

**Building-Block Supply in
Genetic Programming**

**Kumara Sastry
Una-May O'Reilly
David E. Goldberg
David Hill**

IlliGAL Report No. 2003012
April, 2003

Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue, Urbana, IL 61801

Building-Block Supply in Genetic Programming

Kumara Sastry

Illinois Genetic Algorithms Laboratory, and
Department of Material Science & Engineering
University of Illinois at Urbana-Champaign
ksastry@uiuc.edu

Una-May O'Reilly
Artificial Intelligence Lab
Massachusetts Institute of Technology
unamay@mit.edu

David E. Goldberg
Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
deg@uiuc.edu

David Hill
Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign
djh111@uiuc.edu

May 5, 2003

Abstract

This paper analyzes building block supply in the initial population for genetic programming. Facetwise models for the supply of a single schema as well as for the supply of all schemas in a partition are developed. An estimate for the population size, given the size (or size distribution) of trees, that ensures the presence of all raw building blocks with a given error is derived using these facetwise models. The facetwise models and the population sizing estimate are verified with empirical results.

1 Introduction

Genetic programming (GP) practitioners are often frustrated by the lack of theory available to guide them in selecting parameters for applied problems. They also lack a foundation of knowledge to explain many of their empirical findings regarding effective population sizes. To date, fundamental reasons for why varying population size while maintaining a constant number of fitness evaluations for a run results in different evolutionary trajectories and endpoints, have not been revealed.

The purpose of this paper is to start addressing this lack of theory by providing an estimate of the population size necessary to solve a given GP problem. It is hoped that like in genetic algorithm (GA) theory (Goldberg, 2002), the availability of a population sizing equation will be a valuable tool to aid GP practitioners in their efforts to understand how GP processes information. In addition, it may indicate how to adapt GP to be more competent.

The first step towards understanding population sizing is to tackle the issue of building-block (BB) supply. We forgo a temporal approach which would assume that the recombination, mutation and other diversity-generating operators will create and maintain sufficient BB diversity on an

appropriate time scale. Instead, using a spatial approach, we estimate the population size required to ensure diversity and the number of BBs present in the initial population.

The objective of this study is to develop facetwise models for supply of BBs and to estimate the population size required to guarantee the presence of all raw BBs for a given tree size (or size distribution) in the initial GP population. Though ensuring BB growth supersedes BB supply in the subsequent population, BB growth will be extremely difficult if BB supply is not ensured. Also, while decision making usually governs population sizing, it is sometimes governed by BB supply. In such cases a facetwise model of BB supply is necessary for ensuring a successful GP design. Furthermore, understanding initial supply of BBs is essential for developing a practical population-sizing model.

This paper is structured as follows. We start with a brief literature review of BB supply. Section 3 provides background and states key assumptions made in this study. Details of an expression mechanism and test problems are provided in section 4, followed by facetwise models for BB supply in section 5. Section 6 outlines some thoughts on handling BB supply for real GP expressions. Finally, summary and conclusions are presented.

2 Brief Literature Review

The GP community is interested in identifying strategies to size populations, in order to estimate the computational effort required to solve particular problems with GP; however, few studies have addressed this topic, thus far. One approach has been suggested by Langdon and Poli (2002), but it has not been fully developed. This approach employs the methodology used by Poli (2000) for the sizing of populations in GA. In this method, Poli used Stephens and Waelbroeck’s (1999) concept of transmission probability to develop a recursive conditional schema theory that allows for the prediction of the probability of reaching a solution to a problem in a fixed number of generations. An expression for the transmission probability for standard GP was developed by Langdon and Poli (2002). However, the expression is very difficult to evaluate.

The methodological and analytic foundation for our approach to deciphering selectorecombinative GA (Goldberg, 2002) and GP (Goldberg & O’Reilly, 1998; O’Reilly & Goldberg, 1998) has been stated before. Put succinctly, our approach is to analyze and understand GP’s simple mechanisms before its complex ones. We predict that lessons learned from experimentation and theory on a simple case will lead to insight, and possibly, carry over to more complex cases. Therefore, we start by analyzing building-block supply in GP’s initial population, before the activity of crossover and selection.

While building-block supply has been largely ignored in GP literature, many researchers have studied the BB supply in GAs. Holland (1975) estimated the number of BBs that receive at least a specified number of trials using Poisson distribution. A later study (Goldberg, 1989) calculated the same quantity more exactly using binomial distribution and studied their effects on population sizing in serial and parallel computation. Reeves (1993) proposed a population sizing model for supply of alphabets with fixed cardinality. Recently, Goldberg, Sastry, and Latoza (2001) developed facetwise models for ensuring BB supply in the initial population for genetic algorithms. They considered a population of fixed-length strings consisting alphabets of arbitrary cardinality χ . They predicted that the population size required to ensure the presence of all competing building blocks with a tolerance of $\epsilon = 1/m$ is given by $n = \chi^k (k \log \chi + \log m)$, where χ is the alphabet cardinality, k is BB size, and m is the number of BBs.

This paper follows a similar methodology along the lines of Goldberg, Sastry, and Latoza (2001) and develops facetwise models for predicting the probability of the presence of a single schema as

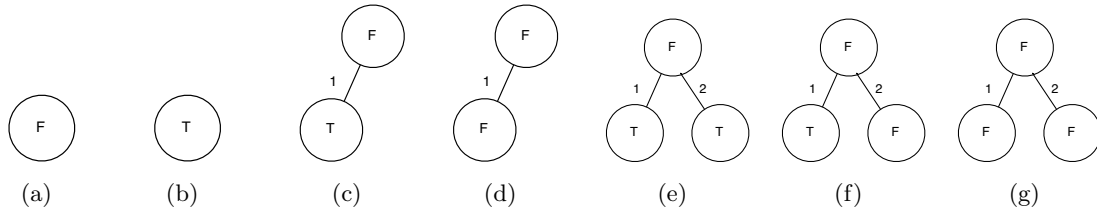


Figure 1: The smallest tree fragments in GP. Fragments (c) and (d) have mirrors where the child is 2nd parameter of the function. Likewise, fragment (f) has mirror where 1st and 2nd parameters of the function are reversed.

well as all schemas in a given partition. Before developing the models, we present some background and state assumptions used in the modeling procedure.

3 Preliminaries

In this section, we present definitions and concepts that underpin our analysis of BB supply in GP.

3.1 GP Tree Composition

Most GP implementations reported in the literature use parse trees to represent candidate programs in the population (Langdon & Poli, 2002). We have assumed this representation in our analysis. To simplify the analysis further, we consider the following:

1. A primitive set of the GP tree is $\mathcal{F} \cup \mathcal{T}$, where \mathcal{F} denotes the set of functions (interior nodes to a GP parse tree), and \mathcal{T} denotes the set of terminals (leaf nodes in a GP parse tree).
2. The cardinality of $\mathcal{F} = \chi_f$ and the cardinality of $\mathcal{T} = \chi_t$.
3. The arity of all functions in the primitive set is two: All functions are binary and thus parse trees generated from the primitive set are binary.

We believe that our analysis could be extended to primitive sets containing functions with arity greater than two (non-binary trees). We also note that our assumption closely matches a common GP benchmark, symbolic regression, which frequently has arithmetic functions of arity two.

3.2 Translating GA Schemas to GP Schemas Isn't Straightforward

Schemas are similarity templates that describe sets of solutions that share a common feature. The GP literature contains several alternative definitions of schemas (Koza, 1992; Altenberg, 1994; O'Reilly & Oppacher, 1995; Whigham, 1995; Rosca, 1997; Langdon & Poli, 2002). Per O'Reilly and Oppacher (1995), a GP schema is a multiset of subtrees and tree fragments with nodes denoted as functions, terminals or don't care symbols. Tree fragments are trees with at least one leaf that is a "don't care" symbol which can be matched by any subtree (including subtrees with only one node).

3.2.1 Tree Fragments

While in general tree fragments refer to a multiset of tree patterns or tree templates, we restrict ourselves to a single tree pattern. A tree fragment pattern has each of its nodes labeled with the

function symbol, \mathcal{F} , or terminal symbol, \mathcal{T} . However, it does not have an absolute position or positional anchor. Figure 1 shows the fragments our analysis focuses on. Along the edge between a function and its child node, a numeral denotes what parameter of the function the child node is (i.e. the first or second argument in the case of a binary function). A tree fragment has a length or size; that is, its number of nodes, $k = N_t + N_f$, where N_t and N_f are the number of terminal and functional nodes in the tree fragment, respectively. Furthermore, the total number of possible instances of a tree fragment is given by

$$\kappa = \chi_f^{N_f} * \chi_t^{N_t} \quad (1)$$

For example, for the tree fragment P_b (fragment with only terminal), $N_f = 0$, and $N_t = 1$, and therefore, the total number of instances of P_b is χ_t .

Since a tree fragment is not anchored to a position of a tree, there can be none or more than one instance of a fragment in a single tree. Yet, the smallest fragments P_a and P_b appear at least once or twice in a tree respectively. Assuming a single tree of size s^1 and the tree properties listed in section 3.1, Table 3.2.1 provides estimates (derived by probability of frequency) of the average quantity of tree-fragment instances, ϕ . In other words, ϕ counts the expected number of tree-fragment instances, given the tree size (or size distribution), in the population.

P	Description	ϕ	κ
P_a	function	$\frac{1}{2}(s-1)$	χ_f
P_b	terminal	$\frac{1}{2}(s+1)$	χ_t
P_c	one terminal that is the first parameter of a binary function	$\frac{1}{4}(s+1)$	$\chi_f \cdot \chi_t$
P_d	a function at the root and a function as its first parameter	$\frac{1}{4}(s-3)$	χ_f^2
P_e	a function at the root and 2 terminals as its parameters	$\frac{1}{8} \frac{(s+1)^2}{(s-1)}$	$\chi_f \cdot \chi_t^2$
P_f	a function at the root and 1 terminal as the first parameter and one function as its second parameter.	$\frac{1}{8} \frac{(s+1)(s+3)}{(s-1)}$	χ_f^3
P_g	a binary function at the root and 2 functions as its parameters	$\frac{1}{8} \frac{(s-3)^2}{(s-1)}$	$\chi_f^2 \cdot \chi_t$

Table 1: Designations, P_i , and descriptions of tree fragments considered in the BB supply models, the quantity of fragments, ϕ , and the number of competing schemas in the fragments, for a binary tree of size s . See also Figure 1.

3.2.2 The Tree Fragments are not Enough: How are They Expressed?

While tree fragments are the parts of a physical tree, and counting number of instances of tree fragments can itself be important, what is more important are those tree fragments that get *expressed*. The expression mechanism dictates what the building blocks of a problem are and therefore affects the BB supply. Specifically, we are interested in expression of small tree fragments into partially correct subfunctions. Let us consider, for example, symbolic regression of $1 + x + x^2 + x^3$. Early on

¹It should be noted here that the average tree size of a population can be calculated for popular initialization schemes (Koza, 1992), or initialization schemes such as PCT1 or PCT2 (Luke, 2000) can be used to generate a population which conform to an expected tree size.

in the GP run, it is important to get the constant and the linear part of the symbolic equation right. Therefore, all the tree fragments that contribute to the correct constant and linear subfunctions are important and their supply is critical in the initial population.

We illustrate the methodology to incorporate expression mechanism in BB supply models by using a simple expression mechanism, called **ORDER**, which is explained in the next section. We choose **ORDER** because while it models some of the GP behavior (Goldberg & O’Reilly, 1998; O’Reilly & Goldberg, 1998), the expression mechanism can be analyzed in a straightforward manner.

4 ORDER Expression Mechanism

ORDER is a simple, yet intuitive expression mechanism which makes it amenable to analysis and modeling (Goldberg & O’Reilly, 1998; O’Reilly & Goldberg, 1998). The primitive set of **ORDER** consists of the primitive **JOIN** of arity two and complimentary primitive pairs (X_i, \bar{X}_i) , $i = 0, 1, \dots, \ell$ of arity one. A candidate solution of the **ORDER** problem is a binary tree with **JOIN** primitive at the internal nodes and either X_i ’s or \bar{X}_i ’s at its leaves. The candidate solution’s expression is determined by parsing the program tree in order (from left to right). The program expresses the value X_i if, during the in order parse, a X_i leaf is encountered before its complement \bar{X}_i . Furthermore, only unique primitives are expressed in **ORDER** during the in order parse.

Building blocks in **ORDER** are the sets of primitives that are part of the subfunctions that reduce error (alternatively improve fitness). In this study, we consider two test problems that use **ORDER** expression mechanism: 1. **UNITATION**: where each primitive X_i is a BB, and 2. **DECEPTION**: where k primitives form a BB. The following sections describe these two test problems.

4.1 UNITATION

In **UNITATION**, for each X_i (or \bar{X}_i) that is expressed, an equal unit of fitness value is accredited. That is,

$$f_1(x_i) = \begin{cases} 1 & \text{if } x_i \in \{X_1, X_2, \dots, X_\ell\} \\ 0 & \text{otherwise} \end{cases} . \quad (2)$$

The fitness function for **ORDER** is then defined as

$$F(\mathbf{x}) = \sum_{i=1}^{\ell} f_1(x_i), \quad (3)$$

where \mathbf{x} is the set of primitives expressed by the tree. The output for optimal solution of a ℓ -primitive **UNITATION** problem is $\{X_1, X_2, \dots, X_\ell\}$, and its fitness value is ℓ .

4.2 DECEPTION

In **DECEPTION**, the primitives are divided into m subgroups, each subgroup consisting of k primitives. The fitness of each subgroup is computed using the following trap function (Goldberg, 1987; Deb & Goldberg, 1993):

$$f_k(u(x_1, x_2, \dots, x_k)) = \begin{cases} 1.0 & u = k \\ (1.0 - \delta) \left(1 - \frac{u}{k-1}\right) & u < k \end{cases} , \quad (4)$$

where u is the unitation, or the number of primitives, X_i , in a subgroup:

$$u(x_1, x_2, \dots, x_k) = \sum_{i=1}^k f_1(x_i), \quad (5)$$

where x_i is the i th primitive, δ is the difference in the functional value between the correct BB and its deceptive attractor. The fitness function of a candidate solution (tree) is then given by

$$F(\mathbf{x}) = f_k(u(x_1, x_2, \dots, x_k)) + f_k(u(x_{k+1}, \dots, x_{2k})) + \dots + f_k(u(x_{(m-1)k+1}, \dots, x_{mk})), \quad (6)$$

where, F is the fitness function, \mathbf{x} is the expressed primitives, m is the number of BBs, and $\ell = mk$.

5 Facetwise Models of Building-Block Supply

In this section, we develop facetwise models for building-block supply for ORDER expression. First we start with addressing the supply of a single BB in a given partition. Then we extend the model to ensure the supply of all schemas in a partition. We then use the facetwise models to derive a population-sizing model dictated by BB supply. The models developed in this section are verified with empirical results for UNITATION and DECEPTION along the way.

5.1 Supply of a Single Building Block

Assuming trees of size, s , and that the expression mechanism used is ORDER, the probability that a primitive expressed by a tree is given by

$$\begin{aligned} p_{X_i^{\text{exp}}} &= p(\#of X_i \geq 1, \#of \bar{X}_i = 0) + p(X_i \text{ appears before } \bar{X}_i), \\ &= \sum_{j=1}^{n_l} \binom{n_l}{n_l - j} 2^{j-1} \left(\frac{\ell - 2}{\ell}\right)^{n_l - j} \left(\frac{1}{\ell}\right)^j, \\ &= \frac{1}{2\ell^{n_l}} [\ell^{n_l} - (\ell - 2)^{n_l}], \\ &= \frac{1}{2} \left[1 - \left(1 - \frac{2}{\ell}\right)^{n_l} \right] \end{aligned} \quad (7)$$

where $n_l = (s + 1)/2$, is the number of leaf nodes in the tree, and $\ell = \chi_t$.

Assuming that primitives are expressed independent of each other, the probability that a order k BB (without loss of generality, we will consider $X_1 X_2 \dots X_k$) is expressed by a tree is given by

$$\begin{aligned} p_{X_{1 \dots k}^{\text{exp}}} &= p(X_i \text{ is expressed})^k, \\ &= \left[\frac{1}{2} \left\{ 1 - \left(1 - \frac{2}{\ell}\right)^{n_l} \right\} \right]^k. \end{aligned} \quad (8)$$

The probability that the BB is not expressed by a tree is then given by

$$\begin{aligned} p_{X_{1 \dots k}^{\text{not exp}}} &= 1 - p_{X_{1 \dots k}^{\text{exp}}}, \\ &= 1 - \left[\frac{1}{2} \left\{ 1 - \left(1 - \frac{2}{\ell}\right)^{n_l} \right\} \right]^k \end{aligned} \quad (9)$$

The probability that a BB is not expressed by any of the n individuals in the population is given by

$$\begin{aligned} p_{X_{1 \dots k}^{\text{exp}=0}} &= \left(p_{X_{1 \dots k}^{\text{not exp}}} \right)^n, \\ &= \left[1 - \left[\frac{1}{2} \left\{ 1 - \left(1 - \frac{2}{\ell}\right)^{n_l} \right\} \right]^k \right]^n \end{aligned} \quad (10)$$

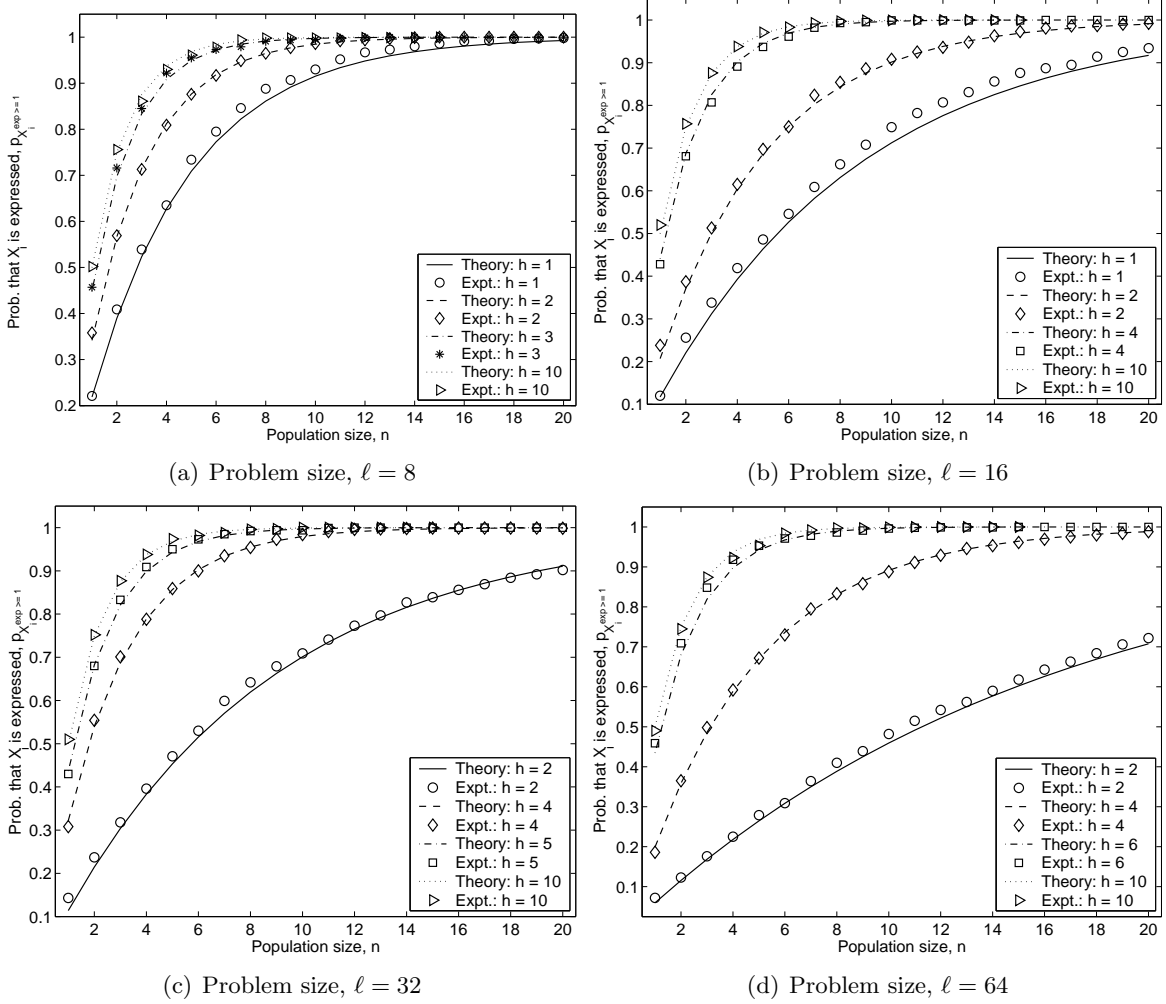


Figure 2: Verification of the facetwise model for a single BB supply (Equation 11) with empirical results for the UNITATION problem for different tree heights, h , as a function of population size, n . The empirical results depict the proportion of runs having at least one copy of a particular schema out of 1000 trials.

Therefore the probability that a order- k BB is expressed by at least one individual in the population is given by

$$\begin{aligned}
 p_{X_{1 \dots k}^{\text{exp}} \geq 1} &= 1 - p_{X_{1 \dots k}^{\text{exp}} = 0}, \\
 &= 1 - \left[1 - \left[\frac{1}{2} \left\{ 1 - \left(1 - \frac{2}{\ell} \right)^{n_l} \right\} \right]^k \right]^n
 \end{aligned} \tag{11}$$

The model for single BB success given by Equation 11 is compared to empirical results for the UNITATION problem ($k = 1$) in Figure 2, and for the DECEPTION problem ($k = 4$) in Figure 3. The empirical results are for full trees, therefore, $n_l = 2^h$. The results show that the empirical results agree with the models.

Using the approximation, $(1 - r/s)^s \approx e^{-r}$, and recognizing that this approximation is suffi-

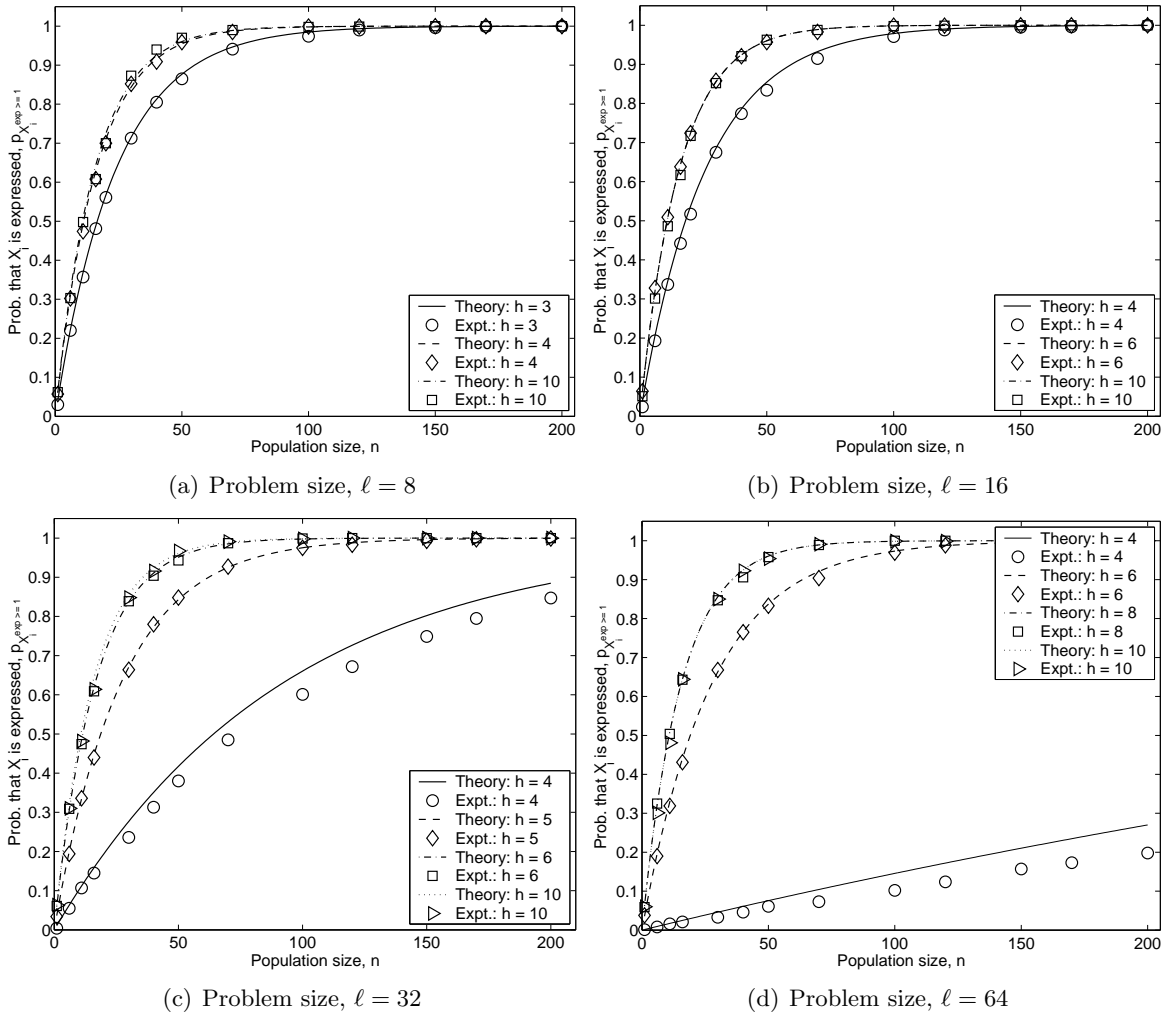


Figure 3: Verification of the facetwise model for a single BB supply (Equation 11) with empirical results for the DECEPTION problem for different tree heights, h , as a function of population size, n . The empirical results depict the proportion of runs having at least one copy of a particular schema out of 1000 trials.

ciently accurate even for modest values of s , we can simplify Equation 11 as follows:

$$p_{X_{1 \dots k}^{exp} \geq 1} \approx 1 - \exp \left[-n 2^{-k} \exp \left\{ -k \exp \left(-\frac{2n_l}{\ell} \right) \right\} \right]. \quad (12)$$

When $n_l \gg \ell$, $p_{X_i^{exp} \geq 1} \approx 1 - \exp(-n 2^{-k})$. In other words, the probability of a BB being expressed by at least one individual, given a population size, increases with the tree size and saturates as $2^h > \ell$, as shown in Figure 4 for UNITATION problem.

5.2 Supply for Partition Success

When solving real-world problems, one does not have prior knowledge about a particular schema being superior to others in a partition. Hence it is necessary to ensure that all competing schemas in a partition are present. The decision process would then be able to consider all the relevant

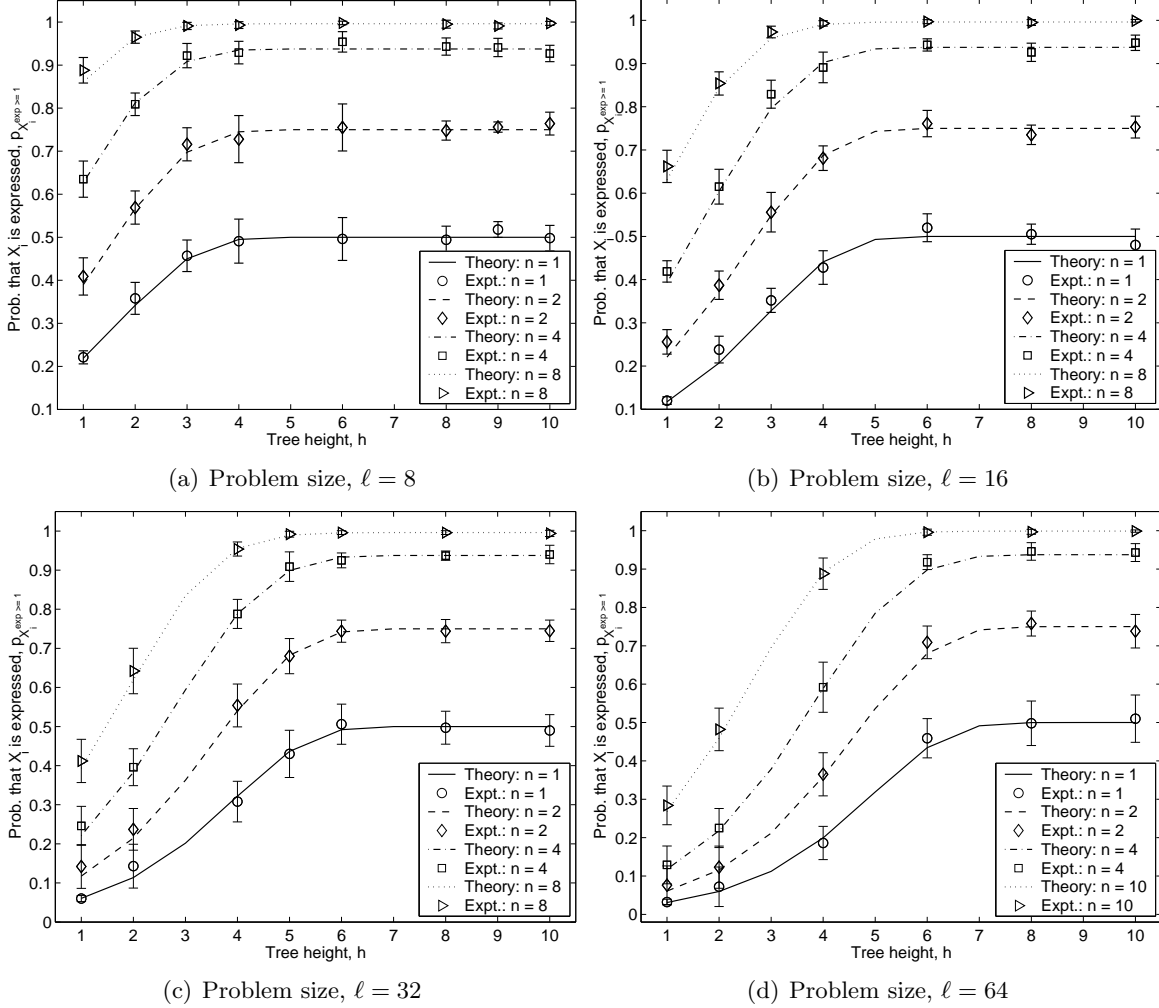


Figure 4: Verification of the facetwise model for a single BB supply (Equation 11) with empirical results for the UNITATION problem for different population size, n , as a function of tree height, h . The empirical results depict the proportion of runs having at least one copy of a particular schema out of 1000 trials.

alternative schemas. Therefore, in this section we extend the model developed in the previous section to ensure the presence of at least one copy of all the competing schemas (both the primitive and its complement) in a partition.

For ORDER, we are interested in the probability that all the 2^k possible schemas are present in the population. Assuming that individual schema success values are independent, the probability for partition success is given by

$$\begin{aligned}
 p_s &= \left(p_{X_{1 \dots k}^{\text{exp}} \geq 1} \right)^{2^k}, \\
 &= \left[1 - \left[1 - \left[\frac{1}{2} \left\{ 1 + \left(1 - \frac{2}{\ell} \right)^{n_l} \right\} \right]^k \right]^{n_l} \right]^{2^k}.
 \end{aligned} \tag{13}$$

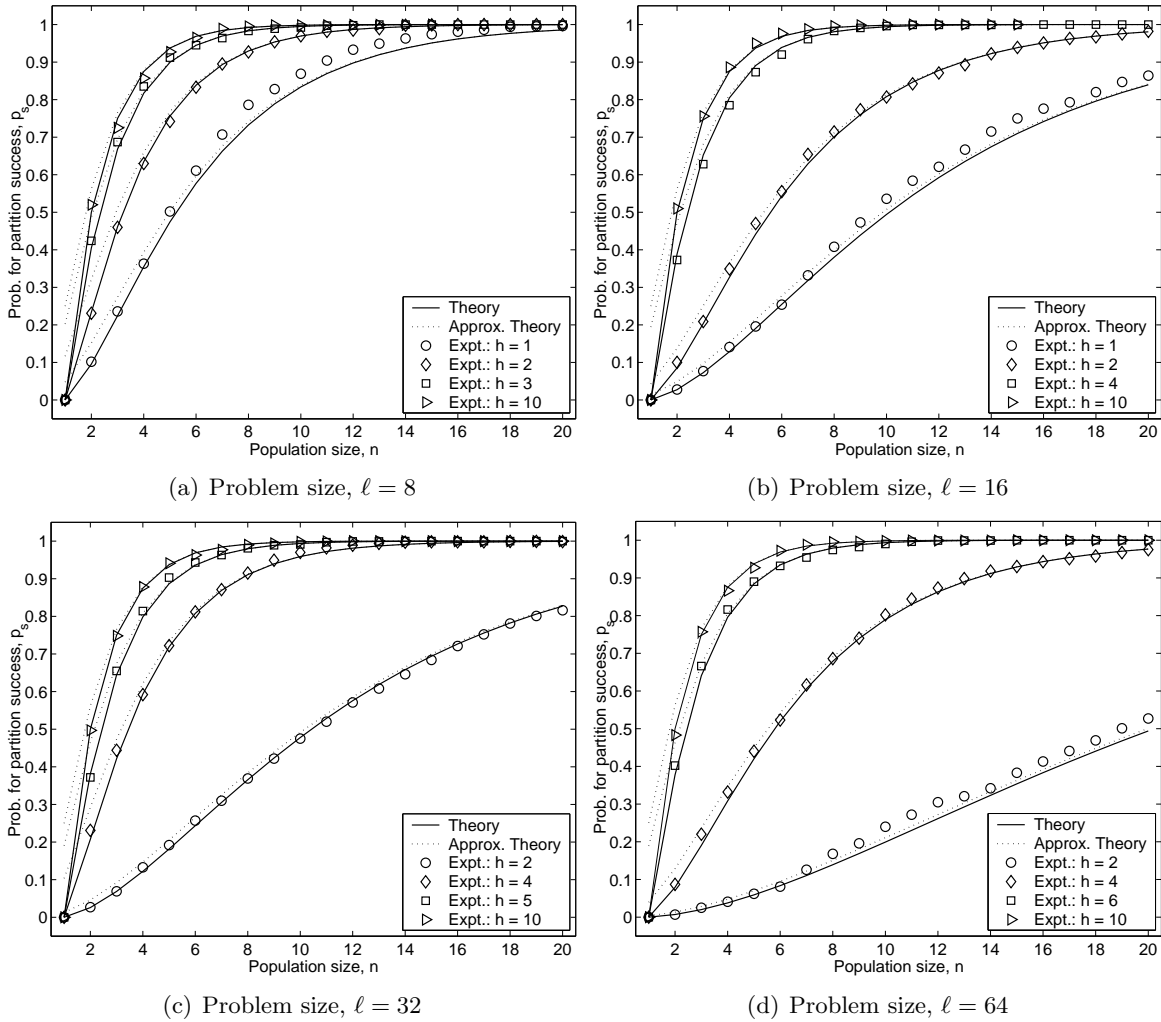


Figure 5: Verification of the models for BB partition success (Equations 16, and 14) with empirical results for UNITATION problem for different tree heights, h , and problem sizes, ℓ , as a function of population size, n . The empirical results depict the proportion of runs having at least one copy of a primitive and its complement in the population out of 1000 trials.

Using the approximation $(1 - r/s)^s \approx e^{-r}$, the above equation can be further approximated as

$$p_s \approx \exp \left[-2^k \exp \left[-n2^{-k} \exp \left\{ -k \exp \left(-\frac{2^{h+1}}{\ell} \right) \right\} \right] \right]. \quad (14)$$

It should be noted that the independence assumption of individual schema success is an approximation and a more exact model can be derived which is illustrated for BB of unit size ($k = 1$).

$$p_s = \sum_{i=2}^n (2^i - 2) \binom{n}{n-i} (p_{X_i^{\text{exp}}})^i (1 - 2p_{X_i^{\text{exp}}})^{n-i}, \quad (15)$$

The above equation can be rearranged as follows:

$$p_s = \sum_{j=0}^{n-2} \binom{n}{j} (2p_{X_i^{\text{exp}}})^{n-j} (1 - 2p_{X_i^{\text{exp}}})^j$$

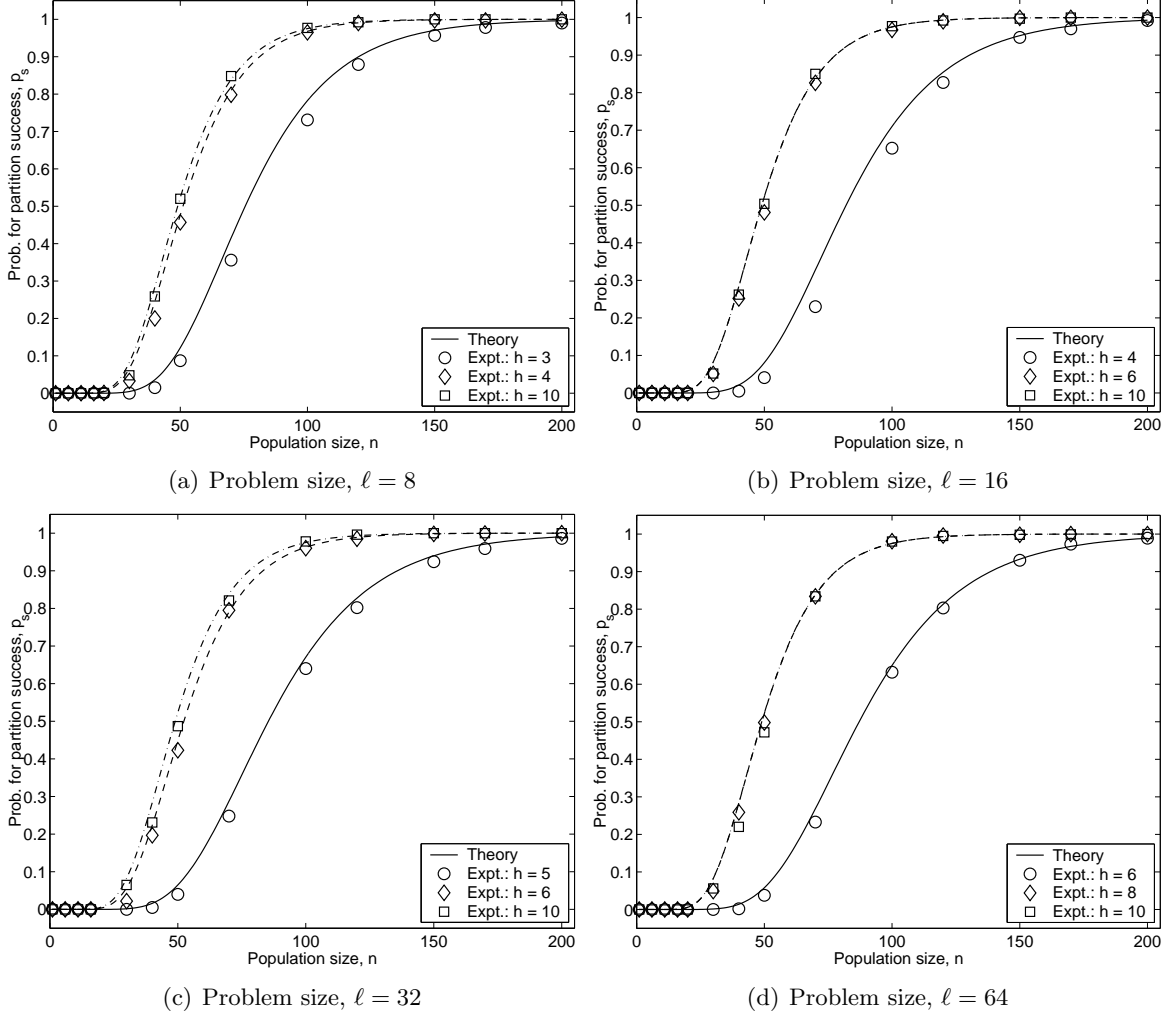


Figure 6: Verification of the BB partition success model (Equation 14) with empirical results for DECEPTION problem for different tree heights, h , and problem sizes, ℓ , as a function of population size, n . The empirical results depict the proportion of runs having at least one copy of a primitive and its complement in the population out of 1000 trials.

$$\begin{aligned}
& - 2 \sum_{j=0}^{n-2} \binom{n}{j} (p_{X_i^{\text{exp}}})^{n-j} (1 - 2p_{X_i^{\text{exp}}})^j, \\
& = 1 + (1 - 2p_{X_i^{\text{exp}}})^n - 2(1 - p_{X_i^{\text{exp}}})^n.
\end{aligned} \tag{16}$$

Equations (16), and (13) are compared with empirical results in Figure 5. The figures show that the approximate model (Equation 13) agrees with Equation 16 for higher population sizes and larger tree sizes. The partition success model (Equation 13) is compared with the empirical results for DECEPTION with $k = 4$, in Figure 6. Both Figures 5, and 6 clearly validate the BB supply model.

5.3 Population Sizing for Building-Block Supply

The facetwise model derived in the previous section will be rearranged in this section to estimate the population size required to ensure the presence of all BBs of a partition for ORDER, given the

problem size is ℓ , and the tree height is h . Assuming that we can tolerate a probability ϵ of not having all BBs in a given partition, and setting p_s to $1 - \epsilon$, we can rewrite Equation 14,

$$1 - \epsilon = \exp \left[-2^k \exp \left[-n2^{-k} \exp \left\{ -k \exp \left(-\frac{2n_l}{\ell} \right) \right\} \right] \right]. \quad (17)$$

Taking logarithms on both sides of the above equation and using the approximation, $\ln(1 - \epsilon) \approx -\epsilon$, we get

$$\epsilon = 2^k \exp \left[-n2^{-k} \exp \left\{ -k \exp \left(-\frac{2^{h+1}}{\ell} \right) \right\} \right]. \quad (18)$$

After taking logarithms on both sides of the above equation and rearranging the resulting equation, we can write

$$n = 2^k (k \ln 2 - \ln \epsilon) \exp \left[-k \exp \left(-\frac{2n_l}{\ell} \right) \right]. \quad (19)$$

If we assume tree size to be big enough ($n_l \gg \ell$), then the above equation can be simplified as $n \approx 2^k (k \ln 2 - \ln \epsilon)$. Furthermore, if we assume that the supply error is inversely proportional to the number of BBs, m , i.e., $\epsilon = 1/m$,

$$n \approx 2^k (k \ln 2 + \ln m). \quad (20)$$

It is interesting to note that the above population-sizing equation for BB supply in DECEPTION is identical to that developed by Goldberg, Sastry, and Latoza (2001) for selectorecombinative GAs.

6 Some Thoughts On Modeling Realistic GP Expressions

The last section developed BB supply models for ORDER expression mechanism and verified it for two test problems for different parameter values. This section provides a brief outline on how to develop BB supply models for realistic GP expressions. First we start by addressing the supply of raw tree fragments, or in other words, we consider that every tree fragment in the tree is *expressed*.

6.1 Tree Fragment Supply

6.1.1 Single BB Success

The probability that a tree does not contain a partition, P_i , is given by

$$p(\#of P_i = 0) = \left(1 - \frac{1}{\kappa}\right)^\phi \quad (21)$$

Recall that the values for κ , and ϕ for different partitions are given in Table 3.2.1. From the above equation, we can write the probability that the population contains at least one copy of the partition, P_i , as

$$p_k = 1 - \left[\left(1 - \frac{1}{\kappa}\right)^\phi \right]^n. \quad (22)$$

Using the approximation, $(1 - r/s)^s \approx e^{-r}$, and recognizing that this approximation is sufficiently accurate even for modest values of s , we can write

$$p_k \approx 1 - \exp \left(-\frac{n\phi}{\kappa} \right). \quad (23)$$

Furthermore, from table 3.2.1, we can see that $\phi \approx 2^{-k}s$, where $k = N_t + N_f$. Substituting this approximation for ϕ in the above equation, we get

$$p_k \approx 1 - \exp\left(-\frac{n \cdot s}{\kappa \cdot 2^k}\right). \quad (24)$$

It should be noted that that the approximation for ϕ is an underestimation for the tree fragments, P_b, P_c, P_e and P_f , and an overestimation for the tree fragments, P_a, P_d , and P_g .

6.1.2 Partition Success

Similar to the previous section, we assume that the schema partition success values are independent. Then the probability of at least one success of each of the κ schemas, p_s is given by $p_s = p_k^\kappa$:

$$p_s = \left[1 - \exp\left(-\frac{n \cdot s}{\kappa \cdot 2^k}\right)\right]^\kappa, \quad (25)$$

$$\approx \exp\left[-\kappa \exp\left(-\frac{n \cdot s}{\kappa \cdot 2^k}\right)\right]. \quad (26)$$

6.1.3 Population Sizing for Partition Success

We now proceed to model the population size required to ensure the presence of all order- k tree fragments. Assuming that we can tolerate a probability ϵ of not having all BBs in a given partition, and setting p_s to $1 - \epsilon$, we can rewrite equation 26,

$$1 - \epsilon = \exp\left[-\kappa \exp\left(-\frac{n \cdot s}{\kappa \cdot 2^k}\right)\right] \quad (27)$$

Taking logarithm on both sides and using the approximation $\ln(1 - \epsilon) \approx -\epsilon$, for small values of ϵ , gives

$$\epsilon = \kappa \exp\left(-\frac{n \cdot s}{\kappa \cdot 2^k}\right) \quad (28)$$

Solving the above equation for n yields

$$n = \frac{1}{2} 2^k \kappa (\log \kappa - \log \epsilon). \quad (29)$$

Recall that $\kappa = \chi_f^{N_f} \chi_t^{N_t}$, and $k = N_f + N_t$. Then we can rewrite the above equation as

$$n = \frac{1}{s} (2\chi_f)^{N_f} (2\chi_t^{N_t}) [N_f \ln \chi_f + N_t \ln \chi_t - \ln \epsilon] \quad (30)$$

This relation can be further simplified if we assume that the supply error is inversely proportional to the number of BBs, m , i.e., $\epsilon = 1/m$. Then the equation may be rewritten as

$$n = \frac{1}{s} (2\chi_f)^{N_f} (2\chi_t^{N_t}) [N_f \ln \chi_f + N_t \ln \chi_t + \ln m] \quad (31)$$

6.2 Incorporating Expression

While counting the tree fragments may be useful enroute with proper expression model as in section 5, on its own it is not realistic. Therefore, we have to compute the combined probability that a tree fragment is present in the population and that it expresses a correct subfunction:

$$p(\text{BB is present}) = p(\text{fragment is present})p(\text{expression}) \quad (32)$$

In the above equation we assume that the events that a tree being present in the population and it being expressed are independent. It should be noted that this assumption becomes more accurate as the population size increases. The probability of a tree fragment being present in the population, $p(\text{fragment is present}) = p_k$, and is given by equation 24, and the expression model is incorporated by the term $p(\text{expression})$. For example, in the symbolic regression example of $1 + x + x^2 + x^3$, the probability of expression incorporates the probability of different tree fragments expressing the linear and constant subfunctions.

7 Conclusions

In this paper, a detailed analysis of building-block supply in the initial population of GP using ORDER expression has been presented. Two facetwise models are derived, one for ensuring the supply of a single schema in a partition, and the other for ensuring the supply of all competing schemas in a partition for problems which employ ORDER expression mechanism. The latter model has been employed to estimate the population size required to ensure the presence of at least one copy of all raw BBs of a partition in the initial population. The population sizing model indicates that there is a minimum tree size dependent on the problem size. Furthermore, the models suggest that when the tree size is greater than the problem size, the population size required on BB supply grounds is $2^k (k \ln \chi + \ln m)$. This study also shows that the population size required to ensure the presence of all instances of tree fragments (assuming that all of them are expressed) is $\frac{1}{s} (2\chi_f)^{N_f} (2\chi_t)^{N_t} [N_f \ln \chi_f + N_t \ln \chi_t + \ln m]$.

Acknowledgments

We thank Martin Martin, Sean Luke, Terry Soule, and Bill Langdon. We also thank Gerulf Pedersen, Ying-Ping Chen, and Tian-Li Yu for their insightful comments and suggestions.

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163 and F49620-03-1-0129, and the National Science Foundation under grant DMI-9908252, and CSE fellowship, UIUC. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the AFOSR, the NSF, or the U.S. Government.

References

- Altenberg, L. (1994). Emergent phenomena in genetic programming. *Evolutionary Programming — Proceedings of the Third Annual Conference*, 233–241.
- Deb, K., & Goldberg, D. E. (1993). Analyzing deception in trap functions. *Foundations of Genetic Algorithms, 2*, 93–108.

- Goldberg, D. E. (1987). Simple genetic algorithms and the minimal, deceptive problem. In Davis, L. (Ed.), *Genetic algorithms and simulated annealing* (Chapter 6, pp. 74–88). Los Altos, CA: Morgan Kaufmann.
- Goldberg, D. E. (1989). Sizing populations for serial and parallel genetic algorithms. *Proceedings of the Third International Conference on Genetic Algorithms*, 70–79.
- Goldberg, D. E. (2002). *The design of innovation: Lessons from and for competent genetic algorithms*. Boston, Mass.: Kluwer Academic Publishers.
- Goldberg, D. E., & O’Reilly, U.-M. (1998). Where does the good stuff go, and why? How contextual semantics influences program structure in simple genetic programming. *Proceedings of the First European Workshop on Genetic Programming (EuroGP’98)*, 16–36.
- Goldberg, D. E., Sastry, K., & Latoza, T. (2001). On the supply of building blocks. *Proceedings of the Genetic and Evolutionary Computation Conference*, 336–342.
- Holland, J. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.
- Koza, J. R. (1992). *Genetic programming: On the programming of computers by natural selection*. Cambridge, MA: MIT Press.
- Langdon, W. B., & Poli, R. (2002). *Foundations of genetic programming*. Springer-Verlag.
- Luke, S. (2000, September). Two fast tree-creation algorithms for genetic programming. *IEEE Transactions on Evolutionary Computation*, 4(3), 274.
- O’Reilly, U.-M., & Goldberg, D. E. (1998). How fitness structure affects subsolution acquisition in genetic programming. *Genetic Programming 1998: Proceedings of the Third Annual Conference*, 269–277.
- O’Reilly, U.-M., & Oppacher, F. (1995). The troubling aspects of a building block hypothesis for genetic programming. *Foundations of Genetic Algorithms*, 3, 73–88.
- Poli, R. (2000). Recursive conditional schema theorem, convergence and population sizing in genetic algorithms. *Foundations of Genetic Algorithms*, 6.
- Reeves, C. (1993). Using genetic algorithms with small populations. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 92–99.
- Rosca, J. P. (1997). Analysis of complexity drift in genetic programming. *Genetic Programming 1997: Proceedings of the Second Annual Conference*, 286–294.
- Stephens, C., & Waelbroeck, H. (1999). Schemata evolution and building blocks. *Evolutionary Computation*, 7(2), 109–124.
- Whigham, P. A. (1995). A schema theorem for context-free grammars. *Proceedings of the 1995 IEEE Conference on Evolutionary Computation*, 1, 178–181.