# A Bit-Wise Epistasis Measure for Binary Search Spaces

Cyril Fonlupt, Denis Robilliard, Philippe Preux

Laboratoire d'Informatique du Littoral
BP 719
62228 Calais Cedex, France
e-mail: {fonlupt,robillia}@lil.univ-littoral.fr

**Abstract.** The epistatic variance has been introduced by Davidor as a tool for the evaluation of interdependences between genes, thus possibly giving clues about the difficulty of optimizing functions with genetic algorithms (GAs). Despite its theoretical grounding in Walsh function analysis, several studies have shown its weakness as a predictor of GAs results. In this paper, we focus on binary search spaces and propose to measure epistatic effect on the level of individual genes, an approach that we call *bit-wise epistasis.* We give examples of this measure on several well-known test problems, then we take into account this supplementary information to improve the performances of evolutionary algorithms. We conclude by pointing towards possible extensions of this concept to real size problems.

## 1 Introduction

An important issue tackled by epistasis studies, in genetic algorithms (GAs), is understanding and characterizing the difficulty to optimize functions. In other words, how well are epistatic measures able to predict the difficulty of problems? Can they help in finding better solutions? There have been a lot of works dealing with this problem, both theoretical and empirical studies, initiated by a seminal paper by Davidor [1]. His epistasis variance have been shown to have a strong mathematical foundation, based on Walsh functions analysis (see [2–4] and also [5, 6]), but still the correlation between problem hardness and epistasis is not straightforward. As it was argued in [3], even if it may provide some guidance for this matter, no definitive conclusion can be drawn. This is also confirmed by recent works from Rochet and Venturini [7], showing that one can change the representation of a problem, and thus achieve a lower epistasis, *without* changing the problem hardness. We think that this lack of accuracy may come from the global nature of this measure, which computation involves two levels of averaging. In this paper, we do not hope to provide a definitive answer to this question. Rather we propose a different measure of epistatic effects in binary search spaces, that we call *bit-wise epistasis.* We think our measure provides an increased accuracy over Davidor's proposition and we show that this may give another point of vue on the matter of problems hardness. This accuracy is especially clear when dealing with and explaining the unsuccessful remappings of search space proposed in [7]. We present the definition and the principles of our epistasis measure in Sect. 2. We give examples of computations, applied to NK-landscapes, in Sect. 3. Then we study a set of common functions in Sect. 4, and explain why some proposed remapping have failed in Sect. 5. In Sect. 5, we use the information provided by our measures in order to enhance the quality of solutions found by evolutionary algorithms. This improvement is obtained through adding a simple stochastic mutation operator which rate is based on bit-wise epistasis values. This work is too preliminary to allow us to report improvements on real world problems, but we give hints at how one could use the concept of bit-wise epistasis in GAs.

## 2 A Definition of Bit-Wise Epistasis

Our approach is based on the fact that a strong epistatic relation may exist only on some genes within a genotype, the other genes being much more independent. We think that it may be interesting to have a detailed view of such interactions rather than mixing and merging them like it is done in the epistatic variance defined by Davidor. In the following, we assume the reader is familiar with the notion of *schema* (see also [8, 9]).

Let $f$ be the fitness function from a binary search space $B = \{0,1\}^l$ in the set of reals $\mathbb{R}$ with $l$ the length of genotypes.

Let $B' = \{0,1,\#\}^l$ the set of schemata associated with $B$. Let $\Sigma_i$ the set of schemata of order $l-1$ such that their unique undefined loci is at the $i^{\text{th}}$ position in the schema, i.e. $\Sigma_i = \{\sigma_0\sigma_1 \ldots \sigma_i \ldots \sigma_{l-1} \in B' \mid \sigma_{j \neq i} \in \{0,1\} \text{ and } \sigma_{\rm i} = \#\}$.

Let $\alpha = \alpha_0\alpha_1 \ldots \alpha_{i-1}\#\alpha_{i+1} \ldots \alpha_{l+1}$ a schema in $\Sigma_i$. Let $X_\alpha, \bar{X}_\alpha$ be genotypes in $B$, members of $\alpha$, with $X_\alpha = \alpha_0\alpha_1 \ldots \alpha_{i-1}\mathbf{0}\alpha_{i+1} \ldots \alpha_{l-1}$ and $\bar{X}_\alpha = \alpha_0\alpha_1 \ldots \alpha_{i-1}\mathbf{1}\alpha_{i+1} \ldots \alpha_{l-1}$. We call $d_i(\alpha)$ the *fitness difference*[1] *at gene* $i$:

$$d_i(\alpha) = f(X_\alpha) - f(\bar{X}_\alpha)$$

We define the *mean fitness difference at gene* $i$ as the mean $d_i(\alpha)$ for all schemata $\alpha \in \Sigma_i$ :

$$M_i = \frac{1}{2^{l-1}} \sum_{\alpha \in \Sigma_i} d_i(\alpha)$$

We call *bit-wise epistasis at gene* $i$ the variance of the fitness difference at gene $i$:

$$\sigma_i^2 = \frac{1}{2^{l-1}} \sum_{\alpha \in \Sigma_i} [M_i - d_i(\alpha)]^2$$

[1] An alternative definition could be the absolute value $|d_i(\alpha)|$. We intend to discuss the reasons underlying our choice in a forthcoming paper.

When the search space is too big for such a computation (as is usually the case), we approximate bit-wise epistasis on a sample of schemata. This raises the same kind of problems as for Davidor's variance relatively to sampling error and cost of epistasis computation. To allow comparison between different problems, we have followed a suggestion in [2], and normalized numerical results on the fitness variance of our samples. Obviously, a bit-wise epistasis equals to 0 for all genes is associated with a problem where there is no dependency between genes.

## 3 An Illustration of Bit-Wise Epistasis: the *NK* Landscapes

In this section we use *NK* landscapes as an illustration of bit-wise epistasis measures. We do not define *NK* landscapes but refer the reader to [10]. Let us say that these are artificial problems where the degree of epistasis can be tuned in a wide range of values. We generate different NK landscapes using the guidelines provided in [11]. Our measures were computed on landscapes characterized by $N = 24$ and four different values for parameter $K$ : 1, 4, 12, 23 (the higher the value of $K$, the higher the level of epistatic dependencies). Bit-wise epistasis measures are plotted in Fig. 1 and consistent with what we expected: the higher the value of $K$, the higher the curve on the plot; also notice that the level of bit-wise epistasis is evenly distributed on the whole range of 24 bits with no peak of epistasis.

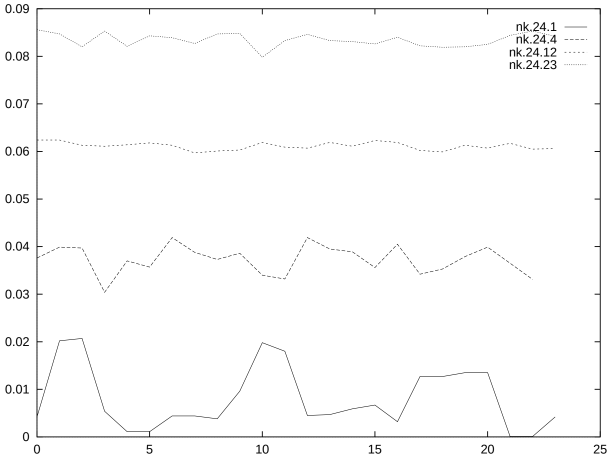


**Fig. 1.** Comparison of bit-wise epistasis in different NK landscapes.

## 4 Bit-Wise Epistasis Measures for a Set of Common Problems

In this section, we look at 4 functions taken from a set of problems widely studied in the GA community, proposed in [12, 13]. Figure 2 sums-up the characteristics of the functions we have studied. We have added the Davidor's epistasis variance computed on a sample of 2000 points and normalized on the fitness variance of the sample (as proposed in [2]).

| | Functions | Range of $x_i$ | $x_i$ coded on | Davidor's Epistasis |
|---|---|---|---|---|
| f1 | $x_1^2 + x_2^2 + x_3^2$ | $[-5.12, 5.11]$ | 10 bits | 0.984 |
| f2 | $100(x_1^2 - x_2)^2 + (1 - x_1)^2$ | $[-2.048, 2.047]$ | 12 bits | 0.728 |
| f6 | $0.5 + \frac{\sin^2(\sqrt{x_1^2+x_2^2})-0.5}{1+0.001(x_1^2+x_2^2)}$ | $[-100, 100]$ | 10 bits | 0.994 |
| f9 | $x_1 + 2x_2 + 3x_3 + x_4^2$ | $[0, 100]$ | 10 bits | 0.082 |

**Fig. 2.** The set of test functions.

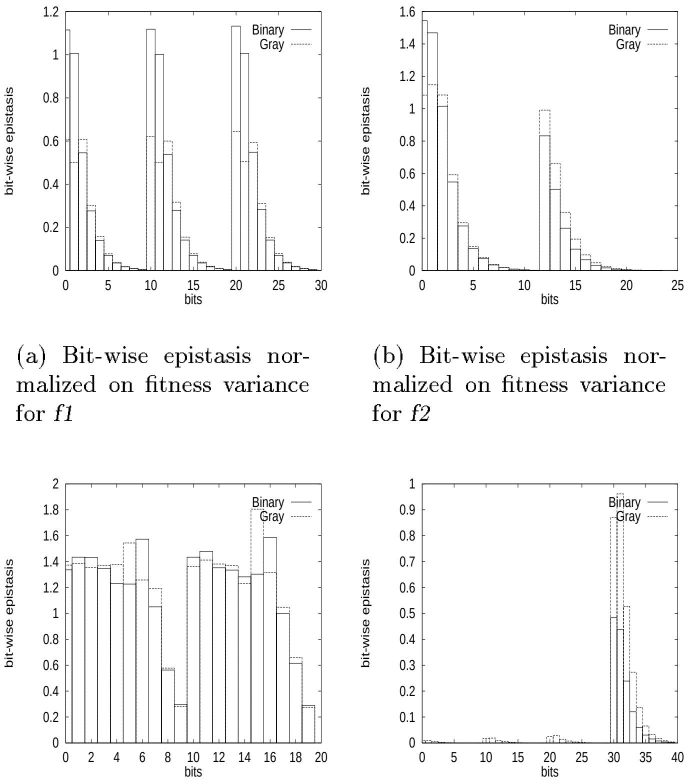
There have been for some time, a debate about the interest of using Gray coding as a general way to improve GAs performances (a general presentation can be found in [14]). Gray coding greatly reduces the "Hamming cliffs" that exist between consecutive integers around powers of 2 (e.g. $2^x - 1$ and $2^x$ are separated by a Hamming distance equal to $x + 1$ in binary coding, and only 1 in Gray coding). To allow the monitoring of differences introduced by this alternative representation scheme, we compute bit-wise epistasis using both standard binary coding and gray coding. Results are shown in Fig. 3, plots corresponding to binary coding are shown in continuous lines, those for gray coding are in dotted lines.

Compared with previous measures, these plots allow a closer view at epistatic effects. It is especially true for function $f9$. We already know that its overall epistatic variance was low (0.082), now we can see that only a few genes are involved in epistatic dependence: clearly, from its definition, we recognize the influence of the most significant bits of its last variable $x_4$. For function $f1$, epistasis is also quite concentrated on a few number of genes, corresponding to the most significant bits of the three variables involved in its definition. On the opposite, function $f6$ shows a large amount of dependencies on almost every bit.

It is clear from plots in Fig. 3 that using Gray code does slightly change the epistatic dependencies at the bit level. Notice anyway that this does not always amounts to a reduction in epistasis as can be seen in the case of function $f9$.

## 5 What is Bit-Wise Epistasis Good for ?

Through this section we propose to use the bit-wise epistasis measure to:



(a) Bit-wise epistasis normalized on fitness variance for *f1*

(b) Bit-wise epistasis normalized on fitness variance for *f2*

(c) Bit-wise epistasis normalized on fitness variance for *f6*

(d) Bit-wise epistasis normalized on fitness variance for *f9*

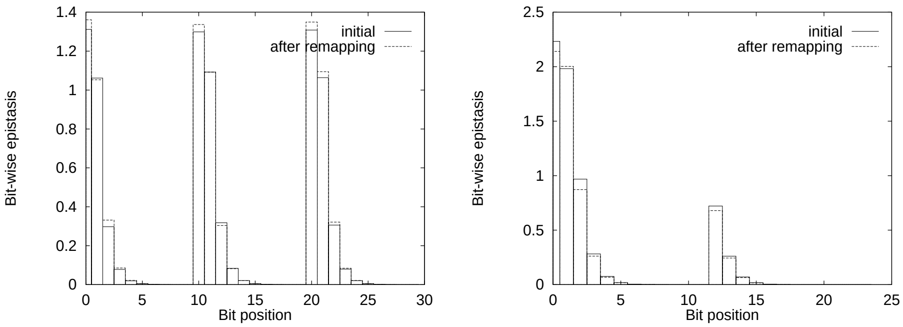
**Fig. 3.** Bit-wise epistasis for some test functions using binary and Gray coding.

- understand why remapping the binary search space does not always improve GA performance;
- improve the evolutionary algorithms by using the supplementary information gained through bit-wise epistasis computation.

### 5.1 Remapping

Remapping a function is known to be a very difficult problem (see [15]), but nonetheless some interesting attempts have been made to find other representations, notably in [16, 17]. Here we focus on the method proposed in [7], where Rochet *et al* have proposed to randomly generate transformations in binary search spaces that remap bits 3 by 3. A random sample of such remapping functions is generated, their associated epistasis variance (or the derived epistasis correlation) is computed, and then the remapping function associated with lowest epistasis is chosen to remap the search space. As was shown in their paper, the results of genetic algorithms were not improved, and, more strikingly, experiments have shown no correlation between Davidor's epistasis variance level and the difficulty of the problem.

Here, we have generated transformations for functions *f1* and *f2*, according to the method of Rochet *et al.* and have found remapping functions such that Davidor's epistasis variance drops from 0.975 to 0.513 for *f1*, and from 0.713 to 0.431 for *f2* (to limit the influence of sampling error, the level of epistasis variance after remapping has been verified on two independent random samples). Then we have computed bit-wise epistasis before and after remapping the space (plots shown respectively in continuous and dotted lines on Fig. 4). It clearly appears that the amount of epistatic change at the bit level is extremely small. So, failing to improve GAs results this way seems to us hardly surprising.



(a) Influence of remapping on bit-wise epistasis for *f1*

(b) Influence of remapping on bit-wise epistasis for *f2*

**Fig. 4.** Bit-wise epistasis before and after remapping.

### 5.2 Hints at Building Epistasis-Based Hybrid Algorithms

Now we intend to show how our epistasis measure can be used to improve evolutionary algorithms. First, we have focused our experiments on a basic hill-climber exploring the standard bit flipping neighborhood, to which a probabilistic mutation phase is added. The mutation rate is set independently for every gene, proportionally to the bit-wise epistasis value. Next we apply the same idea to a genetic algorithm, where we replace the usual unique mutation rate by computing for every gene a rate that depends on its bit-wise epistasis value. We do not claim to have devised algorithms that fully exploit bit-wise epistasis information.

We only show that a very simple scheme, changing probability of mutation on the basis of this information, is enough to obtain better results.

Accordingly, we don't have formal proofs of the underlying mechanics behind our scheme. An intuitive idea is that genes with few epistatic dependencies don't need to be changed very often. Indeed if the algorithm finds an allele that increases fitness, it is not useful to question this choice in the future: changing others genes values doesn't have much impact on this choice. On the opposite, highly epistatic genes should be mutated at a higher rate, due to the fact that their contribution to the global fitness is very dependent on other genes values: the more combinations we evaluate, the more chances we have to find a good set of alleles and jump away from local optimum (see also [18]).

**Experimental Setting** We present here three other problems that were used as benchmarks in our experiments.

The two first functions, taken from Sebag and Schoenauer in [19], are known to be difficult to solve.

$$\begin{aligned} y_1 &= x_1 \\ y_i &= x_i + y_{i-1},\ i \geq 2 \end{aligned} \qquad F'_1 = \frac{100}{10^{-5}+\sum_i |y_i|}$$

$$F'_2 = \frac{100}{10^{-5}+\sum_i |.024\times(i+1)-x_i|}$$

All these two functions involve 100 numerical variables $x_i$ coded on 9 bits each and varying in $[-2.56, 2.56]$. The maximum for *F'1* is $10^7$, the maximum for *F'2* is 416.64.

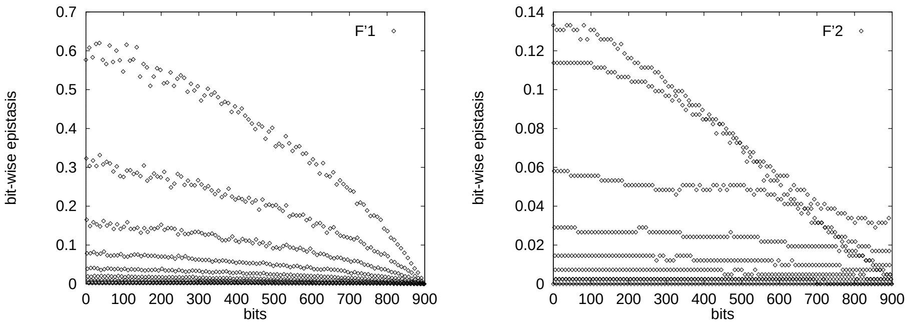
The third problem is a special version of the NK problem, with the following modifications:

- each even bit is involved in the NK epistatic dependencies;
- each odd bit brings an independent fitness contribution (like in the well-known OneMax problem).

We call this last problem *semi-nk*. In our experiments, the length of the chromosome was 100 and $K = 50$.

**Hill-Climber Experiments** In this section, a gene is said to be *epistatic* if its bit-wise epistatic measure is over the average bit-wise epistasis measure for all genes. In order to improve the hill-climber, we hybridize the standard hill-climbing method with a high rate mutation operator working only on epistatic genes. We do not detail this algorithm, due to lack of space.

Shapes of the *F'1* and *F'2* problems in Fig. 5 clearly indicate that we face very epistatic problems. The fitness contribution of each bit except the last one depends on the other ones. Furthermore, *F'1* problem can be seen as a deceptive problem because for each numerical variable coded on 9 bits the optimal solution is 100000000 (binary coding) while a basic hill-climbing method will improve the fitness by flipping a 1 to the last eight positions.



(a) Bit-wise epistasis for the *F'1* problem. The average epistasis is 0.11

(b) Bit-wise epistasis for the *F'2* problem. The average epistasis is 0.026

**Fig. 5.** Bit-wise epistasis for our experiments

Our results are compared with a basic multi-start hill-climber (*HC*), which explores the neighborhood defined by flipping one bit of the current solution. We allow 200000 functions evaluations.

| Problems | HC | Hybrid HC |
|---|---|---|
| *F'1* binary | 1.04 | 1.37 |
| *F'2* binary | 8.07 | 9.7 |
| *semi-nk* binary | 1.605 | 1.63 |

**Table 1.** Average best fitness for the 3 problems

Using bit-wise epistasis information improves the results. For the *semi-nk* problem, the smallness of the improvement may be explained by the relative easiness of this problem.

**GAs Experiments** In this section we show how bit-wise epistasis information can be used within the GAs frame. Experiments have been done on problems known to be hard for GAs : maximizing Schoenauer's *et al.* function $F'1$, minimizing Whitley's functions $F3$ and $F7$ (in [12]). We have used a standard GA, with ranked selection, 1-points crossover with a probability of 0.8, the size of the population being respectively 50, 20 and 30, the number of generations respectively 1000, 2000 and 5000, and the mutation operator the classical bit-flipping. There was two versions of this GA that differed from each other by mutation rates. In version 1 of our algorithm, half of the genes, those with lower bit-wise epistasis values, were associated with a mutation rate of $2e-3$, the other half (higher bit-wise epistasis genes) using a rate of $1e-3$. Version 2 of the GA used the opposite setting : lower mutation rate associated with lower bit-wise epistasis, higher rates with higher epistasis. The average best results other 20 independent runs of each GA is shown in Tab. 2, with other statistics.

| | maximizing $F'1$ | | minimizing $F3$ | | minimizing $F7$ | |
|---|---|---|---|---|---|---|
| | GA 1 | GA 2 | GA 1 | GA 2 | GA 1 | GA 2 |
| Min | 1.107 | 1.306 | 4.431 | 3.647 | 0.599 | 0.599 |
| Average | 1.349 | 1.568 | 11.471 | 8.542 | 1.059 | 0.834 |
| Max | 1.533 | 1.770 | 25.394 | 16.044 | 3.093 | 2.511 |
| Std Deviation | 0.106 | 0.112 | 4.769 | 3.613 | 0.864 | 0.580 |

**Table 2.** Best results on 20 independent runs: GA1 works with low mutation on epistatic genes, GA2 with high mutation on epistatic genes.

We can see that GA2 is ahead of GA1, so putting higher mutation rates on genes with high bit-wise epistasis is worthwhile. Nonetheless we warn the reader that this effect appears on rather long runs (1000 generations or more). Work is still needed to show if easier problems or shorter runs may benefit from "bit-wise epistasis aware" schemes.

## 6 Conclusion

In this article, we have introduced the notion of bit-wise epistasis measure. We think this notion is a tool that is useful to understand epistatic interactions at a detailed level. This measure gives clues in explaining the failure of some remapping techniques which decreases the amount of Davidor's epistasis but seemingly with very few effects at the bit level. We have applied this tool to devise simple improvement schemes to a hill-climber and a GA. Work is still needed to understand how to use this information in better ways.

We are currently working on an extension of this technique to address non binary problems, like Traveling Salesman and Quadratic Assignment Problems. We think this concept deserves an in-depth theoretical study, in order to precise its relationship with Davidor's epistasis and also with the notion of ruggedness in fitness landscapes.

Note: we acknowledge the anonymous referree's suggestions that helped us in improving the presentation of our results.

## References

1. Yuva Davidor. Epistasis variance: A viewpoint on GA-hardness. In *[20]*, pages 23–35, 1991.
2. Mauro Manela and J.A. Campbell. Harmonic analysis, epistasis and genetic algorithms. In *[21]*, 1992.

3. C.R. Reeves and C.C. Wright. Epistasis in genetic algorithms: An experimental design perspective. In *[22]*, pages 217–224, 1995.
4. S. Rochet, M. Slimane, and G. Venturini. Epistasis for real encoding in genetic algorithms. In *IEEE ANZIIS'96*, pages 268–271, 1996.
5. David E. Goldberg. Genetic algorithms and Walsh functions: Part I, a gentle introduction. *Complex Systems*, 3:129–152, 1989.
6. David E. Goldberg. Genetic algorithms and Walsh functions: Part II, deception and its analysis. *Complex Systems*, 3:153–171, 1989.
7. S. Rochet, G. Venturini, M. Slimane, and E. M. El Kharoubi. A critical and empirical study of epistasis measures for predicting GA performances: a summary. In *Evolution Artificielle 97*, pages 331–341, Nimes, Frances, October 1997.
8. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
9. John H. Holland. *Adaptation in Natural and Artificial Systems*. Michigan Press University, 1975.
10. S.A. Kauffman. Adaptation on rugged fitness landscapes. *Lecture in the Sciences of Complexity*, pages 527–618, 1989.
11. Kenneth A. De Jong, Mitchell A. Potter, and William M. Spears. Using problem generators to explore the effects of epistasis. In *[23]*, pages 338–345, 1997.
12. D. Whitley, K. Mathias, S. Rana, and J. Dzubera. Building better test functions. In *in [22]*, pages 239–246, 1995.
13. K. A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, MI, USA, 1975.
14. David Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
15. G.E. Liepins and M.D. Vose. Representational issues in genetic optimization. *Journal of Experimental and Theoretical AI*, 2:1–15, 1990.
16. Keith Mathias and Darrell Whitley. Remapping hyperspace during genetic search: Canonical delta folding. In *[24]*, pages 167–186, 1993.
17. Keith E. Mathias and Darrell Whitley. Changing representations during search: A comparative study of delta coding. *Evolutionary Computation*, 2, 1994.
18. J. David Schaffer and Larry J. Eshelman. On crossover as an evolutionarily viable strategy. In *[25]*, pages 61–67, 1991.
19. Michèle Sebag and Marc Schoenauer. A society of hill-climbers. In *[26]*, 1997.
20. Gregory J.E. Rawlins, editor. *Workshop on the Foundations of Genetic Algorithms and Classifiers*, Bloomington, IN, USA, July 1991. Morgan Kaufmann.
21. Reinhard Manner and Bernard Manderick, editors. *Proceedings of the second Conference on Parallel Problem Solving from Nature*, Free University of Brussels, Belgium, September 1992. Elsevier Science.
22. Philips Laboratories Larry J. Eshelman, editor. *Proceedings of the 6th International Conference on Genetic Algorithms*, University of Pittsburgh, USA, July 1995. Morgan Kaufmann.
23. *Proceedings of the 7th International Conference on Genetic Algorithms*, East Lansing, Michigan, USA, July 1997. Morgan Kaufmann.
24. Darrell Whitley, editor. *Proc. of the Workshop on Foundations of Genetic Algorithms*, Vail, CO, USA, 1993. Morgan Kaufmann.
25. Richard K. Belew and Lashon B. Booker, editors. *Proceedings of the 4th International Conference on Genetic Algorithms*, La Jolla, California, USA, July 1991. Morgan Kaufmann.
26. *International Conference on Evolutionary Computation*, Anchorage, USA, 1997.