# Comparing Evolutionary Computation Techniques via Their Representation

Boris Mitavskiy

Department of Mathematics
University of Michigan
Ann Arbor MI 48109
bmitavsk@umich.edu

**Abstract.** In the current paper a rigorous mathematical language for comparing evolutionary computation techniques via their representation is developed. A binary semi-genetic algorithm is introduced, and it is proved that in a certain sense any reasonable evolutionary search algorithm can be re-encoded by a binary semi-genetic algorithm (see corollaries 15 and 16). Moreover, an explicit bijection between the set of all such re-encodings and the collection of certain $n$-tuples of invariant subsets is constructed (see theorem 14). Finally, all possible re-encodings of a given heuristic search algorithm by a classical genetic algorithm are entirely classified in terms of invariant subsets of the search space in connection with Radcliffe's forma (see [9] and theorem 20).

## 1 Introduction

Over the past 25 years evolutionary algorithms have been widely exploited to solve various optimization problems. In order to apply an evolutionary algorithm to attack a specific optimization problem, one needs to model the problem in a suitable manner. That is, one needs to construct a search space $\Omega$ (the set whose elements are all possible solutions to the problem) together with a computable positive valued fitness function $f : \Omega \to (0, \infty)$ and an appropriate family of "mating" and "mutation" transformations. One can say, therefore, that a representation of a given problem by an evolutionary algorithm is an ordered 4-tuple $(\Omega, \mathcal{F}, \mathcal{M}, f)$ where $\Omega$ is the search space, $\mathcal{F}$ is a family of binary operations on $\Omega$ and $\mathcal{M}$ is the family of unary transformations on $\Omega$, that is, $\mathcal{M}$ is just a family of functions from $\Omega$ to itself. Intuitively $\mathcal{F}$ is the family of mating transformations: every element of $\mathcal{F}$ takes two elements of $\Omega$ (the parents) and produces one element of $\Omega$ (the child).[1] while $\mathcal{M}$ is the family of mutations (or asexual reproductions) on $\Omega$. For theoretical purposes it is usually assumed that $\mathcal{M}$ is ergodic in the sense that the only invariant subsets under $\mathcal{M}$ are the $\emptyset$ and

[1] In general there is no reason to assume that a child has exactly two parents. All of the results in this paper are valid for the families of $m$-ary operations on $\Omega$. The only reason $\mathcal{F}$ is assumed to be the family of binary transformations is to simplify the notation.

E. Cantú-Paz et al. (Eds.): GECCO 2003, LNCS 2723, pp. 1196–1209, 2003.

the entire search space $\Omega$. (The ergodicity assumption ensures that the Markov process modelling the algorithm is irreducible ( see, for instance, [4] ). A typical evolutionary algorithm works as follows: A population $P = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2m} \end{pmatrix}$ with $x_i \in \Omega$ is selected randomly. The algorithm cycles through the following stages:

**Evaluation:**

Individuals of $P$ are evaluated:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2m} \end{pmatrix} \begin{matrix} \rightarrow & f(x_1) \\ \rightarrow & f(x_2) \\ \vdots & \vdots \\ \rightarrow & f(x_{2m}) \end{matrix}$$

**Selection:**

A new population

$$P' = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{2m} \end{pmatrix}$$

is obtained where $y_i = x_j$ with probability $\frac{f(x_j)}{\Sigma_{l=1}^{2m} f(x_l)}$.

In other words, all of the individuals of $P'$ are these of $P$, and the expectation of the number of occurrences of any individual of $P$ in $P'$ is proportional to the number of occurrences of that individual in $P$ times the individual's fitness value. In particular, the fitter the individual is, the more copies of that individual are likely to be present in $P'$. On the other hand, the individuals having relatively small fitness value are not likely to enter into $P'$ at all. This is designed to imitate the natural survival of the fittest principle.

**Partition:**

The individuals of $P'$ are partitioned into $m$ pairwise disjoint couples for mating according to some probabilistic rule: For instance the couples could be

$$Q_1 = \begin{pmatrix} y_{i_1^1} \\ y_{i_2^1} \end{pmatrix} \; Q_2 = \begin{pmatrix} y_{i_1^2} \\ y_{i_2^2} \end{pmatrix} \; \ldots \; Q_j = \begin{pmatrix} y_{i_1^j} \\ y_{i_2^j} \end{pmatrix} \; \ldots \; Q_m = \begin{pmatrix} y_{i_1^j} \\ y_{i_2^j} \end{pmatrix}$$

**Reproduction:**

Replace every one of the selected couples $Q_j = \begin{pmatrix} y_{i_1^j} \\ y_{i_2^j} \end{pmatrix}$ with the couples

$$Q' = \begin{pmatrix} T_1(y_{i_1^j}, y_{i_2^j}) \\ T_2(y_{i_1^j}, y_{i_2^j}) \end{pmatrix}$$

for some couple of transformations $(T_1, T_2) \in \mathcal{F}^2$. The couple $(T_1, T_2)$ is selected according to a fixed probability distribution on $\mathcal{F}^2$. This gives us a new population

$$P'' = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_{2m} \end{pmatrix}$$

**Mutation:**

Finally, with small probability we replace $z_i$ with $F(z_i)$ for some randomly chosen $F \in \mathcal{M}$. This, once again, gives us a new population $P''' = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{2m} \end{pmatrix}$

Upon completion of mutation start all over with the initial population $P'''$. The cycle is repeated a certain number of times depending on the problem. A more general and extensive description is given in [18]. The importance of choosing a reasonable representation for a specific problem is emphasized in some of the modern research. See, for instance, [10]. A few special evolutionary algorithms are introduced in the next section.

## 2 Special Evolutionary Algorithms

**Classical Genetic Algorithm with Masked Crossover:**

Let $\Omega = \prod_{i=1}^{n} A_i$. For every subset $M \subseteq \{1, 2, \ldots, n\}$, define a binary operation $L_M$ on $\Omega$ as follows:

$$L_M(\mathbf{a}, \mathbf{b}) = (x_1, x_2, \ldots, x_i, \ldots, x_n)$$

where $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n) \in S$ and $x_i = \begin{cases} a_i & \text{if } i \in M \\ b_i & \text{otherwise} \end{cases}$.
The reader will recognize $L_M$ as a masked crossover operator with mask $M$.

Let $\mathcal{F} = \{L_M \mid M \subseteq \{1, 2, \ldots, n\}\}$. That is, $\mathcal{F}$ in this example is simply the family of masked crossover transformations. The probability distribution on the set $\mathcal{F}^2$ is concentrated on the pairs of the form $(L_M, L_{\bar{M}})$ where $\bar{M}$ denotes the complement of the set $M$ in $\{1, 2, \ldots, n\}$. Most often the pairs are equally likely to be chosen from that diagonal-like subset.

**Example:** Let $n = 5$ and $A_i = \{0, 1, \ldots, i+1\}$. Suppose a given population $P$ consists of 6 individuals which are the rows of the matrix

$$\begin{pmatrix} 2 & 3 & 4 & 5 & 6 \\ 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 0 & 0 & 1 & 2 & 3 \\ 1 & 1 & 0 & 1 & 2 \\ 1 & 2 & 1 & 5 & 4 \end{pmatrix}$$

Say, after selection stage is complete one obtains the following population

$$P' = \begin{pmatrix} 2\,3\,4\,5\,6 \\ 2\,3\,4\,5\,6 \\ 1\,2\,3\,4\,5 \\ 0\,0\,1\,2\,3 \\ 0\,1\,2\,3\,4 \\ 1\,2\,3\,4\,5 \end{pmatrix}$$

Now the following individuals are paired for mating: (masked crossover in this case)

$$Q_1 = \begin{pmatrix} 2\,3\,4\,5\,6 \\ 0\,0\,1\,2\,3 \end{pmatrix},\; Q_2 = \begin{pmatrix} 2\,3\,4\,5\,6 \\ 1\,2\,3\,4\,5 \end{pmatrix},\; \text{and } Q_3 = \begin{pmatrix} 0\,1\,2\,3\,4 \\ 1\,2\,3\,4\,5 \end{pmatrix}$$

Suppose we have chosen the masks $M_1 = \{1, 4, 5\}$, $M_2 = \{1, 2\}$ and $M_3 = \{3, 4\}$ for the crossover of pairs $Q_1$, $Q_2$ and $Q_3$ respectively. In the language of this paper it means we have chosen the pairs of transformations $(T_{M_1}, T_{\bar{M}_1})$ for $Q_1$, $(T_{M_2}, T_{\bar{M}_2})$ for $Q_2$ and $(T_{M_3}, T_{\bar{M}_3})$ for $Q_3$ respectively. Upon applying these we obtain

$$Q_1 \to \begin{pmatrix} T_{M_1}((2,\,3,\,4,\,5,\,6),(0,\,0,\,1,\,2,\,3)) \\ T_{\bar{M}_1}((2,\,3,\,4,\,5,\,6),(0,\,0,\,1,\,2,\,3)) \end{pmatrix} = \begin{pmatrix} 2\,0\,1\,5\,6 \\ 0\,3\,4\,2\,3 \end{pmatrix}$$

Likewise

$$Q_2 \to \begin{pmatrix} T_{M_2}((2,\,3,\,4,\,5,\,6),(1,\,2,\,3,\,4,\,5)) \\ T_{\bar{M}_2}((2,\,3,\,4,\,5,\,6),(1,\,2,\,3,\,4,\,5)) \end{pmatrix} = \begin{pmatrix} 2\,3\,3\,4\,5 \\ 1\,2\,4\,5\,6 \end{pmatrix}$$

and, finally,

$$Q_3 \to \begin{pmatrix} T_{M_3}((0,\,1,\,2,\,3,\,4),(1,\,2,\,3,\,4,\,5)) \\ T_{\bar{M}_2}((0,\,1,\,2,\,3,\,4),(1,\,2,\,3,\,4,\,5)) \end{pmatrix} = \begin{pmatrix} 1\,2\,2\,3\,5 \\ 0\,1\,3\,4\,4 \end{pmatrix}$$

The family of mutation transformations, $\mathcal{M}$ in this (and in all of the following examples) consists of the transformations $M_{\boldsymbol{a}} : \Omega \to \Omega$ where $\boldsymbol{a} \in \bigcup_{S \subseteq \{1,\,2,\ldots,n\}} \prod_{i \in S} A_i$ so that $\boldsymbol{a} = (a_{i_1}, a_{i_2}, \ldots, a_{i_k})$ for $i_1 \leq i_2 \leq \ldots \leq i_k \in S_{\boldsymbol{a}} \subseteq \{1, 2, \ldots, n\}$ defined as follows: $\forall\, \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Omega$ we have $M_{\boldsymbol{a}}(\mathbf{x}) = \mathbf{y} = (y_1, y_2 \ldots, y_n)$ where $y_q = \begin{cases} a_q & \text{if } q = i_j \text{ for some } j \\ x_q & \text{otherwise} \end{cases}$ In other words, $M_{\boldsymbol{a}}$ simply replaces the $q^{\text{th}}$ coordinate of its argument with $a_q \in A_i$ whenever $q \in S_{\boldsymbol{a}}$.

**Random Respectful Recombination**

This type of algorithm first appeared in [9] under the name of Random Respectful Recombination, but it didn't seem to be useful at first[2]. Here the search space $\Omega$ and the family of mutation transformations, $\mathcal{M}$, are exactly the same as

[2] Recently a variation of this technique, known as "gene pool recombination" has been considered in [16], [7] and [19]

in the example of classical genetic algorithm, and the family of mating transformations is described below: In [8] these were named Holland transformations (because their corresponding fixed family of subsets is precisely the collection of subsets of $\Omega$ determined by the classical Holland schemata together with the empty set. See examples following corollary 12 in the next section). For every given point $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \Omega$ define a Holland transformation $T_{\mathbf{u}} : \Omega^2 \to \Omega$ as follows: for every $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and $\mathbf{b} = (b_1, b_2, \ldots, b_n) \in S$

$$T_{\mathbf{u}}(\mathbf{a}, \mathbf{b}) = (x_1, x_2, \ldots, x_n)$$

where

$$x_i = \begin{cases} a_i \text{ if } a_i = b_i \\ u_i \text{ otherwise} \end{cases}$$

In other words, if the $i^{\text{th}}$ coordinates of $\mathbf{a}$ and $\mathbf{b}$ coincide, then the $i^{\text{th}}$ coordinate of $T_{\mathbf{u}}(\mathbf{a}, \mathbf{b})$ also coincides with them. If, on the other hand, the $i^{\text{th}}$ coordinates of $\mathbf{a}$ and $\mathbf{b}$ differ, then the $i^{\text{th}}$ coordinate of $T_{\mathbf{u}}(\mathbf{a}, \mathbf{b})$ is that of $\mathbf{u}$, namely, $u_i$. Let $\mathcal{F} = \{T_{\mathbf{u}} \mid \mathbf{u} \in S\}$ be the family of all Holland transformations. At every iteration of the algorithm, once a new population $P$ is obtained, a new probability distribution on $\mathcal{F}$ is defined: $T_{\mathbf{u}}$ is chosen from $\mathcal{F}$ so that $u_i$ occurs in $\mathbf{u}$ with the probability proportional to its fitness in $P$. Every transformation in the pair $(T_{\mathbf{u}}, T_{\mathbf{v}})$ is chosen independently.

**Binary Genetic Algorithm with Masked Crossover:**

When every $A_i = \{0, 1\}$ (which means that $\Omega = \{0, 1\}^n$) in the example above, one obtains the classical binary genetic algorithm.

**Binary Random Respectful Recombination**

The search space $\Omega$ and the family of mating transformations $\mathcal{F}$ and the family of mutations $\mathcal{M}$ are exactly the same as these for the binary genetic algorithm with masked crossover described above. The only difference is that the probability distribution on $\mathcal{F}^2$ is now completely uniform. (rather than being concentrated on the diagonal-like subset described in the classical genetic algorithm example) For instance, if $n = 5$, $M_1 = \{2,\ 3,\ 4\}$, $M_2 = \{1,\ 3,\ 5\}$ and the pair $(T_{M_1}, T_{M_2})$ is selected for mating, we have, for instance,

$$\begin{pmatrix} 1\,0\,0\,1\,1 \\ 1\,1\,0\,0\,1 \end{pmatrix} \longmapsto \begin{pmatrix} T_{M_1}((1,\ 0,\ 0,\ 1,\ 1),\ (1,\ 1,\ 0,\ 0,\ 1)) \\ T_{M_2}((1,\ 0,\ 0,\ 1,\ 1),\ (1,\ 1,\ 0,\ 0,\ 1)) \end{pmatrix} = \begin{pmatrix} 1\,1\,0\,0\,1 \\ 1\,1\,0\,0\,1 \end{pmatrix}$$

This type of a binary search algorithm can be classified by the following property: If both parents have a 1 in the $i^{\text{th}}$ position then the offspring also has a 1 in the $i^{\text{th}}$ position. Likewise, if both parents have a 0 in the $i^{\text{th}}$ position then the offspring also has a 0 in the $i^{\text{th}}$ position. If, on the other hand, the alleles of the $i^{\text{th}}$ gene don't coincide, then the $i^{\text{th}}$ allele could be either a 0 or a 1.

The following type of algorithm may seem useless at first. Its importance will become clear in the next section when we present the binary embedding theorem which shows that the binary semi-genetic algorithm (described below) possesses an interesting universal property.

**Binary Semi-genetic Algorithm:**

The search space $\Omega = \{0, 1\}^n$, just as in the case of the binary genetic algorithm. The family of mating transformations $\mathcal{F}$ is defined as follows: Fix $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \Omega$. Define a semi-crossover transformation $F_{\mathbf{u}} : \Omega^2 \to \Omega$ as follows: For any given pair $(\mathbf{x}, \mathbf{y}) \in \Omega^2$ with $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ we have $F_{\mathbf{u}}(\mathbf{x}, \mathbf{y}) = \mathbf{z} = (z_1, z_2, \ldots z_n) \in \Omega$ where

$$z_i = \begin{cases} 1 & \text{if } x_i = y_i = 1 \\ u_i & \text{otherwise} \end{cases}$$

In other words, $F_{\mathbf{u}}$ preserves the $i^{\text{th}}$ gene if it is equal to 1 in all of the rows of $P$, and replaces it with $u_i$ otherwise. Let $\mathcal{F} = \{F_{\mathbf{u}} \,|\, \mathbf{u} \in \Omega\}$ be the family of all semi-crossover transformations. The family of mutation transformations $\mathcal{M}$ is exactly the same as in the examples above.

**Example:** With $n = 5$ and $\mathbf{u}_1 = (0, 1, 1, 0, 1)$, $\mathbf{u}_2 = (0, 1, 0, 0, 1)$ we have

$$\begin{pmatrix} 1\,0\,0\,1\,1 \\ 1\,1\,0\,0\,1 \end{pmatrix} \longmapsto \begin{pmatrix} F_{\mathbf{u_1}\,2}((1, 0, 0, 1, 1), (1, 1, 0, 0, 1)) \\ F_{\mathbf{u_2}\,2}((1, 0, 0, 1, 1), (1, 1, 0, 0, 1)) \end{pmatrix} = \begin{pmatrix} 1\,1\,1\,0\,1 \\ 1\,1\,0\,0\,1 \end{pmatrix}$$

Notice, that if 1 is present in the $i^{th}$ position of both parents, then it remains in the $i^{\text{th}}$ position of both offsprings. There are absolutely no other restrictions, though.

In practice the choice of the search space $\Omega$ is primarily determined by the specific problem and related circumstances. The general methodology for the construction of the search spaces first appeared in the work of Radcliffe (see, for instance, [9]). Radcliffe introduced the notion of a forma which captures the essential properties of the Holland schemata in a representation independent setting. A forma is simply a partition of the search space into equivalence classes. A given collection of forma with suitable properties (see [9]) is, in a sense, no different from the collection of the classical Holland schemata provided that one encodes the search space using the "genetic representation function" which is also introduced in [9]. The connection between all of the possible families of mating transformations on a given search space $\Omega$ and the corresponding families of invariant subsets established in [8] will allow us to extend Radcliffe's notion of the genetic representation function to compare various evolutionary algorithms via possible encodings of their search spaces. This idea will be made clear in the following section.

## 3 The Binary Embedding Theorem

As we have seen in the introduction, a given evolutionary heuristic search algorithm is determined primarily by the ordered 4-tuple $(\Omega, \mathcal{F}, \mathcal{M}, f)$. In the current paper we shall only be concerned with the search space $\Omega$, the family of mating transformations $\mathcal{F}$ and the family of mutations $\mathcal{M}$. As mentioned in the introduction, the family of mutation transformations is ergodic, meaning that the only invariant subsets under $\mathcal{M}$ is the $\emptyset$ and the entire search space $\Omega$. This motivates the following definitions:

**Definition 1** For a given family of $m$-ary operations $\Gamma$ on a set $\Omega$ (that is, functions from $\Omega^m$ into $\Omega$) a subset $S \subseteq \Omega$ is invariant under $\Gamma$ if and only if $\forall T \in \Gamma$ we have $T(S^m) \subseteq S$. We shall denote by $\Lambda_\Gamma$ the family of all invariant subsets of $\Omega$ under $\Gamma$. In other words, $\Lambda_\Gamma = \{S \mid S \subseteq \Omega,\ T(S^m) \subseteq S\ \forall\, T \in \Gamma\}$.

**Definition 2** A heuristic 3-tuple $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$ is a 3-tuple where $\Omega$ denotes an arbitrary set, $\mathcal{F}$ is a family of binary operations on $\Omega$ (in other words, a family of functions from $\Omega^2$ to $\Omega$) and $\mathcal{M}$ is a family of unary transformations on $\Omega$ (in other words, a family of functions from $\Omega$ to itself) such that $\Lambda_\mathcal{M} = \{\emptyset,\ \Omega\}$.

It is easy to verify (see Proposition A1 of [8]) that the family $\Lambda_\Gamma$ is closed under arbitrary intersections and contains $\Omega$. It then follows that for every element $x \in \Omega$ there is a unique element of $\Lambda_\Gamma$ containing $x$ (namely the intersection of all the members of $\Lambda_\Gamma$ containing $x$.)

**Definition 3** Given a heuristic 3-tuple $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$, denote by $S_x^\Omega$ the smallest element of $\Lambda_\mathcal{F}$ containing $x$.

The following definition is a natural extension of the notion of a genetic representation function.

**Definition 4** Given two heuristic 3-tuples $\Omega_1 = (\Omega_1,\, \mathcal{F}_1,\, \mathcal{M}_1)$ and $\Omega_2 = (\Omega_2,\, \mathcal{F}_2,\, \mathcal{M}_2)$, a morphism[3] $\delta : \Omega_1 \to \Omega_2$ is just a function $\delta : \Omega_1 \to \Omega_2$ which respects the mating transformations in the following sense: $\forall\, T \in \mathcal{F}_1$ and $\forall\, \mathbf{x} = (x_1,\, x_2) \in \Omega^2\ \exists\, F_\mathbf{x} \in \mathcal{F}_2$ such that $\delta(T(x_1,\, x_2)) = F_{(x_1,\, x_2)}(\delta(x_1), \delta(x_2))$. Analogously, we must have $\forall\, M \in \mathcal{M}_1$ and $\forall\, x \in \Omega\ \exists\, H_x \in \mathcal{M}_2$ such that $\delta(M(x)) = H_x(\delta(x))$. We shall denote by $Mor(\Omega_1,\, \Omega_2)$ the collection of all morphisms from $\Omega_1$ into $\Omega_2$.

A morphism $\delta : \Omega_1 \to \Omega_2$ provides the means for encoding the heuristic 3-tuple $\Omega_1$ by the heuristic 3-tuple $\Omega_2$. Unless the underlying function $\delta$ is one to one, there is some nontrivial coarse graining involved. We, therefore have a special name for these morphisms whose underlying functions are injective.

**Definition 5** We say that a morphism $\delta : \Omega_1 \hookrightarrow \Omega_2$ is an embedding if the underlying function $\delta : \Omega_1 \to \Omega_2$ is one-to-one.

Already at this stage one can see the importance of the family of invariant subsets $\Lambda_\mathcal{F}$:

**Proposition 6** *Let $\delta : \Omega_1 \to \Omega_2$ be a morphism of heuristic 3-tuples. Then $S \in \Lambda_{\mathcal{F}_2} \Longrightarrow \delta^{-1}(S) \in \Lambda_{\mathcal{F}_1}$. In words, a preimage of an invariant set under a morphism is invariant.*

[3] Heuristic 3-tuples along with the morphisms between them do form a mathematical structure called a Category (see [6] for a detailed exposition). Some properties of the Category of heuristic $k$-tuples will be presented in the forthcoming paper.

*Proof.* Fix $S \in \Lambda_{\mathcal{F}_2}$. Let $(x_1, x_2) \in \delta^{-1}(S)$. Then $\forall T \in \mathcal{F}_1$ we have $\delta(T(x_1, x_2)) = F_{(x_1, x_2)}(\delta(x_1), \delta(x_2))$ for some $F_{(x_1, x_2)} \in \mathcal{F}_2$. But $S \in \mathcal{F}_2$ by assumption so that $\delta(T(x_1, x_2)) = F_{(x_1, x_2)}(\delta(x_1), \delta(x_2)) \in S \Longrightarrow T(x_1, x_2) \in \delta^{-1}(S)$. This shows that $\delta^{-1}(S)$ is, indeed, invariant under $\mathcal{F}_1$.

Although the converse of proposition 6 is not true in general, the mathematical apparatus developed in Appendix A of [8] allows us to establish a partial converse of 6. First we need the notion of a composition closed family which is studied in appendix A of [8]. For the sake of completeness we include the definition below:

**Definition 7** We say that a given family of $m$-ary operations $\Gamma$ on a set $\Omega$ (that is a family of functions from $\Omega^m$ to $\Omega$) is composition closed if the following two conditions hold:

1. $\forall\ T_0, T_1, T_2, \ldots, T_m \in \Gamma$ the operation $T : \Omega^m \to \Omega$ sending any given $\mathbf{x} = (x_1, x_2, \ldots, x_m) \in \Omega^m$ to $T(\mathbf{x}) = T_0(T_1(\mathbf{x}), T_2(\mathbf{x}), \ldots, T_m(\mathbf{x}))$ is also a member of $\Gamma$.
2. $S \subseteq \Omega$ we have $\bigcup_{T \in \Gamma} T(S^m) \supseteq S$.

**Remark 8** Notice that if a given family of mating transformations $\Gamma$ is pure in the sense of [9] (meaning that $\forall\ T \in \Gamma$ and $\forall\ x \in \Omega$ we have $T(x, x, \ldots, x) = x$. See also [11] and [8]) then condition 2 of definition 7 is satisfied automatically. Every one of the families of mating transformations for the algorithms introduced in section 2 is pure.

It is fairly straightforward to verify that every one of the families of mating transformations involved in the examples of section 2 is composition closed. In fact, it has been already shown in [8] that the families of masked crossover transformations and gene Holland transformations (these which are convenient for modelling random respectful recombination) are composition closed (see proposition 2.1 and Theorem 3.6 of [8]). It only remains to show that the family of semi-crossover transformations is composition closed:

**Proposition 9** *The family of binary semi-crossover transformations as defined in the description of the binary semi-genetic algorithm is composition closed.*

*Proof.* Fix arbitrary $\mathbf{u}, \mathbf{v}, \mathbf{w} \in \Omega = \{0, 1\}^n$. We want to show that the transformation $F : \Omega^2 \to \Omega$ sending any given pair $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \Omega^2$ to $T_{\mathbf{u}}(T_{\mathbf{v}}(\mathbf{z}), T_{\mathbf{w}}(\mathbf{z}))$ is of the form $T_{\mathbf{t}}$ for some $\mathbf{t} \in \Omega$. It is routine to verify using the definition that $\mathbf{t} = T_{\mathbf{u}}(\mathbf{v}, \mathbf{w})$ does the job.

The following fact justifies the importance of definition 7:

**Proposition 10** *Let $\Omega_1 = (\Omega_1, \mathcal{F}_1, \mathcal{M}_1)$ and $\Omega_2 = (\Omega_2, \mathcal{F}_2, \mathcal{M}_2)$ denote heuristic 3-tuples with $\mathcal{F}_2$ and $\mathcal{M}_2$ being composition closed. Now given any function $\delta : \Omega_1 \to \Omega_2$ such that $\forall\ S \in \Lambda_{\mathcal{F}_2}$ we have $\delta^{-1}(S) \in \Lambda_{\mathcal{F}_1}$, $\delta$ is a morphism of heuristic 3-tuples.*

*Proof.* Fix arbitrary $x_1, x_2 \in \Omega$ and a mating transformation $T \in \mathcal{F}_1$. Our goal is to find a transformation $F \in \mathcal{F}_2$ such that $F(\delta(x_1), \delta(x_2)) = \delta(T(x_1, x_2))$. Consider the smallest element of $\Lambda_{\mathcal{F}_2}$ containing both, $\delta(x_1)$ and $\delta(x_2)$, call it $K_{\{\delta(x_1), \delta(x_2)\}}$. ($K_{\{\delta(x_1), \delta(x_2)\}}$ is simply the intersection of all the members of $\Lambda_{\mathcal{F}_2}$ containing $\delta(x_1)$ and $\delta(x_2)$. Since $\Lambda_{\mathcal{F}_2}$ is closed under arbitrary intersections, $K_{\{\delta(x_1), \delta(x_2)\}} \in \Lambda_{\mathcal{F}_2}$.) Since $K_{\{\delta(x_1), \delta(x_2)\}} \in \Lambda_{\mathcal{F}_2}$, by assumption $\delta^{-1}(K_{\{\delta(x_1), \delta(x_2)\}}) \in \Lambda_{\mathcal{F}_1}$. But then $\delta(T(x_1, x_2)) \in K_{\{\delta(x_1), \delta(x_2)\}}$. Since $\mathcal{F}_2$ is composition closed, by Lemma A.8 of [8], $\exists \in \mathcal{F}_2$ such that $F(\delta(x_1), \delta(x_2)) = \delta(T(x_1, x_2))$ which is exactly what we were after. Notice that condition $\delta(M(x)) = H_x(\delta(x))$ for some $H_x \in \mathcal{M}_2$ is fulfilled automatically since by definition 2 $\Lambda_{\mathcal{M}} = \{\emptyset, \Omega_2\}$ and, by assumption $\mathcal{M}_2$ is composition closed so, by Lemma A.8 of [8], $\forall\, y \in \Omega_2\ \exists H_y \in \mathcal{M}_2$ such that $H_y(\delta(x)) = y$.

As noted before, for any family of $m$-ary operations on $\Omega$ the corresponding family of invariant subsets $\Lambda_\Gamma$ is closed under arbitrary intersections. Moreover, for any function $\delta : \Omega_1 \to \Omega_2$ the inverse image of the intersection of two subsets of $\Omega_2$ is the intersection of the inverse images of these subsets: $\delta^{-1}(U \cap V) = \delta^{-1}(U) \cap \delta^{-1}(V)$. This motivates the following definition:

**Definition 11** Given a family of $m$-ary operations $\Gamma$ on $\Omega$, we say that a family of subsets $\widetilde{\Lambda_\Gamma} \subseteq \Lambda_\Gamma$ is a base of $\Lambda_\Gamma$ if for every $K \in \Lambda_\Gamma$ there exists a collection $\Lambda_K \subseteq \widetilde{\Lambda_\Gamma}$ such that $K = \bigcap_{S \in \Lambda_K} S$. (Equivalently, if $K = \bigcap_{S \in \widetilde{\Lambda_\Gamma},\, S \supset K} S$).

**Corollary 12** *Let $\Omega_1 = (\Omega_1, \mathcal{F}_1, \mathcal{M}_1)$ and $\Omega_2 = (\Omega_2, \mathcal{F}_2, \mathcal{M}_2)$ denote heuristic 3-tuples with $\mathcal{F}_2$ and $\mathcal{M}_2$ being composition closed, and $\delta : \Omega_1 \to \Omega_2$ be a function. Then the following are equivalent:*

*1. $S \in \widetilde{\Lambda_{\mathcal{F}_2}} \implies \delta^{-1}(S) \in \Lambda_{\mathcal{F}_1}$.*
*2. $S \in \Lambda_{\mathcal{F}_2} \implies \delta^{-1}(S) \in \Lambda_{\mathcal{F}_1}$.*
*3. $\delta : \Omega_1 \to \Omega_2$ is a morphism of heuristic 3-tuples.*

*Proof.* An immediate consequence of propositions 6 and 10 together with the discussion preceding definition 11.

Below we list the families of invariant subsets together with a naturally chosen bases for each of the examples presented in section 2.

**Classical Genetic Algorithm.** In this case, the family of invariant subsets $\Lambda_{\mathcal{F}} = \{\prod_{i=1}^n T_i \mid T_i \subseteq A_i\}$. This is precisely the family of subsets determined by Antonisse's schemata (see corollary 2.4 of [8]). A bases for $\Lambda_{\mathcal{F}}$ is the family $\widetilde{\Lambda_{\mathcal{F}}} = \{\prod_{i=1}^n T_i \mid T_i = A_i$ for all but one $i\}$. The reader can see that $|\widetilde{\Lambda_{\mathcal{F}}}| = 2^{\Sigma_{i=1}^n |A_i|}$. Every element of $\widetilde{\Lambda_{\mathcal{F}}}$ can be thought of as a union of subsets determined by the Holland schemata having exactly one fixed position at the same gene.

**Random Respectful Recombination.** $\Lambda_{\mathcal{F}} = \{\prod_{i=1}^n T_i \mid T_i = \{a_i\}$ or $T_i = A_i\} \cup \{\emptyset\}$. This is precisely the family of subsets determined by the Holland schemata together with the empty set (see corollary 3.5 of [8]). A bases for $\Lambda_{\mathcal{F}}$ is the family $\widetilde{\Lambda_{\mathcal{F}}} = \{\prod_{i=1}^n T_i \mid \exists!$ with $T_j = \{a_j\}$. For $i \neq j$ $T_i = A_i\}$. This is precisely the family of subsets determined by Holland schemata having exactly one fixed position.

**Binary Semi-genetic Algorithm.** It is not hard to verify that $\Lambda_{\mathcal{F}} = \{\prod_{i=1}^{n} T_i \mid T_i = \{1\} \text{ or } T_i = \{0, 1\}\} \cup \{\emptyset\}$. This is precisely the family of subsets determined by Holland schemata whose fixed positions can only equal to 1 (can't be equal to 0). A bases for $\Lambda_{\mathcal{F}}$ is the family $\widetilde{\Lambda_{\mathcal{F}}} = \{\prod_{i=1}^{n} T_i \mid \exists! \text{ with } T_j = \{1\}$. For $i \neq j$ $T_i = \{0, 1\}\}$ which is precisely the family of subsets determined by Holland schemata having exactly one fixed position, and that fixed position is equal to 1.

Corollary 12 allows us to characterize all possible morphisms from various heuristic 3-tuples to the standard types described in section 2. Our first result, the Binary Embedding Theorem, establishes an explicit one-to-one correspondence between the set of all embeddings of a given heuristic 3-tuple into a binary semi-genetic algorithm of length $n$ and a certain collection of ordered $n$-tuples of $\Omega$-invariant subsets.

**Definition 13** Fix any heuristic 3-tuple $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$. We say that collection

$$\Upsilon_n = \{\mathcal{I} \mid \mathcal{I} = (I_1, I_2, \ldots, I_n)\ I_j \in \Lambda_\Omega,\ \forall\, x, y \in \Omega \text{ with } x \neq y\ \exists\, 1 \leq j \leq n$$
$$\text{such that either } (x \in I_j \text{ and } y \notin I_j) \text{ or vise versa: } (y \in I_j \text{ and } x \notin I_j)\}$$

is a family of separating $n$-tuples. Notice that $\Upsilon_n \subseteq (\Lambda_{\mathcal{F}})^n$.

**Theorem 14** *Fix a heuristic 3-tuple $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$. We now have the following bijection $\phi : \Lambda_{\mathcal{F}} \to Mor(\Omega, \mathcal{S}_n)$ (here $\mathcal{S}_n$ denotes the binary semi-genetic heuristic 3-tuple with the search space $\{0, 1\}^n$, see also definition 4 for the meaning of $Mor(\Omega, \mathcal{S}_n)$) which is defined explicitly as follows: Given an ordered $n$-tuple of sets from $\Lambda_\Omega$, call it $\mathcal{I} = (I_1, I_2, \ldots, I_n) \in (\Lambda_{\mathcal{F}})^n$ let $\phi(\mathcal{I}) = \delta_{\mathcal{I}}$ where $\delta_{\mathcal{I}}(x) = (x_1, x_2, \ldots, x_n) \in S = \{0,1\}^n$ with $x_j = \begin{cases} 1 & \text{if } x \in I_j \\ 0 & \text{otherwise} \end{cases}$ $\forall x \in \Omega$. Moreover, $\delta_{\mathcal{I}}$ is an embedding (injective) if and only if $\mathcal{I} \in \Upsilon_n$ (see definition 13). In other words, the restriction of $\phi$ to $\Upsilon_n$ is a bijection onto the collection of all embeddings of $\Omega$ into $\mathcal{S}_n$.*

*Proof.* Given any map $\delta : \Omega \to \{0, 1\}$, for $1 \leq j \leq n$ let $I_j = \delta^{-1}(\prod_{i=1}^{n} T_i)$ with $T_i = \{0, 1\}$ if $i \neq j$ and $T_j = \{1\}$. Recall from examples following definition 11 that $\{\prod_{i=1}^{n} T_i \mid \exists! \text{ with } T_j = \{1\}$. For $i \neq j$ $T_i = \{0, 1\}\}$ forms a bases for the family of subsets invariant under semi-crossover transformations. Therefore, according to corollary 12, $\delta : \Omega \to \mathcal{S}_n$ is a morphism of heuristic 3-tuples if and only if $(I_1, I_2, \ldots, I_n) \in (\Lambda_{\mathcal{F}})^n$. This shows that $\phi : \Lambda_{\mathcal{F}} \to Mor(\Omega, \mathcal{S}_n)$ is a well-defined bijection. It is routine to check that $\delta_{\mathcal{I}}$ is injective if and only if $\mathcal{I} \in \Upsilon_n$ (see definition 13).

It turns out that the conditions under which a given heuristic 3-tuple can be embedded into a binary semi-genetic heuristic 3-tuple are rather mild and naturally occurring as the following two corollaries demonstrate:

**Corollary 15** *Given a heuristic* 3*-tuple* $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$*, the following are equivalent:*

1. *$\Omega$ can be embedded into an $n$-dimensional semi-genetic heuristic $k$-tuple for some $n$.*
2. *$\forall\, x, y \in \Omega$ with $x \neq y$ we have either $x \notin S_y^\Omega$ (see definition 3) or vise versa: $y \notin S_x^\Omega$.*
3. *$\forall\, x, y \in \Omega$ with $x \neq y$ we have $S_x^\Omega \neq S_y^\Omega$. (Another way to say this, is that the map sending $x$ to $S_x^\Omega$ is one-to-one.)*

*Moreover, if an embedding exists for some $n$, then there exists one for $n = |\Omega|$. We also must have $n \geq \lceil \log_2 |\Omega| \rceil$.*

*Proof.* One simply shows that $\forall\, x, y \in \Omega$ with $x \neq y$ we have either $x \notin S_y^\Omega$ or $y \notin S_x^\Omega$ if and only if $|\Omega|$-tuple $\mathcal{S} = (S_{x_1}^\Omega, S_{x_2}^\Omega, \ldots, S_{x_{|\Omega|}}^\Omega)$ where $\{x_i\}_{i=1}^n$ is an enumeration of all the elements of $\Omega$ is separating ( i. e. $\mathcal{S} \in \Upsilon_n$, see definition 13 ) if and only if $\Upsilon_n \neq \emptyset$ which, in turn, according to theorem 14, happens if and only if $\Omega$ can be embedded into an $n$-dimensional semi-genetic heuristic $k$-tuple for some $n$. This establishes the equivalence of 1 and 2. Clearly 2 implies 3. To see the converse, we show that "Not 2" implies "Not 3". Indeed, if $x \in S_y^\Omega$ and $y \notin S_x^\Omega$, then, by minimality, (see definition 2) we have $S_x^\Omega \subseteq S_y^\Omega$ and $S_y^\Omega \subseteq S_x^\Omega$, so that $S_x^\Omega = S_y^\Omega$.

**Corollary 16** *Given a heuristic* 3*-tuple* $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$*, if for every $T \in \mathcal{F}$, $T$ is pure in the sense of [9] ( in other words, $\forall\, x \in \Omega\ T(x, x) = x$ ) then $\Omega$ can be embedded into a binary semi-genetic heuristic $k$-tuple of dimension less than or equal to $|\Omega|$.*

*Proof.* The desired conclusion follows immediately from corollary 15 by observing that $\forall\, x, y \in \Omega$ with $x \neq y$ we have $S_x^\Omega = \{x\}$ so that $x \in \{x\} = S_x^\Omega$ while $y \notin \{x\} = S_x^\Omega$.

Notice that purity by itself is sufficient for the existence of an embedding of a given heuristic 3-tuple into a binary semi-genetic heuristic 3-tuple. Of course, the embedding may not be surjective by any means. The main virtue of theorem 14 is not so much the results such as corollary 16, but rather the explicit bijective correspondence between $Mor(\Omega, \mathcal{S}_n)$ and the collection $(\Lambda_{\mathcal{F}})^n$. The main tool involved in the proof of theorem 14 is corollary 12. In the next section we demonstrate how corollary12 can be applied to establish a similar bijective correspondence between certain kinds of sequences of Radcliffe's forma and $Mor(\Omega, \mathcal{G}_{\{A_i\}_{i=1}^n})$ where $\mathcal{G}_{\{A_i\}_{i=1}^n}$ denotes the heuristic 3-tuple corresponding to the genetic algorithm with $\prod_{i=1}^n A_i$ as its underlying search space.

## 4 Characterizing the Morphisms from a Given Heuristic 3-Tuple into a Genetic Heuristic 3-Tuple in Terms of Radcliffe's Forma

For the reader's convenience we restate a few basic notions considered in [9]:

**Definition 17** Given a set $\Omega$, denote by $\mathbb{E}(\Omega)$ the set of all possible partitions of $\Omega$ into disjoint nonempty subsets. (Partitions and equivalence relations are in a natural bijective correspondence. See, for instance [9], or any standard textbook on basic mathematical structures and concepts for details) Given an element $\Xi \in \mathbb{E}(\Omega)$, the elements of $\Xi$ are called forma. Given an $n$-tuple $\Psi = (\Xi_1, \Xi_2, \ldots, \Xi_n)$ of elements of $\mathbb{E}(\Omega)$, let $\Xi(\Psi) = \prod_{i=1}^{n} \Xi_i$. A genetic representation function $\rho : \Omega \to \Xi(\Psi)$ sends a given $x \in \Omega$ to $(X_1, X_2, \ldots, X_n) \in \Xi(\Psi)$ where $x \in X_i$ (remember that such an $X_i \in \Xi_i$ exists and is unique since $\Xi_i$ is a partition of $\Omega$ so that $\rho$ is, indeed, well-defined).

The following definition sets the stage for the main theorem of this section:

**Definition 18** Given a heuristic 3-tuple $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$, and a function $\delta : \Omega \to \prod_{i=1}^{n} A_i$, let $\Psi_\delta = (X_1^\delta, X_2^\delta, \ldots, X_n^\delta)$ where $X_j$ is the collection of all nonempty preimages under $\delta$ of the subsets of $\prod_{i=1}^{n} A_i$ which are determined by the classical Holland schemata having exactly one fixed position in the $j^{\text{th}}$ gene. Explicitly, $X_j^\delta = \{\delta^{-1}(\prod_{i=1}^{n} T_i) \mid T_j = \{a_j\}$ for some $a_j \in A_j$ and $T_i = A_i$ for $i \neq j\} - \{\emptyset\}$.

**Definition 19** To shorten the notation we shall denote by $\mathcal{G}_{\{A_i\}_{i=1}^n}$ the heuristic 3-tuple representing the classical genetic algorithm, and by $\mathcal{P}_{\{A_i\}_{i=1}^n}$ the the heuristic 3-tuple representing the random respectful recombination algorithm with the underlying search space $\prod_{i=1}^{n} A_i$.

The following theorem is an immediate consequence of corollary 12:

**Theorem 20** *Given a heuristic 3-tuple $\Omega = (\Omega, \mathcal{F}, \mathcal{M})$, and a function $\delta : \Omega \to \prod_{i=1}^{n} A_i$, the following are true:*

*1. $\delta : \Omega \to \mathcal{P}_{\{A_i\}_{i=1}^n}$ is a morphism of heuristic 3-tuples if and only if $\forall\, 1 \leq j \leq n$ every forma in $X_j^\delta$ (see definition 18) is invariant under $\mathcal{F}$ (if and only if every forma in $X_j^\delta$ is a member of $\Lambda_{\mathcal{F}}$).*

*2. $\delta : \Omega \to \mathcal{G}_{\{A_i\}_{i=1}^n}$ is a morphism of heuristic 3-tuples if and only if $\forall\, 1 \leq j \leq n$ every union of forma in $X_j^\delta$ (see definition 18) is invariant under $\mathcal{F}$ (if and only if for every subset of forma $Y \subseteq X_j^\delta$ we have $(\bigcup_{S \in Y} S) \in \Lambda_{\mathcal{F}}$).*

*Proof.* In case of a random respectful recombination all forma in $X_j^\delta$, and, in case of a classical genetic algorithm, all unions of forma in $X_j^\delta$ are precisely the preimages under $\delta$ of the sets in $\widetilde{\Lambda_\Gamma}$ where $\Gamma$ is the family of Holland transformations in case of random respectful recombination, and the family of masked crossover transformations in case of a classical genetic algorithm (see examples following corollary 12). The desired conclusion now follows at once from corollary 12.

The difference between theorem 20 and results like theorem 25 of [9] is that theorem 20 classifies all possible re-encodings of a given evolutionary search algorithm in terms of a given genetic algorithm (or in terms of a "random respectful recombination") while Radcliffe's results provide a foundation for designing a genetic algorithm to model a specific problem in question.

## 5 Conclusions

In the current paper the following contributions have been made:

1. An appropriate notion for comparing evolutionary computation techniques via their representation (a morphism between heuristic 3-tuples) has been introduced. (See definition 4)

2. An important connection between the family of invariant subsets of the search space (see definition 1) and the morphisms of heuristic 3-tuples has been established (see corollary 12).

3. A binary semi-genetic algorithm has been introduced and it was shown that virtually any evolutionary heuristic search algorithm can be embedded into a binary semi-genetic algorithm (see theorem 14 and corollaries 15 and 16).

4. All possible morphisms (re-encodings and coarse graining) of a particular evolutionary heuristic search tuple by a classical genetic algorithm, or by a random respectful recombination have been characterized in terms of Radcliffe's forma (see theorem 20).

**Acknowledgements.** I want to thank Professor John Holland for the helpful discussions and for the encouragement I've received from him to write this paper. I also want to thank my thesis advisor, Professor Andreas Blass for the numerous helpful advisor meetings which have stimulated many ideas for this and for my future work. In addition I would like to thank my fellow graduate student of mathematics, Ronald Walker and the University of Michigan Complex Systems Group for helpful discussions. Finally, I want to thank Jon Rowe and the other anonymous referees for the helpful suggestions and remarks they have made.

## References

1. Antonisse, J. (1989). A new interpretation of schema notation that overturns the binary encoding constraint. In Ed. J. D. Schaffer *Procedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann, pages 86–97.
2. Booker, L. (1993). Recombination distributions for genetic algorithms. In L. Darrell Whitley, editor, *Foundations of Genetic Algorithms 2*, pages 29–44, Morgan Kaufmann.
3. Geiringer, H. (1944). On the probability of linkage in Mendelian heredity. *Annals of Mathematical Statistics*, 15:25–57.
4. Coffey, S. (1999) An Applied Probabilist's Guide to Genetic Algorithms. *A Thesis Submitted to The University of Dublin for the degree of Master in Science.*
5. Liepins, G. and Vose, M. (1992). Characterizing Crossover in Genetic Algorithms. *Annals of Mathematics and Artificial Intelligence*, 5: 27–34.
6. Mac Lane, S. (1971) Categories for the working mathematician. *Graduate Texts in Mathematics 5*, Springer-Verlag.
7. Mühleinbeim, H. and Mahnig, T. (2001). Evolutionary computation and beyond. In Y. Uesaka, P. Kanerva, and H. Asoh, editors, *Foundations of Real-World Intelligence*, CSLI Publications, pp. 123–188.

8. Mitavskiy B. (Resubmitted). Crossover Invariant Subsets of the Search Space for Evolutionary Algorithms. *Evolutionary Computation.* http://www.math.lsa.umich.edu/~bmitavsk/
9. Radcliffe, N. (1992). The algebra of genetic algorithms. *Annals of Mathematics and Artificial Intelligence*, 10:339–384.
10. Rothlauf F., Goldberg D., Heinzl A. (2002). Network random keys - a tree representation scheme for genetic and evolutionary algorithms. *Evolutionary Computation*, 10(1): 75–97.
11. Rowe, J., Vose, M., and Wright, A. (2002). Group properties of crossover and mutation. *Evolutionary Computation*, 10(2): 151–184.
12. Stephens, C., Poli, R., Wright, A., and Rowe, J. (2002). Exact results from a coarse grained formulation of the dynamics of variable-length genetic algorithms. In W. B. Langdon, E. Cantu-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M. A. Potter, A. C. Schultz, J. F. Miller, E. Burke, and N. Jonoska, editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference*, pages 578–585, Morgan Kaufmann Publishers.
13. Stephens, C. (2001). Some exact results from a coarse grained formulation of genetic dynamics. *Proceedings of the Genetic and Evolutionary Computation (GECCO) conference*, pages 631–638, Morgan Kaufmann.
14. Stephens, C. and Waelbroeck, H. (1999). Schemata evolution and building blocks. *Evolutionary Computation*, 7(2):109–124.
15. C. Stephens. (2002). The Renormalization Group and the Dynamics of Genetic systems, to be published in *Acta Physica Slovaka.* http://arXiv.org/abs/cond-mat/0210271/.
16. Syswerda, G. (1993). Simulated crossover in genetic algorithms. In L. Darrel Whitley, editor, *Foundations of Genetic Algorithms 2*, Morgan Kaufmann.
17. Vose, M. (1991). Generalizing the notion of a schema in genetic algorithms. *Artificial Intelligence*, 50(3): 385–396.
18. Vose, M. (1999). The simple genetic algorithm: foundations and theory. *MIT Press, Cambridge, Massachusetts.*
19. Wright, A., Rowe, J., Poli, R., and Stephens C. (2002). A fixed point analysis of a gene pool GA with mutation. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)* Morgan Kaufmann. http://www.cs.umt.edu/u/wright/.