A Survey on Chromosomal Structures and Operators for Exploiting Topological Linkages of Genes

Dong-Il Seo and Byung-Ro Moon

School of Computer Science & Engineering, Seoul National University
Sillim-dong, Kwanak-gu, Seoul, 151-742 Korea
{diseo,moon}@soar.snu.ac.kr
http://soar.snu.ac.kr/~{diseo,moon}/

Abstract. The building block hypothesis implies that the epistatic property of a given problem must be connected well to the linkage property of the employed representation and crossover operator in the design of genetic algorithms. A good handling of building blocks has much to do with topological linkages of genes in the chromosome. In this paper, we provide a taxonomy of the approaches that exploit topological linkages of genes. They are classified into three models: static linkage model, adaptive linkage model, and evolvable linkage model. We also provide an overview on the chromosomal structures, encodings, and operators supporting each of the models.

1 Introduction

A genetic algorithm maintains a population of solutions and repeatedly applies genetic operators to find optimal or near-optimal solutions. Through the genetic process, various patterns of genes, called *schemata*, are created, destroyed, and recombined in parallel. Holland named the phenomenon *implicit parallelism* [1]. The schemata of short defining length, low order, and high quality play an important role in the process. They are called *building blocks* and the genetic process is well explained as the juxtaposition of building blocks [2]. It means that the power of a genetic algorithm lies in its ability to find building blocks and to grow them efficiently to larger ones.

The solutions in the population have a structure of genes called *chromosomal structure* for the crossover operator. A typical chromosomal structure is one-dimensional array. But, for many problems, it is known that non-linear chromosomal structures are more advantageous than such a simple structure [3,4,5, 6,7]. One of the reasons of adopting such non-linear structures is that we can minimize the loss of information contained in the given problem with more natural structures for the problem. Another reason is that many crossovers require special type of chromosomal structures. The reasons are ultimately related to the creation and growth of the building blocks. That is, the first one helps the chromosomal structure to reflect well the epistatic property of the problem and

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2724, pp. 1357–1368, 2003.

the second one allows us to be able to apply a crossover operator that creates and recombines well the building blocks.

There has been a significant amount of efforts for designing efficient chromosomal structures, encodings, and operators. In this paper, we provide a survey and bibliography of such studies.

The rest of this paper is organized as follows. The basic concepts are introduced in Sect. 2 and the chromosomal structures are explained in Sect. 3. The approaches concerned with the topic are classified and explained in Sect. 4. Finally, the summary of this paper is provided in Sect. 5.

2 Preliminaries

Building blocks appear in interactive gene groups. We can easily observe that the contribution of a gene to the chromosomal fitness depends on the values of other genes in the chromosome. This phenomenon is called *gene interaction* or *epistasis* [1,8,9]. For example, a bat must be able to hear high frequency ultrasonic waves if it generates high frequency ultrasonic squeaks, and it must be able to hear low frequency ultrasonic waves if it generates low frequency ultrasonic squeaks. So, for bats, the genes related to the organs that generate ultrasonic squeaks and those related to the organs that hear ultrasonic waves have strong interactions.

Building blocks are generated by the crossover and mutation. In this paper, we focus on the crossover. A crossover operator generates an offspring by recombining two parents. There are the number of genes minus one operators in one-point crossover with one-dimensional array representation. The creativity of a crossover is strongly related to the diversity of the operators. Generally, the more operators a crossover has, the more diverse new schemata it can generate [5]. However, existing schemata do not always survive through the crossover. A gene group is said to have a *strong linkage* if the genes are arranged so that the gene group has relatively high survival probability, and it is said to have a *weak linkage*, otherwise [1]. For example, in the case of one-point crossover on a chromosome of one-dimensional array, the strength of the linkage of a gene group is inversely proportional to the defining length of the corresponding schema. For the growth of building blocks to an optimal or near-optimal solution, they have to survive through the genetic process, specifically through the crossover. Consequently, we come to a proposition that mutually interactive genes need to have strong linkages [1,2,10]. In the case of one-point crossover, for example, it is helpful to make the strongly epistatic gene groups have short defining lengths.

The linkage of a gene group is, in many cases, dependent on the distribution of the genes in the chromosome and the crossover scheme. Particularly, there is a group of approaches where each gene is placed in an Euclidean or non-Euclidean space, called *chromosomal space*, to represent the linkages between genes. The linkage in this context is called *topological linkage*. There are many other problem-specific or analytic methods to exploit the linkages between genes. Estimation-of-distribution algorithms (EDAs) [11,12], also called probabilistic model-building genetic algorithms (PMBGAs), are examples. But, they are out of the scope of this paper.

3 Chromosomal Structures

In order to make the topological linkages reflect well the epistatic structure of a given problem, we need to choose an appropriate chromosomal structure. Generally, in representing a graph geometrically, it is known that considerably high dimensions of representation space are necessary to alleviate the degree of distortion [13]. This is a good reason that multi-dimensional representations are more advantageous than simple one-dimensional representations for highly epistatic problems. However, multi-dimensional representations require sophisticated crossover operators.

Figure 1 shows an illustration of several representative chromosomal structures. One-dimensional array (a) is a typical and the most popular chromosomal structure. By linking the two ends, a ringed array (b) is generated. This structure has a virtue of treating all the positions of genes symmetrically. We can also increase the dimension of chromosomes (c)–(d). Two-dimensional array (c) is frequently used. The distortion level in using an array may yet be lowered by adopting a real space. For example, a one-dimensional real space (line) (e) or a two-dimensional real space (plane) (f) may be used. A complete graph (g) is also possible. In the case of a complete graph, the genes are linked by weighted edges. The edge weight is sometimes called *genic distance*.

The chromosomal structures can be classified into two categories: *Euclidean chromosomal structures* and *non-Euclidean chromosomal structures*. The Euclidean chromosomal structures include arrays and real spaces and the non-Euclidean chromosomal structures include complete graphs. It should be noted that the chromosomal structure is not necessarily the same as the data structure used in the encoding. The term means only the conceptual structure of genes used in the crossover operator in this paper.

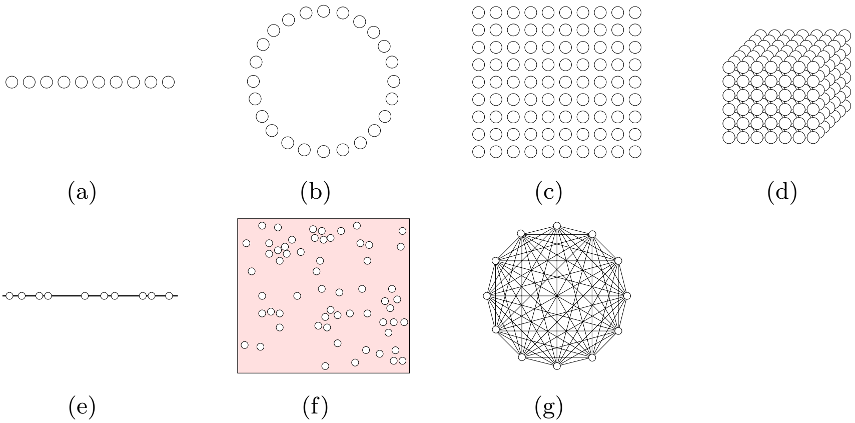

Fig. 1. Various chromosomal structures. (a) 1D array. (b) ringed array. (c) 2D array. (d) 3D array. (e) 1D real space. (f) 2D real space. (g) complete graph.

4 Linkage Models

In general, the interactions between genes are not simple. If they are, the genes can be divided into mutually independent subgroups. It means that we can divide the whole problem into a number of subproblems and can conquer them independently. In most cases, the epistatic structure is so complicated that it is extremely hard to find an optimal arrangement of the genes in the chromosome. For example, when a problem is encoded into n binary genes, the chromosome has totally $n!$ arrangements in an array representation. It means that the problem is more difficult than the original problem with a total of 2^n solutions. To make matters worse, we cannot exactly grasp the epistatic properties of the given problem unless we explore the whole solution space [8].

For decades, various approaches to the problem have been tried. They are divided into two classes: *static linkage model* and *dynamic linkage model.* The linkage is fixed statically during the genetic process in the static linkage model, while it changes dynamically in the dynamic linkage model. The dynamic linkage model is divided again into *adaptive linkage model* and *evolvable linkage model.* In the evolvable linkage model, the linkage itself evolves in parallel with the alleles of genes through the competition of individuals in the genetic process. In the adaptive linkage model, however, the linkage alters but does not evolve in the genetic process. Generally, the linkages are assigned analytically or heuristically using the prior information about the epistatic property of the given problem in the static linkage model. Thus, efficient methods for the assignment are needed. The evolvable linkage model is advantageous for the problems where prior information is not available. Such problems are sometimes called *black box optimization* [32]. The model, however, experienced many failures initially. The failures seem to be due to the loss in the race against allele selection because of the slow evolution speed. As mentioned before, the linkage assignment problem is at least as difficult as the original problem in many cases. So, the model requires efficient methods to speed up the evolution. The model also requires implementation overheads for the evolution of the linkage. The adaptive linkage model is a trade-off between the other two models.

The encodings directly concerned with the linkage issues are “encapsulated” encoding and locus-based encoding. The encapsulated encoding, used by Bagley [14] first, is usually used in the evolvable linkage model with array representation. In the encoding, each gene is an encapsulation of a position indicator and an allele value, i.e., each gene is encoded as (index, value) pair. The encoding allows the crossover operator to contribute to the evolutions of both alleles and linkages at the same time. In the locus-based encoding, the allele values have meaning only when associated with specific positions in the chromosome. In order to use the locus-based encoding in the array representation and the real space representation, a mapping table that maps each gene to a position in the chromosomal space is needed. In order to use the locus-based encoding in the complete graph representation, a table for the genic distances is needed.

The approaches that exploit topological linkages are summarized in Table 1. For each approach, chromosomal structure (CStr), linkage model (Mod), and

Table 1. A summary of the topological linkage-based approaches. For each approach, chromosomal structure (CStr), linkage model (LMod), and encoding (Enc) are given.

Approach	CStr	LMod	Enc	Remark
Bagley, 1967 [14]	1D A	E	E	inversion operator
Goldberg *et al.*, 1989 [15]	1D A	E	E	messy GA
Goldberg *et al.*, 1990 [16]	1D A	E	E	messy GA
Levenick, 1991 [17]	1D A	S	L	non-coding segment
Goldberg *et al.*, 1993 [18]	1D A	E	E	fast messy GA
Bui & Moon, 1993 [19], 1996 [20]	1D A	S	L	DFS/BFS-reordering
Bui & Moon, 1994a [21]	1D A	S	L	DFS-reordering
Bui & Moon, 1994b [22]	1D A	S	L	local optimum reordering
Wu *et al.*, 1994 [23]	1D A	S	L	non-coding segment
Wu & Lindsay, 1995 [24]	1D A	S	L	non-coding segment
Levenick, 1995 [25]	1D A	S	L	metabits
Kargupta, 1996 [26]	1D A	E	E	GEMGA
Bui & Moon, 1998 [27]	1D A	S	L	weighted-DFS reordering
Levenick, 1999 [28]	1D A	S	L	non-coding segment
Knjazew & Goldberg, 2000 [29]	1D A	E	E	OMEGA
Kwon & Moon, 2002 [30]	1D A	A	L	correlation coefficient
Harik & Goldberg, 1996 [31]	RA	E	E	LLGA
Harik, 1997 [32]	RA	E	E	LLGA
Lobo *et al.*, 1998 [33]	RA	E	E	compressed intron LLGA
Harik & Goldberg, 2000 [34]	RA	E	E	prob. expression LLGA
Chen & Goldberg, 2002 [35]	RA	E	E	start expression gene LLGA
Cohoon & Paris, 1986 [3]	2D A	S	L	rectangle-style cross. (ψ_2)
Anderson *et al.*, 1991 [4]	2D A	S	L	block-uniform crossover
Bui & Moon, 1995 [36]	2D A	S	L	Z3
Moon & Kim, 1997 [37]	2D A	S	L	geo. attraction-based embed.
Moon *et al.*, 1998 [38]	2D A	S	L	geographic crossover, DFS-zigzag embedding
Moon & Kim, 1998 [39]	2D A	E	L	two-layered crossover
Kim & Moon, 2002 [40]	2D A	S	L	geographic crossover, neuron reordering
Kahng & Moon, 1995 [5]	2/3D A	S	L	geographic crossover, DFS-row-major embedding
Lee & Antonsson, 2001 [41]	1D RS	E	L	adaptive non-coding seg.
Jung & Moon, 2000 [42], 2002a [6]	2D RS	S	L	natural crossover
Jung & Moon, 2002b [43]	2D RS	S	L	natural crossover
Greene, 2002 [44]	CG	A	L	cutoff-value based crossover
Seo & Moon, 2002 [7], 2003 [45]	CG	S	L	Voronoi quantized crossover

encoding (Enc) are summarized in the table. The chromosomal structures are abbreviated as A (array), RA (ringed array), RS (real space), and CG (complete graph). The linkage models are abbreviated as S (static linkage model), A (adaptive linkage model), and E (evolvable linkage model). The encodings are abbreviated as E (encapsulated encoding) and L (locus-based encoding).

4.1 Static Linkage Models

Raising Dimensions. Cohoon and Paris's work [3] seems to be the first approach published employing multi-dimensional representation. They proposed a rectangle-style crossover (ψ_2) using two-dimensional array representation for VLSI circuit placement. In the crossover, the gene values in a $k \times k$ square section of one parent and the remaining gene values of the other parent are copied into the offspring, where k has a truncated normal distribution with mean 3 and variance 1.

Another approach based on two-dimensional array is the block-uniform crossover proposed by Anderson *et al.* [4]. The crossover divides the chromosome into $i \times j$ blocks where i and j are chosen at random. Then each block of one parent is interchanged randomly with the corresponding block of the second parent based on a preassigned probability. They applied it to Ising problem.

Bui and Moon [36], proposed an extension of the traditional k-point crossover to multi-dimensional array representation, called multi-dimensional k-point crossover (Z3) for grid-placement and graph partitioning. In the crossover, the chromosomal space is divided into hyper-subspaces by k random crossover points, i.e., k_i crossover points are chosen at random for each i^{th} dimension such that $k_i \geq 0$ and $\sum_{i=1}^{n} k_i = k$, and, by them, the chromosomal space is divided into $\prod_{i=1}^{n}(k_i + 1)$ hyper-subspaces. Then the gene values in each adjacent subspaces are copied alternately from one of the two parents into the offspring.

Although multi-dimensional array representations can preserve more epistatic relations between genes than simple one-dimensional array representations, naive multi-dimensional crossovers have a potential weakness in the relatively small number of recombinations. In the case of one-point crossover for n genes, for example, one-dimensional array representation has $n - 1$ operators, while the Z3 on two-dimensional and three-dimensional array representations have only $2(\sqrt{n} - 1)$ and $3(\sqrt[3]{n} - 1)$ operators, respectively. Kahng and Moon [5] proposed a flexible framework to deal with the problem. In their geographic crossover, the chromosomal space is divided into subspaces with monotonic hypersurfaces instead of hyperplanes. The crossover have been applied to graph partitioning [5], neural network optimization [40], and fixed channel assignment [46] recently.

Although multi-dimensional array representations and crossovers have the advantages mentioned above, array-based chromosomal structures have limits to represent non-regular epistatic relationships between genes. Jung and Moon [42,6] focused on the point and proposed 2D real space representation and the natural crossover. In the approach, the 2D image of the problem instance itself is used for crossover and arbitrary simple curves are used for chromosomal cutting. They are applied to 2D traveling salesman problem [6,42] and vehicle routing problem [43].

Figure 2(a)–(e) illustrates the crossovers described above.

Reordering. Bui and Moon [19] proposed the static linkage model that reorders the positions of genes in an array representation before starting the ge-

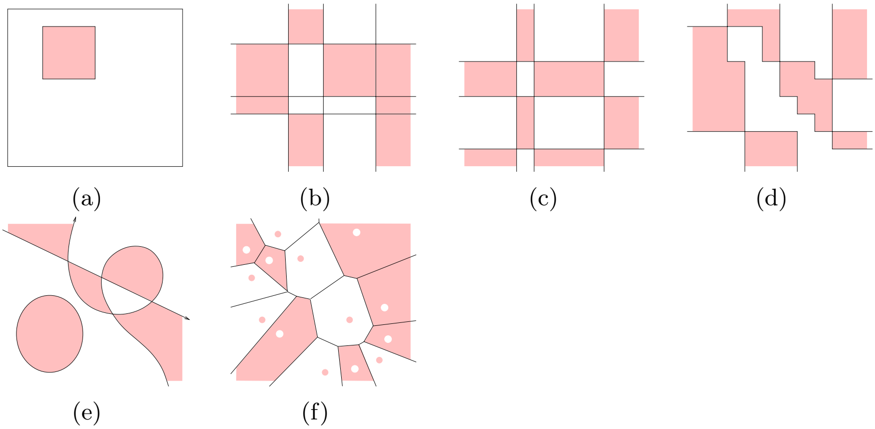

Fig. 2. Illustrations of the non-linear crossover operators. (a) Rectangle-style crossover. (b) Block-uniform crossover. (c) Multi-dimensional k-point crossover (Z3). (d) Geographic crossover. (e) Natural crossover. (f) Voronoi quantized crossover with genic distances defined on 2D Euclidean space.

netic algorithm. The approach is sometimes called *schema preprocessing* or *static reordering*.

Various reordering methods were proposed recently. Many of them are for graph problems and VLSI circuit design. They are breadth-first search (BFS) reordering [19,20], depth-first-search (DFS) reordering [19,20], and weighted-DFS reordering [27] for one-dimensional array representation, and DFS-row-major embedding [5], Euclidean embedding [5], DFS-zigzag embedding [38], and geographic attraction-based embedding [37] for multi-dimensional array representation. Using the order of cities in a local optimum solution for traveling salesman problem [22] and using the correlation coefficients of variables for function approximation [30] are other examples of reordering.

Other Approaches. Another remarkable approach is that of complete graph representation. In the representation, each gene pair is linked by a weighted edge that reflects explicitly the strength of the epistasis between the genes. The edge weights are also called *genic distances*. Seo and Moon [7,45] proposed heuristics for assigning genic distances and Voronoi quantized crossover (VQX) for traveling salesman problem and sequential ordering problem. The heuristics are based on the fact that the epistasis reflects the topological locality of the given cities. In VQX, the chromosomal space defined by the genic distances is divided into k Voronoi regions (nearest neighbor regions) determined by k randomly selected genes, then a sort of block-uniform crossover [4] is performed on the regions. This approach is discriminated from the others in this section in that the chromosomal structure is not necessarily based on a metric space. The crossover is illustrated visually in Fig. 2(f) with the assumption that genic distances are defined on a two-dimensional Euclidean space. The assumption is merely for the visualization.

It is known that the non-coding DNA, sometimes called *introns*, found in biological systems helps the evolution [47]. Several approaches motivated by the observation were proposed recently [17,23,24,25,28]. In the approaches, the chromosome includes non-coding segments that hold positions in the representation space but do not contribute to the evaluation of the solutions. LLGA [31,32] and its variants [33,34,35] described in Sect. 4.2 also use non-coding segments.

4.2 Dynamic Linkage Models

Adaptive Linkage Model. In the adaptive linkage model, the linkages are assigned similarly with the way in the static linkage model. But, it differs from the static linkage model in that the linkages are adjusted dynamically during the process of the genetic algorithm. An application that does not have prior information about the given problem is also possible by assigning the linkages based only on the analysis of the solutions in the population.

There are several approaches based on the adaptive linkage model. Kwon and Moon [30] applied the model to a function optimization problem. They adopted one-dimensional array representation and periodically rearranged the genes by a heuristic based on the correlation coefficients between the corresponding function variables.

Greene [44] proposed an adaptive linkage model employing complete graph representation. In the approach the genic distances are dynamically assigned based on the normalized entropy of the gene values. He proposed a crossover based on a cutoff value where an uncopied gene is repeatedly chosen at random and, at the same time, the values of the genes whose genic distances from the chosen gene are less than a preassigned cutoff value are copied together from one of the two parents alternately.

Evolvable Linkage Model. The beginning of the evolvable linkage model is Bagley's inversion operator [14]. The operator works by reversing the order of the genes lying inbetween a pair of randomly chosen points. To make the genes movable, the "encapsulated" encoding was used. The result of the approach was not successful. One of the reasons might be that the evolution of the linkage in the approach was so slow that it lost the race against allele selection. Messy genetic algorithm (mGA) [15,16] and fast messy genetic algorithm (fast mGA) [18,29] proposed by Goldberg *et al.* are the approaches dealing with the difficulty found in the Bagley's study. They use cut-and-splice operator for recombination. By the operator, genes in a solution migrate in group to other solutions in the population. The operator may cause over-specification and under-specification. The former occurs when a particular gene position is specified by two or more allele values and the other occurs, on the other hand, when a particular gene position is not specified by any allele value in the chromosome. They resolved the over-specification problem by scanning the chromosome in a left-to-right manner

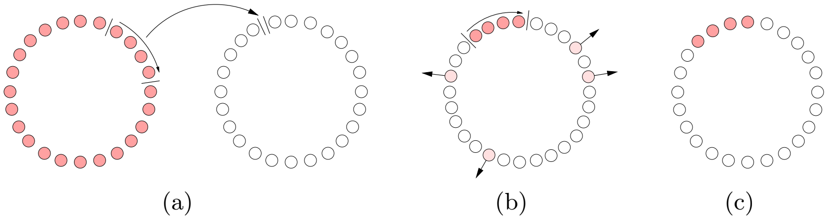

Fig. 3. An illustration of exchange crossover. (a) A randomly chosen segment of the donor is grafted to the recipient. (b) The duplicated genes are removed from the intermediate solution. (c) The final solution.

and choosing the allele that occurred first for each position, and resolved the under-specification problem by random assignment or a heuristic constructive filling in. The problems were avoided in Kargupta's gene expression messy genetic algorithm (GEMGA) [26], and Harik and Goldberg's linkage learning genetic algorithm (LLGA) [31,32]. In LLGA, the exchange crossover was used. It grafts a randomly chosen segment of one chromosome (called donor) on the other chromosome (called recipient) and deletes the genes in the recipient that have the same gene positions with the grafted genes to avoid duplication. Figure 3 shows an illustration of the crossover. There are a number of variants for the exchange crossover [33,34,35]. One-dimensional array representations were used in inversion, mGA, fast mGA, and GEMGA, and a ringed array representation was used in LLGA.

The adaptive non-coding segment representation proposed by Lee and Antonsson [41] is an approach that uses a one-dimensional real space. In the representation, genes are linked by edges with variable lengths. The edges represent the strengths of linkage between genes and perturbed by Gaussian random variable for adaptation. It is a variant of the approaches employing non-coding segments described in Sect. 4.1.

Moon and Kim [39] proposed a genetic algorithm that maintains two populations: one for the evolution of solutions and the other for the evolution of embeddings. In the approach, a solution is generated by a two-layered crossover; two embeddings are crossovered first and then the resultant embedding is used for the recombination of the solutions.

5 Summary

The results of previous theoretical and experimental studies suggest that the performance of a genetic algorithm highly depends on its ability to create and grow the building blocks. Various genetic algorithms motivated by the hypothesis have been proposed recently. A large part of them is based on the topological linkage of genes in the chromosome. In this paper, we summarized the approaches that exploit the topological linkages of genes and provided a taxonomy of them. The various chromosomal structures are byproducts of such studies. They include, for example, 1D array, ringed array, 2D array, 3D array, 1D real space (line), 2D real space (plane), and complete graph.

Acknowledgments. This work was partly supported by Optus Inc. and Brain Korea 21 Project. The RIACT at Seoul National University provided research facilities for this study.

References

1. J. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, 1975.
2. D.E. Goldberg. *Genetic Algorithms in Search, Optimization, Machine Learning.* Addison-Wesley, 1989.
3. J. Cohoon and D. Paris. Genetic placement. In *IEEE International Conference on Computer-Aided Design*, pages 422–425, 1986.
4. C. Anderson, K. Jones, and J. Ryan. A two-dimensional genetic algorithm for the Ising problem. *Complex Systems*, 5:327–333, 1991.
5. A.B. Kahng and B.R. Moon. Toward more powerful recombinations. In *International Conference on Genetic Algorithms*, pages 96–103, 1995.
6. S. Jung and B.R. Moon. Toward minimal restriction of genetic encoding and crossovers for the 2D Euclidean TSP. *IEEE Transactions on Evolutionary Computation*, 6(6):557–565, 2002.
7. D.I. Seo and B.R. Moon. Voronoi quantized crossover for traveling salesman problem. In *Genetic and Evolutionary Computation Conference*, pages 544–552, 2002.
8. Y. Davidor. Epistasis variance: Suitability of a representation to genetic algorithms. *Complex Systems*, 4:369–383, 1990.
9. D.I. Seo, Y.H. Kim, and B.R. Moon. New entropy-based measures of gene significance and epistasis. In *Genetic and Evolutionary Computation Conference*, 2003.
10. D. Thierens and D.E. Goldberg. Mixing in genetic algorithms. In *International Conference on Genetic Algorithms*, pages 38–45, 1993.
11. P. Larrañaga and J.A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Kluwer Academic Publishers, 2002.
12. M. Pelikan, D.E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.
13. N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. In *Foundations of Computer Science*, pages 577–591, 1994.
14. J.D. Bagley. *The Behavior of Adaptive Systems which Employ Genetic and Correlation Algorithms.* PhD thesis, University of Michigan, 1967.
15. D.E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5):493–530, 1989.
16. D.E. Goldberg, K. Deb, and B. Korb. Messy genetic algorithms revisited: Studies in mixed size and scale. *Complex Systems*, 4:415–444, 1990.
17. J.R. Levenick. Inserting introns improves genetic algorithm success rate: Taking a cue from biology. In *International Conference on Genetic Algorithms*, pages 123–127, 1991.

18. D.E. Goldberg, K. Deb, H. Kargupta, and G. Harik. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. In *International Conference on Genetic Algorithms*, pages 56–64, 1993.
19. T.N. Bui and B.R. Moon. Hyperplane synthesis for genetic algorithms. In *International Conference on Genetic Algorithms*, pages 102–109, 1993.
20. T.N. Bui and B.R. Moon. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers*, 45(7):841–855, 1996.
21. T.N. Bui and B.R. Moon. Analyzing hyperplane synthesis in genetic algorithms using clustered schemata. In *Parallel Problem Solving from Nature*, pages 108–118. 1994.
22. T.N. Bui and B.R. Moon. A new genetic approach for the traveling salesman problem. In *IEEE Conference on Evolutionary Computation*, pages 7–12, 1994.
23. A.S. Wu, R.K. Lindsay, and M.D. Smith. Studies on the effect of non-coding segments on the genetic algorithm. In *IEEE Conference on Tools with Artificial Intelligence*, pages 744–747, 1994.
24. A.S. Wu and R.K. Lindsay. Empirical studies of the genetic algorithm with non-coding segments. *Evolutionary Computation*, 3(2):121–147, 1995.
25. J.R. Levenick. Metabits: Generic endogenous crossover control. In *International Conference on Genetic Algorithms*, pages 88–95, 1995.
26. H. Kargupta. The gene expression messy genetic algorithm. In *IEEE Conference on Evolutionary Computation*, pages 814–819, 1996.
27. T.N. Bui and B.R. Moon. GRCA: A hybrid genetic algorithm for circuit ratio-cut partitioning. *IEEE Transactions on CAD*, 17(3):193–204, 1998.
28. J.R. Levenick. Swappers: Introns promote flexibility, diversity and invention. In *Genetic and Evolutionary Computation Conference*, pages 361–368, 1999.
29. D. Knjazew and D.E. Goldberg. OMEGA – Ordering messy GA: Solving permutation problems with the fast messy genetic algorithm and random keys. In *Genetic and Evolutionary Computation Conference*, pages 181–188, 2000.
30. Y.K. Kwon, S.D. Hong, and B.R. Moon. A genetic hybrid for critical heat flux function approximation. In *Genetic and Evolutionary Computation Conference*, pages 1119–1125, 2002.
31. G.R. Harik and D.E. Goldberg. Learning linkage. In *Foundations of Genetic Algorithms 4*, pages 247–262. 1996.
32. G.R. Harik. *Learning Gene Linkage to Efficiently Solve Problems of Bounded Difficulty Using Genetic Algorithms*. PhD thesis, University of Michigan, 1997.
33. F.G. Lobo, K. Deb, and D.E. Goldberg. Compressed introns in a linkage learning genetic algorithm. In *Third Annual Conference on Genetic Programming*, pages 551–558, 1998.
34. G.R. Harik and D.E. Goldberg. Learning linkage through probabilistic expression. *Computer Methods in Applied Mechanics and Engineering*, 186:295–310, 2000.
35. Y.P. Chen and D.E. Goldberg. Introducing start expression genes to the linkage learning genetic algorithm. Technical Report 2002007, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, 2002.
36. T.N. Bui and B.R. Moon. On multi-dimensional encoding/crossover. In *International Conference on Genetic Algorithms*, pages 49–56, 1995.
37. B.R. Moon and C.K. Kim. A two-dimensional embedding of graphs for genetic algorithms. In *International Conference on Genetic Algorithms*, pages 204–211, 1997.
38. B.R. Moon, Y.S. Lee, and C.K. Kim. GEORG: VLSI circuit partitioner with a new genetic algorithm framework. *Journal of Intelligent Manufacturing*, 9(5):401–412, 1998.

39. B.R. Moon and C.K. Kim. Dynamic embedding for genetic VLSI circuit partitioning. *Engineering Applications of Artificial Intelligence*, 11:67–76, 1998.
40. J.H. Kim and B.R. Moon. Neuron reordering for better neuro-genetic hybrids. In *Genetic and Evolutionary Computation Conference*, pages 407–414, 2002.
41. C.Y. Lee and E.K. Antonsson. Adaptive evolvability via non-coding segment induced linkage. In *Genetic and Evolutionary Computation Conference*, pages 448–453, 2001.
42. S. Jung and B.R. Moon. The natural crossover for the 2D Euclidean TSP. In *Genetic and Evolutionary Computation Conference*, pages 1003–1010, 2000.
43. S. Jung and B.R. Moon. A hybrid genetic algorithm for the vehicle routing problem with time windows. In *Genetic and Evolutionary Computation Conference*, pages 1309–1316, 2002.
44. W.A. Greene. A genetic algorithm with self-distancing bits but no overt linkage. In *Genetic and Evolutionary Computation Conference*, pages 367–374, 2002.
45. D.I. Seo and B.R. Moon. A hybrid genetic algorithm based on complete graph representation for the sequential ordering problem. In *Genetic and Evolutionary Computation Conference*, 2003.
46. E.J. Park, Y.H. Kim, and B.R. Moon. Genetic search for fixed channel assignment problem with limited bandwidth. In *Genetic and Evolutionary Computation Conference*, pages 1172–1179, 2002.
47. A.S. Wu and R.K. Lindsay. A survey of intron research in genetics. In *Parallel Problem Solving from Nature*, pages 101–110, 1996.