Hyperschema Theory for GP with One-Point Crossover, Building Blocks, and Some New Results in GA Theory

Riccardo Poli
(R.Poli@cs.bham.ac.uk)

School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK

Abstract. Two main weaknesses of GA and GP schema theorems are that they provide only information on the expected value of the number of instances of a given schema at the next generation $E[m(H, t+1)]$, and they can only give a lower bound for such a quantity. This paper presents new theoretical results on GP and GA schemata which largely overcome these weaknesses. Firstly, unlike previous results which concentrated on schema survival and disruption, our results extend to GP recent work on GA theory by Stephens and Waelbroeck, and make the effects and the mechanisms of schema creation explicit. This allows us to give an exact formulation (rather than a lower bound) for the expected number of instances of a schema at the next generation. Thanks to this formulation we are then able to provide in improved version for an earlier GP schema theorem in which some schema creation events are accounted for, thus obtaining a tighter bound for $E[m(H, t+1)]$. This bound is a function of the selection probabilities of the schema itself and of a set of lower-order schemata which one-point crossover uses to build instances of the schema. This result supports the existence of building blocks in GP which, however, are not necessarily all short, low-order or highly fit. Building on earlier work, we show how Stephens and Waelbroeck's GA results and the new GP results described in the paper can be used to evaluate schema variance, signal-to-noise ratio and, in general, the probability distribution of $m(H, t+1)$. In addition, we show how the expectation operator can be removed from the schema theorem so as to predict with a known probability whether $m(H, t+1)$ (rather than $E[m(H, t+1)]$) is going to be above a given threshold.

1 Introduction

Since John Holland's seminal work in the mid seventies and his well known schema theorem [1, 2], schemata are traditionally used to explain why GAs and more recently GP [3, 4, 5] work. Schemata are similarity templates representing entire groups of points in the search space. The schema theorem describes how schemata are expected to propagate generation after generation under the effects of selection, crossover and mutation.

The usefulness of the schema theorem has been widely criticised (see for example [6, 7, 8, 9]), and many people in the evolutionary computation field nowadays seem to believe that the theorem is nothing more than a trivial tautology of no use whatsoever. So, recently, the attention of GA theorists has moved away from schemata to land onto Markov chains [10, 11, 12, 13, 14, 15]. However, many of the problems attributed to the schema theorem are probably not due to the theorem itself, rather to its over-interpretations [16].

The main criticism for schema theorems is that they cannot be used easily for predicting the behaviour of a GA over multiple generations. One reason for this is that schema theorems give only a *lower bound* for the *expected value* of the number of instances of a schema H at the next generation $E[m(H, t+1)]$ as a function of quantities characterising the schema at the previous generation. The presence of an expectation operator means that it is not possible to use schema theorems recursively to predict the behaviour of a genetic algorithm over multiple generations unless one assumes the population is infinite. In addition, since the schema theorem provides only a lower bound, some people argue that the predictions of the schema theorem are not very useful even for a single generation ahead.

Clearly there is some truth in these criticisms. However, this does not mean that schema theorems are useless. As shown by our and other researchers' recent work [17, 18, 19, 20, 21] schema theorems have not been fully exploited nor fully developed, and when this is done they become very useful.

This paper presents new theoretical results on GP and GA schemata which largely overcome some of the weaknesses of the schema theorem. Firstly, unlike previous results which concentrated on schema survival and disruption, our results extend to GP recent work on GA theory by Stephens and Waelbroeck, and make the effects and the mechanisms of schema creation explicit. This allows us to give an exact formulation (rather than a lower bound) for the expected number of instances of a schema at the next generation. Thanks to this formulation we are then able to provide in improved version for an earlier GP schema theorem in which some schema creation events are accounted for, thus obtaining a tighter bound for $E[m(H, t+1)]$. This bound is a function of the selection probabilities of the schema itself and of a set of lower-order schemata which one-point crossover uses to build instances of the schema. This result supports the existence of building blocks in GP which, however, are not necessarily all short, low-order or highly fit. Building on earlier work, we show how Stephens and Waelbroeck's GA results and the new GP results described in the paper can be used to evaluate schema variance, signal-to-noise ratio and, in general, the probability distribution of $m(H, t+1)$. In addition, we show how the expectation operator can be removed from the schema theorem so as to predict with a known probability whether $m(H, t+1)$ (rather than $E[m(H, t+1)]$) is going to be above a given threshold.

The structure of the paper is the following. Since the work presented in this paper extends our and other people's work, before we introduce it we will extensively review earlier relevant work on GP and GA schemata in Section 2. Then, in Section 3 we connect the recent GA results by Stephens and Waelbroeck to previous theoretical work on schemata, indicating how this leads to a series of new theoretical results for GAs. Section 4 presents perhaps the main contribution of this paper: the hyperschema theory for GP, its uses and interpretations. We draw some conclusions in Section 5.

2 Background

2.1 Holland's GA Schema Theory

In the context of GAs operating on binary strings, a schema (or similarity template) is a string of symbols taken from the alphabet {0,1,#}. The character # is interpreted as a "don't care" symbol, so that a schema can represent several bit strings. For example the schema #10#1 represents four strings: 01001, 01011, 11001 and 11011. The number of non-# symbols is called the *order* $\mathcal{O}(H)$ of a schema H. The distance between the furthest two non-# symbols is called the *defining length* $\mathcal{L}(H)$ of the schema. Holland obtained a result (the schema theorem) which predicts how the number of strings in a population matching (or belonging to) a schema is expected to vary from one generation to the next [1]. The theorem can be reformulated as follows:

$$E[m(H,t+1)] \geq Mp(H,t) \cdot (1-p_m)^{\mathcal{O}(H)} \cdot \left[1 - p_{xo}\frac{\mathcal{L}(H)}{N-1}(1-p(H,t))\right] \quad (1)$$

where p_m is the probability of mutation per bit, p_{xo} is the probability of crossover, N is the number of bits in the strings, M is the number of strings in the population, $E[m(H,t+1)]$ is the expected number of strings matching the schema H at generation $t+1$, and $p(H,t)$ is the probability of selection of the schema H. In fitness proportionate selection this is given by $p(H,t) = \frac{m(H,t)f(H,t)}{M\bar{f}(t)}$ where $m(H,t)$ is the number of strings matching the schema H at generation t, $f(H,t)$ is the mean fitness of the strings matching H, and $\bar{f}(t)$ is the mean fitness of the strings in the population.[1]

2.2 GP Schema Theories

One of the difficulties in obtaining theoretical results on GP using the idea of schema is that the definition of schema is much less straightforward than for GAs and a few alternative definitions have been proposed in the literature. All of them define schemata as composed of one or multiple trees or fragments of trees. In some definitions [23, 24, 25, 26, 27] schema components are *non-rooted* and, therefore, a schema can be present multiple times within the same program. This, together with the variability of the size and shape of the programs matching the same schema, leads to considerable complications in the calculations necessary to formulate schema theorems for GP. In more recent definitions [3, 5] schemata are represented by *rooted* trees or tree fragments. These definitions make schema theorem calculations easier. We describe these schema definitions below.

[1] This is a slightly different version of Holland's original theorem which applies when crossover is performed taking both parents from the mating pool [2, 22].

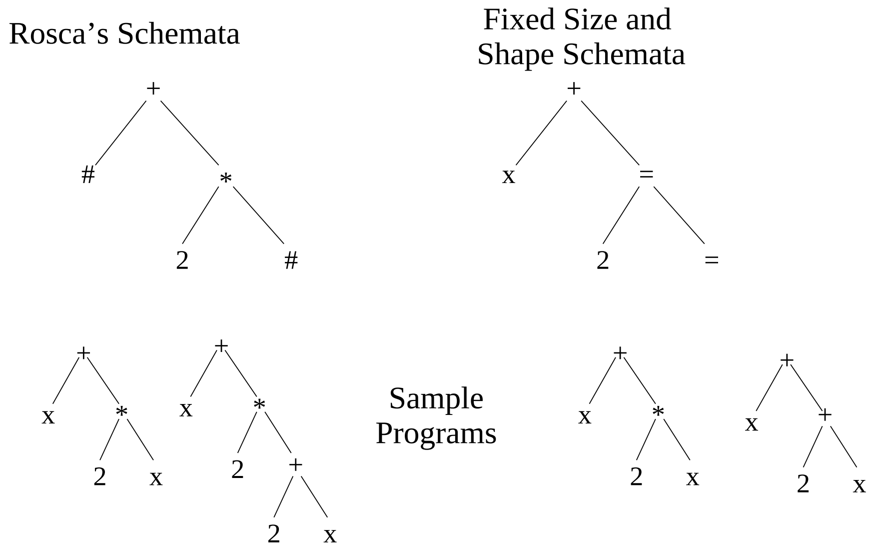

Fig. 1. Examples of rooted schemata (top) and some instances of programs sampling them (bottom).

Rosca's GP Schema Theory for Standard Crossover Rosca [5] has proposed a definition of schema, called *rooted tree-schema*, in which a schema is a rooted contiguous tree fragment. For example, the rooted tree-schema `(+ # x)` represents all the programs whose root node is a `+` the second argument of which is `x`. Another example of schema with some of its instances is shown in Figure 1 (left). With this definition, schemata divide the space of programs into subspaces containing programs of different sizes and shapes.

Rosca derived a schema theorem for GP with standard crossover which provided a lower bound for the expected number of instances of a schema at the next generation as a function of the schema order, and the fitness and size of its instances in the population.

GP Fixed-Size-and-Shape Schema Theory for One-point Crossover In [3] a simpler definition of schema for GP was proposed in which a *schema* is a tree composed of functions from the set $\mathcal{F} \cup \{=\}$ and terminals from the set $\mathcal{T} \cup \{=\}$, where $\mathcal{F}$ and $\mathcal{T}$ are the function set and the terminal set used in a GP run. The symbol $=$ is a "don't care" symbol which stands for a *single* terminal or function. In line with the original definition of schema for GAs, a schema H represents programs having the same shape as H and the same labels for the non-$=$ nodes. For example, if $\mathcal{F}$=`{+, -}` and $\mathcal{T}$=`{x, y}` the schema `(+ (- = y) =)` would represent the four programs `(+ (- x y) x)`, `(+ (- x y) y)`, `(+ (- y y) x)` and `(+ (- y y) y)`. Another example of schema with some of its instances is shown in Figure 1 (right). This definition of schema partitions the program space into subspaces of programs of fixed size and shape. For this reason in the following we will refer to these schemata as *fixed-size-and-shape schemata*. The definition is lower-level than those described above, as a smaller number of trees can be represented by schemata with the same number of "don't care" symbols and it is possible to represent other types of schemata by using a collection of these. This is quite important because it makes it possible to export some results of the fixed-size-and-shape schema theory to other kinds of schemata.

The number of non-$=$ symbols is called the *order* $\mathcal{O}(H)$ of a schema H, while the total number of nodes in the schema is called the *length* $N(H)$ of the schema. The number of links in the minimum tree fragment including all the non-$=$ symbols within a schema H is called the *defining length* $\mathcal{L}(H)$ of the schema (see [3, 4] for more details). For example the schema `(+ (- = =) x)` has order 3 and defining length 2. These definitions are independent of the shape and size of the programs in the actual population.

In order to derive a GP schema theorem for these schemata non-standard forms of mutation and crossover, namely point mutation and one-point crossover, were used. *Point mutation* is the substitution of a node in the tree with another node with the same arity. *One-point crossover* works by selecting a common crossover point in the parent programs and then swapping the corresponding subtrees, like standard crossover, as illustrated in Figure 2(a). In order to account for the possible structural diversity of the two parents, one-point crossover analyses the two trees from the root nodes and considers for the selection of the crossover point only the parts of the two trees (common region) which have the same topology (i.e. the same arity in the nodes encountered traversing the trees from the root node) [3, 4] as illustrated in Figure 2(b).

The resulting schema theorem is the following:

$$E[m(H,t+1)] \geq Mp(H,t)(1-p_m)^{\mathcal{O}(H)} \cdot \tag{2}$$
$$\left\{1 - p_{xo}\left[p_{\text{diff}}(t)\,(1-p(G(H),t)) + \frac{\mathcal{L}(H)}{(N(H)-1)}(p(G(H),t)-p(H,t))\right]\right\}$$

where p_m is the mutation probability (per node), $G(H)$ is the zero-th order schema with the same structure of H where all the defining nodes in H have been replaced with "don't care" symbols, M is the number of individuals in the population, $p_{\text{diff}}(t)$ is the conditional probability that H is disrupted by crossover when the second parent has a different shape (i.e. does not sample $G(H)$), and the other symbols have the same meaning as in Equation 1 (see [3, 4] for the proof). The zero-order schemata $G(H)$'s represent different groups of programs all with the same shape and size.

The probability $p_{\text{diff}}(t)$ is hard to model mathematically, but its expected variations during a run were analysed in detail in [3].

2.3 General Schema and Schema Variance Theorems in the Presence of Schema Creation

After the work on GP schemata in [3, 28, 4] we started asking ourselves whether the estimate (lower bound) for the expected value of the number of individuals

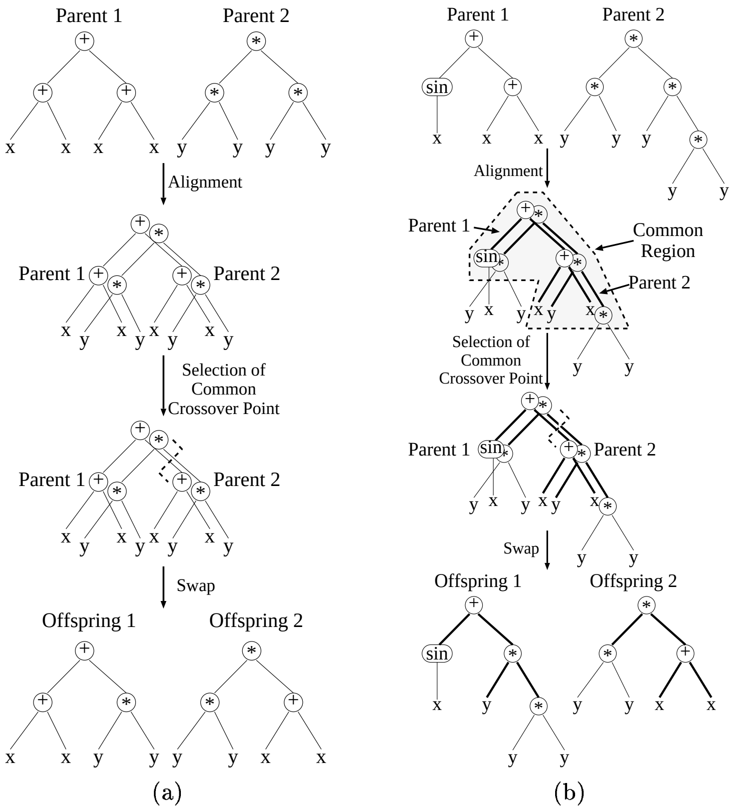

Fig. 2. (a) One-point crossover for GP when both parents have the same shape, and (b) general form of one-point crossover for GP. In (b) the links that can be selected as common crossover points are drawn with thick lines.

sampling a given schema at the next generation provided by the schema theorem is reliable. In order to investigate this issue in [29] we analysed the impact of variance on schema transmission and we obtained a schema-variance theorem which is general and applicable to GAs as well as GP. The analysis revealed the relative dependencies between schema transmission, population size, schema measured fitness, schema fragility and schema creation. In the rest of this subsection the main mathematical results of that work are summarised.

Crossover will cause some of the individuals matching H in the mating pool to produce offspring not sampling H and vice versa. Let us call $p_s(H,t)$ the probability that individuals in H will survive crossover (in the sense that their offspring will still sample H) and $p_c(H,t)$ the probability that offspring which sample H are *created* by parents not sampling H. Then the total schema transmission probability for the schema H is[2]

$$\alpha(H,t) = p_s(H,t)p(H,t) + p_c(H,t)(1 - p(H,t)).$$

Since we can see the selection/crossover process as a Bernoulli trial (each time an individual is produced, either the individual samples H or it does not), $m(H,t+1)$ can be seen as a binomial stochastic variable. Therefore,

$$\Pr\{m(H,t+1) = k\} = \binom{M}{k} \alpha(H,t)^k (1 - \alpha(H,t))^{M-k}, \tag{3}$$

whereby

$$E[m(H,t+1)] = M\alpha(H,t), \tag{4}$$

$$Var[m(H,t+1)] = M\alpha(H,t)(1 - \alpha(H,t)), \tag{5}$$

and

$$\left(\frac{S}{N}\right) \stackrel{def}{=} \frac{E[m(H,t+1)]}{\sqrt{Var[m(H,t+1)]}} = \sqrt{M}\sqrt{\frac{\alpha(H,t)}{1 - \alpha(H,t)}}, \tag{6}$$

which represents the signal-to-noise ratio of the schema. When the signal-to-noise ratio is large, the propagation of a schema will occur nearly exactly as predicted. When the signal-to-noise ratio is small, the actual number of instances of a schema in the next generation may be essentially random with respect to its current value.

From Equation 3 it is also possible to compute the probability of schema extinction in one generation:

$$\Pr\{m(H,t+1) = 0\} = (1 - \alpha(H,t))^M \tag{7}$$

It should be noted that although the results reported above include the probability of schema creation $p_c(H,t)$, such a quantity was not modelled mathematically in more detail. It is also worth noting that these results are valid independently of the representation adopted for the individuals in the population, the operators used, and the definition of schema. Therefore, they are valid for GAs as well as for GP.

2.4 General Schema Theorems without Expected Values

Building on the work described in the previous section in [20] two new schema theorems were introduced in which expectations are not present. The theorems were obtained using Chebychev inequality in conjunction with Equations 4 and 5. These theorems are a first step towards removing one of the most criticised components of the schema theorem: the expectation operator. One of the theorems states that for any given constant $k > 0$

$$\Pr\{m(H,t+1) > M\alpha(H,t) - k\sqrt{M\alpha(H,t)(1 - \alpha(H,t))}\} \geq 1 - \frac{1}{k^2}. \tag{8}$$

[2] The total schema transmission probability α at time t corresponds to the expected proportion of population sampling H at generation $t+1$. Thus, Equation 4 can be seen as a reformulation of the "gain and losses" equation in [30].

This result is necessarily weaker than the obvious consequence of Equation 3

$$\Pr\{m(H,t+1) \geq h\} = \sum_{k=h}^{M} \binom{M}{k} \alpha(H,t)^k (1-\alpha(H,t))^{M-k}, \tag{9}$$

but it is easier to handle mathematically when $\alpha(H,t)$ is unknown [19].

Interestingly, Equation 8 can be used to provide an upper bound for the probability of schema extinction in one generation as a simple function of the signal-to-noise ratio. This is achieved by solving for k the equation $M\alpha(H,t) - k\sqrt{M\alpha(H,t)(1-\alpha(H,t))} = 0$ obtaining $k = \sqrt{M}\sqrt{\frac{\alpha(H,t)}{1-\alpha(H,t)}} = \left(\frac{S}{N}\right)$. This, substituted into Equation 8, after simple calculations leads to

$$\Pr\{m(H,t+1) = 0\} \leq \frac{1}{\left(\frac{S}{N}\right)^2}. \tag{10}$$

(This is a small new result, but we report it here to make it easier to see where this comes from.)

Again, these results are valid for GAs as well as for GP.

2.5 Stephens and Waelbroeck's GA Schema Theory

The results in the previous two sections require the knowledge of the total transmission probability α, i.e. they require that not only schema survival and schema disruption events be modelled mathematically but also that schema creation events are. This is not be an easy task if one wants to do that using only the properties of the schema H (such as the number of instances of H and the fitness of H) and those of the population when expressing the quantity α. Indeed, to the best of our knowledge, none of the schema theorems presented to date in the literature have succeeded in doing this. This is the reason why all of schema theorems provide upper bounds. However, thanks to the recent work of Stephens and Waelbroeck [17, 18] it is now possible to express exactly $\alpha(H,t)$ for GAs operating on fixed-length bit strings by using properties of lower-order schemata which are supersets of the schema under consideration.

For a GA with one point crossover applied with a probability p_{xo}, $\alpha(H,t)$ is given by the following equation:

$$\alpha(H,t) = (1-p_{xo})p(H,t) + \frac{p_{xo}}{N-1}\sum_{i=1}^{N-1} p(L(H,i),t)p(R(H,i),t) \tag{11}$$

where $L(H,i)$ is the schema obtained by replacing with "don't care" symbols all the elements of H from position $i+1$ to position N, $R(H,i)$ is the schema obtained by replacing with "don't care" symbols all the elements of H from position 1 to position i, and i varies over the valid crossover points. The symbol L stands for "left part of", while R stands for "right part of". For example, if $H =$`1*111`, $L(H,1) =$`1****`, $R(H,1) =$`**111`, $L(H,3) =$`1*1**`, $R(H,3) =$`***11`.

It should be noted that Equation 11 is in a considerably different form with respect to the equivalent results in [17, 18]. This is because we developed it using our own notation following a simpler approach with respect to that used by Stephens and Waelbroeck. In our approach we assume that while producing each individual for a new generation one flips a biased coin to decide whether to apply selection only (probability $1 - p_{xo}$) or selection followed by crossover (probability p_{xo}). If selection only is applied, then there is a probability $p(H, t)$ that the new individual created sample H (hence the first term in Equation 11). If instead selection followed by crossover is selected, we use the unusual idea of first choosing the crossover point and then the parents. When selecting the crossover point, one has to choose randomly one of the $N - 1$ crossover points each of which has a probability $1/(N - 1)$ of being selected. Once this decision has been made, one has to select two parents. Then crossover is executed. This will result in an individual that samples H only if the first parent has the correct left-hand side (with respect to the crossover point) *and* the second parent has the correct right-hand side. These two events are independent because each parent is selected with an independent Bernoulli trial. So, the probability of the joint event is the product of the probabilities of the two events. Assuming that crossover point i has been selected, the first parent has the correct left-hand side if it belongs to $L(H, i)$ while the second parent has the correct right-hand side if it belongs to $R(H, i)$. The probabilities of these events are $p(L(H, i), t)$ and $p(R(H, i), t)$, respectively (whereby the terms in the summation in Equation 11, the summation being there because there are $N - 1$ possible crossover points). Combining the probabilities all these events one obtains Equation 11.

By substituting Equation 11 into Equation 4 and performing some minor additional calculations[3] the GA schema theorem described in [17, 18] is obtained. Stephens and Waelbroeck reported a number of other important ideas (including a refinement of the concept of effective fitness firstly defined and applied to GP in [31, 32, 33]) and results on the behaviour of a GA over multiple generations in the assumption of infinite populations.

2.6 Building Block Hypothesis in GAs and GP

The *building block hypothesis* [2] is typically stated informally by saying that a GA works by combining short, low-order schemata of above average fitness (building blocks) to form higher-order ones over and over again until it converges to an optimum or near-optimum solution.

The building block hypothesis and the related concept of deception have been strongly criticised in [34], in the context of binary GAs, where it was suggested that these ideas can only be used to produce first-order approximations of what really happens in a GA. In [25] the criticisms to the building block hypothesis have been extended to the case of GP with standard crossover, arguing that the hypothesis is even less applicable to GP because of the highly destructive effects

[3] In [17, 18] the summation in Equation 11 is performed only over the crossover points between the extreme defining bits in a schema.

of crossover. The analysis of Equation 2 in [3, 4] and the experimental results in [28] suggest that the latter criticism does not apply to GP with one-point crossover. Therefore, we cannot exclude that the building block hypothesis might be a good first approximation of the behaviour of GP with one-point crossover.

Stephens and Waelbroeck analysed their schema theorem (equivalent to Equation 11) and stated that the presence of the terms (equivalent to) $p(L(H,i),t)p(R(H,i),t)$ is a clear indication of the fact that, indeed, GAs build higher-order schemata (H) by juxtaposing lower-order ones (the $L(H,i)$'s and $R(H,i)$'s) [17, 18]. However, these building blocks are not necessarily all fitter than average, short or even low-order.

3 New Results on GA Schemata

The availability of Equation 11 makes it possible and easy to produce a series of new results which go beyond the schema theorem obtained by Stephens and Waelbroeck and provide additional statistical information on schema behaviour for finite populations. This can be done by simply substituting the expression of $\alpha(H,t)$ given in Equation 11 into Equations 3, 5, 6, 7, 8, 9 and 10.

We believe that the equations resulting from Equations 8 and 9 are particularly important, at least in principle, because they show that it is possible to remove the expectation operator from exact schema theories without having to assume infinite populations. We have started studying the consequences of this fact in [19].

4 GP Hyperschema Theory

A question that immediately comes to mind is whether it would not be possible to extend Rosca's or the fixed-size-and-shape GP schema theories to emulate what Stephens and Waelbroeck have done for GAs. This is important because it would make it possible to exploit the recent and new GA results summarised above in GP, too. As described below we were able to extend the fixed-size-and-shape GP schema theory for one-point crossover, but that required to generalise the definition of GP schema.

4.1 Hyperschemata

In order to do obtain for GP with one-point crossover results similar to those obtained Stephens and Waelbroeck for binary GAs we need to introduce a new, more general definition of schema. The definition is as follows:

Definition 1 (GP hyperschema). *A* GP hyperschema *is a rooted tree composed of nodes from the set $\mathcal{F} \cup \mathcal{T} \cup \{=, \#\}$, where $\mathcal{F}$ and $\mathcal{T}$ are the function set and the terminal set used in a GP run, and the operators* = *and* # *are polymorphic functions with as many arities as the number of different arities of the elements of $\mathcal{F} \cup \mathcal{T}$, the terminals in $\mathcal{T}$ being 0-arity functions. The operator* = *is* *a "don't care" symbols which stands for exactly one node, while the operator* # *stands for any valid subtree. The internal nodes of a hyperschema can be chosen from* $\mathcal{F} \cup \{=\}$*, while the leaves of a hyperschema are elements of* $\mathcal{T} \cup \{=, \#\}$.

This definition presents some of the features of the schema definition we used for our GP schema theory [4]: hyperschemata are rooted trees, and they include "don't care" symbols which stand for one node only. However, Definition 1 also includes one of the features of the schema definitions proposed by other authors [25, 5]: hyperschemata also include "don't care" symbols which stand for entire subtrees. Indeed, the notion of hyperschema is a generalisation of both Rosca's schemata (which are hyperschemata without = symbols) and fixed-size-and-shape schemata (which are hyperschemata without # symbols). So, a hyperschema can represent a group of schemata in essentially the same way as such schemata represent groups of program trees (hence the name "hyperschema"). An example of hyperschema is `(* # (+ x =))`. This hyperschema represents all the programs with the following characteristics: a) the root node is a product, b) the first argument of the root node is any valid subtree, c) the second argument of the root node is `+`, d) the first argument of the `+` is the variable `x`, e) the second argument of the `+` is any valid node in the terminal set.

4.2 Theory for Programs of Fixed Size and Shape

If one had a population of programs all having exactly the same size and shape, thanks to the definition of hyperschema, it would be possible to express the total transmission probability of a fixed-size-and-shape schema using the properties of lower-order hyperschemata, in the presence of one-point crossover, in exactly the same way as in Equation 11, i.e.

$$\alpha(H,t) = (1 - p_{xo})p(H,t) + \frac{p_{xo}}{N(H)-1} \sum_{i=1}^{N(H)-1} p(L(H,i),t)p(U(H,i),t) \quad (12)$$

where: $N(H)$ is the number nodes in the schema H (which is assumed to have exactly the same size and shape of the programs in the population); $L(H,i)$ is the hyperschema obtained by replacing with = nodes all the nodes on the path between crossover point i and the root node, and with # nodes all the subtrees connected to the nodes replaced with =; $U(H,i)$ is the hyperschema obtained by replacing with a # node the subtree below crossover point i; i varies over the valid $N(H)-1$ crossover points. The symbol L now stands for "lower part of", while U stands for "upper part of". For example, if $H =$`(* = (+ x =))` and the crossover points are numbered as in Figure 3 (top left) then $L(H,1)$ is obtained by first replacing the root node with a $=$ symbol and then replacing the subtree connected to the right of root node with a $\#$ symbol obtaining `(= = #)`, as indicated in the second column of Figure 3. The schema $U(H,1)$ is instead obtained by replacing the subtree below the crossover point with a $\#$ symbol obtaining `(* # (+ x =))`, as illustrated in the third column of Figure 3. The fourth and fifth columns of Figure 3 show how $L(H,3) =$`(=

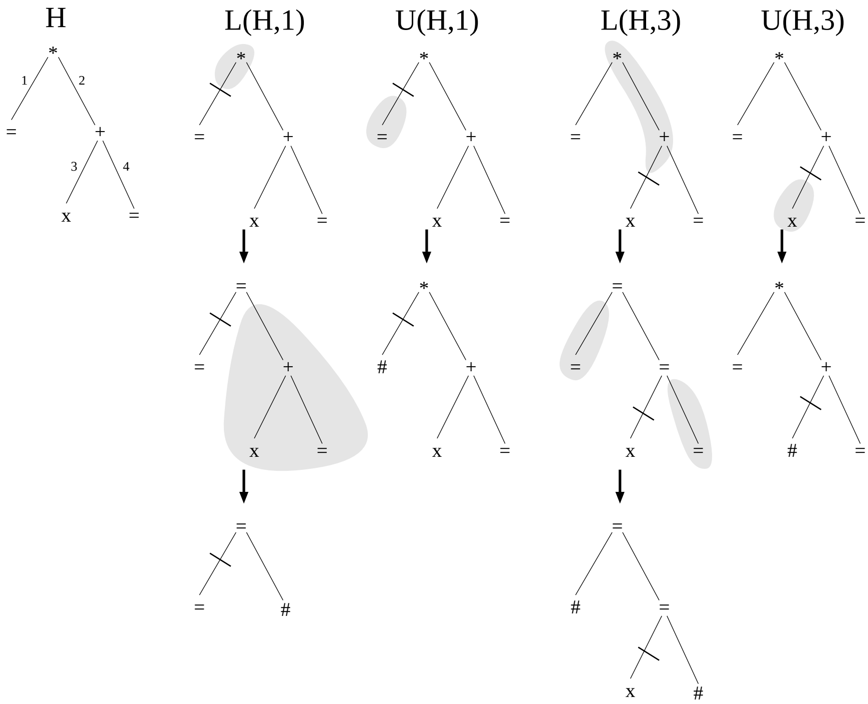

Fig. 3. Example of hyperschema and some of its potential building blocks.

`# (= x #))` and $U(H,3) =$`(* = (+ # =))` are obtained. The remaining building blocks for the schema H are: $L(H,2) =$`(= # (+ x =))`, $U(H,2) =$`(* = #)`, $L(H,4) =$`(= # (= # =))`, and $U(H,4) =$`(* = (+ x #))`.[4]

Formally this result can be obtained by proceeding like in Section 2.5. Again we assume that while producing each individual for a new generation one first decides whether to apply selection only (probability $1-p_{xo}$) or selection followed by crossover (probability p_{xo}). If selection only is applied, the new individual created samples H with a probability $p(H,t)$ (hence the first term in Equation 12). If selection followed by crossover is selected, we first choose the crossover point (to be interpreted as a link between nodes) randomly out of one of the $N(H)-1$ crossover points available. Then we select two parents and perform crossover.

[4] One might think that the definitions of $L(H,i)$ and $U(H,i)$ given above are overly complicated with respect to their GA counterparts. Indeed, if all the programs in the population have the same size and shape, one could equivalently define $L(H,i)$ as the schema obtained by replacing with `=` nodes all the nodes above crossover point i and $U(H,i)$ as the schema obtained by replacing with a `=` nodes all the nodes below crossover point i. So, it would not be necessary to introduce the notion of hyperschema altogether. However, as clarified below, the notion of hyperschema can be used to produce more general results which apply when the population contains programs of different size and shape.

This will result in an individual that samples H only if the first parent has the correct lower part (with respect to the crossover point) *and* the second parent has the correct upper part. Assuming that crossover point i has been selected,[5] the parents recreate H if they belong to $L(H,i)$ and $U(H,i)$, respectively (whereby the terms in the summation in Equation 12). Combining the probabilities all these events one obtains Equation 12.[6]

4.3 General Case

Alternatively, Equation 12 can be obtained by specialising the following general result which is valid for populations of programs of any size and shape:

Theorem 1 (Exact GP Schema Theorem). *The total transmission probability for a fixed-size-and-shape GP schema H under one-point crossover and no mutation is*

$$\begin{aligned}\alpha(H,t) &= (1-p_{xo})p(H,t) \\ &\quad + p_{xo}\sum_{h_1}\sum_{h_2}\frac{p(h_1,t)p(h_2,t)}{NC(h_1,h_2)-1}\sum_{i\in C(h_1,h_2)}\delta(h_1\in L(H,i))\delta(h_2\in U(H,i))\end{aligned} \tag{13}$$

where the first two summations are over all the individuals in the population, $NC(h_1,h_2)$ is the number of nodes in the tree fragment representing the common region between program h_1 and program h_2, $C(h_1,h_2)$ is the set of indices of the crossover points in such a common region, and $\delta(x)$ is a function which returns 1 if x is true, 0 otherwise.

The theorem can easily be proven by: a) considering all the possible ways in which parents can be selected for crossover and in which the crossover points can be selected in such parents, b) computing the probabilities that each of those events happens *and* the first parent has the correct lower part (w.r.t. the crossover point) to create H while the second parent has the correct upper part, and c) adding up such probabilities (whereby the three summations in Equation 14). Obviously, in order for this equation to be valid it is necessary to assume that if a crossover point i is in the common region between two programs but outside the schema H under consideration, then $L(H,i)$ and $U(H,i)$ are empty sets (i.e. they cannot be matched by any individual).

Equation 14 allows one to compute the exact total transmission probability of a GP schema. Therefore, by proceeding as indicated in Section 3 a series of new results for GP schemata can be obtained which provide important new statistical information on schema behaviour (e.g. schema probability distribution, schema variance, schema signal-to-noise ratio, extinction probability, etc.). In this paper

---

[5] Any numbering scheme for the crossover points produces the same result.

[6] A simpler version of this equation, applicable when crossover is applied with 100% probability, was presented in [21] where we failed to state the fact that, in this form, the equation is only applicable to populations of programs of fixed size and shape.

we will not study these results. Instead we will concentrate on understanding Equation 14 in greater depth.

If one restricts the first two summations in Equation 14 to include only the individuals having the same shape and size of the schema H, i.e. which belong to $G(H)$, one obtains:

$$\alpha(H,t) \geq (1-p_{xo})p(H,t)+$$
$$\frac{p_{xo}}{N(H)-1}\sum_{i=1}^{N(H)-1}\sum_{h_1\in G(H)} p(h_1,t)\delta(h_1\in L(H,i))\sum_{h_2\in G(H)} p(h_2,t)\delta(h_2\in U(H,i))$$

where we used the following facts: $NC(h_1,h_2) = N(H)$ since $h_1, h_2 \in G(H)$, all common regions have the same shape and size as H, and so $C(h_1,h_2) = \{1,2,\ldots,N(H)-1\}$ (assuming that the crossover points in H are numbered 1 to $N(H)-1$). This equation can easily be transformed into the following

Theorem 2 (GP Schema Theorem with Schema Creation Correction). *A lower bound for the total transmission probability for a fixed-size-and-shape GP schema H under one-point crossover and no mutation is*

$$\alpha(H,t) \geq (1-p_{xo})p(H,t)+\frac{p_{xo}}{N(H)-1}\sum_{i=1}^{N(H)-1} p(L(H,i)\cap G(H),t)p(U(H,i)\cap G(H),t), \tag{14}$$

the equality applying when all the programs in the population sample $G(H)$.

Let us compare this result to the one described in Section 2.2.

It is easy to see that by dividing the right-hand side of Equation 2 by M one obtains a lower bound for $\alpha(H,t)$. If we consider the case in which $p_m = 0$ and we assume the worst case scenario for $p_{\text{diff}}(t)$, i.e. $p_{\text{diff}}(t) = 1$, we obtain[7]

$$\alpha(H,t) \geq p(H,t)\left\{1-p_{xo}\left[1-p(G(H),t)+\frac{\mathcal{L}(H)}{(N(H)-1)}(p(G(H),t)-p(H,t))\right]\right\}$$

By subtracting the right-hand side of this equation from the lower bound provided by Equation 14 one can show that the difference is:

$$\Delta\alpha(H,t) = \frac{p_{xo}}{N(H)-1}\left(\sum_{i\in B(H)} p(L(H,i),t)p(U(H,i),t) - \mathcal{L}(H)p(H,t)^2\right),$$

where $B(H)$ is the set of crossover points in the tree fragment used in the definition of $\mathcal{L}(H)$. Since such tree fragment contains exactly $\mathcal{L}(H)$ crossover points, we can rewrite this equation as

$$\Delta\alpha(H,t) = \frac{p_{xo}}{N(H)-1}\sum_{i\in B(H)}[p(L(H,i),t)p(U(H,i),t) - p(H,t)^2].$$

[7] We are forced to assume $p_{\text{diff}}(t) = 1$ to maintain the generality of our results.

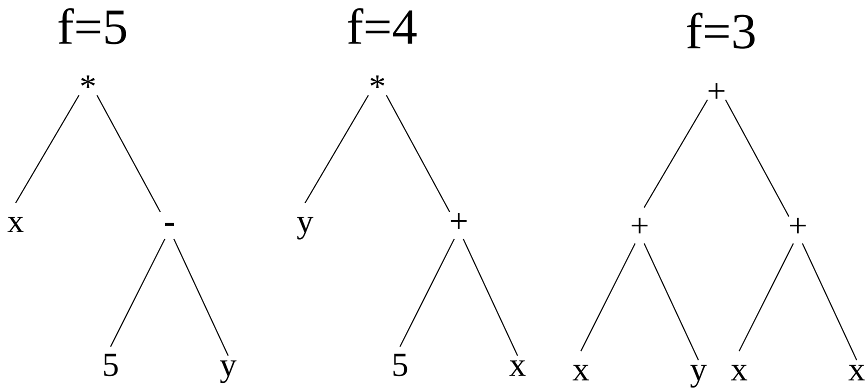

Fig. 4. Example population of programs with their fitnesses.

Since, for $i \in B(H)$, $L(H,i)$ and $U(H,i)$ are supersets of H, it follows that $p(L(H,i),t) \geq p(H,t)$ and $p(U(H,i),t) \geq p(H,t)$, whereby we see that $\Delta\alpha(H,t) \geq 0$, i.e. Theorem 2 provides a better estimate of the true transmission probability of a schema in the assumption that $p_{\text{diff}}(t) = 1$. This is why Theorem 2 is called "GP Schema Theorem with Schema Creation Correction".

4.4 Example

Let us consider an example. Let us imagine that the population contains 10 copies of each of the 3 programs in Figure 4, and let us consider the propagation of the schema $H' =$`(= = (= = =))`. Since $\mathcal{L}(H') = 0$ and $B(H') = \emptyset$ then $\Delta\alpha(H',t) = 0$. So, the new GP schema theorem cannot provide a better bound than the previous one.

However, if one considers the schema $H'' =$`(* = (- 5 =))`, with simple calculations (assuming fitness proportionate selection) one obtains $\Delta\alpha(H',t) \approx 0.07 \times p_{xo}$. This might look like a small difference, but the new schema theorem can provide a lower bound for $E[m(H,t+1)]$ of up to about 2.1 higher than the old theorem, the maximum value being reached when $p_{xo} = 1$. This is a big difference considering that the correct value for $E[m(H,t+1)]$ obtained from Equation 14 is 10.5 (for reference: the value obtained if selection only was acting is 12.5), and the lower bound provided by the old schema theorem is 7.3. So, in this example the "schema creation correction" improves the estimate by nearly 30% providing a very tight lower bound of 9.4 for $E[m(H,t+1)]$.

4.5 Hyper Building Blocks?

The presence of the terms $p(L(H,i),t)p(U(H,i),t)$ in the results presented in this section seems to suggest that some hyperschemata are a form of building blocks for fixed-size-and-shape schemata and that, indeed, also GP with one-point crossover builds higher-order schemata by juxtaposing lower-order ones.

The results also suggest that for this to happen the building blocks need not necessarily be all fitter than average, short or even low-order. The question is whether this is true for general hyperschemata (remember that fixed-size-and-shape schemata are only a special kind of hyperschemata, since they do not include # symbols). More work is needed to see if Equation 14 provides support for the existence of natural building blocks for general hyperschemata.

5 Conclusions

In this paper, for the first time, we provide an exact schema theorem for genetic programming which models the effects of schema creation due to crossover, in addition to schema survival and disruption. This theorem is the result of extending recent GA theory to GP through the definition of a new, more general notion of GP schema: the hyperschema.

In the paper we have also derived a simplified version of the exact GP schema theorem: the GP schema theorem with schema creation correction. This provides a simple way of calculating a lower bound for the total transmission probability of a schema which is significantly more accurate then the one provided by earlier schema theorems.

In addition, the paper indicates how recent general results reported in the GP literature can be applied to extend the most advanced results reported to date on GA schemata as well as the exact GP schema theorem presented in this paper. This extension provides much more information on the probabilistic behaviour of the number of instances of a schema at the next generation $m(H, t+1)$.

This paper shows how the theory of genetic algorithms and theory of genetic programming are converging more and more and how they can mutually support the development of each other.

Acknowledgements

The author would like to thank the members of the EEBIC (Evolutionary and Emergent Behaviour Intelligence and Computation) group at Birmingham for useful discussions and comments.

References

[1] J. Holland, *Adaptation in Natural and Artificial Systems.* Cambridge, Massachusetts: MIT Press, second ed., 1992.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, Massachusetts: Addison-Wesley, 1989.

[3] R. Poli and W. B. Langdon, "A new schema theory for genetic programming with one-point crossover and point mutation," in *Genetic Programming 1997: Proceedings of the Second Annual Conference* (J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, eds.), (Stanford University, CA, USA), pp. 278–285, Morgan Kaufmann, 13-16 July 1997.

[4] R. Poli and W. B. Langdon, "Schema theory for genetic programming with one-point crossover and point mutation," *Evolutionary Computation*, vol. 6, no. 3, pp. 231–252, 1998.

[5] J. P. Rosca, "Analysis of complexity drift in genetic programming," in *Genetic Programming 1997: Proceedings of the Second Annual Conference* (J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, eds.), (Stanford University, CA, USA), pp. 286–294, Morgan Kaufmann, 13-16 July 1997.

[6] L. Altenberg, "The Schema Theorem and Price's Theorem," in *Foundations of Genetic Algorithms 3* (L. D. Whitley and M. D. Vose, eds.), (Estes Park, Colorado, USA), pp. 23–49, Morgan Kaufmann, 31 July–2 Aug. 1994 1995.

[7] W. G. Macready and D. H. Wolpert, "On 2-armed gaussian bandits and optimization." Sante Fe Institute Working Paper 96-05-009, March 1996.

[8] D. B. Fogel and A. Ghozeil, "Schema processing under proportional selection in the presence of random effects," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 4, pp. 290–293, 1997.

[9] D. B. Fogel and A. Ghozeil, "The schema theorem and the misallocation of trials in the presence of stochastic effects," in *Evolutionary Programming VII: Proc. of the 7th Ann. Conf. on Evolutionary Programming* (D. W. V.W. Porto, N. Saravanan and A. Eiben, eds.), pp. 313–321, 1998.

[10] A. E. Nix and M. D. Vose, "Modeling genetic algorithms with markov chains," *Annals of Mathematics and Artificial Intelligence*, vol. 5, pp. 79–88, 1992.

[11] T. E. Davis and J. C. Principe, "A markov chain framework for the simple genetic algorithm," *Evolutionary Computation*, vol. 1, no. 3, pp. 269–288, 1993.

[12] G. Rudolph, "Stochastic processes," in *Handbook of Evolutionary Computation* (T. Baeck, D. B. Fogel, and Z. Michalewicz, eds.), pp. B2.2–1–8, Oxford University Press, 1997.

[13] G. Rudolph, "Genetic algorithms," in *Handbook of Evolutionary Computation* (T. Baeck, D. B. Fogel, and Z. Michalewicz, eds.), pp. B2.4–20–27, Oxford University Press, 1997.

[14] G. Rudolph, "Convergence analysis of canonical genetic algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, pp. 96–101, 1994.

[15] G. Rudolph, "Modees of stochastic convergence," in *Handbook of Evolutionary Computation* (T. Baeck, D. B. Fogel, and Z. Michalewicz, eds.), pp. B2.3–1–3, Oxford University Press, 1997.

[16] N. J. Radcliffe, "Schema processing," in *Handbook of Evolutionary Computation* (T. Baeck, D. B. Fogel, and Z. Michalewicz, eds.), pp. B2.5–1–10, Oxford University Press, 1997.

[17] C. R. Stephens and H. Waelbroeck, "Effective degrees of freedom in genetic algorithms and the block hypothesis," in *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)* (T. Bäck, ed.), (East Lansing), Morgan Kaufmann, 1997.

[18] C. R. Stephens and H. Waelbroeck, "Schemata evolution and building blocks," *Evolutionary Computation*, vol. 7, no. 2, pp. 109–124, 1999.

[19] R. Poli, "Probabilistic schema theorems without expectation, left-to-right convergence and population sizing in genetic algorithms," Tech. Rep. CSRP-99-3, University of Birmingham, School of Computer Science, Jan. 1999.

[20] R. Poli, "Schema theorems without expectations," in *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference* (W. Banzhaf, J. Daida, A. E. Eiben, M. H. Garzon, V. Honavar, M. Jakiela, and R. E. Smith, eds.), (Orlando, Florida, USA), Morgan Kaufmann, 13-17 July 1999. Forthcoming.

[21] R. Poli, "Schema theory without expectations for GP and GAs with one-point crossover in the presence of schema creation," in *Foundations of Genetic Programming* (T. Haynes, W. B. Langdon, U.-M. O'Reilly, R. Poli, and J. Rosca, eds.), (Orlando, Florida, USA), 13 July 1999.

[22] D. Whitley, "A genetic algorithm tutorial," Tech. Rep. CS-93-103, Department of Computer Science, Colorado State University, August 1993.

[23] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[24] U.-M. O'Reilly, *An Analysis of Genetic Programming.* PhD thesis, Carleton University, Ottawa-Carleton Institute for Computer Science, Ottawa, Ontario, Canada, 22 Sept. 1995.

[25] U.-M. O'Reilly and F. Oppacher, "The troubling aspects of a building block hypothesis for genetic programming," in *Foundations of Genetic Algorithms 3* (L. D. Whitley and M. D. Vose, eds.), (Estes Park, Colorado, USA), pp. 73–88, Morgan Kaufmann, 31 July–2 Aug. 1994 1995.

[26] P. A. Whigham, "A schema theorem for context-free grammars," in *1995 IEEE Conference on Evolutionary Computation*, vol. 1, (Perth, Australia), pp. 178–181, IEEE Press, 29 Nov. - 1 Dec. 1995.

[27] P. A. Whigham, *Grammatical Bias for Evolutionary Learning.* PhD thesis, School of Computer Science, University College, University of New South Wales, Australian Defence Force Academy, 14 Oct. 1996.

[28] R. Poli and W. B. Langdon, "An experimental analysis of schema creation, propagation and disruption in genetic programming," in *Genetic Algorithms: Proceedings of the Seventh International Conference* (T. Back, ed.), (Michigan State University, East Lansing, MI, USA), pp. 18–25, Morgan Kaufmann, 19-23 July 1997.

[29] R. Poli, W. B. Langdon, and U.-M. O'Reilly, "Analysis of schema variance and short term extinction likelihoods," in *Genetic Programming 1998: Proceedings of the Third Annual Conference* (J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, eds.), (University of Wisconsin, Madison, Wisconsin, USA), pp. 284–292, Morgan Kaufmann, 22-25 July 1998.

[30] D. Whitley, "An executable model of a simple genetic algorithm," in *Foundations of Genetic Algorithms Workshop (FOGA-92)* (D. Whitley, ed.), (Vail, Colorado), July 1992.

[31] P. Nordin and W. Banzhaf, "Complexity compression and evolution," in *Genetic Algorithms: Proceedings of the Sixth International Conference (ICGA95)* (L. Eshelman, ed.), (Pittsburgh, PA, USA), pp. 310–317, Morgan Kaufmann, 15-19 July 1995.

[32] P. Nordin, F. Francone, and W. Banzhaf, "Explicitly defined introns and destructive crossover in genetic programming," in *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications* (J. P. Rosca, ed.), (Tahoe City, California, USA), pp. 6–22, 9 July 1995.

[33] P. Nordin, F. Francone, and W. Banzhaf, "Explicitly defined introns and destructive crossover in genetic programming," in *Advances in Genetic Programming 2* (P. J. Angeline and K. E. Kinnear, Jr., eds.), ch. 6, pp. 111–134, Cambridge, MA, USA: MIT Press, 1996.

[34] J. J. Grefenstette, "Deception considered harmful," in *FOGA-92, Foundations of Genetic Algorithms*, (Vail, Colorado), 24–29 July 1992. Email: gref@aic.nrl.navy.mil.