

Portland State University
PDXScholar

Systems Science Faculty Publications and
Presentations

Systems Science

2004

Reconstructability Analysis Detection of Optimal Gene Order in Genetic Algorithms

Martin Zwick

Portland State University, zwick@pdx.edu

Stephen Shervais

Eastern Washington University

Let us know how access to this document benefits you.

Follow this and additional works at: http://pdxscholar.library.pdx.edu/sysc_fac

 Part of the [Systems Architecture Commons](#), and the [Theory and Algorithms Commons](#)

Citation Details

Zwick, M., and Shervais, S. (2004). "Reconstructability Analysis Detection of Optimal Gene Order in Genetic Algorithms." *Kybernetes*, vol. 33, No. 5/6, pp. 1052-1062.

This Post-Print is brought to you for free and open access. It has been accepted for inclusion in Systems Science Faculty Publications and Presentations by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

RECONSTRUCTABILITY ANALYSIS DETECTION OF OPTIMAL GENE ORDER IN GENETIC ALGORITHMS

Martin Zwick
Portland State University
Systems Science Ph.D. Program
Portland, OR 97201
zwickm@pdx.edu

Stephen Shervais
Eastern Washington University
College of Business and Public Administration
Cheney, WA 99004
sshervais@ewu.edu

KEYWORDS

Reconstructability analysis, transposition, genetic algorithms, crossover, optimization

ABSTRACT

The building block hypothesis implies that genetic algorithm efficiency will be improved if sets of genes that improve fitness through epistatic interaction are near to one another on the chromosome. We demonstrate this effect with a simple problem, and show that information-theoretic reconstructability analysis can be used to decide on optimal gene ordering.

I. INTRODUCTION

Holland's schema theorem and the building block hypothesis suggest that the performance of a genetic algorithm (GA) might be improved if genes exhibiting epistasis, i.e., genes having a strong interaction effect in their effect on fitness, are near one another on the chromosome. Genes that are close are less likely to be separated by the crossover operator, and alleles that have high fitness can constitute a building block for further evolution. Epistasis may thus imply the existence of an optimal gene order for a GA. This suggests two questions. First, can it be shown that GAs work better if epistatically linked genes are close to one another? Second, if such an effect exists, is it possible to extract information from data produced by the GA, so that we can modify the gene order (using a new genetic operator) to improve GA performance?

In this paper, after describing the schema and building block hypotheses and their relevance to epistasis, we demonstrate the existence of a gene order effect in a very simple problem (Section II). We then show that the methodology of reconstructability analysis can be used to discover preferred gene orders even from GA data produced by less preferred gene orders (Section III). Finally, we discuss the results of these preliminary experiments and point to areas for future exploration (Section IV).

II. SCHEMA THEOREM, BUILDING BLOCKS, AND EPISTASIS

The schema theorem was first proposed by Holland (1975) as a description of how adaptive systems "persistently test and incorporate structural properties associated with better performance (p. 66)." Although there is now some doubt as to how well it describes the dynamics of the GA search process (Thornton, 1997; Mitchell, 1995), it is still useful as a conceptual device, and we use it that way here. According to the theorem, GAs work by parallel testing of multiple combinations of bit strings made up of the available alleles. In the typical binary chromosome, the alleles may be represented as 1, 0, and * (don't care). Thus, 110***11 is a schema (call it S1) of defining length eight and cardinality five. Note that S1 contains a shorter schema (S2), 110*****, with a defining length and cardinality of three, and a third schema (S3), *****11. In fact, a schema of defining length eight has 3^8 possible schemata embedded in it, but we shall here discuss just these three.

If strings containing S2 have a higher-than-average fitness, then they will be preferentially selected, and S2 will act as a *building block* that can be assembled with other building blocks to create longer schemata and higher fitness bitstrings. Since the ratio of the defining length to the cardinality is low, S2 is not likely to be broken up by the crossover operator. The same argument applies to S3. Now consider S1. If bitstrings containing this schema have a higher than average fitness, they will be preferentially selected as well. However, since the defining length of S1 is large relative to its cardinality, it also stands a higher chance of being broken up during crossover. If S2 and S3 are both important to the fitness of S1, we would be better off changing the representation so that S2 and S3 are close together. In other words, if S1 has high fitness, it would be more likely to survive recombination if we had some good reason to move the 11 alleles over to be adjacent to the 110 alleles, i.e., to recode the genome so that this schema was 11011***.

Although the usefulness of short building blocks has long been understood, only a few researchers have

addressed the issue of how changing gene order might facilitate reaching enhanced fitness. Barbara McClintock is credited with discovering the importance of gene transposition in nature (McClintock, 1987). This established transposition as a possible genetic operator available for use by GA researchers. Goldberg, et al. (1993) developed the “fast messy” GA, which, among other things, allows the GA to evolve gene locations on the chromosome. They did this by coding stretches of the chromosome with a gene identifier, which specified the gene that part of the chromosome represented. A given gene might start out overexpressed in a chromosome, because its identification code appears at two different locations. The program selects the first instance of the gene and ignores the rest. Alternatively, a gene might be underexpressed if it does not appear in the bitstring at all. The program then applies a default template to supply the missing gene values. As evolution proceeds, and the length of the GA is allowed to change from long to short and back to long again, those bitstrings with efficient gene orders will be preferentially selected. Beasley et al. (1993) used *a priori* knowledge to code interactive genes into *sub-problems*, which are subject to separate evolutionary processes and are recombined each generation. This requires that some exogenous process identify the sub-problems. Simoes and Costa (1999) examined the usefulness of McClintock’s transposons as a replacement for the crossover operator. In their work, randomly selected runs of bitstrings were moved about on the chromosome. No effort was made to record which bitstrings worked best together.

The impact of one gene on the fitness contribution of another is called *epistasis*. In the schema S1 discussed above, assume that the high fitness of S1 derives from an epistatic interaction between S2 and S3, and not merely from the separate high fitnesses of these two schemas. This would be all the more reason for S2 and S3 to be adjacent to one another and constitute a compact building block.

The matter might be more complex, however. While the tight coupling of high epistatic genes into building blocks might seem at first glance to be an unalloyed good, further reflection shows the advantage of repositioning genes on our illustrative chromosome accrues only *after* the good 110 and 11 alleles first occur on the genome, after which preservation of these alleles as a building block becomes advantageous. A different, indeed opposite, argument might apply to the process of searching for high fitness schemata. During the early generations, the GA is still searching for good combinations of alleles, and crossover is the primary tool for searching out novel combinations. If the 110 and 11 alleles exist on two different parental chromosomes, they are more likely to be recombined as the result of crossover if the genes are distant

from one another. In the work reported in this paper, however, we have observed only the benefits gained by placing epistatically linked genes close to one another. This issue is addressed further in the Discussion section.

III. DEMONSTRATING GENE ORDER EFFECTS IN A GENETIC ALGORITHM

We here demonstrate the possibility of a gene order effect by using an extremely simple fitness function, namely the function (to be maximized) specified by equation 1.

$$F = \min(A/B, B/A) * C \quad (1)$$

where A, B, and C take on values between 0 and 3.0. The minimization operation thus constrains the AB term to values less than or equal to 1.0 and fitness, F, to the range 0.0 to 3.0. The epistatic nature of the problem arises from the fact that the AB term is maximized (at 1.0) only if A and B are equal. The C variable has no impact on the AB term, and contributes to overall fitness in simple proportion to its value. From a theoretical standpoint, focusing exclusively on the imperative of retaining good building blocks, one would expect that a chromosome where the variables A and B were side by side would allow the GA to perform more efficiently than on with A and B separated by C. Thus, in the six ways of ordering A, B, and C, four are expected to be *good* orders (ABC, BAC, CAB, CBA), and two are expected to be *bad* orders (ACB, BCA).

The Genetic Algorithm we used employed standard binary encoding, with three 8-bit genes and a chromosome length that varied depending upon the requirements of the experiment. The GA parameters for all experiments included: population 30, generations 30, mutation rate 0.01, crossover rate 1.0, and repetitions 100, with a new random seed for each repetition. Crossover was single point, and occurred either at the gene boundary only, or any place on the chromosome, depending upon the experiment. All six possible gene orders (four good, two bad) were tested.

Results of the experiments are shown in figures 1-3. For each experiment, the results from the 100 runs of the six gene orders hypothesized to be *good* (those like CBA, that kept A and B together) were averaged together. Results from runs of the two gene orders hypothesized to be *bad* were also averaged.

Three experimental setups were used. In the first setup, the chromosome length was set to 24 (short chromosome), and crossover was only allowed at the gene boundaries. All of the genes were *exons*, that is, they all expressed values used in the solution of the problem. For the second and third setups, the chromosome length was

increased by the introduction of 108 bits of non-coding *introns* to the right of each gene (long chromosome).

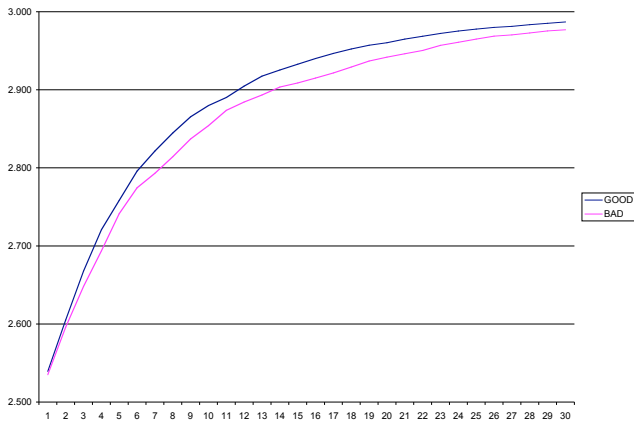


Figure 1. GA effectiveness on short chromosomes when crossover is allowed at any place on the bitstring. Effectiveness of good orders, where linked genes A and B are adjacent, is compared to effectiveness of bad orders, where linked genes are separated by gene C. All genes are exons.

The second experiment retained crossover at the gene boundary only, while the third allowed crossover anywhere on the length of the chromosome. Only the twenty-four bits in the three genes were exons.

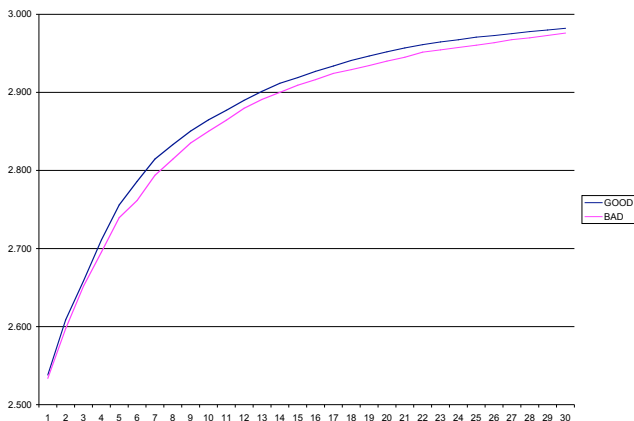


Figure 2. GA effectiveness on long chromosomes with introns and crossover allowed at any place on the bitstring. Effectiveness of good orders, where linked genes A and B are adjacent, is compared to effectiveness of bad orders, where linked genes are separated by gene C. Expressed genes (exons) are separated by 108 bits of non-expressed genes (introns).

Figures 1-3 demonstrate that a small but definite improvement in the performance of the GA can be attained if genes are ordered optimally, i.e., if A and B are not separated by C. The effect is small, but the genome itself is small, so a large gene order effect is not to be expected.

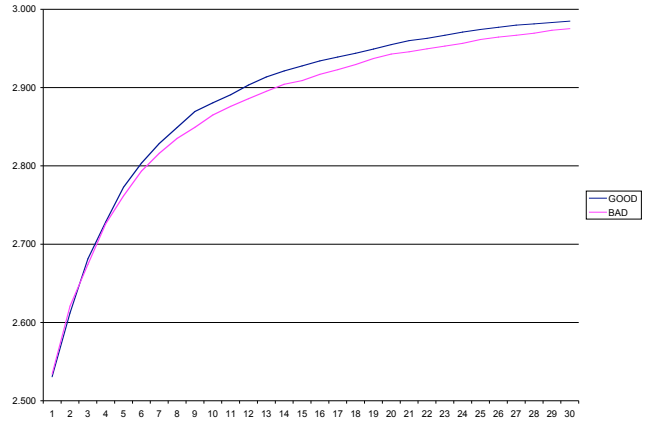


Figure 3. GA effectiveness on long chromosomes with crossover at gene boundaries. Effectiveness of the good orders, where linked genes A and B are adjacent, is compared to effectiveness of bad orders, where linked genes are separated by gene C. Expressed genes (exons) are separated by 108 bits of non-expressed genes (introns).

IV. DETECTING OPTIMAL GENE ORDER BY RECONSTRUCTABILITY ANALYSIS

Assuming then that gene order matters, and that it might matter more dramatically for more complex genomes and fitness functions, the challenge is to find out what the optimum gene order is. In this section we show that this determination is in fact achievable. Information on $F(A,B,C)$ is generated by the GA, and this information can be analyzed to find the optimal gene order, *even when the GA is initially implemented with a non-optimal order*. We here show that this can be done using the methods of reconstructability analysis.

Reconstructability Analysis

Reconstructability analysis (RA) derives from Ashby (1964), and was developed by Broekstra, Cavallo, Cellier, Conant, Jones, Klir, Krippendorff, and others; an extensive bibliography is available in (Klir, 1986), and a compact summary of RA is available in (Zwick, 2000). RA resembles log-linear (LL) methods (Bishop et al, 1978; Knoke & Burke, 1980), used widely in the social sciences, and where RA and LL methodologies overlap they are equivalent (Knoke & Burke, 1980; Krippendorff, 1986). In

RA (Klir, 1985), a probability or frequency distribution or a set-theoretic relation is decomposed (compressed, simplified) into component distributions or relations. ABC might thus be decomposed into AB and BC projections, written as the structure, AB:BC. The two linked bivariate distributions (or relations) constitute a model of the data.

RA can model problems both where “independent variables” (inputs) and “dependent variables” (outputs) are distinguished (*directed* systems) and where this distinction is not made (*neutral* systems). In the present case, we have a directed system, with independent variables (genes) A, B, and C, where the dependent variable is the fitness value, F. Consider “regression models” (Krippendorff, 1986) where there is no overlap between genes in the separate components of the model. These models are ABF:CF, ACF:BF, BCF:AF, and AF:BF:CF. Interaction between two variables (i.e. an epistatic link, an “interaction effect”), is indicated when the model places the two input variables next to one another along with output F. So ABF:CF indicates that A and B together contribute to the fitness F separately from the contribution made by C to F. It is also useful to look at “chain models” (Krippendorff, 1986), which feature overlapping of input variables (like ABF:BCF). Chain models do not yield disjoint subproblems, but they indicate particular orders of variables on the chromosome, e.g., ABF:BCF corresponds to order ABC.

RA Calculations

Calculations were made using the RA software programs developed at Portland State University, now integrated into the package OCCAM (for the principle of parsimony and as an acronym for “Organizational Complexity Computation And Modeling”). The earliest of these programs was developed by Zwick and Hosseini (Hosseini, Harmon, & Zwick, 1986); a review of RA methodology is offered in (Zwick, 2001a); a list of recent RA papers is given in (Zwick, 2001b). The reconstructability analysis was conducted on a dataset generated by multiple runs of the GA, *using the two bad orders only*. Results are shown in Table 1.

Using the same parameters as experiment 1, these runs first saved *all* members of the population, in excess of ten thousand records. Then, to select data associated with the most fit solutions, the highest scoring members of that population were extracted. The cutoff point was a fitness of at least 2.0, and a total of 7,800 records resulted. Values of A,B,C, and F were then discretized into 5 equally spaced bins, and the results were analyzed by the *OCCAM* software package.

Table 1. Reconstructability Analysis Results. I is the information captured in the model, relative to 100% knowledge of F for the top model ABCF where the joint dependence of F on all genes is known, and 0% knowledge of F for the bottom model, where A,B, and C are all unknown. Regression models are in bold, and chain models are in italics. Other models are shown in smaller font.

ABCF	1.000
ABF:ACF:BCF	0.999
<i>ABF:BCF</i>	<i>0.999</i>
<i>ABF:ACF</i>	<i>0.994</i>
ABF:CF	0.980
<i>ACF:BCF</i>	<i>0.659</i>
AF:BCF	0.655
ACF:BF	0.653
AF:BF:CF	0.639
BCF	0.631
BF:CF	0.616
ACF	0.610
AF:CF	0.591
CF	0.554
ABF	0.358
AF:BF	0.092
BF	0.067
AF	0.039
F	0.000

The RA results of Table 1 show that models corresponding to good gene orders are clearly superior to those with bad gene orders. Consider first the regression models, shown in the table in bold. These models assess the possible partitions of the problem into disjoint subproblems. Of the 4 regression models, ABF:CF is clearly the best, indicating that A and B are epistatically linked, while C make an independent contribution to fitness. This suggests that A and B should be placed near one another. Consider now the chain models, shown in the table in italics, which directly indicate how well different gene orders fit the data. ABF:BCF and ABF:ACF, corresponding to orders ABC (and CBA) and BAC (and CAB), respectively, are the best models, in agreement with the implications of the regression models. These models as well support the idea that A and B should be adjacent.

V. DISCUSSION

For the simple test problem shown, where a part of the solution depends on the interaction of epistatic genes, the good orders (those that kept epistatic genes together)

found better solutions faster than did orders that separated the epistatic genes. The gene order effect was small, but in more problems with more variables, it may become more substantial. The relative effectiveness of the two sets of orders changed throughout the experiments. At the beginning, roughly the first five generations, the bad orders performed about the same as the good ones. For the next twenty generations the good orders performed better. In the end game, when both were approaching the solution asymptotically, the bad orders slowly caught up, but were still behind the performance of the good orders at the end.

It was noted above in Section II that the impact of separation on epistatic genes might be more complex than what is suggested by the main results of this paper. Specifically, one might expect that in the early phases of a GA run, epistatically linked genes should best be located far from one another. This is based on the supposition that at the beginning of the search it may be useful for all genes to be mixed as much as possible by the crossover operator. If two epistatic genes are side-by-side from the beginning, then crossover would have less chance of improving them, and the GA would have to depend upon a good initialization and fortunate mutations to create the best gene pair possible. If, on the other hand, two epistatic genes start out well separated, the crossover operation might more easily assemble a larger selection of allele patterns in the two genes. These expectations are under continuing investigation, but so far we have not seen any clear evidence for them, i.e., for the better performance *at the beginning of GA runs* of orders which separate A and B. Our runs start out with good and bad order nearly equivalent in performance. At some point, the separation of A and B by C in "bad" orders definitely becomes a handicap, and that these orders fall behind the others.

Reconstructability analysis allows one to find the simplest models that retain high information about the data. The top model, ABCF, includes interactions among all variables, but Table 1 shows that ABF:CF has virtually complete information (98%), so solving the ABF and CF subproblems separately and merging the answers would probably give a good result. We suggest that this might be a way to solve Beasley et al.'s problem of *a priori* identification of subproblems for expansive coding. Using RA to decompose optimization problems into subproblems might of course also be useful for optimization methods other than the GA.

The success of this experiment means that in principle we have a way to restructure the genome of a GA based on data that the GA itself generates. This might speed up processing in a particular GA run, if the optimum order can be detected early enough for the GA to gain an

advantage from gene reordering. It could also offer a way to build a GA that is optimized for a specific type of problem. To use a real-world example, one of the authors recently studied the use of a GA to solve an inventory and distribution problem (Shervais, 2000a, 2000b). One would expect that any set of such problems with the identical number and structure of nodes and stocks could use information generated by the first specific problem to be addressed. Alternatively, we may be able to apply RA to binned data to prestructure the genome for optimizing the fitness of the original unbinned variables. This may become useful as we search for more complex problems to test this approach on.

VI. BIBLIOGRAPHY

- Ashby, W. R. 1964. "Constraint Analysis of Many-Dimensional Relations." *General Systems Yearbook*, 9, pp. 99-105.
- Beasley, D., D. Bull, and R. Martin "Reducing Epistasis in Combinatorial Problems by Expansive Coding." In: *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufman Publishers, San Mateo, 1993, pp. 400-407.
- Bishop, Y., S. Feinberg, and P. Holland, 1978. *Discrete Multivariate Analysis*. MIT Press, Cambridge.
- Goldberg, D., K. Deb, H. Kargupta, and G. Harik, *Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithms*. IlliGAL Report No. 93004, University of Illinois, 1993.
- Hosseini, J.C., R. R. Harmon, and M. Zwick, "Segment Congruence Analysis Via Information Theory." *Proceedings, International Society for General Systems Research*, Philadelphia, PA, May 1986, pp. G62 - G77.
- Holland, J. 1975. *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor.
- Klir, G. 1985. *The Architecture of Systems Problem Solving*. Plenum Press, New York.
- Klir, G. 1986. "Reconstructability Analysis: An Offspring of Ashby's Constraint Theory." *Systems Research*, 3 (4), pp. 267-271.
- Knoke, D. and P.J. Burke 1980. *Log-Linear Models*. (Quantitative Applications in the Social Sciences Monograph # 20). Sage, Beverly Hills.

Krippendorff, K. 1986. "Information Theory: Structural Models for Qualitative Data." (Quantitative Applications in the Social Sciences #62). Sage, Beverly Hills.

McClintock, B. 1987. *The discovery and characterization of transposable elements: the collected papers of Barbara McClintock*, Garland, New York.

Mitchell, M. 1996. *An Introduction to Genetic Algorithms*. MIT Press, Cambridge.

Shervais, S 2000a. "Developing Improved Inventory And Transportation Policies For Distribution Systems Using Genetic Algorithm And Neural Network Methods." In: *Proceedings of the World Conference on the Systems Sciences*, Toronto, Canada, pp. 200059-1 to 200059-17.

Shervais, S. 2000b. *Adaptive Critic Design of Control Policies for A Multi-Echelon Inventory System*. Ph.D. dissertation, Portland State University. Available at: http://www.cbpa.ewu.edu/~sshervais/Personal_Info/Papers/Dissertation.ps.zip

Simoes, A. and E. Costa 1999. "Transposition: {A} Biologically Inspired Mechanism to Use with Genetic Algorithms." In: *Proceedings of the Fourth International Conference on Neural Networks and Genetic Algorithms (ICANN'99)*, Portoroz, Slovenia. Springer Verlag, Berlin, pp. 178-186.

Stadler, P., R. Seitz, and G.P. Wagner 2000. "Population Dependent Fourier Decomposition of Fitness Landscapes over Recombination Spaces: Evolvability of Complex Characters" *Bulletin of Mathematical Biology*, Vol. 62, No. 3, pp. 399-428.

Thornton, C. "The building block fallacy." *Complexity International* 4 (1997).
<http://www.csu.edu.au/ci/vol04/thornton/building.htm>

Zwick, M. 2001a. "Wholes and Parts in General Systems Methodology." In: *The Character Concept in Evolutionary Biology*, edited by Gunter Wagner. Academic Press, New York, 2001a, pp. 237-256.

Zwick, M. (2001b). "Discrete Multivariate Modeling":
http://www.sysc.pdx.edu/res_struct.html

VII. BIOGRAPHIES

Martin Zwick

Martin Zwick is a Professor of Systems Science at Portland State University. Prior to taking his current position at PSU, he was a faculty member in the Department of Biophysics and Theoretical Biology at the University of Chicago, where he worked in macromolecular structure and mathematical crystallography. In the 1970's his interests shifted to systems theory and methodology. Since 1976 he has been on the faculty of the PSU Systems Science Ph.D. Program and during the years 1984-1989 he was the program Coordinator and then Director. His current research interests are in discrete multivariate modeling (reconstructability analysis), "artificial life" and theoretical/computational biology, and systems philosophy.

Stephen Shervais

Steve Shervais is an Associate Professor of Management Information Systems in the Accounting and Information Systems Department of Eastern Washington University. He was awarded his Ph.D. in Systems Science by Portland State University in 2000. Prior to his shift to academia he was an intelligence analyst in the USAF, and a database developer in industry. His main area of research is the application of systems methodologies to the solution of business problems.