Mixed-Integer NK Landscapes

Rui Li[1], Michael T.M. Emmerich[1], Jeroen Eggermont[2], Ernst G.P. Bovenkamp[2], Thomas Bäck[1], and Jouke Dijkstra[2], and Johan H.C. Reiber[2]

[1] Natural Computing Group, Leiden University,
P.O. Box 9500, 2300 CA Leiden, The Netherlands
{ruili, emmerich, baeck}@liacs.nl
[2] Division of Image Processing, Department of Radiology C2S,
Leiden University Medical Center,
P.O. Box 9600, 2300 RC Leiden, The Netherlands
{J.Eggermont, E.G.P.Bovenkamp, J.Dijkstra, J.H.C.Reiber}@lumc.nl

Abstract. NK landscapes (NKL) are stochastically generated pseudo-boolean functions with N bits (genes) and K interactions between genes. By means of the parameter K ruggedness as well as the epistasis can be controlled. NKL are particularly useful to understand the dynamics of evolutionary search. We extend NKL from the traditional binary case to a mixed variable case with continuous, nominal discrete, and integer variables. The resulting test function generator is a suitable test model for mixed-integer evolutionary algorithms (MI-EA) - i. e. instantiations of evolution algorithms that can deal with the aforementioned variable types. We provide a comprehensive introduction to mixed-integer NKL and characteristics of the model (global/local optima, computation, etc.). Finally, a first study of the performance of mixed-integer evolution strategies on this problem family is provided, the results of which underpin its applicability for optimization algorithm design.

1 Introduction

NK landscapes (NKL, also referred to as NK *fitness* landscapes), introduced by Stuart Kauffman [6], were devised to explore the way that epistasis controls the 'ruggedness' of an adaptive landscape. Frequently, NKL are used as test problem generators for Genetic Algorithms. NKL have two advantages. First, the ruggedness and the degree of interaction between variables of NKL can be easily controlled by two tunable parameters: the number of genes N and the number of epistatic links of each gene to other genes K. Second, for given values of N and K, a large number of NK landscapes can be created at random. A disadvantage is that the optimum of a NKL instance can generally not be computed, except through complete enumeration.

As NKL have not yet been generalized for continuous, nominal discrete, and mixed-integer decision spaces, they cannot be employed as test functions for a large number of practically important problem domains. To overcome this shortcoming, we introduce an extension of the NKL model, *mixed-integer NKL*

T.P. Runarsson et al. (Eds.): PPSN IX, LNCS 4193, pp. 42–51, 2006.
© Springer-Verlag Berlin Heidelberg 2006

(MI-NKL), that capture these problem domains. They extend traditional NKL from the binary case to a more general situation, by taking different parameter types (continuous, integer, and nominal discrete) and interactions between them into account (cf. Figure 1).

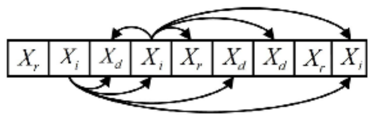


Fig. 1. Example Genes and their interaction

This paper is organized as follows. First, in Section 2, we will give a review of Kauffman's NKL and its variants. In Section 3, we extend NKL to the mixed-integer case , provide theorems on the existence and position of local and global optima, and discuss the implementation of the model. Some initial experimental results are given in Section 4 using a mixed-integer Evolution Strategy. Conclusions and topics for future research are discussed in Section 5.

2 NK Landscapes

Kauffman's NK Landscapes model defines a family of pseudo-boolean fitness functions $F : \{0,1\}^N \to \mathbb{R}^+$ that are generated by a stochastic algorithm. It has two basic components: A structure for gene interaction (using an *epistasis matrix* E), and a way this structure is used to generate a fitness function for all the possible genotypes [1]. The gene interaction structure is created as follows: The genotype's fitness is the average of N fitness components $F_i,\ i = 1, \ldots, N$. Each gene's fitness component F_i is determined by its own allele x_i, and also by K alleles at K $(0 \leq K \leq N-1)$ epistatic genes distinct from i. The fitness function reads:

$$F(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} F_i(x_i; x_{i_1}, \ldots, x_{i_k}),\ \mathbf{x} \in \{0,1\}^N \tag{1}$$

where $\{i_1, \ldots, i_k\} \subset \{1, \ldots, N\} - \{i\}$. There are two ways for choosing K other genes: '*adjacent neighborhoods*', where the K genes nearest to position i on the vector are chosen; and '*random neighborhoods*', where these positions are chosen randomly on the vector. In this paper we focus on the latter case, '*random neighborhoods*'. However, a translation to the first case is straightforward.

The computation of $F_i : \{0,1\}^K \to [0,1),\ i = 1, \ldots, N$ is based on a *fitness matrix* F. For any i and for each of the 2^{K+1} bit combinations a random number is drawn independently from a uniform distribution over $[0,1)$. Accordingly, for the generation of one (binary) NK landscape the setup algorithm has to generate $2^{K+1}N$ independent random numbers. The setup algorithm also creates an epistasis matrix E which for each gene i contains references to its K epistatic genes. Table 1 illustrates the *fitness matrix* and *epistasis matrix* of a NKL. A more detailed description of its implementation can be found in [4].

Table 1. Epistasis matrix E (left) and fitness matrix F (right)

$E_1[1]$	$E_1[2]$	$\cdots$	$\cdots$	$\cdots$	$E_1[K]$
$E_2[1]$	$E_2[2]$	$\cdots$	$\cdots$	$\cdots$	$E_1[K]$
$\cdots$	$\cdots$	$\cdots$	$E_i[j]$	$\cdots$	$\cdots$
$E_N[1]$	$E_N[2]$	$\cdots$	$\cdots$	$\cdots$	$E_N[K]$

$F_1[0]$	$F_1[1]$	$\cdots$	$\cdots$	$\cdots$	$F_1[2^{K+1}-1]$
$F_2[0]$	$F_2[1]$	$\cdots$	$\cdots$	$\cdots$	$F_2[2^{K+1}-1]$
$\cdots$	$\cdots$	$\cdots$	$F_i[j]$	$\cdots$	$\cdots$
$F_N[0]$	$F_N[1]$	$\cdots$	$\cdots$	$\cdots$	$F_N[2^{K+1}-1]$

After having generated the epistasis and fitness matrices, for any input vector $\mathbf{x} \in \{0,1\}^N$ we can compute the fitness in $\mathcal{O}(KN)$ computational complexity via:

$$F(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} F_i[2^0 x_i + 2^1 x_{E_i[1]} + \cdots + 2^K x_{E_i[K]}] \tag{2}$$

Note, that the generation of F has an exponential computational complexity and space complexity in K, while being linear in N. The computational complexity for computing function values is linear in K and N for this implementation.

2.1 Properties of NK Landscapes

Kauffman's model makes two principal assumptions: first, that the fitness of a genotype is the sum of the contributions from each gene, and second, that the effects of polygeny and pleiotropy make these interactions effectively random. Besides Kaufmann, some other researchers, e. g. Weinberger et al. [10,9], did an extensive study on NKL. Some well-known properties are:

1. $K = 0$ (no epistasis): The problem is separable and there exists a unique global optimum. Assuming a Hamming neighborhood-structure, the problem gets unimodal.
2. $1 \le K < N - 1$: For $K = 1$, a global optimum can still be found in polynomial time [10]. For $K \ge 2$, global optimization is NP-complete for the random assignment of neighbors and constant K. However, the problem can always be solved in a computational complexity of 2^N function evaluations and hence can practically be solved for problems of moderate dimension (N around 30). For adjacent neighbors, the problem can be solved in time $O(2^K N)$ (cf. Weinberger [10]).
3. $K = N - 1$: This corresponds to the maximum number of interactions between genes. Practically speaking, to each bitstring of $F : \{0,1\}^N \to [0,1)$ we assign a sum of N values, each of which is drawn independently from a uniform distribution in $[0,1)$. If we choose the Hamming neighborhood on $\{0,1\}^N$ the following results apply:
   - The probability that a random bit string is a local optimum is $\frac{1}{N+1}$
   - The expected number of local optima is $\frac{2^N}{N+1}$

3 Generalized NK Landscapes

As mentioned in the previous section, Kauffman's NKL model is a stochastic method for generating fitness functions on binary strings. In order to use it as a test model for mixed-integer evolution strategies, we extend it to a more general case such that the fitness value can be computed for different parameter types. Here we consider continuous variables in $\mathbb{R}$, integer variables in $[z_{min}, z_{max}] \subset \mathbb{Z}$, and nominal discrete values from a finite set of L values. In contrast to the ordinal domain (continuous and integer variables), for the nominal domain no natural order is given. Mixed-integer optimization problems arise frequently in practise, e.g. when optimizing optical filter designs [2] and the parameters of algorithms [8].

The idea about how to extend NKL to the mixed-integer situation will be described in three steps. First we propose a model for continuous variables, then for those with integer variables and nominal discrete variables. Finally, we will discuss the case of NKL that consists of all these different variable types at the same time and allow for interaction among variables of different types. This defines the full mixed-integer NKL model.

3.1 Continuous NK Landscapes

In order to define continuous landscapes, we choose an extension of binary NKL to an N-dimensional hypercube $[0,1]^N$. Therefore, all continuous variables are normalized between $[0,1]$. In the following we describe the construction of the objective function $F : [0,1]^N \to [0,1)$:

Whenever the continuous variable takes values at the corners of the hypercube, the value of the corresponding binary NKL is returned. For values located in the interior of the hypercube or its delimiting hyperplanes, we employ a *multi-linear interpolation technique* that achieves a continuous interpolation between the function values at the corner. Note that a higher order approach is also possible but we chose a multi-linear approach for simplicity and ease of programming. Moreover, the theory of multi-linear models as used in the design and analysis of experiments, introduces intuitive notions for the effect of single variables and interaction between multiple variables of potentially different types [3]. For each of the N fitness components $F_i : [0,1]^{K+1} \to [0,1)$, we create a multi-linear function

$$F_i(\mathbf{x}) = \sum_{j=0}^{2^{K+1}-1} a^i_j x_i^{[1 \texttt{ AND } j]} \prod_{k=1}^{K} x_{i_k}^{[2^k \texttt{ AND } j]/2^k}, \tag{3}$$

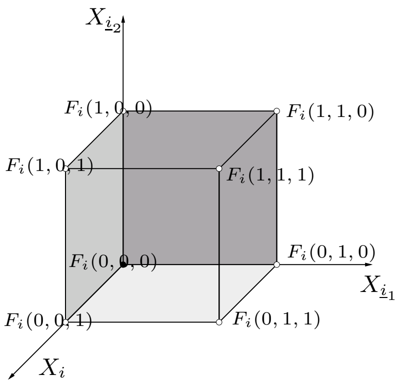
where `AND` is the bitwise *and* operator and x_{i_k} is the k-th epistatic gene of x_i. For instance, in the case $K = 2$ the formula for $F_i(\mathbf{x})$ becomes[1]:
$a^i_{000} + a^i_{001}x_i + a^i_{010}x_{i_1} + a^i_{100}x_{i_2} + a^i_{011}x_i x_{i_1} + a^i_{101}x_i x_{i_2} + a^i_{110}x_{i_1}x_{i_2} + a^i_{111}x_i x_{i_1} x_{i_2}$.

[1] Note, that we use binary instead of decimal numbers for the index to make the construction more clear.

Once uniformly distributed random values have been attached to the corners of the K-dimensional hypercube (cf. Figure 2), we can identify the coefficients $a^i_0, \ldots, a^i_{2^{K+1}-1}$ by solving a linear equation system (LES). However, even for moderate K the computational complexity for applying general LES solvers would be prohibitive high. An advantage of the multi-linear form (as compared to other interpolation schemes like radial basis functions or splines) is, that it allows for an efficient computation of the coefficients by exploiting the diagonal structure of the equation system. Accordingly, a^i_j can be obtained by means of the following formula:

$$a^i_0 = F_i[0], a^i_j = F_i[j] - \sum_{\ell=0}^{j-1} \left[a^i_\ell \mathrm{I}(\ell = (\ell \texttt{ AND } j)) \right], j = 1, \ldots, 2^{K+1} - 1 \qquad (4)$$

In order to compute the values, we have to start with $j = 0$ and increase the value of j. Hence, the number of additions we need for computing all coefficients is proportional to $(2^{K+1} - 1)(2^{K+1})/2 = 2^{2(K+1)-1} - 2^K$.



$$\begin{aligned}
a^i_{000} &= F_i(0,0,0) \\
a^i_{001} &= F_i(0,0,1) - a^i_{000} \\
a^i_{010} &= F_i(0,1,0) - a^i_{000} \\
a^i_{011} &= F_i(0,1,1) - a^i_{000} - a^i_{001} - a^i_{010} \\
a^i_{100} &= F_i(1,0,0) - a^i_{000} \\
&\vdots \\
a^i_{111} &= F_i(1,1,1) - a^i_{000} - a^i_{001} - a^i_{010} - a^i_{011} \\
&\quad - a^i_{100} - a^i_{101} - a^i_{110}
\end{aligned}$$

Fig. 2. Example HyperCube with $K = 2$ and the computation of a^i_j

Once we have the a^i_j values, we can use Equation 1 to compute the model. Of course the domain of the $\mathbf{x}$ values has to be replaced by $[0,1]^N$ in that equation. For the computation of the global optimal value of the continuous NK landscapes the following lemma is useful:

Lemma 1. At least one global optimum of the function F will always be located in one of the corners of the N dimensional hypercube, such that the computation of the optimal function value upper bounds the computational complexity for the binary model.

Proof: The idea of the proof is that there is an algorithm that for any given input $\mathbf{x}^* \in [0,1]^N$ determines a corner of the hypercube, the function value of which is not higher than the function value at F, given that F has a multilinear form. Basically, the proposed algorithm can be described as a path oriented algorithm that searches parallel to the coordinate axis: First we fix all variables except one, say x_1, in F. It is now crucial to see that the remaining form $F(x_1, x_2^*, \ldots, x_N^*)$ is a linear function of x_1. Now, because the form is linear, it is obvious to see that either $(1, x_2^*, \ldots, x_N^*)^T$ or $(0, x_2^*, \ldots x_N^*)^T$ has a function value that is better or equal than the function value at $(x_1^*, \ldots, x_N^*)^T$. We fix x_1 to a value for which this is the case, i. e. we move either to $(1, x_2^*, \ldots, x_N^*)^T$ or to $(0, x_2^*, \ldots x_N^*)^T$ without increasing the function value. For the new position $\mathbf{x}^{1*}$ we again fix all variables except one. This time x_2 is the free variable. Again we can move the value of x_2 either to zero or to one, such that the function value does not increase. Now, the new vector $\mathbf{x}^{12*}$ will either be $(x_1^{1*}, 0, x_3^*, \ldots, x_N^*)^T$ or $(x_1^{1*}, 1, x_3^*, \ldots, x_N^*)^T$. After continuing this process for all remaining variables x_3 to x_N we finally obtain a vector $\mathbf{x}^{12\cdots N*}$, all values of which are either zero or one, and the function value is not worse than that of $\mathbf{x}^*$. □

From Lemma 1 it follows:

Theorem 1. The problem of finding the global optimal value for a continuous NKL is NP-complete for $K \geq 2$.

Proof: Finding the optimum in the corner is equivalent to the NP-complete binary case. By applying Lemma 1, we can reduce the continuous case to the binary case. On the other hand, whenever we find the global optimal solution for the continuous case, in polynomial time we can construct a just as good solution where all optima are located at the corners in linear time. Thus, there exists a polynomial reduction of the binary case to the continuous case. □

3.2 Integer NK Landscapes

Based on our design, NKL on integer variables can be considered to be a special case of continuous NKL. The integer variables can be normalized as follows: Let $z_{min} \in \mathbb{Z}$ denote the lower bound for an integer variable, and $z_{max} \in \mathbb{Z}$ denote its upper bound. Then, for any $z \in [z_{min}, z_{max}] \subset \mathbb{Z}$ we can compute the value of $x = (z - z_{min})/(z_{max} - z_{min})$ in order to get the corresponding continuous parameter in $[0, 1]$, which can then be used in the continuous version of F to compute the NKL. Note that the properties discussed in Lemma 1 and Theorem 1 also hold for integer NKL.

3.3 Nominal NK Landscapes

To introduce nominal discrete variables in an appropriate manner a more radical change to the NKL model is needed. In this case it is not feasible to use interpolation, as this would imply some inherent neighborhood defined on a single variable's domain $x_i \in \{d_1^i, \ldots, d_L^i\}$, $i = 1, \ldots, N$, which, by definition, is not given for the nominal discrete case. We will now propose an extension of NKL that takes into account the special characteristics of nominal discrete variables.

Let the domain of each nominal discrete variable x_i, $i = 1, \ldots, N$ be defined as a finite set of maximal size $L \geq 2$. Then for the definition of a function on a tuple of $K + 1$ such values we would need a table with L^{K+1} entries. Again, we can assign all fitness values randomly by independently drawing values from a uniform distribution. The size of the sample is upper-bounded by L^{K+1}. For $L = 2$ this corresponds to the binary case. After defining N fitness components F_i, we can then sum up the values of these components for the NKL model (eq. 1). The optimum can be found by enumerating all input values, the computational complexity of which is now L^N. The implementation of the function table and the evaluation procedures are similar to that of the binary case. Note, that for a constant value of L and K the space needed for storing the function values is given by NL^{K+1}, so is the computational complexity for generating the matrix. The time for the function evaluations is proportional to $N(K + 1)$.

Equipping the discrete search space with a Hamming neighborhood, in case $K = 0$ the problem remains unimodal. For $K > 0$, we remark, that for the general problem with $L > 2$, the detection of the optimum is more difficult than in the binary case. Hence, the binary case can be reduced to the case $L > 2$, but not vice versa. For the case of full interaction (K=N-1) we show:

Lemma 2. For the nominal discrete NKL with $K = N - 1$, $L \equiv$ *constant*, and Hamming neighborhood defined on the discrete search space, the probability that an arbitrary solution $\mathbf{x}$ gets a local optimum is $\frac{1}{N(L-1)+1}$. Moreover the expected number of local optima is $\frac{L^N}{N(L-1)+1}$.

Proof: Given the preliminaries, $N(L - 1)$ is the number of Hamming neighbors for any solution $\mathbf{x} \in \{1, \ldots, L\}^N$. Since we assign a different fitness value from the interval $[0, 1)$ independently to each neighbor, the probability, that the central solution, i.e. $\mathbf{x}$ itself becomes the best solution, is $1/(N(L-1)+1)$. Since, L^N is the number of search points in $\{1, \ldots, L\}^N$ we can compute the expected number of local optima as $\frac{L^N}{N(L-1)+1}$. □

3.4 Mixed Integer NK Landscapes

It is straightforward to combine these three types of variables into a single NKL with epistatic links between variables of different types (cf. Figure 1). For mixed variables of the integer and continuous types there is no problem, since integers, after normalization, are treated like continuous variables in the formula of F. If there are D nominal discrete variables that interact with a continuous variable, then the values of these discrete variables determine the values at the edges of the $K - D$ dimensional hypercube that is used for the interpolation according to the remaining continuous and integer variables. Note that for different nominal discrete values the values at the corners of the $K - D$ dimensional hypercube will change in almost every case.

Instead of describing the mixed variable case in a formal manner we give an illustrating example (cf. figure 3). This example shows one individual with three parameters (one continuous, one integer and one discrete), and each gene interacts with both other genes. For each gene, a hypercube is created. We assume there are three levels for the discrete gene X_d ($L = 3$), so the hypercube is reduced to three parallel planes, and the value of the discrete gene decides which plane is chosen. More concretely, assuming the individual has the following values: $X_d = 0, X_i = 0.4, X_r = 0.8$, the value of the discrete parameter X_d determines which square is chosen ($X_d = 0$). The value for each corner is based on the fitness matrix in Table 2 (bold displayed). As mentioned in the previous chapter, we calculate the fitness value of this individual as follows:

$$F_r(\mathbf{a}, \mathbf{x}) = a_0 + a_1 X_r + a_2 X_i + a_3 X_i X_r$$
$$a_0 = F_r(0,0) = 0.8, \quad a_1 = F_r(0,1) - a_0 = -0.1$$
$$a_2 = F_r(1,0) - a_0 = -0.1, \quad a_3 = F_r(1,1) - a_0 - a_1 - a_2 = -0.1$$
$$F_r(0.4, 0.8) = 0.648$$

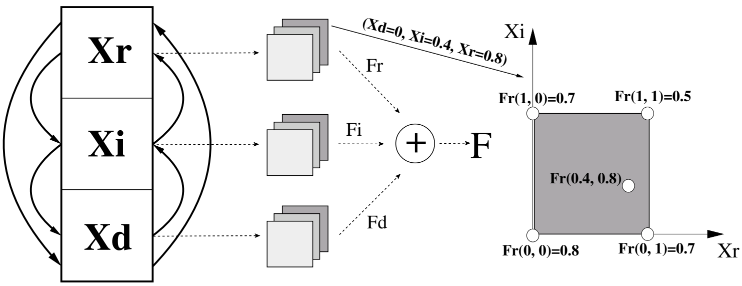


Fig. 3. Example for the computation of a MI-NK landscape

Table 2. Example epistasis matrix (left)and fitness matrix (right)

$E_r[1] = X_i$	$E_r[2] = X_d$
$E_i[1] = X_r$	$E_i[2] = X_d$
$E_d[1] = X_r$	$E_d[2] = X_i$

	0.8	***0.7***	***0.7***	***0.5***		0.3	0.7	0.2	0.9		0.5	0.6	0.3	0.5
F_r	0.5	0.8	0.4	0.7	F_i	0.2	0.3	0.7	0.9	F_d	0.9	0.8	0.2	0.7
	0.2	0.1	0.8	0.4		0.2	0.5	0.4	0.6		0.8	0.7	0.3	0.3

4 Experimental Results

In order to test our mixed-integer NKL problem generator we have tested it using a (μ,κ,λ) mixed-integer evolution strategy (MI-ES) as described in [5]. Here mutation distributions with maximal entropy are employed for the mutation of continuous variables (Gaussian distribution), integer variables (geometric distribution), and nominal discrete variables (uniform distribution). While for the first two types a step-size parameter can be learned, in the latter case a mutation probability is learned. We use a population size μ of 4, offspring size λ of 28 and $\kappa = 1$ (comma-strategie). The stepsize/mutation-probability of each variable was set to 0.1 and the standard MI-ES mutation and crossover operators are used. The maximum number of fitness evaluations is set to 3000.

To see the effect of different values of K we generated 50 problem instantiations for $N = 15$ and for each value $K \leq 14$ (750 MI-NKL problems in total) so that it is still feasable to find the global optimum by evaluating all bitstrings of length 15. Each generated problem consists of 5 continuous, 5 integer and 5 nominal discrete variables. The continuous variables are in the range [-10,10], the integer-valued variables are in the range [0,19] and we used $\{0, 1\}$ for the nominal discrete variables (Booleans). As described previously the continuous and integer-valued variables are normalized to fit in the interval $[0, 1]$ before evaluation. To compare (and average) the results of the different experiments we define the following error-measure:

$$\textbf{error} = \textbf{best found fitness - best possible fitness}$$

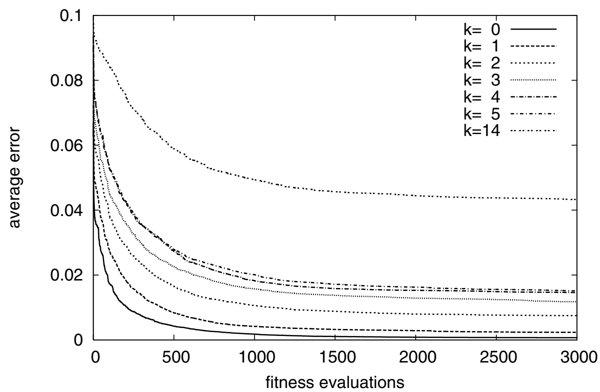


Fig. 4. The error averaged over 50 mixed-integer NK landscape problems with $N = 15$. Each problem contained 5 continuous, 5 integer-valued and 5 Boolean-valued variables.

The results are displayed in Figure 4. The x-axis shows the number of evaluations while the y-axis shows the average *error* (over 50 experiments). As can be seen an increase in K results in an increase in *error* which indicates the problem difficulty increases with K. The fact that even for $K = 0$ the MI-ES algorithm has problems achieving an average *error* of 0 is because in order to find the global optima all variables, including the continuous ones, have to be exactly either 0 or 1 (after normalization). This is hard for the continuous part of MI-ES individuals because of the mutation operator used. In the mutation, we used a reflection at the boundary method for keeping the variables within the $[0, 1]$ intervals [5]. This does not favor solutions that are directly at the boundary, as this is done by other interval treatment methods, like for example logistic transformation [2]. However, the latter mutation operator adds a bias to the search and makes it more easy to locate solutions at the boundary than in the interior, which is why we did not use it here.

5 Conclusion and Outlook

The NK landscape model has been extended to the mixed-integer problem domain. It turns out that a multi-linear interpolation approach for the continuous and integer variables provides a straightforward generalization of this model, that can also be easily implemented. Using Equation 3, function values can be computed in linear time. However, the detection of the global optimum turns out to be a NP-complete problem for $K > 2$ and can be reduced to the problem of detecting the global optimum for the binary case.

An alleged drawback of the interpolation approach is that its optima are always located in the corners of the search space. There are some ways of how this problem could be addressed. One way would be to transform the input variables by means of a periodic function and mapping them back to $[0, 1]$, e.g. to substitute x_i by $s(x_i) = \frac{1}{2} + \frac{1}{2}\cos(\pi x_i + \pi)$ and restrict x_i to the interval $[-0.5, 1.5]$ for $i = 1, \ldots, N$. It is easy to show that the optima for this transformed function are at the same position as for the original model.

For the nominal discrete variables the binary NK landscape was extended to a L-ary representation. For this the amount of random numbers increases exponentially with L. Also, for $N = K - 1$ it has been shown that the number of local optima increases exponentially with L.

One of our intentions for developing MI-NKL was to further improve the MI-ES approach. The experiments demonstrate the applicability of the MI-NKL problem generator and that the difficulty for finding the global-optimum grows with K. Future work will focus on exploring more of the characteristics of the MI-NKL, including its specializations: continuous, integer and discrete NKL.

Acknowledgements. This research is supported by the Netherlands Organisation for Scientific Research (NWO).

References

1. L. Altenberg: NK-Fitness Landscapes, In "the Handbook of Evolutionary Computation", ed. Th. Bäck, D.B. Fogel, and Z. Michalewicz, Oxford Univ. Press, 1997.
2. Th. Bäck and M. Schütz: Evolution Strategies for Mixed-Integer Optimization of Optical Multilayer Systems. Evolutionary Programming, 33 - 51, 1995.
3. G. Box, W. Hunter, and J. Hunter: Statistics for Experiments. Wiley, 1978.
4. A.E. Eiben and J.E. Smith: Introduction to Evolutionary Comp., Springer, 2003.
5. M. Emmerich, M. Groetzner, B. Groß, and M. Schütz. Mixed-integer evolution strategy for chemical plant optimization with simulators. In I. C. Parmee, editor, Proc. of ACDM'00, pages 55-67, Springer, London, 2000.
6. S.A. Kauffman: The origins of order: Self-organization and selection in evolution. Oxford University Press, NY, 1993.
7. S.A. Kauffman. and S. Levin: Towards a general theory of adaptive walks on rugged landscapes. Journey of Theoretical Biology 128: 11—45, 1987.
8. R. Li, M.T.M. Emmerich, E.G.P. Bovenkamp, J. Eggermont, Th. Bäck, J. Dijkstra and J.H.C. Reiber. Mixed-Integer Evolution Strategies and Their Application to Intravascular Ultrasound Image Analysis. In F. Rothlauf et al. eds: Applications of Evolutionary Computing, pp. 415-426, LNCS 3907, 2006, Springer.
9. R.E. Smith and J.E. Smith: An examination of tunable, random search landscapes. In "Foundations of Genetic Algorithms 5", ed. W. Banzhaf and C. Reeves, Morgan Kaufmann, San Francisco, 1999.
10. E.D. Weinberger: NP completeness of Kauffman's N-K model, a tuneable rugged fitness landscape. Working Papers 96-02-003, Santa Fe Institute, Santa Fe, NM, First circulated in 1991.