A Competitive Building Block Hypothesis

Conor Ryan, Hammad Majeed, and Atif Azad

Biocomputing and Developmental Systems Group
Computer Science and Informations Systems Department,
University of Limerick, Ireland.
{Conor.Ryan, Hammad.Majeed, Atif.Azad}@ul.ie

Abstract. This paper is concerned with examining the way in which rooted building blocks grow in GP-like systems. We hypothesize that, in addition to the normal notion of co-operative building blocks, there are also *competitive* building blocks in the population. These competitive building blocks are all of the rooted variety, all share a similar root structure and compete with each other to spread their particular extensions to the common structure throughout the population. We demonstrate that not only do these competitive building blocks exist, but that they work in tandem with non-rooted co-operative building blocks.

1 Introduction

The traditional view of building blocks in Evolutionary Algorithms in general, and Genetic Algorithms in particular, is of their parallel discovery by a population. Different individuals in the same population may contain separate, non-overlapping building blocks which, when recombined with each other, produce a new, larger building block, usually with an associated increase in fitness. Building blocks are looked upon as being *co-operative*, as an individual can increase its fitness simply by accumulating more of them.

There is much work in the literature on building block exchange [5] [4] and the design of the *deceptive* family of problems [2] [3] was motivated in part to examine how GAs first discover, and then exploit and share building blocks.

There has also been some work in this area in GP [8], although the tree structures employed by GP complicate the matter [14], as the manner in which the meaning of nodes often depends on the way in which they are used, and the nodes above them. In particular, the root and other nodes around it dictate how the other nodes operate. This means that a subtree from one part of an individual will not necessarily operate the same way or give the same contribution to fitness if moved somewhere else.

This paper is concerned with testing the *competitive* building block hypothesis [18], which states that, for GP like structures, the most important building blocks are rooted ones. During evolution, as time moves on, an increasing number of individuals will have an increasingly large root structure in common, so the manner in which this common structure increases is vitally important to the

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 654–665, 2004.

success or otherwise of a run. Potential modifications to the common root structure can be looked upon as a *competition*, with different extensions being tried across the population. Once a useful extension has been discovered, it competes with other extensions for control of the population, and, in this manner, the new rooted building block extends through the population.

We use Grammatical Evolution(GE) [16] [12] to test this hypothesis, but also to test whether it is possible that *co-operative* building blocks exist in that system. GE employs a genotype-to-phenotype mapping (GPM) that, like GP, places more importance on genes (nodes) at the start of an individual. However, because of the way the mapping works, genes are interpreted in context, so their meaning can change, in an apparently more abrupt way than in GP, when the context changes. Earlier work [13] has shown that crossover still works in a sensible manner despite this, but did not attempt to capture the manner in which genes required in the final *best-of-run* individual could exist with different interpretations in the population.

2 Building Blocks

Building blocks are the cornerstone on which Genetic Algorithms are built. Most GAs assume the existence of short collections of genes (building blocks) which contribute to the fitness of individuals. Typically, when combining two non-overlapping building blocks in an offspring individual, one can expect to derive some fitness from both of building blocks that have been contributed.

However, constructing increasingly larger building blocks is not always this straightforward. In particular, the *epistatic* effect, where two or more genes contribute to a single trait, makes the discovery of a building block more difficult. For example, consider a genome G of length n. If loci G_0 and G_1 must be set to 1 to achieve an increase in fitness,then one cannot consider either on its own to be a building block. This was described by Goldberg [4] as *minimal superior building blocks*.

Many GA researchers have tried to capture characteristics of search spaces by examining or even designing building blocks. Examples include Royal Road functions from Mitchell et. al. [10], in which the perfect solution is made up of a hierarchical assembly of non-overlapping building blocks. It is argued that these functions are difficult for hill climbing algorithms as the fitness remains unchanged until the correct assembly of the complete building block. Thus, the search algorithm remains in the dark until the sudden discovery of a correct building block. Another, quite different set of examples are Deceptive Functions [2] [3], in which the building blocks are specifically designed to mislead search algorithms. In a typical case the algorithm finds a linear passage towards a local optimum which is a 1's complement of the global optimum. Goldberg [4] also provides a formula to verify if the problem is deceptive on average at schemata of all the orders below that of a fully specified building block. Higher order deceptive functions are generated by concatenating two or more deceptive functions together.

2.1 Competitive or Co-operative Building Blocks

Despite their apparent differences, one crucial trait that the two problems above have in common is their use of non-overlapping building blocks. In fact, virtually all "advanced" GAs (e.g. competent GAs [4]) rely on problems having this characteristic, in order to ensure that exchange and recombination of building blocks is possible.

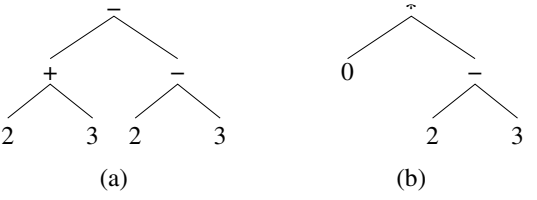

Fig. 1. Simple GP individuals in which the root node affects the outcome of the entire individual.

However, this usually is not the case in GP. Typically, when an individual is evaluated, the last node to be evaluated is the root, and all other nodes are dependent on it. Consider the two individuals in Fig. 1. Individual (a) evaluates to 6, but if the subtrees **(+ 2 3)** from (a) and **(- 2 3)** from (b) are swapped, the result is -6, even though the subtrees still evaluate to the same thing. Thus, it is not possible to assign credit to a particular subtree because, unless it is used in *exactly* the same way, one can not be guaranteed that it will have the same effect. Similarly, individual (b) is such that no subtree swapped into the right hand side of the tree will have an effect on the overall outcome due to the multiplication operator at the root.

Clearly then, the root node is the most important node in a tree, but what are the implications of this for evolution? Work on the "Eve Phenomenon" [9] demonstrated that in around 90% of GP runs there was a *single* ancestor from which *every* individual in the final population descended. Further, they demonstrated that, on average, 70% of the population shared the same *four* levels.

The *Competitive Building Block Hypothesis* follows on from this, and conjectures that a major part of the search effort in GP concentrates on correctly identifying how best to extend this common area. It views every member of the population as an attempt by GP to extend the area, and that much of the useful exploration being carried out by GP at any given time occurs in this area. Fig. 2 indicates the three areas; *(i)* the common area, in which little or no useful exploration happens, *(ii)* the area of discovery, where most of the useful discoveries are made, and *(iii)* the tails, where useful discoveries are less likely to be maintained, due to their distance from the fringe of the common area.

This paper is concerned with measuring the rate at which the common area grows, and the extent to which information in the tails and on the fringe of the area of discovery get incorporated into the common area. We provide evidence

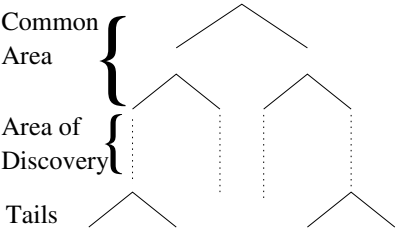

Fig. 2. The different areas of discovery for GP individuals. Root nodes are typically the most stable, while processing carried out at the leaves is likely to be modified by higher nodes.

to show that the competitive building block hypothesis exists, but also that the traditional co-operative building block hypothesis also holds, and that useful information can be held in subtrees lower down in individuals.

To aid our analysis, we employ the Grammatical Evolution system. GE is useful in this case because its linear structures are relatively simple to analyse, and its property of *intrinsic polymorphism* (see the following section) captures the dynamics of subtrees changing meaning with context.

3 Background

Grammatical Evolution [12] is a Genetic Programming system that combines the convenience of a GA style binary string representation with the expressive power of a high level language program. Following a biological metaphor the binary string is termed as the *genotype* that is translated into a program or the *phenotype* through a genotype-phenotype mapping process. The mapping process commonly uses a context free grammar(CFG) represented in Backus Naur Form (BNF). However, the modular design of this evolutionary algorithm permits the use of other types of grammars and notations, for example, Adaptive Logic Programming system (ALP) [6] combines the context sensitive programming features of Prolog logic programs with GE and demonstrates many interesting applications.

As mentioned earlier, CFG is the most widely used grammar in GE and it is represented by a tuple $\{T, N, P, S\}$. T is the set of terminals, the symbols that occur in the valid sentences of the language defined by the grammar. N denotes the set of non-terminals, the interim items that lead to the terminals. P is a set of production rules and $S \in N$ is the start symbol. A grammar typically used by GE for symbolic regression problems is given below.

```
S = <expr>
<expr>    ::= (<expr> <op> <expr>) | <pre-op> (<expr>) | <var>
<op>      ::= / | - | * | +
<pre-op>  ::= Sin | Cos | Log | Exp
<var>     ::= x | 1.0
```

The mapping process chooses different rules from the grammar to arrive at a particular sentence. The genotype is treated as a sequence of 8 bit genes. The mapping starts with the start symbol. At each step during the mapping process a gene is read from the genome to pick a rule for the non-terminal under consideration. Use is made of the **mod** operation to decode a gene into a rule in the following manner.

Rule index = (Gene) Mod (Number of rules for the particular non-terminal)

To elucidate the mapping process consider a sample individual **24 32 14 19 136 7 8 128 19 21**. Start symbol `<expr>` has the following options.

```
<expr> ::= ( <expr> <op> <expr> )    (0)
         | <pre-op> ( <expr> )       (1)
         | <var>                     (2)
```

24 mod 3 = 0. Thus, `(<expr> <op> <expr)` is chosen. The mapping process always resolves the left most non-terminal. `<expr>` being the same non-terminal as before the set of options remains unchanged. The mapping proceeds by reading the next gene **32** that decodes to `<var>`. The expression now becomes `(<var> <op> <expr>)`. For `<var>` the set of choices is:

```
<var> ::= x       (0)
        | 1.0     (1)
```

14 mod 2 = 0. Thus, `<var>` is replaced by `x` in the expression. The mapping continues in this manner until the individual is completely mapped to `(x + Exp (x) )`. The genetic material is reused if the mapping is incomplete at the end of a single pass through the individual. The phenomenon is termed as *wrapping* and discussed comprehensively in [17].

3.1 Ripple Effect and Ripple Crossover

The use of **mod** operation ensures that a gene is always interpreted *in context* of the mapping process. As a result a gene always decodes to a rule that is used immediately. This property is termed *intrinsic polymorphism* [7]. Consequently the meaning of a gene depends on all the genes that precede it in the chromosome. If a change occurs in the earlier part of the chromosome, the effect *ripples* through the rest of the genome. Thus, the one point crossover in GE is termed as the *ripple crossover*. Fig. 3 exemplifies the ripple crossover for the individual discussed previously. When the chromosome is cut at a certain location, it effectively dismantles multiple branches from the derivation tree leaving behind many *ripple sites*. The in-coming fragment may be of different length and may contain the genes used for different non-terminals. The polymorphic interpretation ensures that they are used in context to fill in the ripple sites.

3.2 Building Blocks in GE

The previous section may make it appear as though it is unlikely that building blocks can exist in GE, as they not only have the same property in GP that the

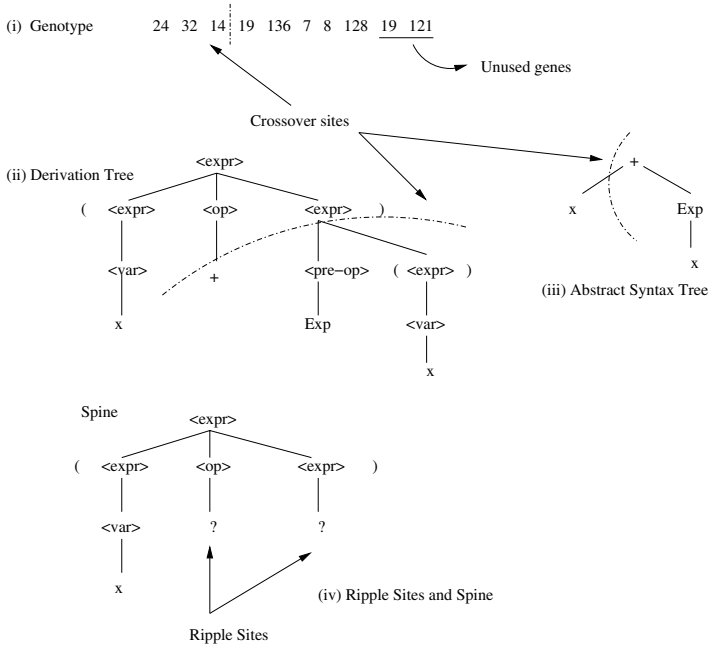

Fig. 3. Ripple crossover in GE. The change in the chromosome in (i) is correlated with the corresponding changes in the derivation tree and the abstract syntax tree in (ii) and (iii) respectively. (iv) depicts the spine and the vacant ripple sites.

operation of their phenotype depends to a large extent on the ones that preceded them (in the same manner that nodes in GP depend on their parent nodes), they can actually code for something entirely different.

An investigation in whether or not building blocks appear in GE was carried out by [11] in which they compared a variety of crossover operators, including the so-called *headless chicken* [1] crossover, in which randomly generated material is inserted into parents rather than having them exchange genes. They showed that a homologous crossover, that prevented individuals from choosing crossover points within their common areas performed approximately the same as the standard one point crossover, suggesting that a type of homologous crossover comes for free with GE. Our belief is that, while this is true, it is more the case that many of the *successful* crossovers are of this variety.

Rooted or unrooted building blocks? Rooted building blocks are clearly of importance to any type of GP system, but the fact that headless chicken crossover usually under-performs compared to standard crossover suggests that there have to be unrooted building blocks at play as well.

The following section presents a suite of experiments designed to test whether crucial building blocks from ideal individuals appear in the wrong position (and possibly with an entirely different meaning).

4 Experimental Design

The first experiment was to test if important building blocks appear in the population, but in the wrong location on individual's chromosomes. Important building blocks are defined as those that appear in the *best-of-run* individual for a particular run.

A second experiment was designed to investigate how newly extended rooted building blocks spread throughout the population. If there really is a competition to discover a rooted building block, we should be able to see evidence of the appearance of increasingly larger rooted building blocks, which then start to take over the population.

We used Koza's quartic polynomial symbolic regression problem, with a population of 500 running for 200 generations, averaged over 30 runs, using steady state replacement with roulette wheel selection. One point crossover was employed and probability was set to 0.9. Mutation was turned off; this was to permit us to focus on the effects of crossover alone. It has been shown [15] that the performance of GE drops off very sharply when mutation is removed, and we experienced a similar drop off in performance. In fact, for all the experiments here, we only experienced two successes, even though they ran for two hundred generations.

4.1 Occurrence of Incorrectly Positioned Building Blocks

These experiments were designed to check the frequency of existence of a building block at different positions throughout the population. The best individual was selected and compared to the entire population of a particular generation. Two counts were maintained, the first shows the frequency of existence of a building block at the *same* position in the whole population as it was in the best individual, while the second count keeps track of the frequency of the building block at *different* positions across the whole population. This test explains how the building blocks are spread across the genome. As the size of the building block was not known before hand, we checked the block sizes starting from one to the maximum size of the best individual. Due to space considerations we only show measures of building block size nine, as this is representative of the trend shown with other sizes. These are shown in Figs. 4 and 5. Every building block of length nine was counted, in an overlapping fashion, so the first building block occupies gene positions 0 to 8, while the second occupies 1 to 9.

As indicated by these figures, as early as after the first generation, the population (Fig. 4) has, on average, produced around seven individuals with the correct top root structure. The further along the genome we examined, the less likely a building block was to be found in its correct position - recall that we are interested in *individual* building blocks in the traditional sense rather than rooted building blocks. Curiously, the trend is that, as the number of building blocks in the correct position decreases, their number in the incorrect position increases, indicating a good mix of diversity in the population.

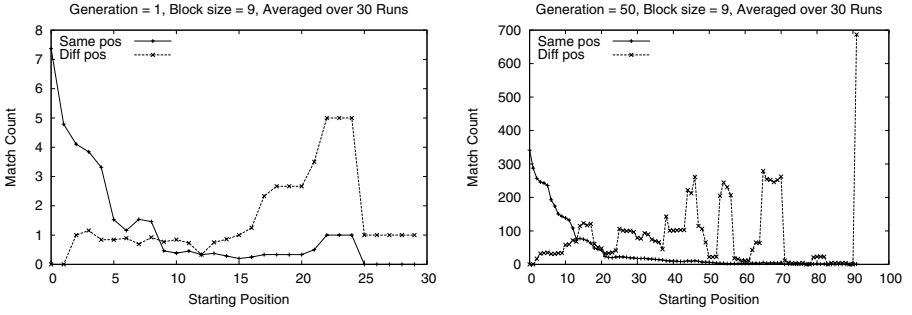

Fig. 4. Count of correctly and incorrectly positioned (overlapping) building blocks of size 9 after the first generation and after 50 generations.

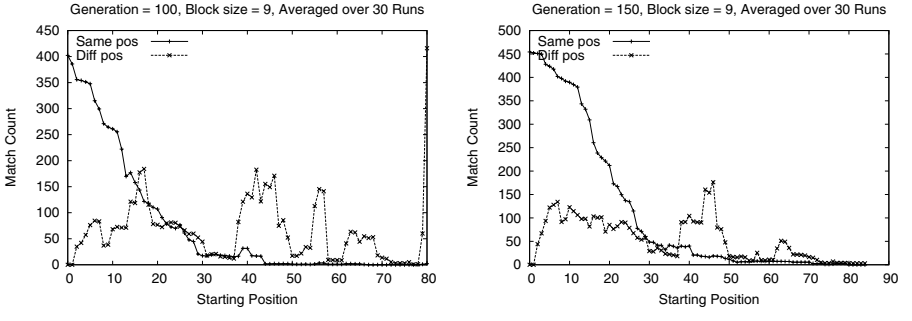

Fig. 5. Count of correctly and incorrectly positioned (overlapping) building blocks of size 9 after the 100 generation and after 150 generations.

By generation 50, also in Fig. 4, we see a very different picture. Notice the difference in the scales and that, on average, around 350 individuals have the same first nine genes. Further, for certain sequences towards the end of the chromosomes, we see relatively high peaks, indicating that many individuals have those sequences, albeit not in the correct position. This seems to support the findings of [17] which postulated the existence of a "stop sequence" in GE. That is, a sequence of genes that can successfully terminate most of the chromosomes in the population. An analogy in GP would be the existence of subtrees that, no matter where they appear in a tree at least don't give a pathological fitness.

This trend continues in Fig. 5, although the lines cross further along, indicating that the population is converging.

4.2 Growth of Rooted Building Blocks

The first batch of experiments here is designed to examine the manner in which the rooted building blocks grow over time. Fig. 6 indicates the average size of rooted building blocks in the population over time, expressed as a proportion of

the *best-of-run* individual for the particular population. It also shows the average size of the common root structure for all individuals in a population.

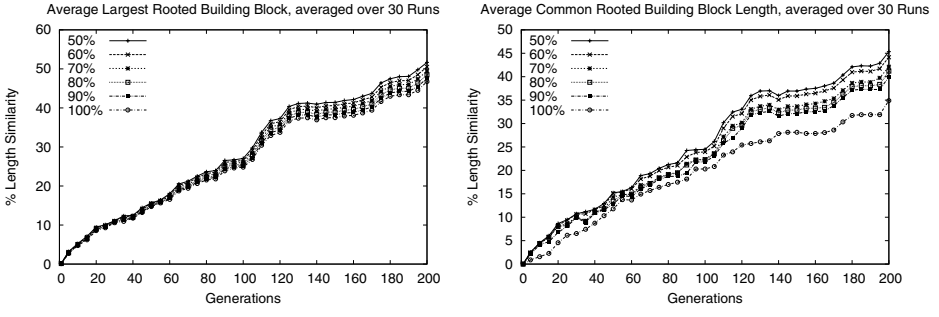

Fig. 6. On the left, the average size of the rooted building blocks, and, on the right, the average size of the common rooted building block.

As it was not clear to what extent a single rooted building block would take over the population, we made several different measures when examining what the average common root node structure was. [9] indicated that around 70% of individuals in GP tended to have a much larger area in common than when the entire population was examined. To reflect this, for this analysis we scored individuals according to how much in common their root structure was with the *best-of-run* individual to find a current *best-match* individual (although these scores were not taken into consideration during the run, this was only done for the retrospective analysis). The measurements that appear in Fig. 6 consider from the top 50% of the population down to the top 100%, the entire population.

Although examining the top 50% only gives a higher measure, one can see from the graph that there is not an enormous difference between it and using other, larger proportions of the population. The biggest drop is when the entire population is considered, but still shows that as early as generation four, *every* individual in the population has the same root node.

Fig. 7 indicates the manner in which newly discovered extended rooted building blocks take over the population. On the right one can see how the size of the largest building block increases over time, while on the left the count of individuals having that building block are shown.

The spiky nature of the counts indicates the constant discovery of new larger blocks, which then propagate throughout the rest of the population. In the earlier stages of the evolution rate of propagation is higher than the latter stages, possibly because the difference in fitness is more marked earlier on in a run. Furthermore, as the size of the root structure becomes larger, there is more space of small errors, and our method of measuring similarity only counts blocks that are *exactly* identical.

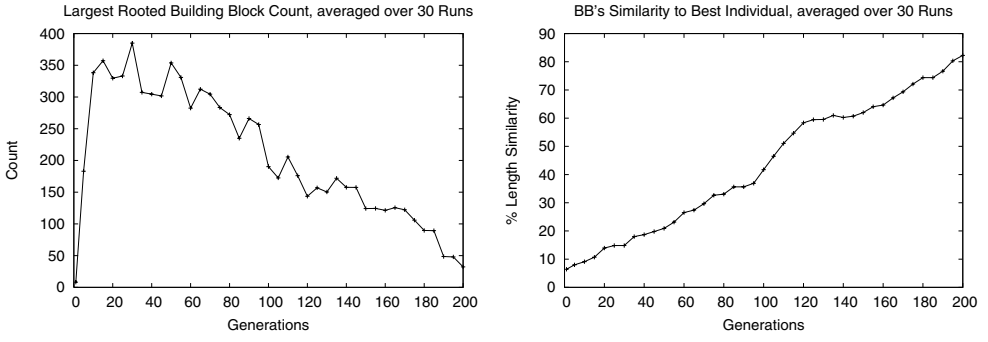

Fig. 7. Left: The count of individuals having the longest rooted building block over time. Right: The average size of the largest rooted building block in the population over time.

5 Discussion and Conclusions

These experiments appear to validate the hypothesis that the evolution is driven by a combination of competitive rooted building blocks and co-operative non-rooted building blocks. Crucially, in Figs. 4 and 5 the count of building blocks in *different positions* never reaches zero, which indicates the presence of the genetic material required for the construction of the building block at that position in the form of a *co-operative* building block, which, at a later stage, may move to the correct position.

The second set of experiments, in Fig. 6, focused on the rooted building blocks and their behavior over time. The linear slope of the curve indicates that, on average, the root structure continually grows towards the *best-of-run* individual. Also in Fig. 6 one can see that the *entire* population shares an increasingly similar root structure as time goes on.

The final set of experiments were designed to measure the convergence rate of the system towards the *best-of-run* individual, as well as the counts of the instances of each *best-match* individual. The counts of the best matching individual provides perhaps the best proof of the competitive nature of the root building blocks. Initially, the counts are very high, but, as the newly discovered root structure gets increasingly longer, the competition intensifies, with fewer and fewer individuals making the discovery each time. However, the undulating nature of the graph indicates that, once a new and longer root structure has been discovered, it very quickly spreads through the population.

In conclusion then, this paper supports a competitive building block hypothesis, and that a single rooted building block grows over time. Whether or not this is a good thing for GP-like systems remains to be seen, as it suggests that if the initial root structure is flawed, possibly through poor diversity or too small a population, then the population becomes trapped. However, to be fore-warned

is to be fore-armed, and future work will look at the implications of this, and ways in which to avoid prematurely converging on suboptimal root structure.

References

1. Peter J. Angeline and Jordan B. Pollack. Competitive environments evolve better solutions for complex tasks. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, pages 264–270, University of Illinois at Urbana-Champaign, 17-21 July 1993. Morgan Kaufmann.
2. David E. Goldberg. Simple genetic algorithms and the minimal, deceptive problem. In L. Davis, editor, *Genetic Algorithms and Simulated Annealing*, pages 74–88. Morgan Kaufmann, 1987.
3. David E. Goldberg. Genetic algorithms and walsh functions: part i, a gentle introduction. *Complex Systems*, 3(2):129–152, 1989.
4. David E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, 2002.
5. John H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Harbor, 1975.
6. Maarten Keijzer. *Scientific Discovery using Genetic Programming*. PhD thesis, Danish Technical University, Lyngby, Denmark, March 2002.
7. Maarten Keijzer, Conor Ryan, Michael O'Neill, Mike Cattolico, and Vladin Babovic. Ripple crossover in genetic programming. In Julian F. Miller, Marco Tomassini, Pier Luca Lanzi, Conor Ryan, Andrea G. B. Tettamanzi, and William B. Langdon, editors, *Genetic Programming, Proceedings of EuroGP'2001*, volume 2038 of *LNCS*, pages 74–86, Lake Como, Italy, 18-20 April 2001. Springer-Verlag.
8. W. B. Langdon and Riccardo Poli. *Foundations of Genetic Programming*. Springer-Verlag, 2002.
9. Nicholas Freitag McPhee and Nicholas J. Hopper. Analysis of genetic diversity through population history. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1112–1120, Orlando, Florida, USA, 13-17 July 1999. Morgan Kaufmann.
10. Melanie Mitchell, Stephanie Forrest, and John H. Holland. The royal road for genetic algorithms: Fitness landscapes and GA performance. In Francisco J. Varela and Paul Bourgine, editors, *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 1991*, pages 245–254, Paris, 11–13 1992. A Bradford book, The MIT Press.
11. Michael O'Neill and Conor Ryan. Crossover in grammatical evolution: A smooth operator? In Riccardo Poli, Wolfgang Banzhaf, William B. Langdon, Julian F. Miller, Peter Nordin, and Terence C. Fogarty, editors, *Genetic Programming, Proceedings of EuroGP'2000*, volume 1802 of *LNCS*, pages 149–162, Edinburgh, 15-16 April 2000. Springer-Verlag.
12. Michael O'Neill and Conor Ryan. *Grammatical Evolution: Evolutionary Automatic Programming in a Arbitrary Language*, volume 4 of *Genetic programming*. Kluwer Academic Publishers, 2003.
13. Michael O'Neill, Conor Ryan, Maarten Keijzer, and Mike Cattolico. Crossover in grammatical evolution. *Genetic Programming and Evolvable Machines*, 4(1):67–93, March 2003.

14. Una-May O'Reilly and Franz Oppacher. The troubling aspects of a building block hypothesis for genetic programming. In L. Darrell Whitley and Michael D. Vose, editors, *Foundations of Genetic Algorithms 3*, pages 73–88, Estes Park, Colorado, USA, 31 July–2 August 1994 1995. Morgan Kaufmann.
15. John O'Sullivan and Conor Ryan. An investigation into the use of different search strategies with grammatical evolution. In James A. Foster, Evelyne Lutton, Julian Miller, Conor Ryan, and Andrea G. B. Tettamanzi, editors, *Genetic Programming, Proceedings of the 5th European Conference, EuroGP 2002*, volume 2278 of *LNCS*, pages 268–277, Kinsale, Ireland, 3-5 April 2002. Springer-Verlag.
16. Conor Ryan, J. J. Collins, and Michael O Neill. Grammatical evolution: Evolving programs for an arbitrary language. In Wolfgang Banzhaf, Riccardo Poli, Marc Schoenauer, and Terence C. Fogarty, editors, *Proceedings of the First European Workshop on Genetic Programming*, volume 1391 of *LNCS*, pages 83–95, Paris, 14-15 April 1998. Springer-Verlag.
17. Conor Ryan, Maarten Keijzer, and Miguel Nicolau. On the avoidance of fruitless wraps in grammatical evolution. In E. Cantú-Paz et al, editor, *Genetic and Evolutionary Computation – GECCO-2003*, volume 2724 of *LNCS*, pages 1752–1763, Chicago, 12-16 July 2003. Springer-Verlag.
18. Conor Ryan and Miguel Nicolau. Doing genetic algorithms the genetic programming way. In Rick L. Riolo and Bill Worzel, editors, *Genetic Programming Theory and Practise*, chapter 12, pages 189–204. Kluwer, 2003.