Intra and Extra-Generation Schemes for Combining Crossover Operators

Dana Vrajitoru
Intelligent Systems Laboratory,
IUSB, Computer and Information Sciences,
1700 Mishawaka Ave, P.O. Box 7111
South Bend, IN 46634
danav@cs.iusb.edu

Abstract

Several studies on the variations of the crossover operator have shown that each of them presents speci£c properties that are interesting under particular circumstances. The advantages of each operator over others are often contradictory and the best operator depends on the problem being solved. This paper is based on the assumption that a combination of several crossover operators can take advantage of their respective qualities and power. Thus, we propose several combination models based on four crossover operators: the 1-point, 2-point, uniform and dissociated. We test the performance of these models through an experimental approach using the problem of the Hamiltonian circuit.

Introduction

The crossover operator is one of the most important and powerful features of the genetic algorithms (GAs), and it has been the object of interest for many researchers in the £eld (Mao *et al.* 2002; Rana 1999; Suzuki & Iwasa 1999).

Holland (1975) and Goldberg (1989) have started the study of the probabilistic properties of the crossover through the schemata theorem. An interesting question that has been risen is between crossover and mutation, which one is more important for GAs (Mühlenbein & Schlierkamp-Voosen 1995; Wu, Lindsay, & Riolo 1997). The general conclusion is that mutation alone is not suf£cient for many problems, and that the role of crossover is essential.

In a different direction, Booker (1987) and Spears (1990) have used a second measure to evaluate the genetic operators : the exploratory power. They have shown that a highly disruptive operator is also highly productive. As the exploiting and the exploratory qualities are contradictory, each of these features could become an advantage for some particular search £elds, and a disadvantage for others.

Some research has been made to combine several crossover operators in the same application of the GA (Hong, Kahng, & Moon 1995; Spears 1995), and our paper follows similar ideas.

Generally, the number of options available for this operator is quite large. We can cite the nonuniform crossover (Maini *et al.* 1994), the spontaneous crossover (Mayer 1998), the selective crossover (Vekaria & Clack 1998). Some of these crossover operators can show a better performance on speci£c £tness landscapes, and be less able to solve other problems.

Considering these issues, it is dif£cult to choose one operator without prior experimenting for a given problem. The classical approach is to test several forms on a limited number of cases, and to base the future use of the crossover on them. The results obtained on a class of problems are in general hard to extend to another class, as the shape of the search space can be completely different. Thus, the choice of the form of crossover is often a subjective one.

We are interested in the situation where a single crossover operator must give good results for most of the problems to study, which is often the case. We would like to obtain a recombination model that shows a better performance than any single crossover on all of the problems we consider, and not only on some of them.

To achieve this goal, we present several models using not one, but four crossover operators (1-point, 2-point, uniform and dissociated) and combining them during the execution of the GA. We expect at least some of these models to show a better performance than each of the considered operators on the test problems. We are testing these models on the problem of the Hamiltonian circuit.

The next section introduces the general parameter settings, the test functions, and the four basic types of crossover that we have used for our experiment. The following section de£nes and tests several combination models based on these operators. The last section summarizes the entire experiment.

Experiment Description

This section presents the test functions and the parameter settings that we have used for the experiment.

Standard Functions Set This set contains ten functions that are used by many researchers to test GAs and their precise equations can be found in (Whitley *et al.* 1996). Each if them is a real function depending on 2 to 30 variables on a given interval. The goal is to minimize each of these functions, so lower £tness values are actually better ones. The genetic representation is a concatenation of the variables occurring in the functions. In our case we have use a discretization of the variables using 10 binary genes for each of them.

The Hamiltonian Circuit Problem The second test function that we have chosen is the problem of the Hamiltonian circuit.

Hamiltonian circuit (HC): Given an oriented graph, £nd a circuit that passes once and only once through each vertex. This problem is known to be NP-complete (Brassard & Bratley 1994).

We have performed our experiments with 11 HC problems with graphs having from 50 to 150 vertices and from 246 to 3136 edges. The direct representation of a HC problem for the GAs is dif£cult. De Jong and Spears (1989) have implemented a genetic representation using the transformation of an instance of HC into an instance of SAT, which is easier to represent in a genetic form.

A detailed description of the reduction of a HC instance into a SAT instance can be found in (Brassard & Bratley 1994) or (Vrajitoru 1999). For any given graph, a Boolean variable corresponds to each arch, and is given the true value if the arch belongs to the circuit. The SAT expression summarizes the fact that, for each vertex, one and only one of the entering edges and of the exiting edges must belong to the circuit.

For example, let us consider the graph in Figure 1. The conversion of the HC instance for this graph into a SAT instance would result in Equation 1, in which a Boolean variable with the same name as edge is true if the edge belongs to the Hamiltonian circuit. The symbols '$\otimes$' and '$\wedge$' represent the Boolean 'xor' and 'and' operators.

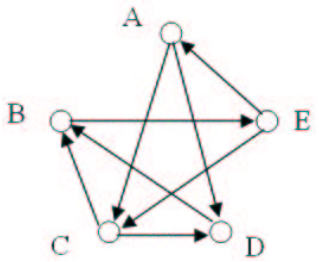

Figure 1: The graph for which Equation 1 represents the instance of SAT corresponding to its HC instance

$$\begin{array}{ll} out & (AC \otimes AD) \wedge BE \wedge (CB \otimes CD) \wedge \\ & \wedge DB \wedge (EA \otimes EC) \wedge \\ in & EA \wedge (CB \otimes DB) \wedge (AC \otimes EC) \wedge \\ & \wedge (AD \otimes AD) \wedge BE \end{array} \quad (1)$$

The £tness function for the HC problems is based on a fuzzy logic interpretation of the logical expression derived from the graph. Thus, each individual represents an assignment of truth values to each of the variables composing the logical expression. The logical and operator is interpreted as the average value of the operands, while the logical or operator is interpreted as the maximum of the values of the operands. If an individual represents a perfect Hamiltonian circuit, then its £tness will be 1.

The questions that we ask based on this problem are the following. First, can the combination of several crossover operators perform better than each of them individually? And second, how does the algorithm react to the problem scaling, especially when going from small problems to big problems?

Parameter Settings

Since we are studying the crossover operator, we have performed 50 runs of the GA for each problem with a crossover rate of 1 and a mutation rate of 0.0005. Previous tests have determined that this is a good setting for both our test problems. For each of the 50 trials, we generate one initial population and run the GA starting from it separately for each operator or combination model. This way, each of the operators has the same chance to £nd the optimal individual.

Each generation contains 100 individuals and the number of generations is limited to 1000. We have used the £tness proportionate selection (Goldberg 1989), and the monotone reproduction (Vrajitoru 1999). Our performance measure is the average of the best £tness value achieved in the last generation.

Crossover Operators

Among the existing variations and forms, we have chosen four crossover operators of comparable behavior: the 1-point, 2-point, uniform with a swap probability of 0.5, and dissociated crossover.

The 1-point crossover has little exploratory power, but can better exploit the knowledge contained in the initial population. On the contrary, the n-point crossover ($n >> 1$) is highly exploratory and is recommended when the population size is small. The uniform crossover has a great exploratory power, which can be controlled by the value of the swap probability. This operator has shown good performance even with a high exchange rate (Syswerda 1989).

The *dissociated* crossover has been introduced in (Vrajitoru 1999). We have used a variation on its original form that splits each parent in two at a different cross site, and swaps the resulting right hand sides of the parents by applying the logical 'or' operator on the overlapping portions for one child, and logical 'and' operator for the other child. More precisely, if $par_{1,2}$ are the parents in the crossover operation, $csite_{1,2}$ are two cross sites, then the children individuals are de£ned by the following equation in which the symbols $\vee$ and $\wedge$ represent the logical operators "or" and "and" respectively:

$$child_1(i) = \begin{cases} par_1(i), & i \leq csite_1 \\ par_1(i) \vee par_2(i), & csite_1 < i \leq csite_2 \\ par_2(i), & csite_2 < i \end{cases}$$

$$child_2(i) = \begin{cases} par_2(i), & i \leq csite_1 \\ par_1(i) \wedge par_2(i), & csite_1 < i \leq csite_2 \\ par_1(i), & csite_2 < i \end{cases} \quad (2)$$

Figure 2 illustrates the way the dissociated crossover builds the children, as speci£ed in Equation 2. This formulation is different from the original one presented in (Vrajitoru 1999) concerning the operator applied to the overlap-

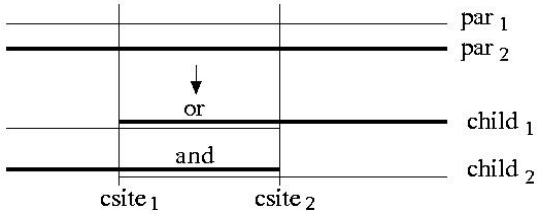

Figure 2: The dissociated crossover

ping part of the parents for the second child. We have experimented with various formulations and the 'and' operator seems to be the best one.

Before starting to combine the crossover operators, it is interesting to know how well each of them performs on the problems we have chosen. Tables 1 and 2 present the average over the 50 trials of the best performance in the last generation for each operator for the set of standard functions and for the HC problem respectively. The highest possible £tness in our case is 1, which would be achieved in the case of the optimal individual. The £rst column represents the graph itself, labeled by the number of vertices. The last column indicates the operator presenting the best performance for that graph.

Table 1: Average £tness of the crossover operators on the standard functions set (minimization problems)

# F	1-point	2-point	Uniform	Dissociated	Best
F1	0.066	0.044	0.344	0.383	uniform
F2	0.414	0.516	0.840	0.734	1-point
F3	12.22	12.34	12.12	10.44	dissoc.
F4	0.981	0.788	1.104	1.526	2-points
F5	5.206	8.175	4.596	5.949	uniform
F6	2.414	2.615	3.916	3.483	1-point
F7	608.37	614.82	670.62	708.84	1-point
F8	1.707	1.501	2.978	3.941	2-points
F9	0.197	0.184	0.243	0.196	2-points
F10	1.887	1.687	2.188	2.527	2-points

From these tables we see that the dissociated crossover presents the best performance on more than half of the problems, especially for the large graphs, and is followed by the 2-point crossover. However, the difference between the performance of these operators is not always visible because we only show the £rst 3 digits of the average £tness. Given the diversity in the results, we can deduce that, in general, we cannot predict with great accuracy that one of the operators will perform better than the others on a new problem.

Combining Crossover Operators

In this section we present several combination models for crossover operators and compare them with single crossover operators.

Table 2: Average £tness the crossover operators on the HC problem (£tness maximum of 1)

Graph	1-point	2-point	Uniform	Dissociated	Best
HC50	0.977	0.979	0.978	0.979	dissoc.
HC60	0.981	0.982	0.984	0.985	dissoc.
HC70	0.977	0.980	0.983	0.986	dissoc.
HC80	0.968	0.972	0.981	0.981	dissoc.
HC90	0.953	0.960	0.973	0.976	dissoc.
HC100	0.936	0.944	0.964	0.973	dissoc.
HC110	0.911	0.923	0.950	0.971	dissoc.
HC120	0.889	0.902	0.936	0.971	dissoc.
HC130	0.873	0.887	0.924	0.969	dissoc.
HC140	0.854	0.867	0.911	0.970	dissoc.
HC150	0.793	0.808	0.859	0.973	dissoc.

The attempts to combine crossover operators have been far less numerous than the developed forms of crossover. Schaffer and Morishima (1987) proposed a crossover scheme based on the n-point crossover where the number and position of the crossover sites where self-adapting over the GA run. Following the adaptation idea, Spears (1995) has developed a model combining the 2-point and uniform crossover, where the proportion between the two is also evolved by the GA. Finally, Hong et al. (1995) propose several adaptive combination strategies based on traditional, cycle, and geographic crossover. In all the cases, using more than one crossover seems to outperform each single crossover.

The three cited approaches use strategies that imply the use of several operators within the construction of each generation. We propose three new models, the £rst two applying several forms of crossover to build every new generation (*intra-generation* models). In the third model, a single crossover is used in each generation, and this operator changes after several generations (*extra-generation* model).

Model Description

Intra-generation schemes. The £rst combination model we propose aims to use several forms of crossover for building each generation. In general, for each crossover operation, the method chooses one of the four operators presented in the previous section by a random function following two schemes.

The £rst model, called *balanced combination*, gives each operator a probability to be selected of 0.25, which means that each operator has equal chances to be employed.

The second model we propose, named *adaptive*, starts the exact same way as the balanced one, by giving each of the four operators an equal chance to be selected. The probabilities associated with the basic operators are modi£ed after completing each generation the following way:

- When the population is too heterogeneous, or when the change in the £tness value is too dramatic from one generation to the next, we increase the probability associated with the conservative operators, which are the 1-point and the 2-point crossover.
- When the population is too homogeneous, or when the change in £tness from the previous generation is too small, we increase the probability associated with the more exploratory operators, which are the uniform and the dissociated crossover.
- Otherwise we may modify all of the probabilities by a small random amount.

Extra-generation Schemes. The third combination model uses a single crossover operator during a certain number of generations, then switches to another. The passage from one operator to the next is accomplished in a round-robin fashion, with the step equal to 50 or 100 generations. We have denoted these strategies by *RR50* and *RR100*.

For the moment, all the experiments start with the 1-point operator, followed in order by the 2-point, then by the uniform, and £nally by the dissociated crossover, and restarting the cycle if needed. The experiment can be extended to see whether the order and the starting point in the cycle are of any importance.

Experimental Results

We still expect the best crossover operator for each problem to be better that the combined models. In the situation we considered, the crossover form has to perform well on several different problems, and not only on a particular one. Thus, we are interested in comparing the combined models with the crossover operator that has shown the best general performance (see Table 2) for each individual problem. The last column marks the combination scheme that has presented the best performance so that it can be compared to the best single operator.

From this table, we can notice that the balanced combination method shows a better performance than the best single operator in most of the cases, except for the last 2 problems where the difference between them is very small. We believe that this scheme proves to be better than the other in most cases because it is the one taking advantage of the strength of each operator for every generation. When the population becomes more homogeneous in the later generations, the probability to spawn a better individual with this method is about equal to the maximum probability considering each operator separately. There is no signi£cant difference between the other combining methods, although they are in general better than most individual operators.

Next, we have thought interesting to observe the evolution of the best £tness through generations. Thus, Figure 3 shows the average performance for the 2-point, dissociated, combined balanced, and combined adaptive operators during the £rst 500 generations for the HC50 problem.

Table 3: Average £tness of the combined models, standard functions set (minimization problems)

# F	Balanced	Adaptive	RR5	RR10	Best
F1	0.066	0.044	0.024	0.025	2-point
F2	0.523	0.361	0.473	0.393	2-point
F3	10.4	11.74	10.6	10.92	dissoc.
F4	0.555	0.272	0.538	0.373	dissoc.
F5	1.581	2.409	2.39	2.985	balanced
F6	1.978	1.374	2.133	1.546	adaptive
F7	607.547	504.569	537.98	521.3	adaptive
F8	1.119	0.664	0.808	0.634	2-point
F9	0.165	0.139	0.154	0.156	2-point
F10	1.453	1.062	1.438	1.269	2-point

Table 4: Average £tness of the combined models, HC problems (£tness maximum of 1)

Graph	Balanced	Adaptive	RR5	RR10	Best
HC50	0.981	0.980	0.980	0.980	balanced
HC60	0.987	0.984	0.987	0.986	RR5
HC70	0.990	0.986	0.990	0.988	RR5
HC80	0.990	0.983	0.989	0.990	balanced
HC90	0.989	0.974	0.988	0.986	balanced
HC100	0.985	0.962	0.985	0.983	balanced
HC110	0.980	0.944	0.980	0.979	RR5
HC120	0.978	0.927	0.978	0.976	RR5
HC130	0.976	0.915	0.976	0.975	RR5
HC140	0.974	0.897	0.975	0.974	RR5
HC150	0.975	0.839	0.975	0.973	dissoc.

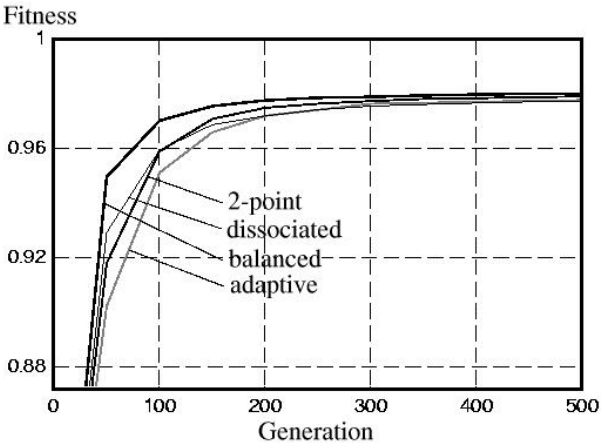

Figure 3: Average performance in 500 generations, HC50 graph

Table 5: Best £tness of 50 runs for the best combined method and the best single operator, HC problems (£tness maximum of 1)

Graph	Best single operator	Best combined method
HC50	0.995 (2-point)	0.991 (adaptive)
HC60	0.995 (dissoc.)	0.995 (RR10)
HC70	0.996 (dissoc.)	0.996 (RR5)
HC80	0.994 (uniform)	0.996 (balanced)
HC90	0.991 (uniform)	0.993 (RR5)
HC100	0.987 (uniform)	0.990 (RR5)
HC110	0.982 (dissoc.)	0.985 (balanced)
HC120	0.979 (dissoc.)	0.982 (RR5)
HC130	0.978 (dissoc.)	0.981 (RR5)
HC140	0.978 (dissoc.)	0.979 (RR5)
HC150	0.978 (dissoc.)	0.979 (RR5)

The Quality of the Solutions

For a different view of the quality of the solutions achieved by each method, we have searched for the best solution obtained in any of the 50 trials. Table 5 show the best solution found by a single crossover operator and by the best combined method for each problem. We can notice that the best solution found by the balanced scheme is in general better than the best solution obtained by any single operator, no matter which one it is.

The last experiment we have performed was intended to study the in¤uence of the mutation rate on the quality of the results for the various crossover operator and combining methods. Thus, we have plotted the average performance (£tness of the best individual) of the best single operator (dissociated), the best combined method (balanced), and the uniform crossover for a set of experiments without mutation and another set with a high mutation rate.

Figures 4 and 5 show these new results in 1000 generations with a mutation rate of 0.0 and of 0.01. The x axis represents the number of vertices in the graph. The size of the individual actually depends on the number of edges, which is signi£cantly larger than the number of vertices.

It is interesting to notice that the difference between the performance of the dissociated crossover and the other single operators increases with the problem size. In these experiments, the dissociated crossover is showing a better performance than the others and the difference increases with the graph size. Still, the performance of the combined balanced model is much closer to the dissociated crossover than the uniform operator.

The three models show a better performance without mutation rather than with a mutation rate of 0.01, which suggests that this mutation rate is too high for this class of problems. The mutation rate of 0.0005 that we have used before is a much better choice.

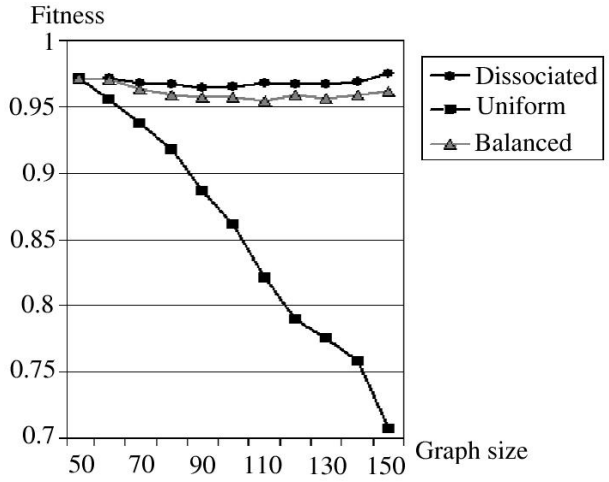

Figure 4: Average performance on HC problems without mutation

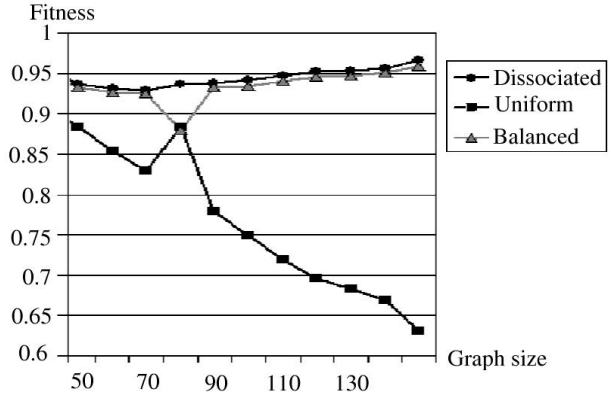

Figure 5: Average performance on HC problems with a mutation rate of 0.01

Conclusions

This paper has presented several combination methods for the crossover operator in an attempt to take advantage of the strength of each individual operator and produce better solutions to the problem in a more consistent way.

In the second section we have compared the four basic crossover operators, which are the 1-point, 2-point, uniform, and dissociated. The £rst experiments presented in Table 2 have shown that the best operator can be different from one problem to another.

In the third section we have de£ned several models combining the four crossover operators and we have tested their performance against the best single operator using the same test problems. The results from Table 4 show that a combined model can be consistently better than any of the single crossover operators. Further tests have con£rmed that the combined balanced model, that chooses between the four operators in a random fashion with equal probabilities for each of them, is also capable of generating solutions that are closer to the optimal one.

We can conclude that a model combining the power of several crossover operators is a more robust choice and can £nd better solutions under various circumstances..

References

Booker, L. 1987. Improving search in genetic algorithms. In Davis, L., ed., *Genetic Algorithms and Simulated Annealing*, 61–73. Morgan Kaufmann Publishers.

Brassard, G., and Bratley, P. 1994. *Fundamentals of Algorithmics*. Prentice-Hall.

De Jong, K., and Spears, M. 1989. Using genetic algorithms to solve NP-complete problems. In *Proceedings of the International Conference on Genetic Algorithms*, 124–132. Fairfax (VA): George Mason University.

Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading (MA): Addison-Wesley.

Holland, J. H. 1975. *Adaptation in Natural and Arti£cial Systems*. Ann Arbor: University of Michigan Press.

Hong, I.; Kahng, A.; and Moon, B. 1995. Exploiting synergies of multiple crossovers: initial studies. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, 245–250.

Maini, H.; Mehrotra, K.; Mohan, C.; and Ranka, S. 1994. Knowledge-based nonuniform crossover. *Complex Systems* 8:257–293.

Mao, J.; Hirasawa, K.; Hu, J.; and Murata, J. 2002. Increasing robustness of genetic algorithm. In Publishers, M. K., ed., *Proceedings of the Genetic and Evolutinary Computation Conference*, 456–462.

Mayer, H. 1998. ptGAs–Genetic algorithms evolving noncoding segments by means of promoter/terminator sequences. *Evolutionary Computation Journal* 6(4):361–386.

Mühlenbein, H., and Schlierkamp-Voosen, D. 1995. Analysis of selection, mutation and recombination in genetic algorithm. In Banuhaf, W., and Eeckman, F., eds., *Evolution as a Computational Process*, 188–214. Berlin: Springer.

Rana, S. 1999. The distributional biases of crossover operators. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 549–556. Orlando (FL): Morgan Kaufmann Publishers.

Schaffer, J. D., and Morishima, A. 1987. An adaptive crossover distribution mechanism for genetic algorithms. In Greffenstette, J., ed., *Proceedings of the Second International Conference on Genetic Algorithms*, 36–40. Hillsdale (NJ): Laurence Erlbaum.

Spears, W. 1990. Using neural networks and genetic algorithms as heuristics for NP-complete problems. Master's thesis, Department of Computer Science, George Mason University, Fairfax, Virginia.

Spears, W. 1995. Adapting crossover in evolutionary algorithms. In *Proceedings of the Fourth Annual Conference on Evolutionary Programming*.

Suzuki, H., and Iwasa, Y. 1999. Crossover accelerates evolution in GAs with a Babel-like £tness landscape: Mathematical analyses. *Evolutionary Computation Journal* 7(3):275–310.

Syswerda, G. 1989. Uniform crossover in genetic algorithms. In Schaffer, J. D., ed., *Proceedings of the International Conference on Genetic Algorithms*. San Mateo (CA): Morgan Kaufmann Publishers.

Vekaria, K., and Clack, C. 1998. Selective crossover in genetic algorithms: an empirical study. In et al., E., ed., *Proceedings of Parallel Problem Solving from Nature V*, 438–447. Springer-Verlag.

Vrajitoru, D. 1999. Genetic programming operators applied to genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 686–693. Orlando (FL): Morgan Kaufmann Publishers.

Whitley, D.; Mathias, K.; Rana, S.; and Dzubera, J. 1996. Evaluating evolutionary algorithms. *Arti£cial Intelligence* 85:245–276.

Wu, A.; Lindsay, R.; and Riolo, R. 1997. Empirical observations on the roles of crossover and mutation. In Bäck, T., ed., *Proceedings of the 7th International Conference on Genetic Algorithms*, 362–369. San Francisco: Morgan Kaufmann.