Crossover in Probability Spaces

Siddhartha Bhattacharyya
Information and Decision Sciences Dept..
College of Business Administration
University of Illinois at Chicago
601 S. Morgan Street (MC 294)
Chicago, IL 60607-7124
sidb@uic.edu

Marvin D. Troutt
Department of Administrative Sciences
Graduate School of Management
Kent State University
P.O. Box 5190
Kent, Ohio 44242-0001
mtroutt@bsa3.kent.edu

Abstract

This paper proposes a new crossover operator for searching over discrete probability spaces. The design of the operator is considered in the light of recent theoretical insights into genetic search provided by forma analysis. A non-trivial test problem in enforcing coherency of probability estimates in cross-impact analysis highlights the utility of the designed operators. The presented operators will be useful for a variety of probability-fitting and optimization applications.

1 INTRODUCTION

Probability estimates are essential input data for many decision support techniques, including expert systems, long range forecasting, data mining, decision analysis and cross-impact analysis, among others. This paper suggests a genetic algorithm (GA) based approach for searching over discrete probability spaces, that will be useful for a variety of probability fitting and optimization applications. The focus, in particular, in on the design of a crossover operator for effective search over a space of probability distributions.

The design of operators is considered in the light of recent theoretical insights into genetic search provided by forma analyses (Radcliffe, 1994). Forma theory provides an extension of the original schema analysis – for bit-string representations -- of genetic algorithms to arbitrary representations, and suggests principles for the design of effective genetic search operators.

We consider a direct real-number encoding of discrete probability distributions. Two crossover operators are developed, one in explicit consideration of forma processing principles, and the other a more “intuitive” operator along the lines of traditional single-point crossover. Both operators are analyzed in terms of the formae that they process, and their performance is evaluated on a non-trivial test problem.

The designed genetic search approach is illustrated on a problem of enforcing coherency of probability estimates in cross-impact analysis (see Dalkey, 1972; Duperrin and Godet, 1975; Gordon and Hayward, 1968; Helmer, 1977; Martino, 1983). The problem also allows solution through mathematical programming techniques, thus providing useful comparison for the genetic algorithm approach.

The next section briefly introduces forma analysis and the related design principles for genetic search operators. The design of the crossover operator and a mutation operator for searching over probability spaces are then considered in the following section. The third section then explains and develops the test problem, and is followed by a performance evaluation of the genetic search operators implemented

2 FORMA AND DESIGN PRINCIPLES

Schema analyses and the schema theorem (Holland, 1975) form the basis for most traditional theoretical analyses of genetic algorithms. In a recent generalization, Radcliffe (1994) extends the notion of schema to general equivalence classes, called forma. The schema theorem and intrinsic parallelism, key to the power of genetic search, is shown to apply to such general-purpose equivalence classes of solutions also. Developed in a series of papers (Radcliffe, 1991a, 1991b, 1992, 1994), forma theory focuses on the design of representations and search operators such that the schema (forma) theorem makes accurate estimates of the forma fitness, and thereby guides the search appropriately. Domain knowledge is brought to bear on the problem through specification of formae that group together solutions of related performance. Forma theory then establishes certain properties for the genetic search operators so as to foster effective search over the given space, considering the defined forma as building blocks in the search process. These properties can be used to analyze "heuristic" operators too.

Forma theory differs from traditional schema theory where the approach taken is to define a linear string representation of the search space so that standard crossover and mutation operators can be applied. Forma theory instead emphasizes an explicit characterization of appropriate forma in consideration of problem domain information, and considers search operators working directly on the search space, rather than on a fixed string encoding of it. Radcliffe (1994) explains how the two approaches are not equivalent; the former is shown to be more general.

In forma theory, domain knowledge is expressed through equivalence relations defined on the search space. Any relation between pairs of solutions that satisfy reflexivity, symmetry and transitivity can be used. These equivalence relations then induce a partitioning of the search space into equivalence classes, or formae. Solutions within a forma are thus equivalent under the equivalence relation. Equivalence relations define the genes, with corresponding forma defining the alleles in the string representation. The following example, adapted from Radcliffe, 1994) illustrates.

Example: For a space of humans, eye color could be an equivalence relation, useful if found to be an important determinant of some performance. Formae then specify sets of people with the same eye-color. Assuming only three colors, $\{x_{brown}, x_{bgreen}, x_{bblue}\}$ defines the set of formae induced by this equivalence relation. Here x_x denotes the set of people satisfying x. Similarly, considering an equivalence relation for hair-type gives another set of forma $\{x_{bcurly}, x_{stringht}\}$. We can now have some composite equivalence classes like $x_{brown,curly}$ defined by the equivalence relation of people with brown eyes and curly hair. Such composite equivalence relations do not form genes; but note that composite equivalence relations can be formed as the intersection of the *basic* equivalence relations (genes). If these equivalence relations uniquely identify each individual in the search space, they form a representation of the space. Obviously, eye-color and hair-type are inadequate for defining a representation since, in general, two people can have same eye-color and hair-type.

Radcliffe (1994) provides a rigorous definition of forma theory. Here we consider only those aspects necessary for our analysis in this paper. Consider *S* the search space, let x denote some forma on S and let Ξ be the set of formae considered. Then Ξ is a subset of the power set of S.

A set of formae Ξ is said to *cover* a search space if any solution is uniquely identified by the forma. Formally, Ξ covers S if and only if $\forall x \in S, \exists x_x \subseteq \Xi$ with $\bigcap_x x_x = x$.

Coverage is an important property since it ensures that forma analysis can distinguish between solutions.

Two formae x_1 and x_2 are *compatible* if a solution can belong to both. Two formae will thus be compatible if their intersection is non-empty, i.e. if $x_1 \bigcap x_2 \neq \emptyset$. It is desirable that compatible formae be *closed under intersection.* This allows formae to specify solutions with different degrees of accuracy and thereby gradually refine the search.

Any genetic operator R that recombines two parents in S to produce an offspring in S can be defined through a function $R\colon S \text{ X } S \text{ x } P_R \longrightarrow S$ where P_R is the control set that determines exactly which of the various possible offspring results from application of the operator. The non-deterministic nature of typically used genetic operators can be specified through some appropriate choice of control parameters in P_R. The control set for an operator need not be explicitly defined; any recombination operator that manipulates two parents to produce offspring in a reasonable manner can be used.

Note that this definition of operators is representation independent, and thus allows the formulation of operators directly on the search space. Such operators, however, need careful design so as to process the defined formae in a meaningful manner – operators should be able to exploit the domain knowledge inherent in the forma specification. In addition to the coverage and closure properties defined above, forma analysis provides three design principles for the genetic recombination operators: *respect, assortment, and ergodicity.* These are defined below.

Respect: A recombination operator *R* is said to respect a set of formae Ξ if recombining two instance of a particular forma produces an instance of the same forma. Formally *R* respects Ξ if and only if

$$\forall x \in \Xi, \forall f, g \in x, \forall p \in P_R : R(f, g, p) \in x \ .$$

Respect seeks to preserve genes common to both parents in their offspring as well. It is effective at reducing forma disruption, specially towards the later stages of the search when population diversity is low.

Assortment: A recombination operator *R* is said to assort a set of formae Ξ if, given two compatible formae, it is possible to recombine any two instances of these to produce an offspring that is in instance of both formae. Formally, *R* assorts Ξ if and only if $\forall x_1, x_2 \in \Xi$ such that $x_1 \bigcap x_2 \neq \emptyset$,
$\forall f \in x_1, \forall g \in x_2, \exists p \in P_R : R(f, g, p) \in x_1 \bigcap x_2$.

The assortment property relates to the building block hypothesis, and seeks to ensure that good building blocks from two parents *can be* effectively combined.

Ergodicity: Given any population, it should be possible to access any point in the search space through a finite sequence of applications of the genetic search operators. The mutation operator is generally used to provide for ergodicity.

A set of formae Ξ is said to be *separable* if a recombination (crossover) operator capable of respecting and assorting it exists. i.e. if the properties of respect and assortment are not contradictory. While desirable, the twin properties of respect and assortment, as indicated earlier, are incompatible in certain practical problems. Noting that most standard crossover operators respect schemata, Radcliffe (1994) writes:

"Thus if a respectful operator is used, the non-separability immediately means that assortment must fail to be achieved. The significance of this is that even when parents are chosen that contain all the genetic material (apparently) necessary to build some given child, it may be impossible for a respectful operator to construct that child. As well as being in conflict with reason, this would seem to make navigation around the search space unnecessarily difficult".

Thus, where formae are non-separable, assortment would seem to be the more desirable property sought; respect increases in relevance towards the later stages of search, as the population begins to converge.The forma properties and GA design principles given above, however, should not be considered as providing necessary or sufficient conditions for effective genetic search (Radcliffe, 1994). Rather, they provide a set of conditions which, if satisfied, is expected to yield improved search capabilities.

3 CROSSOVER AND FORMA IN PROBABILITY DISTRIBUTIONS

The operators are defined directly on the search space. A population member thus represents a solution directly as a string of real-numbers, each number corresponding to the probability of a specific event. Probability distributions impose the constraint that each value be in [0,1] and that all values sum to one. Thus, considering a problem with N events, a solution is represented as the string

$< s_1, s_2, \ldots, s_N >$ with $s_i \in [0, 1], i = 1,..,N$, and $\sum_{i=1}^{N} s_i = 1$.

We consider a crossover operator, obtained intuitively along the lines of traditional GA crossover operators that swap alleles in two parents to obtain two offspring. Here, the offspring are formed by exchanging the "front" and "tail" ends of the parent distributions.

A cumulative distribution function (CDF) representation facilitates the design of this operator. Each population member represents a CDF corresponding to the probability distribution $x = < x_1, x_2, \ldots, x_n >$. Then from any population member *S(x),* the probabilities can be determined as

$$\mathrm{x_i} = S\left(\frac{i}{n}\right) - S\left(\frac{i-1}{n}\right), \mathrm{i} = 1,\ldots,\mathrm{n}$$

with *S(0)=0* and *S(1)=1* being fixed. Thus each population member is represented as a n-component vector $(x_1,\ldots x_n)$ whose coordinates are monotone increasing to the right.

3.1 DISTRIBUTION SWAP CROSSOVER

Given two parents *F(x)* and *G(x),* the *distribution-swap crossover* operator obtains two offspring *P(x)* and *Q(x)* as follows:

1. Select a crossover site x_c = U[0,1] (uniformly in [0,1]).

$$2.\ P(x) = \begin{cases} \mathrm{F(x)\ if\ x \le x_c} \\ \mathrm{F(x_c) + (1 - F(x_c))[\frac{G(x) - G(x_c)}{1 - G(x_c)}]\ if\ x > x_c} \end{cases}$$

$$3.\ Q(x) = \begin{cases} \mathrm{G(x)\ if\ x \le x_c} \\ \mathrm{G(x_c) + (1 - G(x_c)[\frac{F(x) - F(x_c)}{1 - F(x_c)}]\ if\ x > x_c}. \end{cases}$$

(This CDF decomposition was employed earlier, in non-GA related work (Troutt and Paine, 1990).

Note that this operator considers distributions as continuous, and crossover sites can fall in between two defined positions on a string. This is easily avoided, where desirable, by choosing $x_c = (r/n)$ where $r = U[1,n]$; this ensures that the crossover site corresponds to some string position. The operator also exhibits a high positional bias as associated with traditional single-point crossover. This can be overcome in a uniform version of the operator:

1. Considering probabilities $(x_1,\ldots,x_n)$, select crossover positions uniformly ($[0,1]^n$). Let C be the set of selected crossover positions.
2. Rearrange the ordering of the probabilities x_i such that $\{x_i : i \in C\}$ appears to the left of $\{x_i : i \notin C\}$. Transform this rearranged probability distribution to CDF and perform crossover with crossover site $\mathrm{x_c} = \frac{\max\{i : i \in C\}}{n}$. If x_c is not required to correspond to any defined string position, it may be selected uniformly in $[x_c, x_{c+1}]$.

The Distribution Swap crossover preserves the probability ratios amongst events in the front and back ends of a distribution, i.e., on either side of the crossover site x_c. For example, given two parent probability distributions $f = < f_1, f_2, f_3, f_4 >$ and $g = < g_1, g_2, g_3, g_4 >$, and with the crossover site chosen to be, say, at the second string position, offspring $p = < p_1, p_2, p_3, p_4 >$ will maintain

$p_1/p_2=f_1/f_2$ and $p_3/p_4=g_3/g_4$, and similarly for the second offspring. This operator may then be considered as processing *Probability-Ratio forma*.

3.2 PROBABILITY-RATIO FORMA

Probability-ratio formae relate distribution strings having the same probability-ratios for the defining events. A probability distribution $f=<f_1,..,f_n>$ can be represented with its probability-ratio equivalent $f' =< ..., \frac{f_i}{f_{i+1}}, \frac{f_i}{f_{i+2}}, .. \frac{f_i}{f_n}, ... >$. The set of probability-ratio forma can be defined as $\Xi = \prod_j F_j$ where $F_j = \{\frac{f_k}{f_r}, k, r \in P\} \cup \{*\}$; here P denotes the set of events with specified probabilities in the distribution, and * is a don't care symbol corresponding to events with unspecified probabilities. Then, two strings f and g (distributions, or their probability-ratio equivalents) are equivalent or instances of a forma x^R according to:

$$f, g \in x^R \Leftrightarrow \exists P \subseteq \{1,..,n\}, \frac{f_p}{f_q} = \frac{g_p}{g_q} \; \forall p, q \in P .$$

For example, considering probability distributions f = <.2, .3, .4, .1> and g = <.1, .6, .2, .1>, their probability-ratio equivalents are

$$f' =< \frac{2}{3}, \frac{2}{4}, \frac{2}{1}, \frac{3}{4}, \frac{3}{1}, \frac{4}{1} > \quad \text{and} \quad g' =< \frac{1}{6}, \frac{1}{2}, \frac{1}{1}, \frac{6}{2}, \frac{6}{1}, \frac{2}{1} > .$$

Thus, both f and g are instances of the forma $< *, \frac{1}{2} *, *, *, * >$.

Distribution-swap crossover does not respect probability-ratio forma, as seen in the following example: consider parents *f':<0.1,0.3, 0.4, 0.2>* and *g:<0.2, 0.1, 0.3, 0.4>*. Note that $f_1/f_4=g_1/g_4= 0.5$, and a probability-ratio respecting operator will preserve this ratio in offspring as well. However, distribution swap crossover, with crossover site at, say, the second position yields offspring *<0.1, 0.3, 0.26, 0.34>* and *<0.2, 0.1, 0.47, 0.23>*, none of which maintain the ratio between the first and fourth event probabilities. Respect here, however, rises with increasing similarity in gene values as the search progresses towards convergence. Since respect is an important property in the later stages of search, distribution-swap crossover can thus still be expected to perform well.

The uniform version of the distribution swap crossover operator also assorts probability-ratio forma. Consider parents *f':<0.1,0.3, 0.4, 0.2>* and *g:<0.2, 0.1, 0.3, 0.4>*, or their probability ratio equivalents $f' :< \frac{1}{3}, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, \frac{3}{2}, 2 >$ and $g' :< 2, \frac{2}{3}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{3}{4} >$. Here f is an instance of forma $x_1 :< \frac{1}{3}, *, \frac{1}{2}, *, *, 2 >$ and g is an instance of the forma $x_2 :< *, *, \frac{1}{2}, *, \frac{1}{4}, * >$. Now, instances f and g of the above formae can be recombined by distribution-swap crossover to give offspring that are instances of both formae. Note that $x_1 \cap x_2 =< *, *, \frac{1}{2}, *, *, * >$, i.e., having $x_1/x_4=0.5$. Instances of this intersection forma are always obtainable from parents f and g by considering events x_1 and x_4 (first and fourth positions of the string) in the crossover set C (see description of the uniform version of the operator above).

3.3 MUTATION

As noted earlier, mutation plays a key role in implementing the ergodicity property. With both the crossover operators above, mutation is designed to randomly change the probability of any event. The probabilities of the other events will also then need to be adjusted so as to satisfy the probability distribution restriction (summation to unity). These adjustments are made proportional to their existing values.

In addition, noting that the crossover operator carries an inherent bias away from the end-points (0 and 1 probability values), the mutation operator is implemented to insert a 0-value to randomly selected events. This 0-value mutation shares equally with the regular mutation. Since a mutation to a value of 1 would reduce the probabilities of all the other events to 0, and is thus, in general, not expected to yield good solutions, we do not implement a 1-value mutation.

4 TEST PROBLEM: ENFORCING COHERENCY OF PROBABILITY ESTIMATES

We examine the performance of the genetic search operator on a problem of obtaining coherent probabilities estimates for a set of related events. Obtaining high quality probability estimates from decision makers is a problem that occurs with many decision support models and methods. Such estimates, particularly those of related probabilities, can fail to be consistent with the laws of probability theory, a situation known as incoherency. The desirability for coherence has been argued by many authors (Hogarth, 1975; Martino, 1983; Spelzler and von Holstein, 1975; Tversky and Kahneman, 1983). Here, we first develop a search space for representing related probabilities in such a way that coherent scenario and event probability estimates can be obtained. Genetic search then provides a general approach for finding estimates which are exactly coherent while being close to the decision maker's estimates as well. The constructed problem is also amenable to mathematical programming solution approaches, and provides an means for comparing with the designed genetic search operators.

4.1 TEST PROBLEM DESCRIPTION

The problem addressed in this paper may be described as follows. Given data of the type shown in Table 1, find coherent probability estimates for the various possible scenarios and events which are as nearly consistent as possible with the assignments of Table 1. Our starting point is an interpretation of the constraints in Table 1 as relationships between conditional probabilities and marginal probabilities as follows: the first constraint is interpreted as requiring

$$P(B|A) = P(B) + 0.25\,(1-P(B)). \qquad (1)$$

Similarly the next two give rise to

$$P(C|A) = P(C) - 0.10, \qquad (2)$$

and $P(C|\overline{D}) = P(C) + 0.3(1-P(C))$ (3)

respectively. The remaining three equations are respectively

$$P(A|D) = 0.5\,P(A), \qquad (4)$$

$$P(D|C) = P(D) + 0.2\,(1-P(D)), \qquad (5)$$

and $P(D|B) = 0.3\,P(D)$ (6)

Here we use the notation P(.) to denote the probability of event (.) and use the overbar to denote complementary events.

To develop the representation, consider a special type of sample space, *S*, whose sample points are potentially elements of one or more of the event sets *A, B, C,* and *D*. A particular point in *S* can belong to one and only one of the $2^4 = 16$ mutually exclusive subsets or scenarios characterized according to whether the point is or is not an element of each of the four sets *A*, *B*, *C*, and *D*. In fact, we may list and number these scenarios as in Table 2. Further, in Table 2, we associate a variable x_i to represent the relative frequency of sample points in scenario i. We call *S* a *flexible sample space* (FSS). That is, instead of each sample space point having equal probability, these points are allowed to have flexible probabilties denoted by the x_i.[1]

Since $\sum_{i=1}^{16} x_i$ must be unity, the reader may check that the relevant marginal and joint events involved in system (1)-(6) may be expressed as follows:

$$P(A) = \sum_{i=1}^{8} x_i \qquad (7)$$

$$P(C) = \sum_{i=1}^{2} x_i + \sum_{i=5}^{6} x_i + \sum_{i=9}^{10} x_i + \sum_{i=13}^{14} x_i \qquad (8)$$

$$P(B) = \sum_{i=1}^{4} x_i + \sum_{i=9}^{12} x_i \qquad (9)$$

$$P(D) = x_1 + x_3 + x_5 + \ldots + x_{15} \qquad (10)$$

$$P(B \cap A) = \sum_{i=1}^{4} x_i \qquad (11)$$

$$P(C \cap A) = \sum_{i=1}^{2} x_i + \sum_{i=5}^{6} x_i \qquad (12)$$

$$P(C \cap \overline{D}) = x_2 + x_6 + x_{10} + x_{14} \qquad (13)$$

$$P(A \cap D) = x_1 + x_3 + x_5 + x_7 \qquad (14)$$

$$P(D \cap C) = x_1 + x_5 + x_9 + x_{13} \qquad (15)$$

$$P(D \cap B) = x_1 + x_3 + x_9 + x_{11} \qquad (16)$$

The requirements of Table 1 may now be expressed as follows after some simple algebra:

$$\sum_{i=1}^{4} x_i = 0.25\sum_{i=1}^{8} x_i + 0.75\,(\sum_{i=1}^{4} x_i + \sum_{i=9}^{12} x_i)(\sum_{i=1}^{8} x_i) \qquad (17)$$

$$\sum_{i=1}^{2} x_i + \sum_{i=5}^{6} x_i + 0.1\sum_{i=1}^{8} x_i = (\sum_{i=1}^{2} x_i + \sum_{i=5}^{6} x_i + \sum_{i=13}^{14} x_i)(\sum_{i=1}^{8} x_i) \qquad (18)$$

$$x_2 + x_6 + x_{10} + x_{14} = 0.3\,(x_2 + x_4 + \ldots + x_{16}) +$$
$$0.7\,(x_2 + x_4 + \ldots + x_{16})(\sum_{i=1}^{2} x_i + \sum_{i=5}^{6} x_i + \sum_{i=9}^{10} x_i + \sum_{i=13}^{14} x_i) \qquad (19)$$

$$x_1 + x_3 + x_5 + x_7 = 0.5\,(\sum_{i=1}^{8} x_i)(x_1 + x_3 + \ldots + x_{15}) \qquad (20)$$

$$x_1 + x_5 + x_9 + x_{13} = (0.2 + 0.8(x_1 + x_3 + \ldots + x_{15})) \qquad (21)$$

$$x_1 + x_3 + x_9 + x_{11} = 0.3\,(x_1 + x_3 + \ldots + x_{15})(\sum_{i=1}^{4} x_i + \sum_{i=9}^{12} x_i)$$
$$(\sum_{i=1}^{2} x_i + \sum_{i=5}^{6} x_i + \sum_{i=9}^{10} x_i + \sum_{i=13}^{14} x_i) \qquad (22)$$

$$\sum_{i=1}^{8} x_i = 0.5 \qquad (23)$$

$$\sum_{i=1}^{4} x_i + \sum_{i=9}^{12} x_i = 0.4 \qquad (24)$$

$$x_1 + x_3 + \ldots + x_{15} = 0.1 \qquad (25)$$

---

[1] The FSS approach appears to be a relatively straightforward modeling tool which may have been used elsewhere. It appears to be essentially the same as what some have called the minimal relevant sample space.

The last three requirements, (23)-(25), are due to the initial marginal probability estimates in the lower half of Table 1.

Thus the FSS decomposition describes a hypothetical sample space in terms of the possible elementary scenarios related to the particular set of events of interest. All possible joint, marginal and conditional probabilities of these events can therefore be modeled in a coherent way. It is important to emphasize here that coherency is strictly enforced by this scheme. That is, all information about the events *A, B, C, D,* their intersections, complements, and other set operations, is contained in the x_i, i = 1, 16. The reader may check that coherency requirements, such as $P(A \cap B) = P(A|B)\ P(B)$, reduce to identities in the x_i. Moreover the above points remain true even if the system (17)-(25) is solved only approximately. Thus the FSS model used here precisely enforces coherency but permits some possible lack of fit with the decision maker's data such as given by Table 1.

The equations (17)-(25) constitute a set of nine constraints, some of which are nonlinear, in sixteen variables. In addition, it is necessary that $\sum_{i=1}^{16} x_i = 1.0$ and $x_i \geq 0$, i=1,...,16. Hence, it is not known *a priori* whether an exact solution exists. We might therefore seek scenario probabilities x_i which come as close to these conditions as possible in the least squared error sense. Define e_1 as the difference of left and right hand sides of (17), e_2 similarly for (18) and so on to e_9 for (25). This line of reasoning would lead to the constrained programming problem (26)-(27) as follows:

$$\min \sum_{i=1}^{9} e_i^2 \qquad (26)$$

$$\text{s.t.} \sum_{i=1}^{16} x_i = 1, \quad x_i \geq 0, \ i = 1,..16\ . \qquad (27)$$

This problem was coded and solved using GAMS. Table 3 shows the results for both the scenario probabilities and the probabilities of events *A, B, C* and *D.*

Note that the objective function in (26) can not easily be checked for convexity. The GAMS solution thus might not guarantee a global optimum. In order to compute a global minimum of (26), one would need a starting point grid search procedure. However, if the grid mesh is based on dividing the interval [0, 1] into k points, then given the 16 variable problem at hand, the number of such search points is of the order of k^{15}. Thus this strategy is therefore unattractive even for the small example at hand. A genetic algorithm approach proves much more desirable.

4.2 SOLUTION USING GENETIC ALGORITHMS

Since the GA seeks solutions with high fitness, the following fitness-function, consistent with the objective of (26) is used: $f_j = (\sum_{i=1}^{9} e_i^2)^{-2}$ where f_j is the fitness of the j^{th} population member, and e_i , i = 1,..,9 are the errors as determined by (17)-(25). The second power in the above expression provides for greater discrimination between solutions with close values of the total sum of squared errors (SSE).

Table 4 presents the SSE, the scenario probabilities and the probabilities of *A, B, C,* and *D* as obtained using genetic search with the distribution-swap operators. These results correspond to the following settings for the GA parameters: a population size of 100 was maintained for each simulated generation, and the best fitness number retained intact in the next generation (elitist selection). The mutation rate, was set at 0.025 per string position and the crossover rate used was 0.7. The search was terminated after 500 generations. Table 4 presents three different solutions obtained with different random number streams. All the GA solutions are found to outperform the GAMS solution.

5 CONCLUSION

This paper considers the design of a crossover operator for search over discrete probability spaces. A test problem illustrates an application to the estimation of coherent probability assignments. More generally, these procedures apply to optimization over such spaces, or simplexes described by barycentric coordinates. The crossover operator is an intuitive heuristic that forms offspring through an exchange of the front and tail ends of distributions. This distribution swap crossover operator is shown to process what may be called probability-ratio forma. The operator assorts, but does not fully respect probability-ratio forma; however, respect holds to a degree which increases as convergence in the population advances.

Genetic search using the designed crossover operator is found to perform significantly better than a conventional mathematical programming approach for the test problem. It should be noted, however, that the forma design principles of assortment and respect are neither necessary nor sufficient for effective genetic search; rather, they should be looked upon as properties expected to facilitate successful search. The success of the distribution-swap operator adds credence to the argument that in situations where the twin properties of respect and assortment are in conflict, forma assortment is more desirable, with respect gaining in relevance as the population converges. The distribution-swap crossover, analyzed as processing probability-ratio forma, possesses precisely these characteristics.

References

Dalkey, N.C., “An Elementary Cross-Impact Model”, Technol. Forecast. Soc. Change, 3, 1972, 341-352.

Davis, L. ed., Handbook of Genetic Algorithms, 1991, Van Nostrand Reinhold.

Duperrin, J.C. and M. Godet, “SMIC 74 - A Method for Constructing and ranking Scenarios”, Futures, 7 (4), 1975.

Eshelman, L.J., R.A. Caruna and J.D. Schaffer, “Biases in the Crossover Landscape”, Proc. of the Third International Conference on Genetic Algorithms, J. D.Schaffer (ed.), 1989, 10-19.

Gordon, T.J. and H. Hayward, “Initial Experiment with the Cross-Impact Matrix Method of Forecasting”, Futures, 1, No. 2, Dec. 1968.

Hogarth, R.M., “Cognitive processes and the assessment of subjective probability distributions”, Journal of the American Statistical Association, 1975.

Helmer, O., "Problems in Futures Research", Futures, 9 (1), 1977, 2-31.

Holland, J. H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

Michalewicz, Z., Genetic Algorithms + Data Structures = Evolutionary Programs, 2nd Edition, 1994, Springer-Verlag.

Martino, J.P., Technological forecasting for decision making, 2nd Edition, North-Holland, New York, NY, 1983.

Moskowitz, H., and Sarin, R.K., “Improving the consistency of conditional probability assessments for forecasting and decision making”, Management Science, 29, 1983, 735-749.

Radcliffe, N.J., “The Algebra of Genetic Algorithms”, Annals of Maths and Artificial Intelligence, vol. 10, 1994, pp.339-384.

Radcliffe, N.J., “Genetic Set Recombination”, Foundations of Genetic Algorithms, 2, L.D. Whitley (ed.), 1992, Morgan Kaufmann.

Radcliffe, N.J. “Forma Analysis and Random Respectful Recombination”, Proc. Fourth International Conference on Genetic Algorithms, 1991a, pp.221-229, Morgan Kaufmann.

Radcliffe, N.J., “Equivalence Class Analysis of Genetic Algorithms”, Complex Systems, 5, 1991b, pp. 183-205.

Radcliffe, N.J. and P.D. Surry, “Fitness Variance of Forma and Performance Prediction”, Foundations of Genetic Algorithms, III, M.D. Vose and L.D. Whitley (eds.), pp. 51-72, 1994, Morgan Kaufmann.

Spetzler, C.S., and Staël von Holstein C.A.S., “Probability Encoding in Decision Analysis”, Management Science, 1975, 22, 340-58.

Surry, P.D. and N.J. Radcliffe, “Fomal Algorithms + Formal Representations = Search Strategies”, Parallel Problem Solving from Nature, IV, 1996, Springer Verlag.

Syswerda, G., “Uniform Crossover in Genetic Algorithms”, Proc. of the Third International Conference on Genetic Algorithms, 1989, Morgan Kaufmann.

Troutt, M.D., and Paine, T.B., “A monotone variational method for optimal probability distributions”, OR Spektrum, 12:201-206, 1990.

Tversky, A. and Kahneman, D., “Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment”, Psychological Review, 90, 1983, 293-315.

Table 1: Data for four hypothetical interrelated events

1. If event A occurs then the probability of event B is increased by 25% of its allowable increase.
2. If event A occurs then the probability of event C is decreased 10% (or as near to 10% as possible).
3. If event D does not occur, then the probability of event C is increased by 30% of its allowable increase.
4. If event D occurs, then the probability of event A is decreased 50%.
5. If event C occurs, then the probability of event D is increased 20% of its allowable increase.
6. If event B occurs, then the probability of event D is decreased 30% of its allowable decrease.

Initial Probability Estimates: P(A) = .50, P(B) = .40, P(C) = Not Specified, P(D) = .10

Table 2: Scenarios and their relative frequency variables

Scenario Number	Relative Frequency	Scenario*	Scenario Number	Relative Frequency	Scenario*	Scenario Number	Relative Frequency	Scenario*
1	x_1	$A\,B\,C\,D$	7	x_7	$A\,\bar{B}\,\bar{C}\,D$	13	x_{13}	$\bar{A}\,\bar{B}\,C\,D$
2	x_2	$A\,B\,C\,\bar{D}$	8	x_8	$A\,\bar{B}\,\bar{C}\,\bar{D}$	14	x_{14}	$\bar{A}\,\bar{B}\,C\,\bar{D}$
3	x_3	$A\,B\,\bar{C}\,D$	9	x_9	$\bar{A}\,B\,C\,D$	15	x_{15}	$\bar{A}\,\bar{B}\,\bar{C}\,D$
4	x_4	$A\,B\,\bar{C}\,\bar{D}$	10	x_{10}	$\bar{A}\,B\,C\,\bar{D}$	16	x_{16}	$\bar{A}\,\bar{B}\,\bar{C}\,\bar{D}$
5	x_5	$A\,\bar{B}\,C\,D$	11	x_{11}	$\bar{A}\,B\,\bar{C}\,D$			
6	x_6	$A\,\bar{B}\,C\,\bar{D}$	12	x_{12}	$\bar{A}\,B\,\bar{C}\,\bar{D}$			

* For example, $A\,\bar{B}\,C\,\bar{D}$ may be described as the scenario in which events A and C both occur, while B and D fail to occur.

Table 3: GAMS solution

SSE	P(A)	P(B)	P(C)	P(D)	x_i, i=1,…16 (by row order)							
0.0347	0.498	0.402	0.873	0.118	0 .0	0.172	0.0	0.103	0.111	0.105	0.007	0.0
					0.79	0.037	0.0	0.011	0.0	0.374	0.0	0.0

Table 4: Genetic Algorithm solutions (Distribution-Swap Crossover)

SSE	P(A)	P(B)	P(C)	P(D)	x_i, i=1,…16 (by row order)							
0.0203	0.497	0.410	0.753	0.120	0 .02	0.0	0 .19	0.087	0.039	0.086	0.0	0.095
					0.014	0.07	0.0	0.05	0.066	0.288	0.0	0.145
0.0221	0.495	0.410	0.671	0.123	0.022	0.0001	0.054	0.223	0.048	0.167	0.0	0.002
					0.026	0.097	0.0002	0.009	0.019	0.26	0.0295	0.065
0.0198	0.5	0.399	0.774	0.129	0.019	0.008	0.111	0.0	0.156	0.027	0.189	0.0
					0.009	0.003	0.121	0.0	0.09	0.223	0.0	0.062