Chapter 6

CONTINUOUS HIERARCHICAL FAIR COMPETITION MODEL FOR SUSTAINABLE INNOVATION IN GENETIC PROGRAMMING

Jianjun Hu, Erik D. Goodman, Kisung Seo
{Hujianju, goodman, ksseo}@egr.msu.edu
Genetic Algorithm Research & Application Group (GARAGe)
Michigan State University,East Lansing, MI

Abstract: Lack of sustainable search capability of genetic programming has severely constrained its application to more complex problems. A new evolutionary algorithm model named the continuous hierarchical fair competition (CHFC) model is proposed to improve the capability of sustainable innovation for single population genetic programming. It is devised by extracting the fundamental principles underlying sustainable biological and societal processes originally proposed in the multi-population HFC model. The hierarchical elitism, breeding probability distribution and individual distribution control over the whole fitness range enable CHFC to achieve sustainable evolution while enjoying flexible control of an evolutionary search process. Experimental results demonstrate its capability to do robust sustainable search and avoid the aging problem typical in genetic programming.

Key words: Genetic programming, sustainable innovation, HFC, fair competition principle

1. INTRODUCTION

As one of the most natural ways to achieve human competitive intelligence, genetic programming has shown its success in a series of application areas including electric circuit design (Koza, 1999), robot programming, molecular analysis and design (Johnson et al., 2000), etc. Unfortunately, genetic programming is also notorious for its dependence on huge population sizes, its quick convergence after only 50 generations (the aging problem) (Luke, 2001), and the severe bloating phenomenon. Bloating prevents us from achieving solutions of the necessary complexity to solve many engineering problems. The aging problem (early death of evolutionary progress) makes it hard to achieve superior solutions even when more computing power is available. These two issues are related to the scalability of genetic programming. The lack of robustness of genetic programming search makes it not reliable enough to be used as a desktop tool for ordinary design engineers. An ideal genetic programming system should have characteristics such as robustness (reliably achieving good performance), sustainability (getting better performance with more computing effort), and scalability (evolving solutions with higher complexity when needed).

The lack of sustainable search capability of traditional genetic programming comes from two fundamental sources (Hu, Goodman et al., 2003). One is the convergent nature of conventional EA models. The other is the bloating phenomenon caused by the variable length representation and its operators. When compared with the natural process of evolution, which does not normally face the issues of stagnation, bloating and lack of sustainability, two significant differences can be observed. One is that, in natural evolution, an enormous population size and diversity of environment is typically involved, while we can only use much smaller population sizes in artificial evolution, and typically in a much less diverse environment. Another major difference is that in biological evolution, evolution happens at all fitness levels, from the primitive single-cell organism to high-level mammals. This kind of simultaneous multi-level (often corresponding to fitness level) evolution in a multitude of diverse environments may contribute to the sustainability of the biological process. A similar sustainable innovation also happens in human society, in such settings as educational systems. By educating students at all academic levels simultaneously and continuously, better and better students are trained to achieve increasing success. Although the population of students at one instant of time is of finite size, the unlimited timeframe and the continual importing of new students from kindergarten provide a non-depletable source for continuing possible innovation in the school system. In contrast, conventional GA/GP effectively terminate lower-level innovation early, as the average fitness of the population increases. That is, the probability that good building blocks contained in new randomly generated individuals become incorporated in higher-fitness individuals declines rapidly as the average fitness of the population increases.

In education systems, since the competition among students is segregated into different academic levels, the success of high-academic-level students (i.e., those in higher grades) never threatens the survival of students in lower grades, so the students in lower grades have a chance to grow and compete in an environment that is "fair" to them. More importantly, the lower-level knowledge framework assembled in the lower grades supports the possibility of more advanced knowledge frameworks in the higher levels. In terms of genetic programming, the early evolution stage usually establishes a basic framework for later evolution, similar to the speciation process in biological evolution. It is already well established that in tree-based genetic programming, after a certain number of generations, the structure around the root of the genetic programming tree is largely fixed.

Inspired by the sustainable evolution principle in biological and social systems, we proposed the hierarchical fair competition model (HFC) to achieve sustainable evolution for genetic programming (Hu & Goodman, 2002; Hu, Goodman et al., 2002). The continuing search capability of HFC is achieved by ensuring a continuous supply and incorporation of low-level building blocks and by culturing and maintaining building blocks of intermediate levels. HFC employs an assembly-line structure in which subpopulations are hierarchically organized into different fitness levels, reducing the selection pressure within each subpopulation while maintaining the global selection pressure to help ensure exploitation of good building blocks found.

In this paper, we extract the basic ideas of the fair competition principle of HFC (Hu & Goodman, 2002) and apply it to single population evolution, which makes it more widely and readily usable in many existing GP/EA packages. Briefly, we introduce three mechanisms, called a) hierarchical elitism, b) explicit control of the breeding probability distribution over the fitness range, and c) explicit control of the distribution of the individuals over the fitness range, in order to achieve sustainable evolution for single population genetic programming and other EAs. We investigate the robustness and the effectiveness of this continuous HFC and how it helps to ameliorate the classical premature convergence problem in genetic programming. It is demonstrated that, similar to the discrete version of HFC in which the fitness range is segmented into discrete levels, the continuous HFC model (CHFC),with its explicit level maintenance mechanism, can also achieve robust and sustainable search as expected, on the problems explored.

2. SUMMARY OF HFC PRINCIPLE AND HOW IT IS POSTULATED TO HELP GP TO AVOID CLASSICAL PROBLEMS

The HFC concept postulates that an important factor in the the convergent nature of conventional EAs is their over-attention to the high-fitness individuals and neglecting of the importance of recombination and mutation among low-fitness individuals, even at later stages of evolution. The HFC concept is to sustain a supply of low-fitness individuals that represent low levels of "organization" – possessing low-order building blocks that may even be created at random, but not yet (or still) be present in higher-level individuals. These individuals are "sheltered" – allowed to recombine and mutate, regardless of the fact that higher-fitness individuals are present. The goal is not to preserve the low-fitness offspring of bad matings among high-fitness individuals, so hierarchical elitism is employed to prevent that.

High-fitness individuals in conventional EAs usually have difficulty in modifying their basic frameworks without decreasing their fitnesses dramatically, especially in the case of genetic programming. As the average absolute fitness of the population goes up, it becomes increasingly difficult for the degraded offspring of high-fitness parents to survive. Th individuals that do change their basic frameworks just cannot survive with their low fitnesses. So, the probability of framework changes becomes increasingly small as evolution goes on. In addition, the frameworks of current high-fitness individuals are established with very limited testing during the early stages of evolution, which does not guarantee their global superiority. After the potential of these earliest-discovered frameworks is exhausted, no more framework innovation is likely, and genetic programming begins to exhibit stagnation.

This loss-of-exploratory-capability (LEC) hypothesis to explain premature convergence can be understood more easily in genetic programming, where the early-stage evolution usually establishes the framework (or topological structure) of the individuals. However, LEC is also applicable to many GA problems with high epistasis, where disruption of the strongly coupled components of the genome usually leads to dramatic decrease of the fitness of the offspring. It is not the fitness diversity per se that contributes to the sustainable search capability of HFC or any other successful approach to sustainable evolution. In problems amenable to HFC, fitness levels are assumed to correspond to degrees of organization, or coupling among components, from unbiased random individuals to highly coupled, high-fitness individuals.

Sharing some similarity with HFC, Hutter (2002) proposed a Fitness Uniform Selection Scheme (FUSS) to preserve genetic diversity, which, however, is not the motivation of HFC. Compared to the CHFC proposed in this paper, FUSS lacks the explicit control of the breeding probability of all fitness levels and suffers from insufficient selection pressure to exploit high-fitness individuals. FUSS also does not have an explicit mechanism for allocating individuals to different fitness levels and thus easily suffers from unbalanced distribution of individuals. In summary, FUSS is still more like traditional diversity-oriented schemes while the CHFC proposed here is aimed at achieving sustainable search through separate control of the allocation of individuals to fitness levels and selection pressure.

The HFC mechanism aimed at avoiding loss of explorative capability are readily distinguished from many other diversity-oriented techniques. Their lack of sustainability can be attributed to their inability to make large framework modifications of high-fitness individuals at a later evolution stage, due to the fact that they cannot ensure sufficient survival time for low-fitness offspring of high-fitness parents with large modifications of their frameworks. Low fitness individuals face increasing selection pressure as the result of the constant increase of the average fitness of the population. So in the long run, large modifications of the frameworks of high-fitness individuals become increasingly unlikely.

Similarity (either genotypic or phenotypic)-based fitness sharing, crowding and related methods (Goldberg &Richardson, 1987; De Jong, 1975; Li et al., 2002;) tend to spread the individuals to the most prominent peaks, while in fitness sharing, the number of individuals allocated to each peak is proportionate to the peak fitness. First, in both methods, the average fitness of the population is constantly increasing. Second, since more and more higher peaks are identified, the number of individuals allocated to the lower-fitness peaks gradually decreases to almost 0 because of the limited population size. So after a sufficiently long evolution, low-fitness offspring generated via significant modification of the parent individual(s) can't typically survive long enough to be tuned into high-fitness individuals. There are many schemes that can achieve a high degree of fitness diversity, sharing some similarity with HFC – however, maintenance of fitness diversity is not the motivation of HFC. These schemes include Boltzmann-weighted selection (Maza & Tidor, 1991), Boltzmann tournament selection (Goldberg, 1990) and probabilistic crowding (Mengshoel & Goldberg, 1999). First, with all these methods, the average fitness of the population increases constantly. For the Boltzmann schemes, while any low-fitness individuals can survive initially when the temperature is high, the fitness range of surviving individuals becomes increasingly narrow due to the decrease of temperature.

The center of the Boltzmann distribution of individuals over fitness also moves constantly away from lower-fitness toward higher-fitness. As a result, after the average fitness increases to a certain degree and the temperature is reduced to some level, low-fitness individuals with significant modifications cannot survive long enough to allow framework innovation. Similar analysis also applies to probabilistic crowding. This behavior is not entirely unwelcome, of course – in many cases, there is only a known amount of exploratory effort available to solve a problem, and a method that is tuned to exploiting that amount of effort to maximum effect is highly desirable. In contrast, a fixed CHFC strategy, without adaptation of its distributions, does not behave in that manner.

HFC addresses the above loss of exploratory capability problem by explicitly enforcing simultaneous evolution at all fitness levels. Instead of depleting the potentials of the basic frameworks of individuals established in early stages of evolution, the continuing innovation at the low-fitness level provides a constant stream of new frameworks for higher-level exploitation. These bottom-up moving individuals can survive the competition they face (thus allowing sufficient assembly of building blocks) due to the maintenance of fair competition: only individuals with similar performance levels are allowed to compete. From the perspective of avoiding local optima, instead of depending on jumping out of the local basin of attraction occupied by the highly converged population, which is often very improbable, HFC deals with trapping in local basins of attraction by allowing new individuals in other basins of attraction to emerge continuously. When examined in terms of search effort allocation, the competition among individuals is not constrained among the individuals of the current population. Instead, the allocation of computing resources takes a much stronger global view by ensuring that the assembly of building blocks happens at all absolute fitness levels. So the unit of allocation of search effort is not the individuals of the population, but the fitness levels of the whole possible fitness range.

Several forms of multi-population HFC have already been introduced (Hu & Goodman, 2002; Hu, Goodman et al., 2002, Hu, Seo et al., 2003). In those models, the number of fitness levels is specified by the user or may be determined adaptively. The major goals of the HFC framework can be summarized as follows:

a) To ensure fair competition among individuals by segregating competition among different levels of individuals,

   In this way, high-fitness individuals won't threaten the existence of those with lower-level fitness, as in most EAs. Extensive experimental study in Burke et al. (2002) suggested that a phenotypic diversity measure is superior to genotypic ones in predicting the run performance. This suggests that the straightforward and thorough phenotype-based protection of “inferior individuals” by HFC may be more reliable than other techniques such as fitness sharing or the species conserving algorithm SCGA (Li, et al., 2002), in the sense that it doesn’t require a distance measure which is hard to find in genetic programming, and it supports the continuing search for new frameworks at low-levels.

b) To assure that exploration and exploitation occur at all fitness levels

Allocating computing cycles to demes of all fitness levels allows innovation to occur all the way from embryonic primitive individuals to highly evolved individuals. By cascading higher-level beyond lower-level fitness demes, a building block assembly line is established. In analogy to the Cambrian Age, HFC essentially keeps the emergence of new “species” operating, thereby enhancing the capability for sustainable evolution in genetic programming. It also greatly relieves the conventional demand for large population sizes to achieve good performance, as will be shown in the following experimental section.

c) To provide hierarchical elitism (one-way elitism)

In the multi-deme HFC, when the fitness of an offspring is lower than the fitness admission threshold of the current HFC level, the offspring will be discarded. Otherwise, it continues to stay in the current fitness level or is exported to a higher level, if appropriate. This appears as a one-way migration from lower-level demes to higher-level demes. In the context of genetic programming, since higher-fitness individuals usually have higher complexity, the elitism here can effectively remove those individuals which have unnecessary complexity for their fitness level. This mechanism may help to control the bloating phenomenon in genetic programming.

d) To incorporate new genetic material continually to provide a constant influx of evolutionary potential

A random individual generator is configured at the bottom of the HFC deme hierarchy to provide inflow of unbiased genetic material to higher fitness levels. From the bottom level to the highest fitness level, this convergent evolutionary process will never deplete its evolutionary potential. Instead, it provides a mechanism to allow innovation to happen continually at all fitness levels.

Previous work has demonstrated the capability of the HFC model for robust and continuing search capability in a series of genetic programming benchmark problems and a complex real-world circuit synthesis problem (Hu &Goodman, 2002). HFC turns out to be able to get better results more quickly and more robustly for these difficult multimodal problems in which premature convergence is of great concern. By continuing innovation, HFC achieves more robust search, which is not as sensitive to the population size as is conventional genetic programming. It gets better results by its inherent capability to avoid premature convergence.

In the original formulation of the HFC model, the whole fitness range is divided into discrete segments, each with a separate deme. It has the advantage of simplicity and efficiency. But the division is somehow arbitrary. Subsequently, the adaptive HFC model (Hu, Goodman et al., 2002) demonstrated that it is not necessary to find a set of exact admission thresholds to allow fair competition since the performance is not very sensitive to the admission thresholds.

An extension of this idea is to completely remove the discrete segmentation of the demes and to use a single population to do hierarchical fair competition by introducing some special mechanisms to implement the HFC principle. This makes HFC easier to incorporate into many existing EA packages. As a complementary element, the continuous single-population HFC can also be applied to each subpopulation of the original HFC model, which is a naturally parallel model, or to any other parallel evolutionary algorithm.

3. CONTINUOUS SINGLE-POPULATION HFC MODEL (CHFC) FOR SUSTAINABLE EVOLUTION

The CHFC EA model is composed of a set of components as illustrated in Fig. 6-1, including an HFC archive population, a "workshop" population, a random individual generator, a parent selector, a replacement selector, a breeding probability distribution control mechanism, and a removal control mechanism with a density estimator (determining what individuals, of what fitnesses, should be removed when new individuals enter the archive population at their specified fitness levels). In the next section, we first give an overview of the algorithm, and then introduce details of each component.

The Continuous HFC (CHFC) Algorithm

(1) Determine the following parameters besides other GP parameters:

HFC archive size, N_h

workshop size, N_w

Parameters for GP mutation and crossover operators

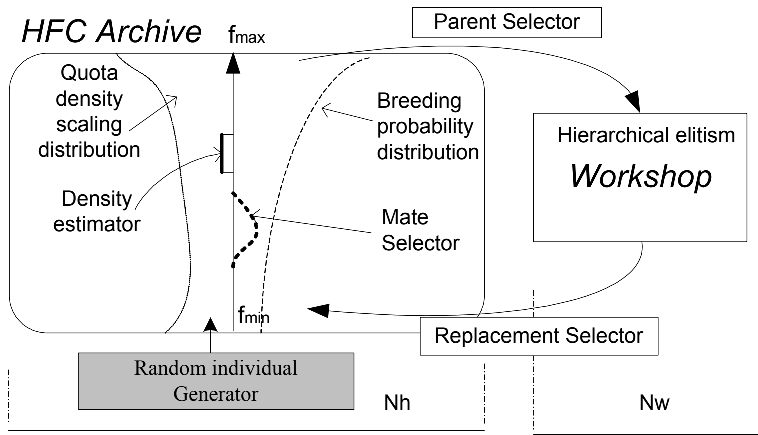

Figure 6-1. Structure of the CHFC model. Nh is the population size of the HFC archive. Nw is the Workshop size

Breeding probability distribution and its parameter (here, based on an exponential distribution)

Density quota scaling distribution and its parameter (here we based it on an exponential distribution with the parameter λ and the maximum ratio K_m)

Pair selection method and its parameters (here, a random neighbor method with $p_{neighborhood} = 0.1$)

(2) Initialize the HFC archive randomly, finding the maximum and minimum fitnesses. Sort the individuals in the population by ascending fitness. The workshop deme is initially empty.

(3)Loop until termination

– According to breeding probability distribution $p_b(x)$, generate a value τ in [0, 1].

– Compute the target sampling fitness $f_t = f_{\min} + \tau\left(f_{\max} - f_{\min}\right)$ according the GP operator probabilities, decide on a current operation

If current operation is mutation, select an individual with index I in the population whose fitness is nearest to the sampling fitness; if current operation is crossover, select one parent as for mutation, then use the pair selector to select the second individual. Create one (if mutation) or two (if crossover) offspring and evaluate their fitnesses. If an offspring has a fitness greater than one of the parents, save it in the workshop deme; else discard this offspring.

Check if the workshop deme is full. If full, update the HFC archive with the new offspring in the workshop deme; else continue at top of loop.

HFC Archive Update Procedure

Calculate the normalized density of each individual in the HFC archive according to Eq. 3.8 below. For each individual in the workshop deme, place it in *the HFC archive according to its fitness. Also, determine the individual with largest normalized density value in the archive, the "victim," and remove it. The exception is that if the victim is among the top K_e (elite group size) fitness genotypes present in the population, it is kept, and another victim is removed.*

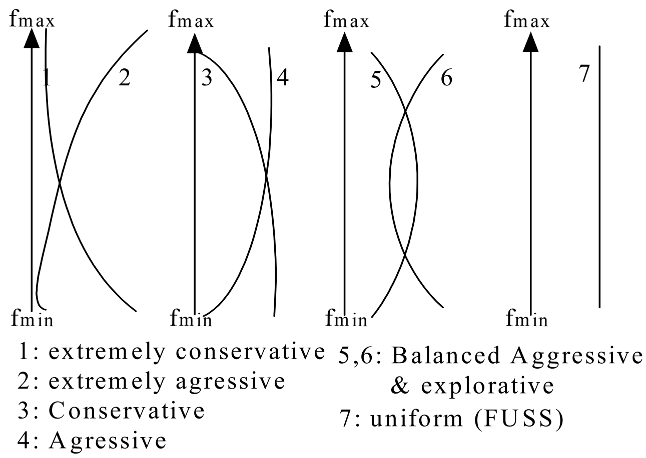

Figure 6-2. Seven types of distribution of the breeding probability of all fitness levels

Explanation of the Continuous HFC model

The continuous HFC model is composed of the following components as shown in Fig. 6-1:

a) Two populations (only one of which is persistent) for storage of individuals. One of them is the HFC "archive" used to accommodate the breeding candidates, and works like a "normal population" in an EA. The other, "workshop," deme is used to hold temporarily the offspring generated during one generation. By varying the size of workshop, we can control the generation gap to be any desired value.

b) A random individual generator that continuously inserts some unbiased random genetic material into the HFC archive. It is the source of the fundamental genetic diversity.

c) A parent selector that is used to select breeding parents according to the breeding probability distribution for recombination or mutation. The breeding probability is introduced to ensure sufficient selection pressure to exploit early-discovered high-fitness individuals, analogous to the allocation of larger subpopulation sizes to higher HFC levels (Hu & Goodman, 2002). Rather than using the fitness uniform selection operator (Hutter, 2002), a higher probability is typically allocated to higher fitness individuals, although that is under user control. However, these higher fitness individuals do not dominate the population as in a conventional EA, because of the individual allocation control mechanism used to prevent uncontrolled growth of high-fitness individuals.

The breeding probability distribution can be linear, exponential or any other distribution; typically, one is chosen that favors higher fitness individuals for breeding. Fig. 6-2 lists some typical distributions; Fitness Uniform Selection (Hutter, 2002) uses the last, uniform, distribution.

In our experiments, the following (truncated, reversed exponential) distribution was used:

$$p_b(x) = \frac{\lambda e^{-\lambda(1-x)}}{1-e^{-\lambda}} \quad for \quad x \in [0, 1] \qquad (3.1)$$

The parameter λ is a parameter to control the selection pressure or the aggressiveness of exploitation vs. exploration.

First a random value *r* between [0, 1] is generated according to the distribution $p_b(x)$; then the target sampling fitness is calculated as

$$f_t = f_{\min} + r * (f_{\max} - f_{\min}) \qquad (3.2)$$

After determining f_t, the individual *I* whose fitness is closest to f_t is determined.

The pair selection method for crossover can be implemented in the following ways:

According to the fair competition principle, the second selection is dependent on the first one and should be selected from the vicinity of the first selected individual. This is called the *Random Neighbor* approach. There are two ways to define the neighborhood of an individual (using the implicit level concept of HFC): 1) an individual (mate) with an index within *K/2* steps (offset of index number in the sorted population) away from the first individual may be randomly selected,

$$[I - \frac{K}{2} * \lambda(f_t),\ I + \frac{K}{2} * \lambda(f_t)] \qquad (3.3)$$

where K is neighborhood size, $\lambda(f_t)$ is a scaling parameter to allow different neighborhood sizes in different fitness levels. (In this paper, $\lambda(f_t)$ is set to 1 for simplicity.)

Or, 2) the entire fitness range may be divided into *L* levels, and the second individual (mate) may be chosen from the set of individuals with fitnesses in the range

$$[f_t - \frac{f_{\max} - f_{\min}}{2L} * \lambda(f_t),\ f_t + \frac{f_{\max} - f_{\min}}{2L} * \lambda(f_t)], \qquad (3.4)$$

using a percentage of the HFC archive size to decide the L value

$$L = p_{neighborsize} * N_h \tag{3.5}$$

where N_h is the HFC archive size.

We have also tested the other two approaches used in FUSS (Hutter 2002) (to be reported elsewhere), the *Next Nearest* approach, where the second individual is the one whose fitness is second closest to the sampling fitness, and *Independent FUSS*, where we select the second individual independently, just as we selected the first individual.

d) A replacement selector, which decides which individuals in the HFC archive are to be replaced by the new offspring. This is based on the density estimator and the quota density scaling distribution.

e) A density estimator, which is used to estimate the density around each individual, with additional scaling by the quota density scaling distribution. By adjusting the scaling distribution, we can achieve any desired proportion of individuals at the various fitness levels. Here, the relative density concept is used. That is, we evaluate by what percentage the current density surpasses its nominal (quota) density at any point on the quota density scaling distribution. The individual with the highest percentage becomes the next individual to be replaced.

The relative density in the neighborhood of an individual can be estimated with either of two approaches:

Approach 1: Select two individuals from the sorted (ascending fitness) HFC archive with indices within K/2 steps of current individual I. The raw density is estimated as

$$D_{raw}(I) = f(I + \frac{K}{2}) - f(I - \frac{K}{2}) \tag{3.6}$$

where $D_{raw}(I)$ is called the raw fitness density of an individual (the implicit denominator, K, is constant, so is ignored).

The raw nominal (quota) density of an individual $D_{rawq}(I)$ is computed as:

$$D_{rawq} = \frac{K}{N_h} * (f_{\max} - f_{\min}) \tag{3.7}$$

where N_h is the population size of the HFC archive. Finally, the normalized density is computed as the relative density:

$$D_{norm} = \frac{D_{raw}(I) - D_{rawq} * r_d}{D_{rawq} * r_d} \tag{3.8}$$

where r_d is a scaling parameter to be described below.

Approach 2: Assume the whole fitness range is divided into *L* levels, and count the number of individuals (M) whose fitnesses are in the range

$$[f(I) - \frac{f_{\max} - f_{\min}}{2L},\ f(I) + \frac{f_{\max} - f_{\min}}{2L}] \tag{3.9}$$

The raw density of an individual I is $D_{raw}(I) = M$
The raw nominal quota density is:

$$D_{rawq} = \frac{N_h}{L} \tag{3.10}$$

where N_h is population size of the HFC archive
The normalized density is then calculated as (3.8)

We use a percentage of the HFC archive size to decide the above L, where $L = p_{neighborsize} * N_h$, and N_h is the HFC archive size.

The parameter r_d in (3.8) is used to bias the allocation of individuals over different fitness levels. It is generated according to a user-specified quota density scaling distribution, which determines the relative size of high-fitness individuals and low-fitness ones. It is important to maintain sufficient high-fitness individuals for effective search. For different problems, this distribution can also have different types, as illustrated in Fig. 6-2. In our experiments, the following distribution was used.

$$r_d(x) = \frac{a + \lambda e^{-\lambda x}}{a + 1 - e^{-\lambda}} \quad where \ \ a = \frac{(e^{-\lambda} - K_m)}{K_m - 1}\lambda \tag{3.11}$$

where λ is used to control the shape of the exponential distribution. K_m is a user-specified maximum ratio of $r_d(1)/r_d(0)$. It is used to control the maximum ratio of individuals that are allowed between the highest fitness level and the lowest fitness level.

f) Hierarchical elitism is enforced during the insertion of offspring into the workshop. When the fitness of the offspring is less than that of both parents, the offspring is discarded (not used to update the HFC archive).

4. EXPERIMENTS AND OBSERVATIONS

The basic motivation of HFC is to ensure sustainable search by avoiding premature convergence. It is most useful in the case of difficult problems where local optima are of great concern. Here the Santa Fe trail artificial ant problem (Koza, 1994) is used to illustrate how CHFC can improve the performance of genetic programming, dealing with the bloating problem and exerting control over the fitness distribution of the population.

The artificial ant problem involves evolving a sequence of movements in a two-dimensional grid to collect as much food (distributed on the grid) as possible. We used the Santa Fe Trail, with a size of 32*32. The maximum amount of food is 89. We defined the fitness as

$$fitness = \frac{1}{1 + \frac{\text{max food} - \text{collected food}}{\text{maxfood}}} \qquad (3.12)$$

We compared the performance of CHFC with both steady-state GP and generational GP. The parameters are listed in Table 6-1.

Table 6-1. Parameters Used in the Various GP Experiments on Santa Fe Trail

Common GP Parameters	Additional CHFC parameter
Function and terminal set	workshop size: 20
{Forward, Right, Left, Prog2, Prog3, IfFoodAhead}	Breeding probability distribution type: exponential (lambda=4.0)
Max evaluation: 100000	
Population size: 1000	Density scaling distribution:
Tournament size: 2 (for steady-state and generational GP)	exponential (lambda=4.0, maxRatio=10)
Crossover rate: 0.9	Elite group size K_e =1
Shrink mutation: 0.05	Probability of introducing random individuals each generation: 0.2
Point mutation: 0.05	
Maxtree depth: 17 mutate maxdepth: 5 Init max depth: 6 Init min depth: 2	Percentage generated randomly in that event: 0.05

Twenty experiments were run for each algorithm. The experimental results are summarized in Table 6-2.

Table 6-2. Peformance comparison of CHFC and conventional GP in Santa Fe Trail artificial ant problem

Algorithm	Best mean fitness	Std. dev.
CHFC-GP	0.92	0.05
Steady state GP	0.76	0.073
Generational GP	0.81	0.089

To gain insight into the evolutionary process, we drew four histograms showing the progression of the fitness distribution of the population during the run. (Since steady-state GP has similar behavior, only generational GP is compared to CHFC-GP). Fig. 6-3 and 6-4 show the fitness distributions of generational-GP and CHFC-GP for the ant problem. It is clear that conventional genetic programming suffers a lot from the domination of early-discovered high-fitness individuals while CHFC, through its controlling mechanism based on fair competition, can always maintain a balanced distribution of both high-fitness individuals for exploitation and low-fitness individuals for "Cambrian innovation"—the innovation of the basic framework.

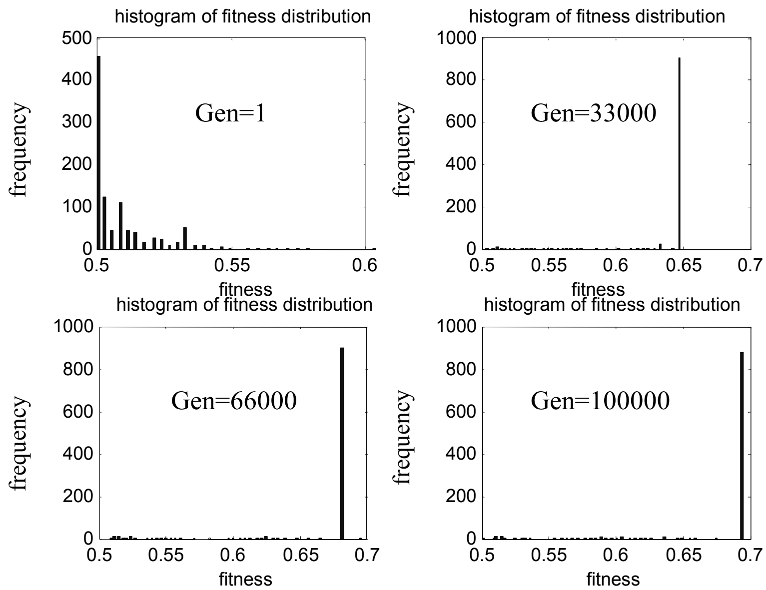

Figure 6-3. The convergent nature of conventional generational GP. The high-fitness individuals rapidly dominate the population of this typical run. The subplots were prepared by sampling at evaluation numbers 1, 33000, 66000, and 100000.

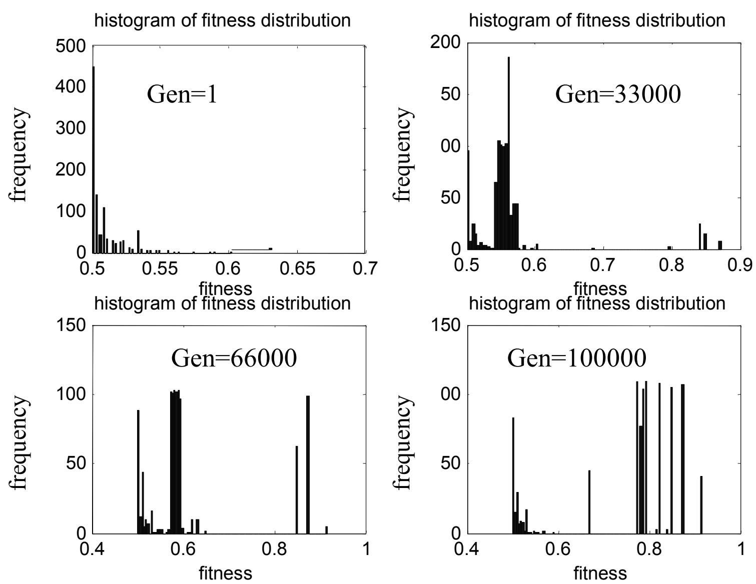

Figure 6-4. Thanks to the breeding probability control and the density control mechanism, CHFC can maintain a balanced distribution of high-fitness and low fitness individuals without any risk of domination in this typical r un, evidently without sacrificing its exploration of the high-fitness range. Subplots are prepared at evaluation numbers 1, 33000, 66000, 100000

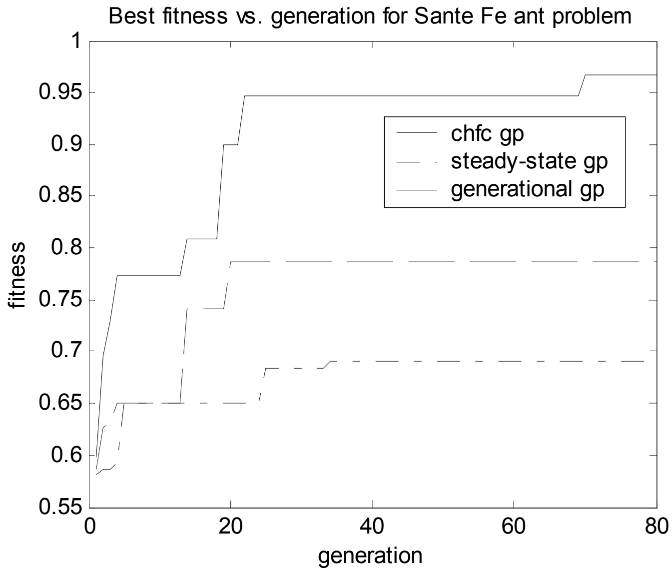

Figure 6-5. Comparison of CHFC with steady-state and generational GP. Having avoided the local optima that trapped the other runs, CHFC makes consistent progress and discovers improved solutions.

Fig. 6-5 plots the fitness growth process of the three algorithms. It is impressive that CHFC, by avoiding premature convergence, continues to generate innovative (and superior) solutions when the steady-state and generational GP have stagnated.

5. CONCLUSIONS AND FUTURE WORK

By extracting the key ideas of the HFC model and combining some ideas from the fitness uniform selection (FUSS) model, this paper proposes the CHFC model for sustainable evolution with single population EAs. CHFC is characterized by its hierarchical elitism to avoid degradation of individuals, explicit control of the distribution of individuals over the fitness range to ensure a continuing supply of building blocks of different orders, and explicit control of breeding probability distribution over the fitness range to enforce strong selection pressure for exploiting early-discovered high-fitness individuals, while lessening the risk of domination and premature convergence. Indeed, CHFC is not a convergent method at all – it seeks to find high-fitness individuals without ever "pulling up the ladder" of building blocks that descends all the way to low-fitness, randomly-generated individuals. CHFC's strength appears to be its providing of separate control of selection pressure from the specification of a desired distribution of individuals across the fitness landscape.

The well-known artificial ant genetic programming benchmark problem is used to investigate the robustness and continuing search capability of CHFC-GP. CHFC provides a simple approach to ensure sustainable and robust search in this genetic programming problem. It obtained better results more robustly and more quickly while avoiding convergence, providing sustainable search. Along with the original HFC model, for which sustainability of evolution has been demonstrated on several more difficult real-world synthesis problems, CHFC adds further evidence that the HFC EA model is able to transform conventional EAs from a convergent framework into a sustainable one.

Our Loss-of-Exploratory Capability (LEC) hypothesis provides an interesting perspective on the aging phenomenon in genetic programming. "50 generations of innovation" seems to come from the inherent convergent nature of the conventional EA framework coupled with a common property in genetic programming: large structural modification by crossover or mutation usually leads to low-fitness individuals that rarely survive in later stages of evolution. HFC algorithms devised on the basis of this theoretical understanding have allowed us to use much smaller population sizes to achieve desired solution qualities on a variety of problems, while enjoying much stronger robustness (Hu, Goodman *et al.,* 2003). The separate control of selection pressure and the individual distribution over fitness range proposed in this CHFC model may provide practitioners with much more flexible control over evolutionary search: one can be aggressive without risk of premature convergence.

Experiments have shown that there are cases in which the fitness distribution is hard to control with the current methods. A better density control mechanism needs to be devised. It will also be interesting to investigate the theoretical properties of CHFC and to seek to eliminate or routinize the setting of many of its parameters. In addition, hybridizing CHFC with the original multi-population HFC may provide a good model for large-scale parallel GP implementation. A topic for future consideration is how fitness sharing can be applied to each fitness level and whether there is any kind of synergy that can thereby be exploited.

Acknowledgement:

This work is developed based on the Open beagle GP C++ framework. The authors sincerely thank its author Christian Gagne for lasting and patient help. We thank David E. Goldberg, Joseph A. Driscoll, Bill Worzel, and Duncan MacLean for their careful and insightful suggestions in their reviews. This work is sponsored by the National Science Foundation under contract DMI 0084934

REFERENCES

De Jong K. A. (1975). An Analysis of the Behavior of a Class of Genetic Adaptive Systems Doctoral Dissertation, University of Michigan *Dissertation Abstracts Int.* 36 5140B (University Microfilms 76-9381)

E. Burke; S. Gustafson and G. Kendall. (2002). A Survey and Analysis of Diversity Measures in Genetic Programming. In Genetic and Evolutionary Computation Conference, New York, July.

Goldberg, D.E. (1990). A not on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. Complex Systems, 4, 445-460.

Goldberg D. E. and Richardson J. (1987). Genetic algorithms with sharing for multimodal function optimization. In Proc. 2nd Int. Conf. on Genetic Algorithms (Cambridge, MA) ed J J Grefenstette. pp 41–9.

Hu, J. and Goodman, E. D. (2002). Hierarchical Fair Competition Model for Parallel Evolutionary Algorithms. In *Proceedings, Congress on Evolutionary Computation*, CEC 2002, IEEE World Congress on Computational Intelligence, Honolulu, Hawaii.

Hu, J.; Goodman, E. D.; Seo, K.; Pei, M. (2002). Adaptive Hierar-chical Fair Competition (AHFC) Model for Parallel Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*: 772-779, GECCO-2002, New York.

Hu, J.; Goodman, E.D.; Seo, K.; Fan, Z.; Rosenberg, R.C. (2003). HFC: a Continuing EA Framework for Scalable Evolutionary Synthesis. In *Proceedings of the AAAI-2003 Spring Symposium on Computational Synthesis.* Stanford, CA: AAAI Press, 2003

Hu, J.; Seo, K.; Fan, Z.; Rosenberg, R.C; Goodman, E.D. (2003). HEMO: A Sustainable Multi-Oobjective Evolutionary Optimization Framework. To appear in In *GECCO-2003: Proceedings of the Genetic and Evolutionary Computation Conference. Chicago.*

Hutter, M. (2002). Fitness Uniform Selection to Preserve Genetic Diversity. In *Proceedings of the 2002 Congress on Evolutionary Computation*: 783—788 (CEC-2002), Hawaii.

Johnson, H.E., Gilbert, R.J., Winson, M.K., Goodacre, R., Smith, A.R., Rowland, J.J., Hall, M.A. & Kell, D.B. (2000). Explanatory analysis of the metabolome using genetic programming of simple, interpretable rules. Genetic Programming and Evolvable Machines 1(3), 243-258.

Koza, J.R. (1994). *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge Massachusetts.

Koza, J.; Bennett,F.H; Andre,D.; Keane, M.A. (1999). *Genetic Programming III*. Morgan Kauffmann, San Francisco, CA, U.S.A.

Langdon, W. B. (1998). Genetic Programming and Data Structures. Dordrecht: Kluwer.

Li, J.P., Balazs, M., Parks, G. T, and Clarkson, P. J. (2002). A Species Conserving Genetic Algorithm for Multimodal Funciton Optimization. Evolutionary Computation, 10(3): 207-234.

Luke, S. (2001). When short runs beat long runs. In *GECCO-2001: Proceedings of the Genetic and Evolutionary Computation Conference*. Lee Spector *et al*, eds. Morgan Kaufmann. 74-80.

Maza, M. de la and Tidor. B. (1991). Boltzmann weighted selection improves performance of genetic algorithms. A.I. Memo 1345. Artificial Intelligence Laboratory, MIT, Cambridge.

Mengshoel, O.J. and Goldberg, D.E. (1999). Probabilistic Crowding: Deterministic Crowding with Probabilisitic Replacement. In *Proceedings of the Genetic and Evolutionary Computation Conference*: 409-416, GECCO-1999, Orlando, Florida, USA.