows: Each individual consists of a real-valued vector $x_1, \ldots, x_n$ and an integer vector $b_1, \ldots, b_n, 1 \leq b_i \leq M$, where M is the number of materials, so each layer of thickness x_i has got its material index b_i. Since both parts of the genome are standard representations, standard mutation and recombination operators can be applied accordingly. For the vector of real numbers, the evolution strategy operators as defined in [10] are used, whereas the bit strings are varied by bit flip mutation and uniform crossover.

This representation allows adjacent layers being of the same material, which can be grouped together to single layers by summing up their thicknesses, thus effectively reducing the dimension of the solution. This representation has a major advantage: even if the dimension is reduced by the genotype-phenotype-mapping, each variable is used to evaluate the fitness of the resulting genotype. This is very important with respect to internal strategy variables, e.g. the step size informations used by evolution strategies, as well as for the smoothness of structural mutations: in the MOC example, a structural mutation does not change the physical thickness of the original coating.

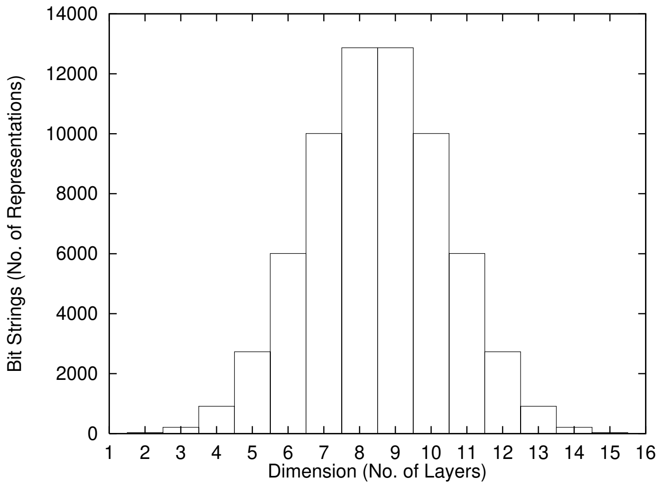

Fig. 2. Distribution of the numbers of layers implied by the MOC coding.

Although this internal representation looks straightforward, it is biased towards medium dimensional solutions. For a MOC design problem with only two materials, the structural information can be encoded by a bit string, as shown in Fig. 1. The numbers of layers implied by the coding is not equally distributed, as shown in Fig. 2. Instead, for a given number of layers k, the number of representations in a bit string of length n is

$$2\binom{n-1}{k-1}$$

As can be seen, a huge number of bit strings represents dimensions around half the given maximum.

2.2 ARMA-Modeling as an Example for Variable Dimensional Optimization

Another example for a variable dimensional optimization problem is the classical statistical approach to time series modeling: The Box-Jenkins method [2]. This method is based on the so-called ARMA-model, which is of the form

$$y_t = \alpha_1 y_{t-1} + \ldots + \alpha_p y_{t-p} + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \ldots - \beta_q \varepsilon_{t-q},$$

where y_t are the observations of the time series, ε_t are the elements of the series of random shocks, p and q are the numbers of considered elements of the two series, $\alpha_1, \ldots, \alpha_p$ and $\beta_1, \ldots, \beta_q$ are the parameter vectors.

Since the parameters as well as their lengths p and q are unknown, this model leads to a mixed-integer optimization problem with variable dimension. In a recent study [6], the simultaneous estimation of model order and parameter of ARMA processes by means of EAs was investigated. As an objective function, the least-squares-function

$$LS(y,\hat{y}) = \sum_{t=k}^{N} [y_t - (\alpha_1 y_{t-1} + \ldots + \alpha_p y_{t-p} - \beta_1 \hat{\varepsilon}_{t-1} - \ldots - \beta_q \hat{\varepsilon}_{t-q})]^2, k \geq max\{p, q\}$$

was chosen, where $y = (y_t)_{t \in \{1,\ldots,N\}}$ is the observed time series, and where $\hat{y} = (\hat{y}_t)_{t \in \{1,\ldots,N\}}$ the estimated time series based on the estimations of $\alpha_1, \ldots, \alpha_p$, $\beta_1, \ldots, \beta_q$, and the estimated error series $(\varepsilon_t)_{t \in \{1,\ldots,N\}}$.

Moreover, a slight penalty term was used to give smaller models an advantage over models of high orders. The resulting objective function F is

$$F(\hat{y}) = LS(y,\hat{y}) \cdot (1 + \frac{p+q}{100}) ,$$

of which the minimum value is sought.

The penalty term in the objective function was introduced to follow one of the principle of ARMA-modeling, which is to find models as sparse as possible. Therefore, it also seems to be reasonable to use a fixed upper limit for the lengths of the parameter vectors p and q. In this study this upper limit was set to $n = 8$. Figure 3 shows the internal representation of the variables as well as the mapping and the resulting vector for the example of the AR-vector α.

The internal representation was taken from the MOC design problem, with one modification: In order to allow zero order models, the dimension is incremented one position after a bit flip was detected.

The feasibility of this combination of statistical methods and evolutionary algorithms has been subject of an elaborate study in which ARMA-models up to the order

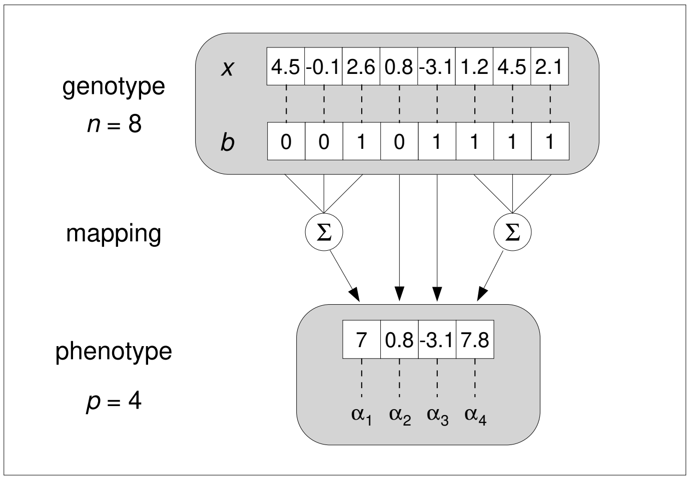

Fig. 3. Mapping from a fixed dimensional internal representation to a variable length AR parameter vector. The resulting order is $p = 4$.

of p and q equal to 3 have been generated and evaluated by this approach. The detailed methods and results of this investigation are documented in [6]. As a short summary it can be said that the optimization of the ARMA-model by means of evolutionary algorithms leads to good results in the fit compared to those reached by common statistical methods. The percentage of correctly identified model orders on the other hand is rather low with about 20%. On the other hand, there is no standard algorithm for the model identification of ARMA processes that guarantees to find the correct model. The Box-Jenkins approach, for example, depends highly on the correct choice of test levels in the model identification step.

3 Analysis of the ARMA Representation

To find out about the reasons of the rather poor model identification abilities of the EA approach to ARMA model estimation, a detailed analysis of the coding has been performed. Under the assumption that all 2^{16} binary vectors are of equal probability and that they are chosen completely by random, the distribution of the resulting model orders p and q has been computed and can be seen in Fig. 4.

The figure shows that the coding does not generate all model orders with equal probability, but prefers models close to the ARMA[4,4]-process. Since in the study only models up to ARMA[3,3] were used, this coding produces a bias into the wrong direction.

According to closer investigations on the coding, the influence of its bias depends very much on the mutation rate used. A high mutation rate (Fig. 5 right) leads to a bad performance in the model identification, the distribution of the chosen models in this case is very close to the theoretical distribution shown in Fig. 4. A low mutation rate on

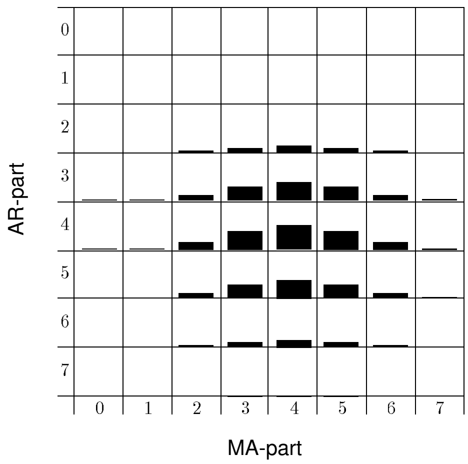

Fig. 4. Distribution of the model orders p and q as implied by the coding.

the other hand leads to quite good results as far as the model identification is concerned. Figure 5 shows the empirical distribution of the model orders in the simulation study which were gained with mutation rates of 0.05 and 0.5 respectively. The lower mutation rate also turned out to achieve a much better fit.

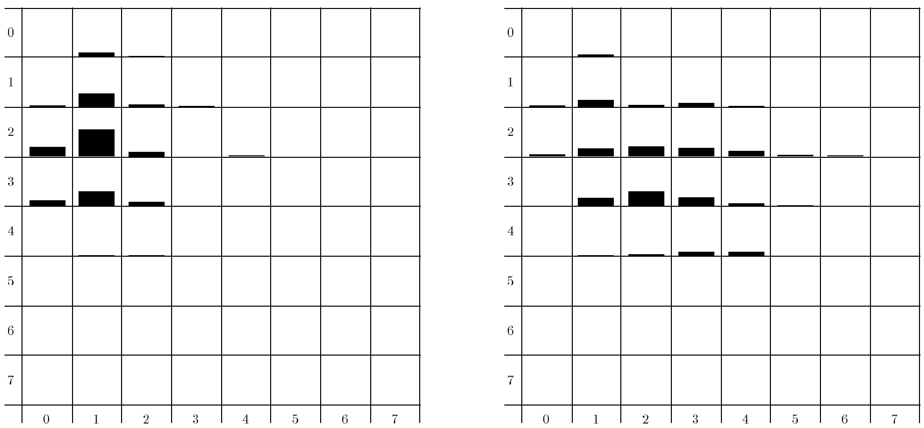

Fig. 5. Empirical distribution of model orders generated with mutation rate 0.05 (left) and 0.5 (right)

4 Alternative Representation

In terms of information theory, it is reasonable to look for the encoding of maximum entropy. On a finite set of parameter values, this is an encoding where each value has the same number of representations. The first ARMA encoding, for example, contains the misleading information that ARMA$[4, 4]$ processes are overrepresented among the solutions.

As a consequence of the obvious influence of the coding on the performance of the method, it seemed to be sensible to take a closer look at the coding itself. As an alternative coding, a neutral representation was chosen, which can be seen in Fig. 6.

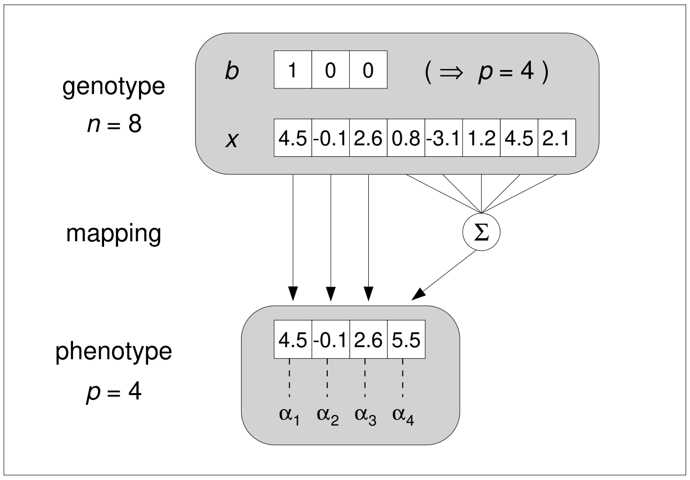

Fig. 6. Genotype-phenotype-mapping in a neutral coding.

The genotype-phenotype mapping in this case works as follows: The binary vector b is treated as a binary number which determines the model order (in the above example p). The first $(p - 1)$ elements of the real vector x are then taken to be the model-parameters α_1 to α_{p-1}. The remaining elements of x are summed up to give α_p in order to keep the entire genome in use. As stated above, this is essential for the step size adaption to work properly. As a side effect, it makes structural mutations smoother: Since the sum of all internal variables is used to explain the past, the cumulated weight of the influence of past values is not changed by dimensional variations.

This coding – in contrast to the former representation – does not favor certain model orders: All combinations of p and q have equal probability in the given range.

To find out whether or not this representation provides an advantage in comparison to the first one, the same simulation study has been carried out for this new coding. It turned out that – as far as the achieved fit is concerned – both representations are of nearly equal quality, just a slight superiority of the first representation could be mentioned.

The much more significant difference between the two codings lies in their model-identification abilities: While the first representation reaches 19.33% of correctly identified models, the second representation does much better with 26.67%. Figure 7 shows the empirical distribution of the model orders, which came from a mutation rate of 0.3.

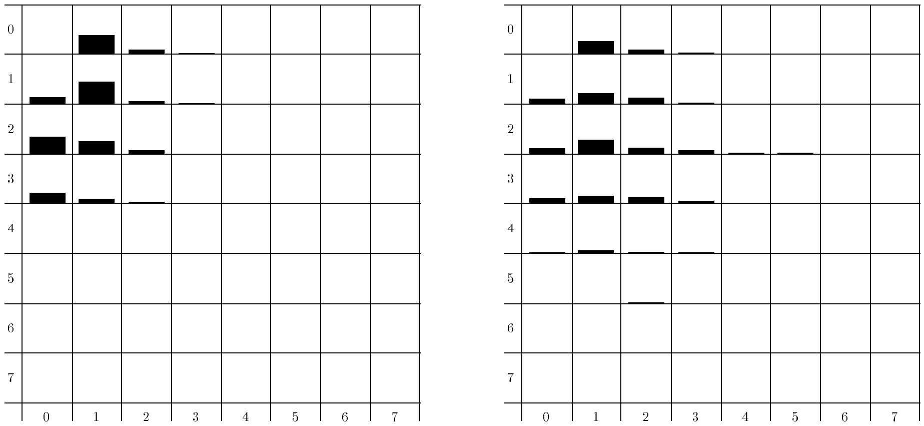

Fig. 7. Empirical distribution of model orders for the second representation.

The difference between the two empirical distributions is not very obvious. As before, a very high percentage of 97% can be found in the correct region. Nevertheless, this coding hits the correct order more often than the first representation does.

5 Conclusions

It has been shown in this paper that a representation, even if it seems to be perfect for one problem, should not be transfered to similar problems without further analysis.

The low absolute identification rates of ARMA processes achieved by both representations are a hint that the fitness criterion itself does not direct the search to the correct models. For this application, statistical tests for the plausibility of model orders, as used in the Box-Jenkins approach, should be integrated either as restrictions or additional objectives.

The analysis of the application examples in this paper shows that obvious or natural representations are not necessarily reasonable. But, nevertheless, the *wrong* representation does not perform significantly worse than the *right* one. This can be partially explained by the robustness of EAs, partially by the fact that the the terms wrong and right depend on our knowledge about the problem. Most probably, the search for the *best* representation is at least as hard as the problem itself.

Acknowledgments

Susanne Rolf and Joachim Sprave gratefully acknowledge support by grant 01 JR 509 A of the German *Federal Ministry of Education, Science, Research and Technology* (BMBF).

References

1. T. Bäck and M. Schütz. Evolution strategies for mixed–integer optimization of optical multilayer systems. In J. R. McDonnell, R. G. Reynolds, and D. B. Fogel, editors, *Evolutionary Programming IV – Proc. Fourth Annual Conf. Evolutionary Programming (EP-95)*, pages 33–51, San Diego CA, March 1–4, 1995. The MIT Press, Cambridge MA.
2. G.E.P. Box and G.M. Jenkins. *Time Series Analysis, Forecasting and Control.* Holden-Day, San Francisco, 1970.
3. Y. Davidor. An evolution standing on the design of redundant robot manipulators. In H.-P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature — Proceedings 1st Workshop PPSN I*, volume 496 of *Lecture Notes in Computer Science*, pages 60–69. Springer, Berlin, 1991.
4. M. Mandischer. Evolving recurrent neural networks with non-binary encoding. In *Proc. Second IEEE Int'l Conf. Evolutionary Computation (ICEC '95), vol. 2*, pages 584–589, Perth, Australia, Nov. 29–Dec. 1, 1995. IEEE Press, Piscataway NJ.
5. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs. 3rd ed.* Springer, Berlin, 1996.
6. S. Rolf, J. Sprave, and W. Urfer. Model identification and parameter estimation of ARMA models by means of evolutionary algorithms. In *Proc. Third Conference on Computational Intelligence for Financial Engineering (IEEE/IAFE CIFEr)*, New York City, March 23–25, 1997.
7. M. Schütz. Eine Evolutionsstrategie für gemischt-ganzzahlige Optimierungsprobleme mit variabler Dimension, diploma thesis, University of Dortmund, September 1994.
8. M. Schütz. Other operators: Gene duplication and deletion. In Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Handbook of Evolutionary Computation*, pages C3.4:8–15. Oxford University Press, New York, and Institute of Physics Publishing, Bristol, 1997.
9. M. Schütz and J. Sprave. Application of parallel mixed-integer evolution strategies with mutation rate pooling. In L. J. Fogel, P. J. Angeline, and T. Bäck, editors, *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, pages 345–354. The MIT Press, Cambridge, MA, 1996.
10. H.-P. Schwefel. *Evolution and Optimum Seeking*. Sixth-Generation Computer Technology Series. Wiley, New York, 1995.