HFC: a Continuing EA Framework for Scalable Evolutionary Synthesis

Jianjun Hu[+§], Erik D. Goodman[+], Kisung Seo[+], Zhun Fan[+], Ronald C. Rosenberg[Ψ]

[+]Genetic Algorithms Research and Applications Group (GARAGe), Michigan State University, East Lansing, MI, 48824
[§]Department of Computer Science & Engineering, Michigan State University, East Lansing, MI, 48824
[Ψ]Department of Mechanical Engineering, Michigan State University, East Lansing, MI, 48824
{hujianju, goodman, ksseo, fanzhun, rosenber}@egr.msu.edu

Abstract

The scalability of evolutionary synthesis is impeded by its characteristic discrete landscape with high multimodality. It is also impaired by the convergent nature of conventional EAs. A generic framework, called Hierarchical Fair Competition (HFC), is proposed for formulation of continuing evolutionary algorithms. This framework features a hierarchical organization of individuals by different fitness levels. By maintaining repositories of intermediate-fitness individuals and ensuring a continuous supply of raw genetic material into an environment in which it can be exploited, HFC is able to transform the convergent nature of current EAs into a sustainable evolutionary search framework. It is also well suited for the special demands of scalable evolutionary synthesis. An analog circuit synthesis problem, the eigenvalue placement problem, is used as an illustrative case study.

1. Introduction

In recent decades, evolutionary algorithms have found increasingly successful applications in many automated system synthesis problems, such as analog and digital circuit design and control system synthesis (Koza 1994; Sripramong and Toumazou 2002), neural network synthesis (Yao 1996), mechatronic system synthesis (Seo et al. 2002), evolvable hardware (Yao & Higuchi 1996), etc. While human designers are strongly constrained by their limited domain knowledge and incomplete understanding of the physical process itself, evolutionary synthesis turns out to be an especially useful tool in exploring many under-exploited design domains such as adaptive systems, fault-tolerant systems, and poorly specified and unstructured systems.

However, just as many EAs may produce results that are surprising initially, but not much later on, evolutionary system synthesis has not generated the expected significant results in terms of the complexity of the evolved solutions. With automatically synthesized analog circuits with a hundred or so components (Koza 1999) and some evolvable hardware circuits with simple functions, evolutionary synthesis seems to be strongly limited by some inherent difficulty, especially considering the huge computing power of a 1000-PC cluster available to Koza (Koza 1999). It is clear that there are some critical issues other than the computational power that need to be handled before the ambition for automated Darwinian invention can be realized to automatically evolve startling results in some "killer" application areas.

The limited scalability of evolutionary synthesis techniques comes from two aspects. One limitation arises from the scalability of the evolutionary algorithm framework. Actually, due to the convergent nature of the conventional EA framework (Thierens 2000), many existing EAs suffer from the phenomenon of premature convergence, which has been extensively studied for a long time (Carter & Park 1994; Ryan 1996; Cantupaz & Goldberg 1999). The second limitation comes from the compositional mechanism for topology and parameter search including topology-encoding scheme, topology modifying operators, etc. Scalability in this aspect has been discussed in (Torreson 2000; Yao & Higuchi 1998). The "biological developmental model" is regarded as a potential technique to improve the scalability of topology synthesis (Gordon & Bentley 2002). It appears that the divide-and-conquer strategy, modularity, parametric abstraction, and hierarchical organization are all important principles that need to be observed to improve the fundamental scalability of evolutionary synthesis systems.

This paper examines the convergent nature of current EAs and its effect on the scalability of evolutionary synthesis involving simultaneous topology and parameter search. We then propose a generic framework named Hierarchical Fair Competition (HFC) for continuing evolutionary algorithms. This framework features a hierarchical organization of individuals by fitness levels. By maintaining repositories of intermediate-fitness individuals and ensuring a continuous supply of raw genetic material, HFC is able to transform the convergent nature of current EAs into a sustainable evolutionary search framework. An analog circuit synthesis problem (the eigenvalue placement problem) is used as the illustrative case study.

2. Premature Convergence, Diversity and Sustainable Evolutionary Synthesis

As one of the main topics of EAs, the premature convergence problem has been addressed by both theoretical and empirical studies. The term "premature convergence" is used to refer to the clustering of the population about a single (or a few) genotypes, resulting in a stagnation of the search process. (Note that if the globally optimal individual is in that population, the convergence is not premature, and if the EA allows the population, although composed of very similar individuals, to continue to make steady (or at least not infrequent) progress toward the global optimum, it is not considered to be prematurely converged. Without stagnation of search, premature convergence has not occurred.) Among these diversified efforts, premature convergence has been attributed primarily to:

- **loss of population diversity**

Loss of diversity is the most popular explanation of premature convergence of EAs (Ryan 1996; Leung et al. 1997; Burke 2002; Ursem 2002). Many related approaches to maintaining or increasing diversity have been proposed, including canonical fitness sharing, crowding, and some explicit diversity-controlled GAs (Ursem 2002). However, in the opinion of the present authors, lack of population diversity is only a *symptom* of premature convergence. The more direct cause is the loss of exploratory capability. Analyzing how the exploratory power gets lost and devising corresponding strategies to maintain it is more fundamental to avoiding premature convergence than is simply maintaining diversity in any form. In this aspect, although diversity-management-oriented approaches can achieve some improvement, generally they are somewhat misguided and won't result in scalable EAs (Thierens 2000). A simple example may help to illustrate the relationship between premature convergence and lack of diversity. Consider a population that has prematurely converged around a single, extremely high-fitness (but not optimal) individual. Now replace all but one copy of that individual with randomly generated individuals (hence, typically having low fitnesses). Such a population has enormous diversity, but its ability to explore is extremely limited, as it has none of the "intermediate-level" building blocks that are alternatives to those contained in the single best individual. It is likely that such a population will quickly be overtaken by (identical or slightly altered) copies of the best individual, or at least that it will not perform well in continuing the search for a global optimum individual. So, although its diversity is high, it may be considered to be "waiting only to meet the convergence criterion" to be labeled prematurely converged.

- **failure of building block mixing in the race of selection and mixing**

Evolutionary search has been vividly modeled as a race between the converging process of selection and the exploratory process of building block mixing (Goldberg 2002; Thierens 2000). On the one hand, some kind of convergence is needed to allocate more computing efforts to promising areas of the search landscape. On the other hand, it is necessary to maintain a continuing exploratory capability to find new promising search areas to avoid being trapped in local optima. A family of building-block-mixing-oriented GAs has been designed which has proved to be competent on some bounded difficulty problems (Goldberg 2002). Unfortunately, they are usually limited to binary coded GAs.

- **"founder effect"**

According to the building-block-oriented explanation of the "founder" effect in GA (Holland 2000), an early-discovered schema tends to occupy most of the population and thus prevent any other incompatible schemata from being incorporated into these individuals. Thus, no other incompatible schemata may be tested and the exploratory capability gets lost. Langdon (1998) provided another macroscopic observation of "founder" effect in the case of GP: deceptive partial solutions discovered early in the evolution of the population with relatively high fitness tend to produce clones at high rates, thus reducing the variety of the population. This is the well-known dominance phenomenon.

In the following subsections, we will examine the characteristics in the search process of evolutionary synthesis and examine how the exploratory capability gets lost.

2.1 Characteristics of the Fitness Landscape in Evolutionary Synthesis

One of the major characteristics of evolutionary synthesis is the inclusion of simultaneous topology search and parameter search. Contrary to the knowledge-based evaluation of design candidates by human designers, the goodness of a topology can only be evaluated in the context of a concrete instance of the topology, with whatever parameters that instance may have. As a result, an effective and fair evaluation of the quality of a particular topology is possible only after sufficient parameter search in the framework of that topology.

Good performance of a solution is the result of the *concordance* of its topology and its parameter, so the fitness landscape of evolutionary synthesis is extremely multimodal. Each topology typically has multiple fitness peaks over its parameter space. Different topologies comprise a second level of fitness peaks (where each peak is the highest fitness of one topology). Among these peaks there are a huge number of low-fitness valleys, created by the parameter space of the topologies. What this means to evolutionary search is an increasing difficulty of moving from one topology peak to another topology peak in the fitness landscape. The reason is, after topology modification either by crossover or mutation, the offspring usually have very low fitness and they need a considerable amount of time to adjust their parameters before the potential performance of their topologies can be exposed.

2.2 Loss of Explorative Capability in Evolutionary synthesis: the Explanation

Instead of the loss of diversity, the loss of exploratory capability turns out to be the more direct reason for premature convergence in EAs. In most EAs, while running, exploratory capability is found to be gradually lost along with the increasing fitness of the population. Premature convergence is also observed to be especially severe in genetic programming, which usually uses a much larger population size to get good results. All these phenomena can be explained in terms of loss of exploratory capability.

In a typical EA, the selection process is more or less based on fitness of individuals. Higher fitness individuals are given a higher probability for generating offspring. The result of this decision is: the average fitness of the population increases continually. However, new offspring created by crossover and mutation usually have low fitness, as the result of disruption of the co-adaptation of closely coupled subcomponents. Thus, it is often only those individuals that differ least significantly from one of the parents that still have reasonably high fitness, and become increasingly abundant in the population. In addition to the already existing fitness valleys specific to evolutionary synthesis, the ever-increasing average fitness of the population makes those valleys even deeper. The consequence is that new exploratory offspring will have increasing difficulty to persist in the population long enough to be sufficiently exploited to demonstrate their actual performance value (i.e., fitness when suitably adapted parameters have been found). It becomes increasingly harder to move from one peak to another peak across those huge fitness valleys. The evolution process will then gradually lose the power to explore new search areas from these new individuals with low fitness, whethercreated by crossover, mutation, or reinitialization.

It is clear now that to maintain explorative capability, there must be some mechanisms to protect new individuals of low fitness and culture them into higher fitness individuals. Individuals of different fitness levels should be segregated to ensure fair competition at all fitness levels. This suggests an assembly-line or pipeline structural organization of subpopulations, as employed in the HFC framework discussed in Section 3.

2.3 The Convergent Nature of Most EAs and their Assumptions

The scalability of evolutionary synthesis is also constrained by the convergent nature of most of the commonly used EAs. Intuitively, as the goal of evolution is to find high fitness individuals, all the individuals should be used to explore the fitness frontier, and indeed they are, as the result of some form of fitness-based selection in most EAs. However, due to the unbalanced sampling and the unbalanced speed of fitness growth of individuals, in any given run, typically some salient building blocks get to propagate, while other potential building blocks getting lost. And as the average population fitness increases, it becomes increasingly difficult for new building blocks to be discovered and exploited.

Traditional EAs typically begin the process of converging from the start of a run, discarding some low-order components of potentially important building blocks (even when they may be critical later). This scheme is most clearly demonstrated in messy GA, where a special initialization phase is used to screen for useful building blocks (Goldberg 2002). However, this scheme is unjustified in the sense that the screening decision is based on very limited evaluations of the possible combinations of building blocks in the first few generations However, it is highly probable that some of the discarded building blocks, when appropriately assembled, may produce the perfect solution, or that some of the discarded building blocks may be essential to forming the best solutions. So, to some extent, premature convergence is the result of premature convergence of intermediate building blocks.

The convergent nature of EAs is caused by the underlying one-epoch assumption of the conventional EA framework: starting from random individuals, all the individuals should move to a higher fitness frontier and the low-level building block assembly stage is terminated. The population-sizing model of Goldberg is a typical example of this assumption (Goldberg 1989). The need to transform this convergent nature of conventional EAs into one of continuing (sustainable) search suggests a *continual* process of building block sampling and assembly, also including at the low and intermediate fitness levels, rather than only at high fitness levels. It also implies the importance of maintaining individuals of all fitness levels, from random ones to the best individuals. By continually introducing random individuals into the low-fitness-level subpopulations, it becomes possible to ameliorate the demand for a huge population size, as required by GA, to ensure sufficient building block sampling (Goldberg 2002). This is especially true for GP, where much larger population sizes are usually required in order to achieve reasonably good results.

The advantage of maintaining individuals distributed across all fitness levels is also reported in (Hutter 2002). Instead of using a hierarchical organization of subpopulations, a fitness uniform selection (FUSS) is used to maintain individuals of all fitness levels. However, FUSS tends to distribute the computing efforts too evenly among all individuals and lacks sufficient selection pressure for healthy convergence to explore the fitness frontier.

2.4 The Developmental/Growth Process in Evolutionary Synthesis

One unique feature of evolutionary synthesis, compared to normal parameter-oriented EAs such as evolution strategies and GA (in most applications), is the variable-length-genotype encoding. One representative example is the case of genetic programming. In typical system synthesis tasks involving topology innovation, some kind of developmental or growth process through many generations is needed to evolve trivial primitive embryo designs into full-fledged functioning systems. This necessary intermediate "growth" process has a tremendous effect on the performance of an EA framework for evolutionary synthesis.

When one examines the fitness progress curve of most GP (or also GA) experiments, the most salient observation is that the largest fitness gains occur in the very early stages. The evolution in the later stages appears to be more like a refining process. In the tree-type genetic programming, the initial stages of evolution usually establish the general framework of the topology. The nodes near the tree root converge relatively quickly. Actually, the higher the fitness, the more constraints the established topology puts on possible later modifications, and the less likely become major innovations. It is very much like the biological evolution of species, in which the emergence of most species on the earth occurred during the age of the Cambrian explosion. Later on, the speciation process continues and reproductive segregation emerges. It is a converging, refining process.

Few people have considered how this phenomenon may guide our design of EAs. What the Cambrian innovation stage tells us is that, to ensure sustainable innovation and evolution, an EA framework for evolutionary synthesis should always keep the early evolution stages running. Since major innovation happens in the early stage, why should we terminate this phase after a very limited assembly process, as the fitness of individuals rises? Since highly evolved individuals have less and less probability of changing their basic frameworks, it seems preferable to maintain, somehow, repositories of representative intermediate individuals, from which further innovation may occur. This corresponds to the conclusions of the analysis of the loss of exploratory capability and the convergent nature of EAs:

- Individuals of all fitness levels should be maintained, from random individuals to the highly evolved ones, to keep evolution going at all levels;
- Random individuals should be continually introduced into low-fitness subpopulations
- Mutation (either topology or parameter modification) rate and magnitude should be higher at lower fitness levels while lower at higher fitness levels, so as to provide highly evolved individuals a more gradual refining process, while not "throwing away" what has been learned at that level.

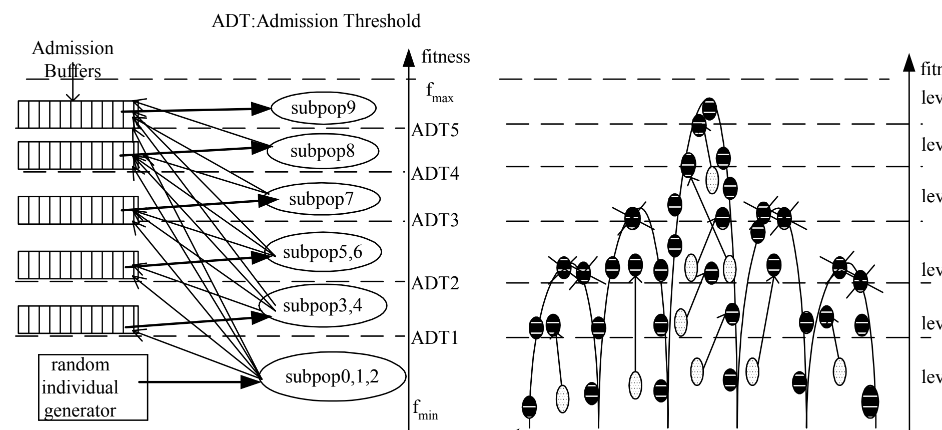
The Hierarchical Fair Competition (HFC) model proposed below is exactly such an EA framework for scalable evolutionary synthesis.

3. The HFC Framework for Scalable Evolutionary Synthesis

Inspired by the fair competition principle observed in societal and economical systems (Hu and Goodman 2002), the HFC model is devised with three interrelated components, as discussed below.

3.1 The Hierarchical Organization of Subpopulations to Establish a Fitness Gradient

In the HFC framework (Figure 1), multiple subpopulations are organized into a fitness hierarchy, in which each subpopulation belongs to a specific fitness level that accommodates immigrating individuals within a specified range of fitness, and that forces emigration of individuals with fitness above that range. The entire range of possible fitnesses is spanned by the union of the fitness ranges of all levels. Each fitness level has an admission buffer that has an admission threshold determined either initially (fixed) or adaptively. The admission buffer is used to collect



(a) In HFC model, subpopulations are organized in a hierarchy with ascending fitness level. Each level (with one or more subpopulations) accomodates individuals within a certain fitness range determined by the admission thresholds

(b) HFC model extends the search horizontally in search space and vertically in fitness dimension and kills bad individuals at appropriate times while allowing promising young individuals grow up continuously

Figure 1: (a) The assembly line structure of HFC framework and (b) HFC from the perspective of fitness landscape.

qualified candidates, synchronously or asynchronously, from the subpopulations of lower levels. Each level also has an export fitness threshold, defined by the admission threshold of the next higher fitness level. Only individuals whose fitnesses are above the admission threshold and below the export threshold of the fitness level that a given subpopulation belongs to are allowed enter, or stay, respectively, in that subpopulation. Otherwise, they are exported to asubpopulation of the appropriate higher fitness level.

A problem that can occur at any fitness level but the lowest (which has no lower fitness limit) is that the children produced via mutation or crossover of individuals at a given fitness level may have fitness below its admission threshold. This could allow the average fitness of individuals at that level to degrade below the admission threshold. While many alternative policies are possible, in the work reported here, this degradation problem is dealt with by allowing these low-fitness offspring to remain in the level despite their degraded fitness -- reminiscent of the occasional backward step allowed in simulated annealing -- with the assumption that the selection mechanism and immigration replacement policy will act strongly enough to maintain a reasonable percentage of "qualified" individuals at each level.

3.2 Random Individual Generator: the Source of Genetic Material

To maintain the Cambrian innovation stage, as illustrated in Figure 1, at the bottom fitness level, there exists a random individual generator that continuously feeds raw building blocks (in the form of individuals) into the bottom processing level. It is important that this generator be unbiased as much as possible to supply a complete set of all possible low-level building blocks, unless there is prior knowledge about the search space that should be used to bias the generator. This inflow of random individuals relieves HFC from depending on a large initial population size to provide sufficient primitive building blocks.

3.3 The Migration Policy from Lower to Higher Fitness Levels

Exchange of individuals can be conducted synchronously after a certain interval or asynchronously as in many parallel models. At each moment of exchange, or, if desired, as each new individual in a subpopulation is evaluated, any individual whose fitness qualifies it for a higher level is exported to the admission buffer of whatever higher level has a fitness range that accommodates the individual. After the exporting processes at all levels finish, subpopulations at each level import an appropriate number of qualified candidates from their admission buffers to replace some worst individuals. If subpopulations at the base level find any open spaces left over by exporting, they fill those spots with random individuals. If subpopulations of higher levels find empty spaces after importing individuals from their admission buffers or the admission buffer is empty, they can either mutate current members or select two members and do crossover to generate the needed number of new individuals. In the experiments reported here, crossover is used to make up any shortfall of individuals at any level except the bottom level. As we use the sub-populations of each level as the repositories, the admission buffer is then cleaned after the migration process.

3.4 Adaptive Allocation of Subpopulations to Fitness Levels

In the previous HFC model (Hu & Goodman 2002), the allocation of subpopulations to fitness levels is configured before the evolution is started. A shortcoming of this method is that in the initial generations, the high –fitness-level subpopulations won't actually contain any qualified individuals, and so, according to this method, are not activated for evolution. This reduces the effective population size. The solution presented here (Figure 5: b) is to adaptively allocate subpopulations to fitness levels. In the beginning, all subpopulations are allocated to the bottom level. Later, once certain higher levels get some qualified individuals, then all intermediate levels are activated and the whole subpopulations on those activated levels are allocated according to some strategy. One simple strategy is illustrated in Figure 3. This HFC framework with adaptive topology works like a rubber band. At the initial stage, it is quite compressed, but gradually, the rubber band stretches to accommodate individuals within a larger range of fitness. In the work reported in this paper, we simply allocate the subpopulations evenly to all activated levels. Details of this procedure are described in (Hu et al. 2002).

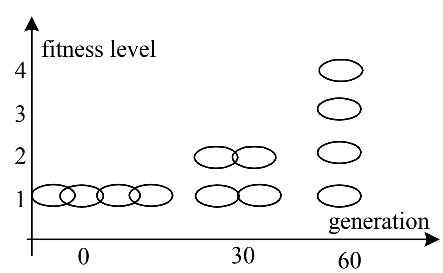


Figure 2: Adaptive allocation of subpopulations to fitness levels

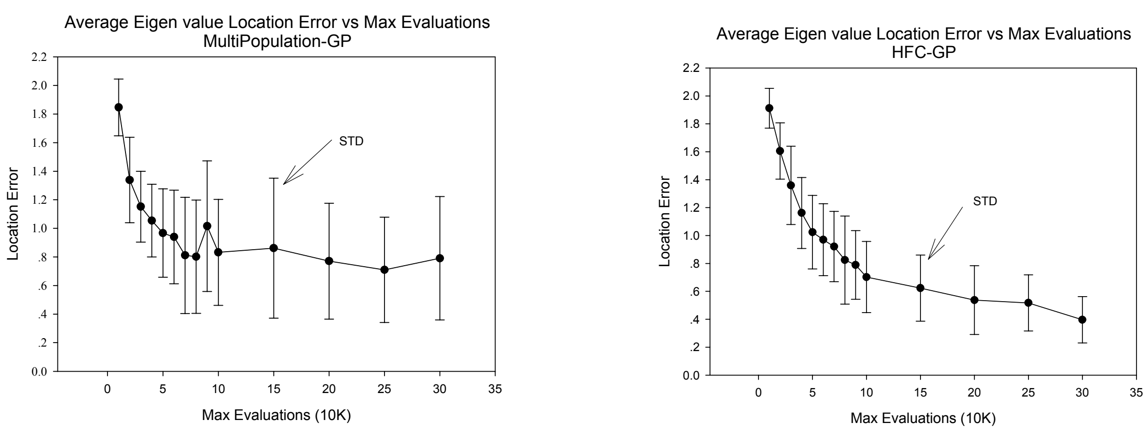
4. Experimental evaluations

To evaluate the performance of the HFC framework, an HFC-based genetic programming system is applied to an analog circuit synthesis problem: the 10-eigenvalue-placement problem. Except for the representation of circuits with bond graphs, the topology synthesis approach employed here is similar to what Koza did for evolving electric circuits with a developmental process. Simply speaking, the design task here is to synthesize a linear system such that the eigenvalues are located at assigned target positions, building the system from capacitors, inductors, resistors, and sources of effort (like a battery). More details can be found in (Seo et al 2002; Fan et al 2002).

Two experiments are conducted here to compare the performance of HFC with island multi-population GP (MulPop). In experiment 1, we investigate the relationship between the average of the best fitness over 40 runs and the maximum number of evaluations, ranging from 10,000 to 300,000. Experiments for each maximum evaluation limit are run with independent random seeds. In experiment 2, we demonstrate the continuing search capability of HFC by examining the influence of population sizes on the average best fitness over 40 runs, each with a maximum of 150,000 evaluations.

The parameters used in experiment 1 are as follows: the 10 target eigenvalues are $\{-0.1\pm5.0j, -1\pm2j, -2\pm j, -3\pm0.7j, -4\pm0.4j\}$. Total population size is 500, with 10 subpopulations for the island model, each having 50 individuals. For HFC, the subpopulation sizes for subpops 1 to 10 are {30,30,40,40,50,50,50,50,60,100}. The maximum number of nodes is 400, while the maximum depth is 12. Crossover probability is 0.95 (individuals chosen using tournament selection with size 7) and reproduction probability is 0.05 with best selection (elitism). No mutation is used. The GP initialization method is half-and-half. Migration frequency is 5 generations with 10 percent migration for MulPop-GP. The parameters of experiment 2 are the same as above except that the whole population size is varied. Individuals are evenly distributed among 10 subpops for the MulPop approach, while for HFC, the proportion of subpop sizes is the same as in experiment 1, again using ten subpops.

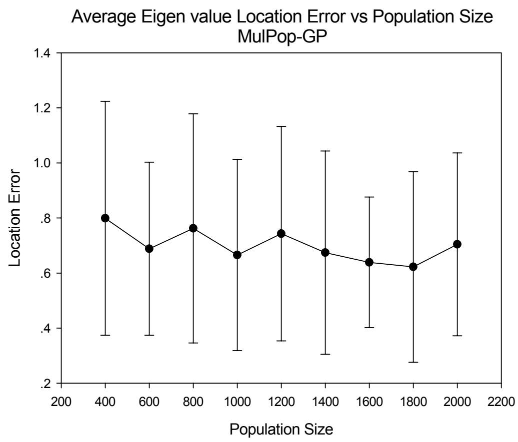
Figure 3 shows the results from the first experiment. It is apparent that HFC can achieve much lower average error of eigenvalue position assignment for different numbers of maximum evaluations and sustainable evolution appears with its steady decrease of the location error. While for conventional multi-population GP, it is expected that beyond a certain evaluation limit, there would be no more progress regardless how many more evaluations one may provide. The results of experiment 2 are illustrated in Figure 4. They illustrate that a multi-population version of "conventional" GP relies on large population sizes to achieve good performance. With limited evaluations, there is an optimal population size. If with unconstrained evaluations, larger populatin size produces better results. With a limitation on maximum number of evaluations (150,000) in this case, HFC clearly achieves better performance with smaller population sizes, which is contrary to the typical behavior of conventional GPs. Combining the evidence from Figure 3, we can expect that with sufficient evaluation limits, HFC-GP will achieve equally good performance regardless of the population sizes, as already confirmed in our other experiments. This is achieved due to the continuing search capability with sustainable innovation at all fitness levels.



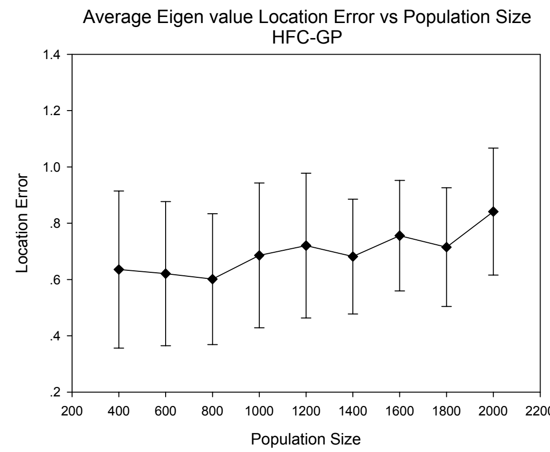
(a) Multi-Population GP

(b) HFC-GP

Figure 3. Comparison of the average best-of-run errors vs. maximum evaluations for multi-population-GP and HFC-GP. Error of 0 is the optimal value. HFC achieves much more robust search with continuous progress, reflected by its much smaller standard deviations. Multi-population-GP is more opportunistic with its large standard deviations of the location error. It also has the tendency that beyond 300 K evaluations, there won't be much progress, while for HFC-GP, sustainable progress appears.



(a) Multi-Population GP



(b) HFC-GP

Figure 4. Comparison of the performance dependence on population size for HFC-GP and Multi-populatio GP. Due to its continuing search capability, HFC works well with small population sizes. With limited evaluations, HFC achieves better results by running more generations. Multi-population GP depends on large population size to get better performance.

5. Discussion and Conclusions

By examining the phenomenon of premature convergence in EAs, this paper attributes its fundamental cause to the loss of exploratory capability, rather than the loss of diversity per se, as has been widely believed. This understanding, along with the analysis of the characteristics of evolutionary synthesis (such as fitness landscapes and developmental processes) and of the convergent nature of current EAs, helps to explain why premature convergence is much more severe in GP-based evolutionary synthesis and why GP usually needs much larger population sizes. It also helps to explain which types of “diversity” are helpful. That is, only when the diversity includes low- and moderate-fitness individuals operating in an environment free of competition from much-higher-fitness individuals and allowing them sufficient opportunity for exploration can the diversity produce maximum benefit.

Based on this understanding of the premature convergence phenomenon, a generic EA framework, called Hierarchical Fair Competition (HFC), is proposed, which is characterized by its: maintaining of individuals of all fitnesses, fair competition at each fitness level, inherent hierarchical elitism, and its continuing search at all fitness levels including the continual introduction of random individuals. By removing the one-epoch assumption of the conventional EA framework, HFC transforms the convergent nature of the current EA framework into a continuing one. Two experiments demonstrate the continuing search capability by showing its independence of population size and its better performance with constant progress for different evolution times. In this framework, population sizing theory as earlier formulated is no longer applicable (Goldberg 1989).

It is interesting to note that HFC seems lie between two extreme strategies to allocate search effort: the conventional EA framework and FUSS (Hutter 2002). The first one allocates essentially all the computing resource to the earliest-found high-fitness individuals, increasing the average population fitness constantly. The latter one allocates the computing resource uniformly to individuals over all fitnesses, and thus has too weak selection pressure to ensure sufficient exploitation of early discovered promising search areas. However, introducing FUSS into each level of HFC might be useful, because of the selection pressure already provided by the HFC’s array of fitness levels.

By considering the apparent innovation period in the early evolution stages and the subsequent refining period, it seems to us that to achieve sustainable innovation in evolutionary synthesis and in all other EAs, the appropriate strategy is not to try to jump out of local optima from highly evolved populations, as is done in conventional EAs, but rather to try to form new local or global optima in a bottom-up way, as is done in a continuing sustainable EA framework like HFC. To achieve scalable evolutionary synthesis, this paradigmatic transition seems to be critical.

Acknowledgements

This work was supported by the National Science Foundation under contract DMI 0084934.

References

Burke, E. S.; Gustafson, G. K. 2002. A Survey and Analysis of Diversity Measures in Genetic Programming. In *Proceeding of Genetic and Evolutionary Computation Conference*, New York.

Cantupaz, E. and Goldberg, D.E. 1999. On the scalability of Parallel Genetic Algorithms. *Evolutionary computation* 7(4): 429-449

Carter, B. and Park, K. 1994. Scalability Problems of Genetic Search. In *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics,* 1591-1596.

Fan, Z.; Seo, K.; Rosenberg, R. C.; Hu, J.; Goodman, E. D. 2001. Exploring Multiple Design Topologies using Genetic Programming and Bond Graphs. In *Proceeding of Genetic and Evolutionary Computation Conference,* San Fransisco;

Goldberg, D.E. 2002. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms* Boston, MA: Kluwer Academic Publishers.

Goldberg, D. 1989. Sizing Populations for Serial and Parallel Genetic Algorithms. In *Proc. Third International Conf. on Genetic Algorithms*. Morgan Kaufmann.

Gordon, T.G.W and Bentley, P. J. 2002. Towards Development in Evolvable Hardware. In *Proceedings of the 2002 NASA/DoD Conference on Evolvable Hardware*: 241-250. Washington D.C., U.S.A.

Holland, J.H. 2000. Building Blocks, Cohort Genetic Algorithms, and Hyperplane-Defined Functions. *Evolutionary Computation*, 8(4): 373-391.

Hu, J., 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.

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

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.

Leung, Y.; Gao, Y. and Xu, Z. 1997. Degree of Population Diversity--- A Perspective on Premature Convergence in Genetic Algorithms and its Markov Chain Analysis, *IEEE Transactions on Neural Networks*, 8(5): 1165-1171.

Ryan. C. 1996. Reducing Premature Convergence in Evolutionary Algorithms. Ph.D. Thesis. Department of Computer Science, National University of Ireland.

Seo, K.; Hu, J.; Fan, Z.; Goodman, E. D.; and Rosenberg, R.C. 2002. Automated Design Approaches for Multi-Domain Dynamic Systems Using Bond Graphs and Genetic Programming. *The International Journal of Computers, Systems and Signals*, 3(1): 55-70.

Sripramong, T; Toumazou, C. 2002. The Invention of CMOS Amplifiers Using Genetic Programming and Current-Flow Analysis. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*. 21(11).

Thierens, D. 2000. Scalability Problems of Simple Genetic Algorithms. *Evolutionary Computation*, 7(4): 331-352

Torresen, J. 2002. A Scalable Approach to Evolvable Hardware. *Genetic Programming and Evolvable Machines*, 3 (3): 259-282.

Ursem, R., K. 2002. Diversity-Guided Evolutionary Algorithms. In: *Proceedings of Parallel Problem Solving from Nature VII* (PPSN-2002): 462-471

Yao X.1993. A Review of Evolutionary Artificial Neural Networks. *International Journal of Intelligent Systems*, 8: 539-567.

Yao, X. and Higuchi, T. 1998. Promises and Challenges of Evolvable Hardware. *IEEE Transactions on Systems, Man, and Cybernetics*, Part C, 28(4).