

On the Nature of Two-Bit Multiplier Landscapes

Vesselin K. Vassilev
School of Computing
Napier University
219 Colinton Road
Edinburgh, EH14 1DJ, UK
v.vassilev@dcs.napier.ac.uk

Julian F. Miller
School of Computing
Napier University
219 Colinton Road
Edinburgh, EH14 1DJ, UK
j.miller@dcs.napier.ac.uk

Terence C. Fogarty
School of Computing
Napier University
219 Colinton Road
Edinburgh, EH14 1DJ, UK
t.fogarty@dcs.napier.ac.uk

Abstract

The two-bit multiplier is a simple electronic circuit, small enough to be evolvable, and practically useful for the implementation of many digital systems. In this paper, we study the structure of the two-bit multiplier fitness landscapes generated by circuit evolution on an idealised model of a field-programmable gate array. The two-bit multiplier landscapes are challenging. The difficulty in studying these landscapes stems from the genotype representation which allows us to evolve the functionality and connectivity of an array of logic cells. Here, the genotypes are simply strings defined over two completely different alphabets. This makes the study of the corresponding landscapes much more involved. We outline a model for studying the two-bit multiplier landscapes and estimate the amplitudes derived from the Fourier transform of these landscapes. We show that the two-bit multiplier landscapes can be characterised in terms of subspaces, determined by the interactions between the genotype partitions.

1. Introduction

Evolving a fully functional two-bit multiplier in a configuration of logic cells is a simple case of digital circuit evolution, recently studied in the nascent field of evolvable hardware [13, 26, 15]. The two-bit multiplier is a simple electronic circuit, small enough to be feasible for evolutionary design, and practically useful as a fundamental building block used in the synthesis of many digital systems. Examples of evolved two-bit multipliers have been given in [13] where it was suggested that the design of digital circuits on a gate array in general can be implemented in terms of the *functionality* and *routing* of the array [12, 11]. These terms refer to the logic functions utilised in the array's cells, and the connections between the inputs, cells

and outputs, respectively, which are then encoded in a *genotype*. This important stage in modeling circuit evolution often predetermines the likelihood of success in solving this optimisation task. Recently, various genotype representations have been proposed and different aspects for attaining efficient evolutionary design of digital circuits have been discussed [12, 11]. Here, we study the evolutionary synthesis of the two-bit multiplier in terms of the structure of its fitness landscapes based on the genotype representation proposed in [11], and explicitly studied in [22]. The motivation behind this research is to understand the feasibility of evolving digital circuits - what are the impediments and the limiting factors?

We consider the evolutionary design of the two-bit multiplier as a search on a fitness landscape. The two-bit multiplier landscapes are quite different from many recently studied landscapes [9, 24, 5]. The difference originates in the structure of the genotypes which are strings defined over two completely different alphabets which are responsible for the functionality and connectivity of the array of logic cells. It gives rise to a complicated *mixture* of epistasis characteristics of the genotype which makes the study of the corresponding landscapes much more convoluted. In [22] a model for studying the structure of circuit evolution landscapes is introduced and autocorrelations are measured. The two-bit multiplier landscapes were defined over three spaces which generate sub-landscapes with different ruggedness. These are strongly dependent on the interactions between the genotype partitions which represent the functionality, internal, and output connectivity of the evolved circuit. In order to attain a greater understanding of the relationships between the genotype partitions, we apply amplitude spectra analysis. This is a method based on the idea that the ruggedness of a fitness landscape can be estimated by studying the amplitude spectrum obtained from the Fourier transform of the landscape [18, 6].

The next section introduces the concept of fitness landscapes and the amplitude spectra analysis used in the re-

mainder of the paper. Section 3 explains the genotype representation used. In section 4 the model of the two-bit multiplier landscapes is described. The structure of the landscapes are studied in section 5 with an analysis of the results. Finally, we give conclusions and suggestions for future work.

2. Fitness Landscapes

The notion of a *fitness landscape*, introduced by Wright, has become an important concept in evolutionary computation [25]. The metaphor is taken from biology and it expresses the idea that evolution can be considered as a population flow on a surface in which the altitude of a point qualifies how well the corresponding organism is adapted to an environment. In evolutionary computation the fitness landscapes are simply search spaces with elements called *phenotypes* which are represented by their *genotype*. The genotype is a sequence of elements taken from a finite alphabet. A *fitness value* is assigned to each genotype and the evolutionary algorithm refers to these values when deciding which phenotypes are to survive and reproduce. The fitness value of a genotype is evaluated by a *fitness function*, f , which measures how *good* the encoded phenotype is. The population of genotypes flow on a landscape guided by an evolutionary operator. So, the genotype representation, the neighbourhood relation provided by the evolutionary operator, and the fitness function define the structure of a fitness landscape [8, 18, 21], which reflects on the ability of the evolutionary algorithm to search. It has been shown that search is difficult on rugged low correlated landscapes [9, 10, 14].

The ruggedness of a fitness landscape can be estimated by studying the amplitude spectrum obtained from the Fourier transform of the landscape [18, 6]. The amplitude spectra analysis was outlined by [18] who suggested that the amplitudes of a landscape convey information about the landscape ruggedness. The idea was adopted by [16] who showed that better evolutionary search might be attained by investigating sub-populations on sub-landscapes which are smoother than the original one. The analysis is based on the graph model of landscapes intensively studied in [18, 21]. A fitness landscape, \mathcal{L} , is formally defined as a graph \mathcal{G}_f with vertices, the genotypes labelled with fitness values, and edges defined by the evolutionary operator, ϕ . If we consider that the genotypes are strings with length n , defined over alphabet with size l then the landscape is $\mathcal{L} = (\mathcal{G}_f, f, \phi)$ where the underlying graph for one-point mutation and crossover are the Hamming hypercube $\mathcal{G}_l^n = (V, E)$, and the P-structure (V, \mathcal{R}) , respectively. The sets V and E represent the vertices and edges of \mathcal{G}_f , respectively. The P-structure is identified by $\mathcal{R}: V \times V \rightarrow P(V)$ which maps each pair of parents into a set of offsprings [21].

In [18] it was shown how a landscape can be represented as a superposition of elementary sub-landscapes. Each sub-landscape contributes to the structure of the original landscape by its component ψ_k , a normalised eigenvector of the Fourier series, where the component is amplified by a certain amplitude B_k . The amplitudes, B_k , uniquely specify the structure of the fitness landscape. Each B_k measures the influence of the interaction of order k on the fitness function. Hence, the landscapes are more rugged when the amplitudes are high for higher values of k . For instance, Stadler and Happel [20] showed that only the first $k + 1$ amplitudes contribute to the spectrum of one-point mutation Nk landscapes (the adjacent neighbourhood model). The amplitudes of a given landscape can be obtained by solving the system

$$\rho(s) = \sum_{k \neq 0} B_k \left(1 - \frac{\lambda_k}{\delta}\right)^s, \quad (1)$$

where λ_k are the eigenvalues, and $\rho(s)$ is the autocorrelation function of the landscape. The parameter δ is the vertex degree of the one-point mutation underlying graph [18]. When a crossover landscape is explored, the parameter δ becomes 1 due to normalisations in the definition of the graph Laplacian of the underlying P-structure [21]. The autocorrelation function can be estimated by measuring the autocorrelations of time series obtained via random walk [24]. The autocorrelation function is

$$\rho(s) = \frac{E[f_t f_{t+s}] - E[f_t]E[f_{t+s}]}{V[f_t]}, \quad (2)$$

where $E[f_t]$ and $V[f_t]$ are the expectation and the variance, respectively, of the time series, $\{f_t\}_{t=0}^n$. Following [18, 21] the eigenvalues λ_k of the graph Laplacian of one-point mutation and uniform crossover graphs can be easily calculated as

$$\lambda_k = kl \quad (3)$$

and

$$\lambda_k = 1 - \left(\frac{l+2}{2l}\right)^n \left(\frac{(l-1)(l+1)-1}{(l-1)(l+2)}\right)^k \quad (4)$$

respectively. In [6] it was suggested that the straightforward solution of the system described in equation 1 does not yield a satisfactory accuracy in the obtained results. In order to obtain accurate approximation, they proposed the steepest descent technique [2] (pp. 614-620) that iteratively minimises the sum of squared errors in the system. The stop criterion of the steepest descent algorithm is related to a certain tolerance which in this paper is 0.00001.

3. Designing the Digital Circuit Evolution

In [13] digital circuits are evolved using a genetic algorithm [4]. In short, a population of individuals is main-

tained, the genotypes of which are initially generated at random. The fitness value of each individual is evaluated, by calculating the percentage of total correct outputs of the encoded electronic circuit in response to appropriate inputs. Then a new population is generated by applying uniform crossover with probability 0.5 and onepoint mutation to individuals selected from the parent population. Tournament selection (size 2) is used in which the winner of the tournament is chosen with a certain probability. Here we consider that the fitness values have been scaled in the interval $[0, 1]$.

In this paper, we consider a genotype representation based on that proposed in [11], and explicitly studied in [22]. The genotype is a composition of two different types of genes which are responsible for functionality and routing of the evolved gate array. It is important to note that the representation is restricted to allow only feed-forward circuits since only combinational design is considered. An example of $n \times m$ gate array with n_I inputs and n_O outputs is given in Figure 1.

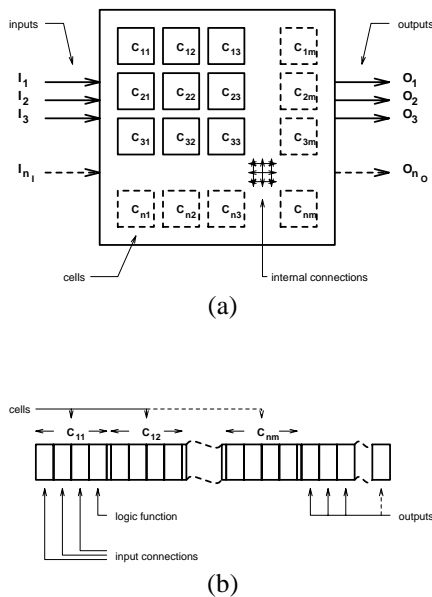


Figure 1. The genotype-phenotype mapping: (a) a $n \times m$ geometry of logic cells with n_I inputs and n_O outputs, and (b) the genotype structure of the array.

The gate array is a composition of cells where each cell may have the function of an allowed two-input logic gate or alternatively a 2-1 multiplexer (two input plus a single control input). The cells allow up to 25 logic functions, where the first sixteen are all boolean functions of up to two arguments (see [1]) and the remainder are the six possible basic

universal-logic-gate 2 logic functions (capable of realising all functions of $n \leq 2$ inputs) as given in [3], some of them taken several times with differently inverted inputs.

The internal connectivity in the gate array is specified by the connections among the array cells. The inputs of each cell are only allowed to be the array inputs or the outputs of cells with lower column numbers. The array connectivity is related to the *levels-back* parameter which determines the neighbourhood size of the cells. Hence, if we assume that the gate array inputs are outputs of cells with column number 0 and the levels-back parameter is L , then the inputs of cells $\{c_{ij}\}_{\forall i,j}$ are chosen from all possible outputs of "preceding" cells with column numbers j' for which $j'' \leq j' < j$ where $j'' = j - L$ if $j > L$, otherwise $j'' = 0$.

The gate array output connections are specified in a similar way. They could be any outputs of cells or array inputs which come from the array outputs neighbourhood given by the levels-back parameter. Consider, for instance, the gate array in Figure 1. If the levels-back is $m + 1$ then the array outputs can be any of the outputs of cells $\{c_{ij}\}_{\forall i,j}$ and the gate array inputs. On the contrary, if the levels-back is 1 then only the cells of column m may contribute to the array outputs.

The genotype construction is straightforward. We number all inputs and cells of the gate array with integers in the following way: each array input I_k is labelled with $k - 1$ where $1 \leq k \leq n_I$, and each cell c_{ij} is labelled with $n_I + n(j - 1) + i - 1$ where $1 \leq i \leq n$ and $1 \leq j \leq m$. For instance, if the gate array has 5 inputs the first cell has number 5 since the last array input is labeled with 4. So, the genotype consists of two parts. The first part is split into groups of four integers where each group codes one cell of the gate array. The first three values of each group represent the points to which the inputs of the gate are connected. If the gate is two-input, then the third connection is not taken into account. The last value of the group represents the logic function of the gate. The second part of the genotype is a sequence of integers that represent the labels of the cells connected to the gate array outputs. An example of the genotype structure is given in Figure 1 (see also [22]). The corresponding *phenotype* of an evolved 2-bit multiplier is illustrated in Figure 2. The figure reveals the construction of the circuit in the gate array with respect to the genotype representation. Each cell in the array is depicted by a box where the number in the top-right corner is the cell's label. Thus, the relationship between the phenotype and its genotype could be easily revealed. Obviously, the circuit is not the optimal one, since the optimisation task is defined on 4×4 array. Examples of much better solutions for 2-bit multiplier are given in [13] where examples of "adders with carry" and other combinatorial circuits are given (for 2-bit and 3-bit multipliers consisting of 7 and 24 two-inputs logic gates, respectively, see [22]).

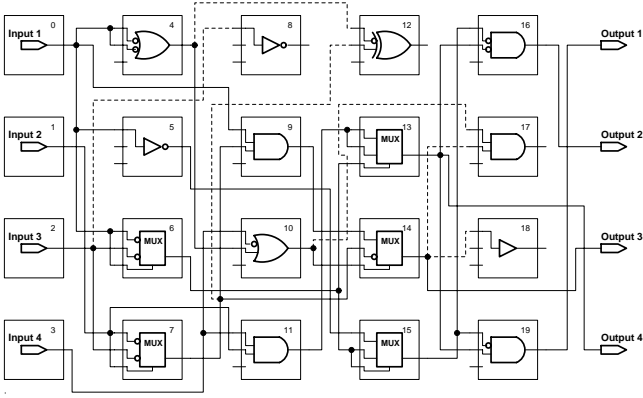


Figure 2. A two-bit multiplier evolved on 4×4 gate array with levels-back set to 2.

4. The Structure of Circuit Evolution Landscapes

We consider that the genotype is a composition of three different parts which are responsible for first, the gates functionality, second, the array internal connectivity, and third, the array outputs. The reason for splitting the genotype into parts is the difference in the purposes of the parts. For convenience, we adopt the term *chromosome* to mean a genotype part. The chromosomes have different lengths and they are defined over two different alphabets. The “gate functionality” chromosomes are strings over alphabet α with length the number of gates. The alphabet size l_α is the number of allowed logic functions used in the circuit design. The “internal connectivity” and “array outputs” chromosomes are defined over alphabet β , and they are strings with length the number of gates and the number of array outputs, respectively. The alphabet β is related to the size of the neighbourhood of the cells and array outputs, which is dependent upon the levels-back parameter. The alphabet is a set of integers, which are reference numbers of the elements of a neighbourhood. Hence, the size of β is

$$l_\beta = \begin{cases} nL, & \text{if } L \leq m \\ nm + n_I, & \text{otherwise} \end{cases}, \quad (5)$$

where n is the number of rows, m is the number of columns, n_I is the number of inputs of the gate array, and L is the levels-back.

A major difficulty in studying the structure of circuit evolution landscapes stems from the genotype structure. It determines a complicated *mixture* of epistasis characteristics of the genotype which is almost insurmountable impediment if we consider the genotype as an inseparable chain.

The landscapes are highly *non-isotropic* which was also revealed by measured autocorrelations of random walks. We performed vast numbers of random walks with length 100,000 and we observed that the autocorrelation functions differed from each other in a significantly wide range. In order to avoid the vagueness in the definition of our landscapes we split the genotype space into three partitions as was done in [19].

Since each genotype consists of three chromosomes we assume that the original landscape for a given genetic operator is a superposition of three configuration spaces defined over alphabets α and β . We define the hypercubes $\mathcal{G}_{l_\alpha}^{nm}$, $\mathcal{G}_{l_\beta}^{3nm}$, and $\mathcal{G}_{l_\beta}^{n_o}$ to represent the configuration spaces of the chromosomes responsible for functionality, connectivity, and output connections, respectively. Let ϕ denote a genetic operator. For each hypercube we define a family of landscapes that represents the genotype space with respect to the hypercube. The landscapes are

$$\begin{aligned} (1) \quad & \{\mathcal{G}_{f_i}, f_i, \phi\}_{i=0}^{l_\beta^{3nm+n_o}-1} \\ (2) \quad & \{\mathcal{G}_{g_i}, g_i, \phi\}_{i=0}^{l_\alpha^{nm} l_\beta^{n_o}-1} \\ (3) \quad & \{\mathcal{G}_{h_i}, h_i, \phi\}_{i=0}^{l_\alpha^{nm} l_\beta^{3nm}-1}. \end{aligned} \quad (6)$$

The graphs \mathcal{G}_{f_i} , \mathcal{G}_{g_i} , and \mathcal{G}_{h_i} are obtained by assigning each vertex from $\mathcal{G}_{l_\alpha}^{nm}$, $\mathcal{G}_{l_\beta}^{3nm}$, and $\mathcal{G}_{l_\beta}^{n_o}$, respectively, with a fitness value. The fitness values are provided by fitness functions $\{f_i\}_{\forall i}$, $\{g_i\}_{\forall i}$ and $\{h_i\}_{\forall i}$ which are defined as follows

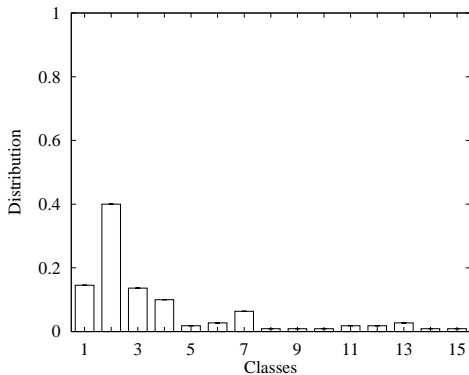
$$\begin{aligned} (1) \quad & \forall i(\mathbf{c}_i \in \mathcal{G}_{l_\beta}^{3nm} \times \mathcal{G}_{l_\beta}^{n_o}), \\ & \forall \mathbf{x} \in \mathcal{G}_{l_\alpha}^{nm} : f_i(\mathbf{x}) = F(\mathbf{x} \cdot \mathbf{c}_i) \\ (2) \quad & \forall i(\mathbf{c}_i \in \mathcal{G}_{l_\alpha}^{nm} \times \mathcal{G}_{l_\beta}^{n_o}), \\ & \forall \mathbf{x} \in \mathcal{G}_{l_\beta}^{3nm} : g_i(\mathbf{x}) = F(\mathbf{x} \cdot \mathbf{c}_i) \\ (3) \quad & \forall i(\mathbf{c}_i \in \mathcal{G}_{l_\alpha}^{nm} \times \mathcal{G}_{l_\beta}^{3nm}), \\ & \forall \mathbf{x} \in \mathcal{G}_{l_\beta}^{n_o} : h_i(\mathbf{x}) = F(\mathbf{x} \cdot \mathbf{c}_i). \end{aligned} \quad (7)$$

The function F is defined over the genotype space (the operator “ \cdot ” is considered to merge the strings in a special way so that the genotype structure as given in Figure 1 is maintained) and it evaluates the percentage of correctness of the represented circuit. Note that for each family of landscapes we have a group of fitness functions, and each fitness function estimates only a part of the genotype. Hence, its index is determined by the constant string \mathbf{c}_i which is the remainder of the genotype.

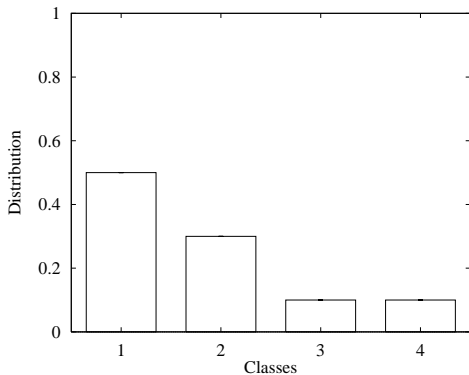
5. The Ruggedness of Two-Bit Multiplier Landscapes

In this section, we study the two-bit multiplier fitness landscapes generated by one-point mutation and uniform

crossover with probability 0.5. The two-bit multiplier is evolved on 4×4 gate array where the levels-back is 2 and the allowed functional gates are the first 21 logic functions. The number of the array inputs is 4. In [22] we measured autocorrelations with respect to the functionality, internal connectivity, and output connectivity of the gate array. We showed that the functionality landscapes are more rugged than the output connectivity landscapes, but smoother than the internal connectivity ones. Our results also revealed that the uniform crossover landscapes are very rugged, and therefore, difficult for evolutionary search. Here we go further. We measure the amplitudes of two-bit multiplier landscapes with respect to functionality, internal connectivity, and output connectivity of the evolved gate array and study the relationships between these gate array characteristics.



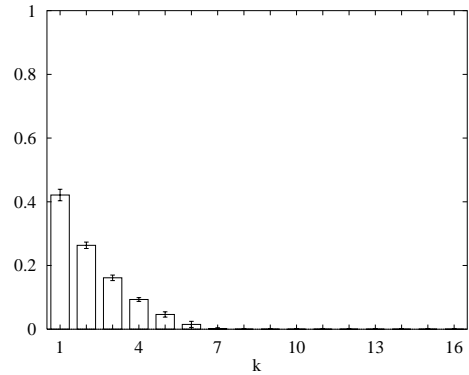
(a)



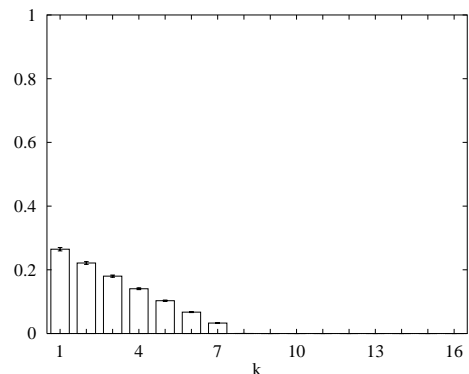
(b)

Figure 3. The structure of functionality landscapes: the distribution of estimated amplitude spectra by classes, for both, (a) one-point mutation and (b) uniform crossover.

To gather statistics of the landscapes on the outlined landscape families we performed 1,000 random walks per



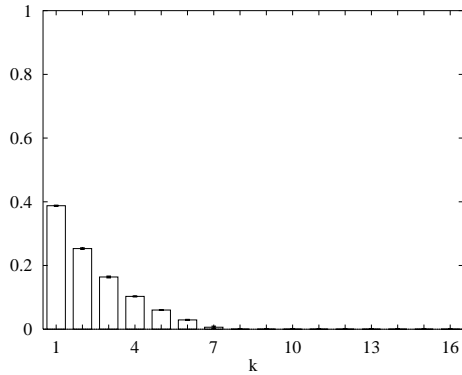
(a)



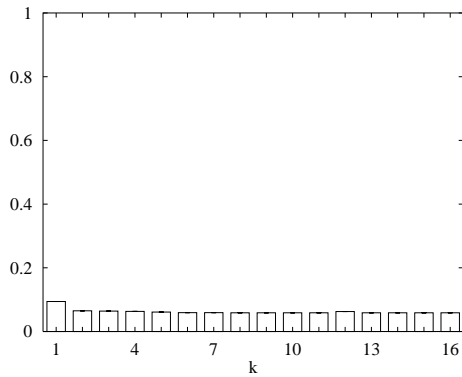
(b)

Figure 4. The structure of functionality landscapes: the representative of the largest class (Figure 3), for both, (a) one-point mutation and (b) uniform crossover.

family, each of them with length 100,000. The initial genotypes were chosen randomly, with fitness values uniformly distributed in the interval $[0, 1]$. Thus, for each family of landscapes, we performed random walks on 1,000 instances. The random walk on the mutation landscape is implemented as follows: start from a randomly chosen point on the landscape, generate all neighbours of the current point by mutation and evaluate their fitness values, choose randomly one neighbour and record its fitness, generate all neighbours of the new point, which becomes “current”, and so on [24]. The random walk on the crossover landscape is different. In this case, the random walk is implemented by mapping \mathcal{R} , (section 2, see also [21] (pp. 248)). The algorithm is given by [23] and can be described as follows: start with an arbitrary pair of parents, generate a set of offsprings by \mathcal{R} (applying the crossover operator) and evaluate their fitness values, from those choose randomly one (record its



(a)

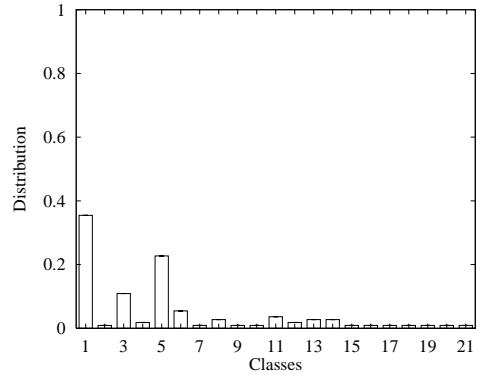


(b)

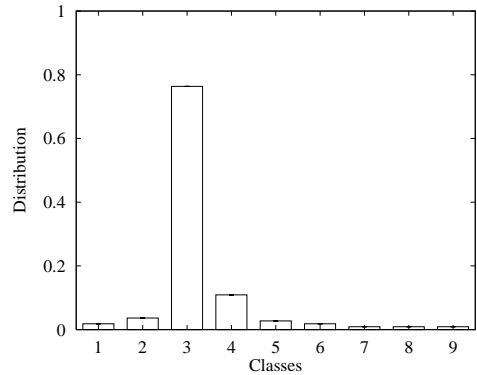
Figure 5. The structure of functionality landscapes: the representative of the most rugged subspaces, for both, (a) one-point mutation and (b) uniform crossover.

fitness) and mate it with a randomly chosen genotype, etc. until the termination conditions are satisfied. The walks were performed with respect to the investigated genotype partitions. We calculated the amplitudes, using the steepest descent method for different starting points, and the results were not noticeably influenced by the start point. Thus, for each landscape family the estimated amplitudes were found to depend solely on the remainder of the genotype. This was expected, since relations between the functionality, internal and output connectivity of the array exist. In order to explore this relationship we employed a classification algorithm by which groups of similar amplitude spectra were identified. The method, similar to that employed in [7], is based on the *crystal algorithm* given by [17]. The algorithm starts with a class containing a single vector (amplitude spectrum), and it expands, when the Euclidean distance between a non-classified vector and a class representative is

less than a certain tolerance. The class representative is a vector with components averaged over all instances of the class. When a vector is classified, a procedure for shrinking the class is started which checks whether the distance between the new representative and the elements of the class is less than the tolerance, mentioned above. In our experiments, the tolerance was set to 0.001.



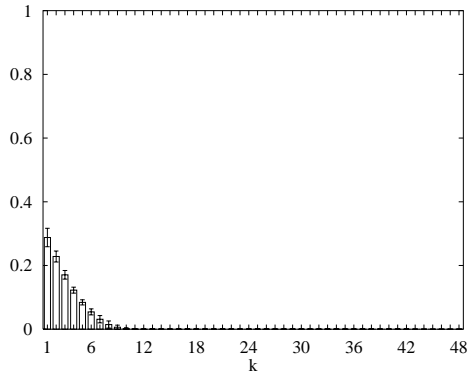
(a)



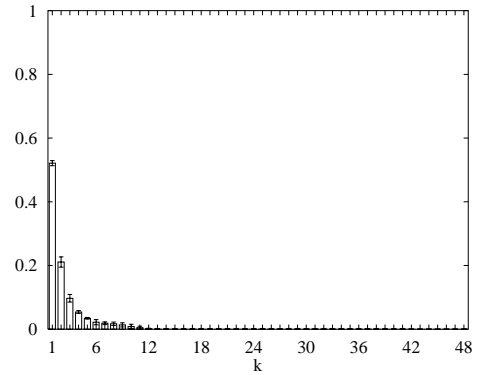
(b)

Figure 6. The structure of internal connectivity landscapes: the distribution of estimated amplitude spectra by classes, for both, (a) one-point mutation and (b) uniform crossover.

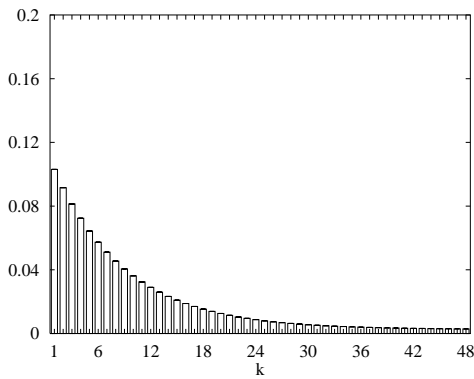
The results for functionality, internal, and output connectivity are depicted in Figures 3–5, 6–8, and 9–11, respectively. The figures represent the distribution of estimated amplitude spectra by classes (Figures 3, 6, and 9), the representative of the largest class (Figures 4, 7, and 10), and the representative of the most rugged subspaces (Figures 5, 8, and 11), for both, (a) one-point mutation and (b) uniform crossover. It is shown that the functionality and input connectivity crossover landscapes are very rugged since spectra with non-zero amplitudes for all values of k were found,



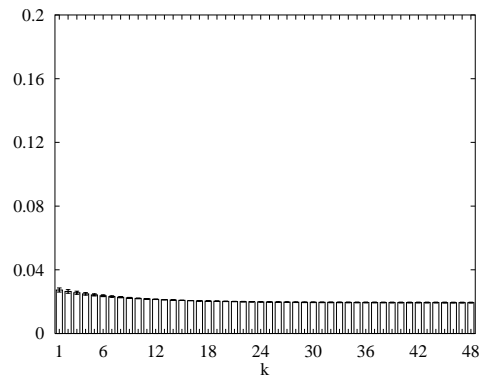
(a)



(a)



(b)



(b)

Figure 7. The structure of internal connectivity landscapes: the representative of the largest class (Figure 6), for both, (a) one-point mutation and (b) uniform crossover.

Figure 8. The structure of internal connectivity landscapes: the representative of the most rugged subspaces, for both, (a) one-point mutation and (b) uniform crossover.

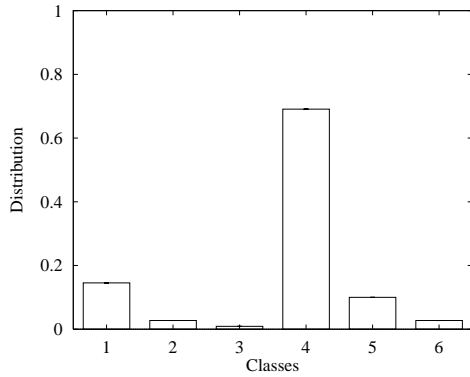
which explains why the autocorrelations of these landscapes were low [22]. On the contrary, the output connectivity crossover landscapes are much smoother since only the first two amplitudes contribute to their ruggedness.

The amplitude spectra of the mutation landscapes are different. The figures reveal that only the first 7, 12, and 2 amplitudes may respectively contribute to the ruggedness of functionality, internal, and output connectivity landscapes. This also explains why the internal connectivity landscapes are much more rugged than the other ones. We surmise that the number of non-zero amplitudes of functionality and internal connectivity landscapes are strongly related to the levels-back parameter. The amplitude spectra of the output connectivity landscapes generated by mutation suggest that the landscapes are smooth, and therefore, easy for search.

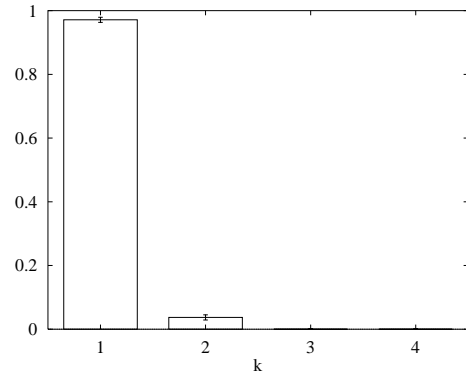
The relationship between the functionality, internal, and output connectivity of the array is revealed by the ratio

of the number of identified classes, which is 15: 21: 6 and 4: 9: 3 for mutation and crossover, respectively. We noticed that the ratio does not change significantly when the classification is performed with a lower tolerance. Therefore, the structure of internal connectivity landscapes most strongly depend on the functionality and output connectivity chromosomes.

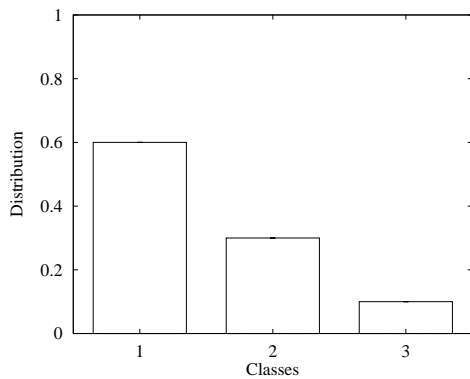
Figures 7 and 8 reveal that the amplitude spectra of the typical and most rugged subspaces of the internal connectivity landscapes differ each other in a significant range. It is also valid for the crossover functionality landscapes (Figures 4 and 5). It indicates that a relationship between the structure of the outlined landscape families and the epistasis of the corresponding genotype partitions exist. For instance, the structure of output connectivity landscapes, characterised in Figures 9–11, is only slightly dependent on the remainder of the genotype. The figures reveal that the



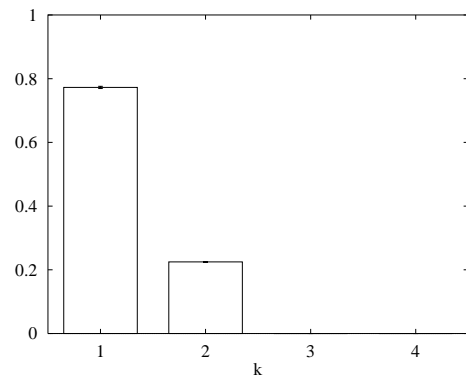
(a)



(a)



(b)



(b)

Figure 9. The structure of output connectivity landscapes: the distribution of estimated amplitude spectra by classes, for both, (a) one-point mutation and (b) uniform crossover.

Figure 10. The structure of output connectivity landscapes: the representative of the largest class (Figure 9), for both, (a) one-point mutation and (b) uniform crossover.

more rugged landscape is, the stronger the dependence on the structure of the other landscapes.

6 Conclusions

The aim of this work is to study the feasibility of evolving digital circuits on an array of logic gates. Figures 3–5 and 6–8 indicate that the uniform crossover is not the right operator for evolving the functionality and internal connectivity of an array of cells, since the corresponding landscapes are extremely rugged. The ruggedness is due to the non-zero contribution of all amplitudes in the entire spectrum, especially noticeable for the more rugged functionality and internal connectivity landscapes. On the contrary, the amplitude spectra of the mutation landscapes reveal that these landscapes have only contributions of the first several amplitudes. Consequently, the mutation landscapes appear

to be relatively smooth, and therefore, they are feasible for evolutionary search.

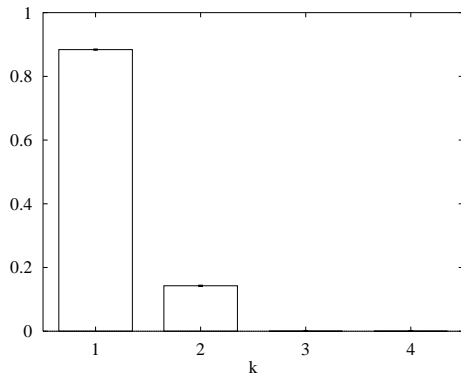
The paper gives a rise of several questions for future research. How are the amplitude spectra of the two-bit multiplier landscapes related to the characteristics of the circuit? How will our findings relate to the much more difficult problem of evolving the three-bit multiplier?

Acknowledgements

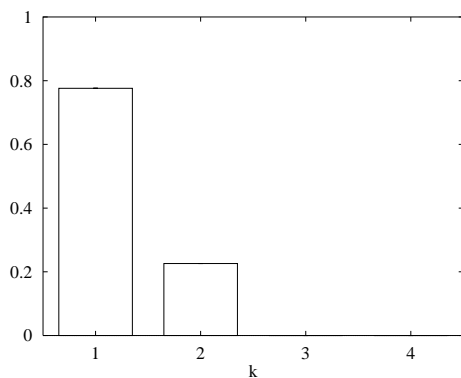
The first author would like to thank Dominic Job (Napier University) for helping to classify the identified amplitude spectra.

References

- [1] P. B. Andrews. *An Introduction to Mathematical Logic and*



(a)



(b)

Figure 11. The structure of output connectivity landscapes: the representative of the most rugged subspaces, for both, (a) one-point mutation and (b) uniform crossover.

Type Theory: To Truth Through Proof. Academic Press, Orlando, Florida, 1986.

- [2] R. L. Burden and J. D. Faires. *Numerical Analysis.* Brooks/Cole Publishing Company, Pacific Grove, CA, 1997. sixth edition.
- [3] X. Chen and S. L. Hurst. A comparison of universal-logic-module realizations and their application in the synthesis of combinatorial and sequential logic networks. *IEEE Transactions on Computers*, C-31(2):140–147, 1982.
- [4] J. Holland. *Adaptation in Natural and Artificial Systems.* MIT Press, Cambridge, MA, 1992. second edition.
- [5] W. Hordijk. Correlation analysis of the synchronising-ca landscape. *Physica D*, 107:255–264, 1997.
- [6] W. Hordijk and P. F. Stadler. Amplitude spectra of fitness landscapes. *J. Complex Systems*, 1:39–66, 1998.
- [7] D. Job, V. Shankaraman, and J. F. Miller. Combining cbr and ga for designing fpgas. In H. Selvaraj and B. Verma, editors, *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*, Piscataway, NJ, 1999. IEEE Computer Society Press. To appear.
- [8] T. Jones. *Evolutionary Algorithms, Fitness Landscapes and Search.* PhD thesis, University of New Mexico, Albuquerque, NM, 1995.
- [9] S. Kauffman. Adaptation on rugged fitness landscapes. In D. Stein, editor, *Lectures in the Sciences of Complexity*, SFI Studies in the Sciences of Complexity, pages 527–618. Addison-Wesley, Reading, MA, 1989.
- [10] B. Manderick, M. de Weger, and P. Spiessens. The genetic algorithm and the structure of the fitness landscape. In R. K. Belew and L. B. Booker, editors, *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 143–150, San Mateo, CA, 1991. Morgan Kaufmann.
- [11] J. F. Miller and P. Thomson. Aspects of digital evolution: Evolvability and architecture. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature V*, volume 1498 of *Lecture Notes in Computer Science*, pages 927–936, Berlin, 1998. Springer.
- [12] J. F. Miller and P. Thomson. Aspects of digital evolution: Geometry and learning. In M. Sipper, D. Mange, and A. Pérez-Uribe, editors, *Proceedings of the 2nd International Conference on Evolvable Systems: From Biology to Hardware*, volume 1478 of *Lecture Notes in Computer Science*, pages 25–35, Heidelberg, 1998. Springer-Verlag.
- [13] J. F. Miller, P. Thomson, and T. Fogarty. Designing electronic circuits using evolutionary algorithms. arithmetic circuits: A case study. In D. Quagliarella, J. Periaux, C. Poloni, and G. Winter, editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science*, pages 105–131. Wiley, Chechester, UK, 1997.
- [14] M. Mitchell, S. Forrest, and J. Holland. The royal road for genetic algorithms: Fitness landscapes and ga performance. In J. Varela and P. Bourguine, editors, *Proceedings of the 1st European Conference on Artificial Life*, pages 245–254, Cambridge, MA, 1991. MIT Press.
- [15] M. Sipper, E. Sanchez, D. Mange, M. Tomassini, A. Pérez-Uribe, and A. Stauffer. A phylogenetic, ontogenetic, and epigenetic view of bio-inspired hardware systems. *IEEE Transactions on Evolutionary Computation*, 1(1):83–97, 1997.
- [16] V. Slavov and N. Nikolaev. Genetic algorithms, fitness sublandscapes and subpopulations. In C. Reeves and W. Banzhaf, editors, *Foundations of Genetic Algorithms*, volume 5. Morgan Kaufmann, San Francisco, CA, 1998. To appear.
- [17] S. Soderland, D. Fisher, J. Asetline, and W. Lehnert. Crystal: Inducing a conceptual dictionary. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, San Francisco, CA, 1995. Morgan Kaufmann.
- [18] P. F. Stadler. Towards theory of landscapes. In R. López-Pena, R. Capovilla, R. García-Pelayo, H. Waelbroeck, and F. Zertuche, editors, *Complex Systems and Binary Networks*, pages 77–163. Springer-Verlag, Berlin, 1995.
- [19] P. F. Stadler and W. Grünter. Anisotropy in fitness landscapes. *J. Theor. Biol.*, 165:373–388, 1993.
- [20] P. F. Stadler and R. Happel. Random field models for fitness landscapes. Technical Report 95-07-069, Santa Fe Institute, 1998. to appear in *J. Math. Biol.*

- [21] P. F. Stadler and G. P. Wagner. Algebraic theory of recombination spaces. *Evolutionary Computation*, 5(3):241–275, 1997.
- [22] V. K. Vassilev, J. F. Miller, and T. C. Fogarty. Digital circuit evolution and fitness landscapes. In *Proceedings of the Congress on Evolutionary Computation*, Piscataway, NJ, 1999. IEEE Press. To appear (available via <http://www.dcs.napier.ac.uk/~vesselin/papers/cec99.ps.gz>).
- [23] G. P. Wagner and P. F. Stadler. Complex adaptations and the structure of recombination spaces. In C. Nehaniv and M. Ito, editors, *Algebraic Engineering*. World Scientific, Singapore, 1998. Santa Fe Institute Report 97-03-029.
- [24] E. D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.
- [25] S. Wright. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In D. F. Jones, editor, *Proceedings of the 6th International Conference on Genetics*, volume 1, pages 356–366, 1932.
- [26] X. Yao and T. Higuchi. Promises and challenges of evolvable hardware. *IEEE Transactions on Systems, Man and Cybernetics*, 28(4), 1998.