C6.3 Parallel Genetic Algorithms Based on Punctuated Equilibria

W N Martin J Lienig and J P Cohoon

Department of Computer Science, University of Virginia

Abstract

The island model genetic algorithm shows promise as a superior formulation based on considerations from theories of natural evolution and from the efficiencies of coarse-grained parallel computer architectures. The theory of "punctuated equilibria" calls for the population to be partitioned into several distinct subpopulations. These subpopulations have extensive periods of isolated evolution, i.e., computation, occasionally interspersed with migration, i.e., communication. It is precisely for this sort of process with its alternating phases of extended computation and limited communication that message-passing multi-processors (implementing coarse-grained parallelism) are best suited. We validate this promise of the island model and illustrate the effects of varying configuration attributes through experiments with a difficult VLSI design problem.

C6.3.1 Parallelization

Research to develop parallel implementations of algorithms has a long history (Slotnick *et al* 1962, Barnes *et al* 1968, Wulf and Bell 1972) across many disparate application areas. The majority of that research has been motivated by the desire to reduce the overall time-to-completion of a task by distributing the work implied by a given algorithm to processing elements working in parallel. More recently some researchers have conjectured that some parallelizations of a task improve the quality of solution obtained for a given overall amount of work, e.g., emergent computation (Forrest 1991), and some even suggest that considering parallelization may lead to fundamentally new modes of thought (Bailey 1992). Note that the benefits of this latter kind of parallelization depend only on concurrency, i.e., the logical temporal-independence, of operations and they can be obtained via sequential simulations of parallel formulations.

The more prevalent motivation for parallelization, i.e., reducing time-to-completion, depends on the specifics of the architecture executing the parallelized algorithm. Very early on, it was recognized that different parallel hardware made possible different categories of parallelization based on the *granularity* of the operations performed in parallel. Typically these categories are referred to as: *fine-grained*, *medium-grained* and *coarse-grained* parallelization. At the extremes of this spectrum, fine-grained (or small-grained) parallelism means that only short computation sequences are performed between synchronizations, while coarse-grained (or large-grained) parallelism means that extended computation sequences are performed between synchronizations. SIMD (single-instruction, multiple-data) architectures are most appropriate for fine-grained parallelism (Fung 1976), while distributed-memory message-passing architectures are most appropriate for coarse-grained parallelism (Seitz 1985).

Two of the earliest parallelizations of a genetic algorithm (GA) were based on a distributed-memory message-passing architecture (Tanese 1987, Pettey *et al* 1987). The resulting parallelization was coarse-grained in that the overall population of the GA was broken into a relatively small number of subpopulations. Each processing element in the architecture was assigned an entire subpopulation and executed a rather standard GA on its subpopulation.

In the same time frame, it was noted that the theory of natural evolution, called *punctuated equilibria*, provided evidence that in natural systems this kind of "parallelization" of evolution had an emergent property of bursts of rapid evolutionary "progress". The resulting parallel GA was shown to have that property on several applications (Cohoon *et al* 1987).

Each of the above systems are examples of what has come to be called *island-model* parallel genetic algorithms (Adamidis 1994). In the next section we discuss theories of natural evolution as they support and motivate island-model formulations. We then discuss the important aspects, parameters and attributes of systems built on this model. Finally, we present results of one such system on a difficult VLSI design problem.

C6.3.2 Theories of natural evolution

In what has been called the *Modern Synthesis* (Huxley 1942), the fields of biological evolution and genetics began to be merged. A major development in this synthesis was Sewall Wright's (1932) conceptualization of the *adaptive landscape*. The original conceptualization proposes an underlying space (two-dimensional for discussion purposes) of possible genetic combinations. At each point in that space an "adaptive value" is determined and specified as a scalar quantity. The surface thus specified is referred to as the "adaptive landscape." A population of organisms can be mapped to the landscape by taking each member of the population, determining the point in the underlying space that its genetic code specifies and marking the associated surface point. The figure used repeatedly by Wright shows the adaptive landscape as a standard topographic map with contour lines of equal adaptive value instead of altitude. The +'s indicate local maxima. A population – in two demes – is then depicted by two shaded regions overlaid on the map.

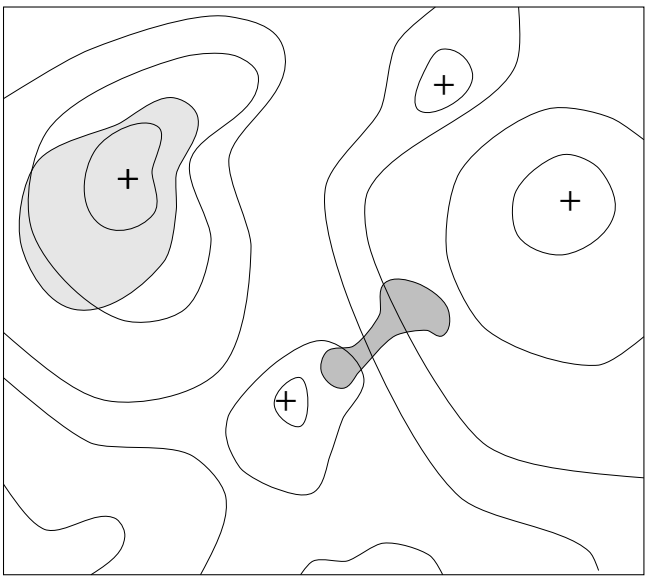

Figure C6.3.1. An adaptive landscape according to Wright, with a sample population – in two demes (shaded areas).

There are several reasons that we used the word "conceptualization" in the previous paragraph. First and foremost, it is not clear what the topology of the underlying space should be. Wright (1932) considers initially the individual gene sequences and connects genetic codes that are "one remove" from each other, implying that the space is actually an undirected connected graph. He then turns immediately to a continuous space with each gene locus specifying a dimension and with units along each dimension being the possible allelomorphs at the given locus. Specifying the underlying space to be an n-dimensional Euclidean space determines the topology. However, if one is to attempt to make inferences from the character of the adaptive landscape (Radcliffe 1991), the ordering of the units along the various dimensions is crucial. With arbitrary orderings the metric

notions of "near by" and "distant" have no clear-cut meaning; similar ambiguities occur in many discrete optimization problems. For instance, given two tours in a travelling salesperson problem, what is the proper measure of their closeness?

The concept of the adaptive landscape has had a powerful effect on both micro- and macro-evolutionary theory, as well as providing a fundamental basis for considering genetic algorithms as function optimizers. As Wright states (1932):

> The problem of evolution as I see it is that of a mechanism by which the species may continually find its way from lower to higher peaks in such a field. In order that this may occur, there must be some trial and error mechanism on a grand scale ...

Wright also used the adaptive landscape concept to explain his mechanism, "the shifting balance theory". In the shifting balance theory the ability for a species to "search" and not be forced to remain at lower adaptive peaks by strong selection pressure is provided through a population structure that allows the species to take advantage of ecological opportunities. The population structure is based upon *demes*, as Wright describes (1964):

> Most species contain numerous small, random breeding local populations (demes) that are sufficiently isolated (if only by distance) to permit differentiation ...

Wright conceives the shifting balance to be a micro-evolutionary mechanism, that is, a mechanism for evolution within a species. For him the emergence of a new species is a corollary to the general operation and progress of the shifting balance. Eldredge and Gould (1972) have contended that macro-evolutionary mechanisms are important and see the emergence of a new species to be associated very often with extremely rapid evolutionary development of diverse organisms. As Eldredge states (1989):

> Other authors have gone further, suggesting that SMRS [SMRS denotes "specific mate recognition system", the disruption of which is presumed to cause reproductive isolation.] disruption actually may *induce* [his emphasis] economic adaptive change, i.e., rather than merely occur in concert with it, ... [Eldredge and Gould] have argued that small populations near the periphery of the range of an ancestral population may be ideally suited to rapid adaptive change following the onset of reproductive isolation. ... Thus SMRS disruption under such conditions may readily be imagined to act as a 'release,' or 'trigger' to further adaptive change the better to fit the particular ecological conditions at the periphery of the parental species's range.

The island-model GA formulation by Cohoon *et al* (1987, 1991a) was strongly influenced by this theory of punctuated equilibria (Eldredge and Gould 1972), so they dubbed the developed system, *genetic algorithm with punctuated equilibria* (GAPE). In general, the important aspect of the Eldredge and Gould theory is that one should look to small disjoint populations, i.e., peripheral isolates, for extremely rapid evolutionary change.

For the analogy to discrete optimization problems, the peripheral isolates are the semi-independent subpopulations and the rapid evolutionary change is indicative of extensive search of the solution domain. Thus, we contend that the island-model genetic algorithm is rightly considered to be based on a population structure that involves subpopulations which have their isolated evolution occasionally punctuated by inter-population communication (Cohoon *et al* 1991b). In Holland's terms (1975), the *exploration* needed in GA's comes from the infusion of migrants, i.e., individuals from neighboring subpopulations, and the *exploitation* comes from the isolated evolution. It is this alternation between phases of communication and computation that holds the promise for island-model GA's to be more than just hardware accelerators for the evolutionary process. In the next section the major aspects of such island models will be delineated.

C6.3.3 The island model

The basic model begins with the islands – the *demes* (Wright 1964), the *peripheral isolates* (Eldredge and Gould 1972). Here the islands will be referred to as *subpopulations*. It is important to note again that while one motivation in parallelization would demand that each subpopulation be assigned to its own processing element, the islands are really a logical structure and can be implemented efficiently on many different architectures. For this reason we will refer to each subpopulation being assigned to a process, leaving the issue of how that process is executed open.

The island model will consider there to be an overall population $\mathcal{P}$ of $M = |\mathcal{P}|$ individuals that is partitioned into N subpopulations $\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_N\}$. For an even partition each subpopulation has $n = \frac{M}{N}$ individuals, but for generality we can have $|\mathcal{P}_i| = n_i$ so that each subpopulation might have a distinct size. For standard GA's the selection of M can be problematic and for island-model GA's this decision is compounded by the necessity to select N (and thereby n). In practice, the decisions often need to be done in the opposite order, that is, n_i is crucial to the dynamics of the trajectory of evolution for $\mathcal{P}_i$ and is heavily problem dependent. We believe that for specific problems there is a threshold size, below which poor results are obtained (as we will show in section C6.3.5). Further, we believe that island-model GA's are less sensitive to the choice of n_i, as long as it is above the threshold for the problem instance. With n_i decided, the selection of N (and thereby M) is often based on the available parallel architecture.

Given N, the next decision is the subpopulation interconnection. This is generally referred to as the *communication topology*, in that the island model presumes migration, i.e., inter-subpopulation communication. The $\mathcal{P}_i$ are considered to be the vertices of a graph (usually undirected or at least symmetric) with each edge specifying a communication link between the incident vertices. These links are often taken to correspond to actual communication links between the processing elements assigned to the subpopulations. In any case, the communication topology is almost always considered to be static.

Given the ability for two subpopulation processes to communicate, the magnitude and frequency of that communication must be determined. Note that if one allows 0 to be a possible magnitude then the communication topology and magnitudes can be specified by a matrix S, where S_{ij} is the number of individuals sent from $\mathcal{P}_i$ to $\mathcal{P}_j$. $S_{ij} = 0$ indicates no communication edge.

As was mentioned in section C6.3.2, the migration pattern is important to the overall evolutionary trajectory. The migration pattern is determined by the degree of connectivity in the communication topology, the magnitude of communication and the frequency of communication. These parameters determine the amount of isolation and interaction among the subpopulations. The parameters are important with regard to both the "shifting balance" and "punctuated equilibria" theories. Note that as the connectivity of the topology increases, i.e., tends toward a completely connected graph, and the frequency of interaction increases, i.e., the isolated evolution time for each $\mathcal{P}_i$ is shortened, the island model approximates more closely a single, large, freely-intermixing population (see section C6.3.5). It is held generally that such large populations quickly reach stable gene frequencies, and thus, cease "progress". Eldredge and Gould (1972) termed this "stasis", while the GA community generally refers to it as "premature convergence".

On the other extreme, as the connectivity of the topology decreases, i.e., tends toward an edgeless graph, and the frequency of interaction decreases, i.e., each $\mathcal{P}_i$ has extended isolated evolution, the island model approximates more closely several independent trials of a sequential GA with a small population. We contend that such small populations "exploit" strongly the area of local optima, but only those local optima extremely "close" to the original population. Thus, intermediate degrees of connectivity and frequency of interaction provide the dynamics sufficient to allow both exploitation and exploration

For our discussion here, the periods of isolated evolution will be called *epochs*, with migration occurring at the end of each epoch (except the last). The length of the epochs determines the frequency of interaction. Often the epoch length is specified by a number G_i of generations that $\mathcal{P}_i$ will evolve in isolation. However, a formulation more faithful to the theories of natural evolution would be to allow each subpopulation process to reach stasis, i.e., reach equilibrium or convergence, on each epoch (see section C6.3.5). From an implementation point of view with a subpopulation assigned to each processing element, this latter formulation allows the work load to become unbalanced and as such may be seen as an inefficient use of the parallel hardware if the processing elements having quickly converging subpopulations are forced to sit idle. The more troublesome problem is in measuring effectively the degree of stasis. "Inefficiency" might occur when reasonably frequent, yet consistently marginal "progress" is being made. Then not only might other processing elements be idle, but also the accumulated progress might not be worth the computation spent. In one of the experiments of the next section, we will present a system that incorporates an epoch-termination criterion. That system yields high-quality results more consistently, while being implemented in an overall parallel computing environment that utilizes the "idle" processing elements.

The overall structure of the island-model process comprises E major iterations called epochs. During an epoch each subpopulation process independently executes the genetic algorithm for G_i generations. After each epoch there is a communication phase during which individuals migrate between neighboring subpopulations. This structure is summarized in the following pseudocode.

```
Island_Model(E,N,n)
{
   Concurrently for each of the i ← 1 to N subpopulations Initialize(P_i, n) ;
   For epoch ← 1 to E do
      Concurrently for each of the i ← 1 to N subpopulations do
                  Sequential_GA(P_i,G_i) ;
      od ;
      For i ← 1 to N do
                  For each neighbor j of i
                        Migration(P_i, P_j) ;
                  Assimilate(P_i) ;
      od
   od
   routing solution = best individual of all subpopulations ;
}

Sequential_GA(P_i, G_i)
{
   For generation ← 1 to G_i do
      P_new ← ∅ ;
      For offspring ← 1 to Max_offspring_i do
         p_α ← Selection(P_i) ;
         p_β ← Selection(P_i) ;
         P_new = P_new ∪ Crossover(p_α, p_β) ;
      od
      Fitness_calculation(P_i ∪ P_new) ;
      P_i ← Reduction(P_i ∪ P_new) ;
      Mutation(P_i) ;
      Fitness_calculation(P_i) ;
   od
}
```

After each phase of migration each subpopulation must assimilate the migrants. This assimilation step is dependent on the details of the migration process. For instance, in the implemented island model presented in the next section, if individual p_k is selected for emigration from $\mathcal{P}_i$ to $\mathcal{P}_j$ then p_k is deleted from $\mathcal{P}_i$ and added to $\mathcal{P}_j$. (The individual p_k itself migrates.) Also, the migration magnitudes, S_{ij}, are symmetric. Thus, the size of each subpopulation remains the same after migration and the assimilation is simply a fitness recalculation.

In other island models (Cohoon *et al* 1991a), if p_k is selected for emigration from $\mathcal{P}_i$ to $\mathcal{P}_j$ then p_k is added to $\mathcal{P}_j$ without being removed from $\mathcal{P}_i$. (A copy of individual p_k migrates.) This migration causes the subpopulation size to increase, so (under an assumption of a constant size subpopulation GA) the assimilation must include a reduction operation.

Still other parallel GA's (Mühlenbein *et al* 1987, Mühlenbein 1989, Gorges-Schleuter 1990) implement overlapping subpopulations, i.e., the "diffusion model". For such systems migration is not really an issue, rather its important effect is attained through the selection process. However, parallel genetic algorithms with overlapping subpopulations are best suited to medium-grained parallel architectures, so we will not discuss them further here.

An important aspect of both the "shifting balance" and "punctuated equilibria" theories is that the demes, or peripheral isolates, evolve in distinct environments. This aspect has two major facets. The first, and most obvious, facet is the restriction of the available breeding population, i.e., the isolated evolution of each subpopulation as we have already discussed.

The second facet is the differing environmental attributes that determine the factors in natural selection. Wright has suggested that this facet provides "ecological opportunity" (1982). For most GA's these factors, and the ways they interrelate to form the basis of natural selection, are encoded in the fitness function. Thus, to follow the fundamental analogy, the island model should have differing fitness functions at the various subpopulations. Of course, for GA's that are used for functions optimization, the fitness function is almost always the objective function to be optimized (or some slight variation, e.g., an inversion to make an original minimization consistent with "fitness"). We are not aware of any systems that have made use of this facet with truly distinct fitness functions among the subpopulations.

The island model presented in the next section (and many others) has a rudimentary form of this facet through *local* normalization of objective scores to yield fitness values. For example, an individual's fitness might be assigned to be its raw objective score divided by the current mean objective score across the given subpopulation. [This is the reason for the several fitness calculations in the pseudocode presented above.] Such normalization does effect differing environments to the degree that the distributions of the individuals in the subpopulations differ.

For optimization problems that are multi-objective, there is usually a rather arbitrary linear weighting of the various objective dimensions to yield a scalar objective score, e.g., see (C6.3.1). This seems to provide a natural mechanism to have distinct objective functions, namely, distinct coefficient sets at each subpopulation. The difficulty of using this mechanism is that it clearly adds another level of evolution control parameters. Eldredge and Gould recognized this additional level when the discuss "punctuated equilibria" as a theory about the evolution of species, not a theory about the evolution of individuals within a species (Eldredge and Gould 1972). This is, indeed, an important facet of the island model, unfortunately, further exploration of it form and implications is beyond the scope of our discussion here.

C6.3.4 Island-model genetic algorithm applied to a VLSI design problem

In this section we present an island-model GA applied to the routing problem in VLSI circuit design. We then present the results from experiments in which important island-model parameters were varied in order to illustrate the effectiveness of this parallel method and the effects of modifying those parameters.

Problem formulation

The VLSI routing problem is defined as follows. Consider a rectangular routing region on a VLSI circuit with *pins* located on two parallel boundaries (*channel*) or four boundaries (*switchbox*). The pins that belong to the same *net* need to be connected subject to certain constraints and quality factors. The interconnections need to be made inside the boundaries of the routing region on a symbolic routing area consisting of horizontal *rows* and vertical *columns* (see figure C6.3.2).

The routing quality of a particular solution involves (for the purposes of the following experiments) three factors: *Netlength* – the shorter the length of the interconnections, the smaller the propagation delay, *Number of vias* – the fewer the number of vias (connections between routing layers), the fewer electrical and fabrication problems occur, *Crosstalk* – in sub-micron regimes, crosstalk results mainly from coupled capacitance between adjacent (parallel routed) interconnections, so the shorter the length of these parallel-routed segments, the less crosstalk occurs and the better the performance of the circuit. Thus, the optimization is to find a routing solution p_i for which $Obj(p_i)$ is minimal, with the objective function Obj specified by

$$Obj(p_i) = w_1 * l_{nets}(p_i) + w_2 * n_{vias}(p_i) + w_3 * l_{par}(p_i) \qquad \text{(C6.3.1)}$$

where $l_{nets}(p_i)$ is the total length of the nets of routing solution p_i, $n_{vias}(p_i)$ is the number of vias of routing solution p_i, $l_{par}(p_i)$ is the total length of adjacent, parallel-routed net segments of p_i (crosstalk segments), and w_1, w_2 and w_3 are weight factors.

For VLSI designers it is important to have the weight factors, i.e., w_1, w_2 and w_3, to enable the designer to easily adjust routing quality characteristics: the netlength, the number of vias, and the tolerance of crosstalk, respectively, to the requirements of a given VLSI technology.

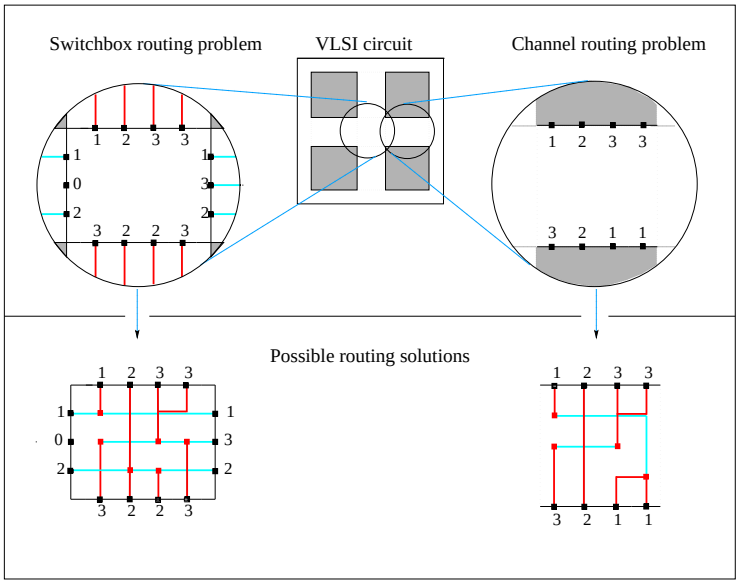

Figure C6.3.2. Example switchbox and channel routing problems ("magnified" in the circles) and possible routing solutions.

Genetic operators

As indicated in the pseudocode in section C6.3.3, each subpopulation process executes a genetic algorithm requiring the following problem-domain specific operators. For these operators, each individual is a complete routing solution, p_i.

Initialization A random routing strategy (Lienig and Thulasiraman 1994) is used to create the initial subpopulations consisting of "non-optimized" routing solutions. These initial routing solutions are guaranteed to be feasible solutions, i.e., all necessary connections exist, but no refinement is performed on them. Thus, we consider them to be random solutions that are distributed throughout the search space.

Fitness calculation The higher quality solutions have smaller objective function values. So, to get a fitness value suitable for maximizing, a raw fitness function is calculated as the inverse of the objective function, $F'(p_i) = \frac{1}{Obj(p_i)}$, then the final fitness $F(p_i)$ of each individual p_i is determined from $F'(p_i)$ by linearly scaling (Goldberg 1989) *local* to the specific subpopulation.

Selection The selection strategy, which is responsible for choosing mates for the crossover procedure, is stochastic sampling with replacement (Goldberg 1989), i.e., individuals are selected with probabilities proportional to their fitness values.

Crossover Two individuals are combined to create a single offspring. The crossover operator gives high-quality routing components of the parents an increased probability of being transferred intact to their offspring (low disruption). The operator is analogous to 1-point crossover, with a randomly positioned line (a horizontal or vertical crossline) that divides the routing area into two sections, playing the role of the crosspoint. For example, net segments *exclusively* on the upper side of a horizontal crossline are inherited from the first parent, while segments *exclusively* on the lower side of the crossline are inherited from the second parent. Net segments intersecting the crossline are newly created for the offspring by means of a random routing strategy (the same strategy used in *Initialization*).

Mutation The mutation operator performs random modifications on an individual, i.e. changes randomly the routing solution. The purpose is to overcome local optima and to exploit new regions of the search space (Lienig and Thulasiraman 1994).

Reduction The reduction strategy combines the current subpopulation with the newly created set of offspring, then simply chooses the fittest individuals from the combined set to be the subpopulation in the next generation, thus keeping the subpopulation size constant.

Parallel structure

The island model used for this application has nine subpopulations ($N = 9$) connected by a torus topology. Thus, each subpopulation has exactly four *neighbors*. The total population ($M = 450$) is evenly partitioned ($n = 50$). (The listed numbers will be changed in the course of the experiments discussed in the following.) The parallel algorithm has been implemented on a network of SPARC workstations (SunOS and Solaris systems). The parallel computation environment is provided by the Mentat system, an object-oriented parallel processing system (Grimshaw 1993, Mentat 1996). The program, written in C++ and Fortran, comprises approximately 10,000 lines of source code. The cost factors in (C6.3.1) are set to: w_1=1.0, w_2=2.0, and w_3=0.01. The experimental results were achieved with the machines running their normal daily loads in addition to this application.

Comparison to other routing algorithms

Any experiment involving a "real" application of a genetic algorithm should begin with a comparison to solution techniques that have already been acknowledged as effective by that application's community. Here we will simply state the results of the comparison because the detailed numbers will only be meaningful to the VLSI routing community and have appeared elsewhere (Lienig 1996). First, eleven benchmark problem instances were selected for channel and switchbox routing problems, for example, "Burstein's Difficult Channel" and "Joo6_16" for channels, and "Joo6_17" and "Burstein's Difficult Switchbox" for switchboxes. These benchmarks were selected because published results were available for various routing algorithms, namely, Yoshimura and Kuh (1982), WEAVER (Joobbani 1986), BEAVER (Cohoon and Heck 1988), PACKER (Gerez and Herrmann 1989), SILK (Lin *et al* 1989), Monreale (Geraci *et al* 1991), SAR (Acan and Ünver 1989), and PARALLEX (Cho *et al* 1994). Note that most of these systems implement deterministic routers.

The island model was run 50 times per benchmark (with varying parameters) and the best-seen solution for each benchmark recorded. A comparison of those best-seen solutions to the previously published best-known solutions indicates that the island-model solutions are qualitatively equal to or better than the best-known solutions from channel and switchbox routers published previously for these benchmarks.

Of course, due to the stochastic nature of a GA, the best-seen results of the island model were not achieved in every run. [All executions were based on arbitrary initializations of the random number generator.] Above we referred to the "best-seen" solutions over all the runs for each benchmark, however, we would like to note that, in fact, in at least 50% of the individual island-model runs solutions equal to these best-seen results were obtained. We judge this to be very consistent behavior for a GA.

C6.3.5 Influence of island-model parameters on evolution

Several experiments have been performed to illustrate the specific effects of important island-model parameters in order to guide further applications of coarse-grained parallel genetic algorithms. The specific parameters varied in the experiments are: the magnitude of migration, the frequency of migration, the epoch termination criterion, the migrant selection strategy, and the number of subpopulations and their sizes.

Five benchmark problem instances, namely, "Burstein's Difficult Channel", "Joo6_13" and "Joo6_16" for channels, and "Joo6_17", and "Pedagogical Switchbox" for switchboxes, were chosen for these experiments. Comparisons were made between various parameter settings for the island model. In addition, runs were made with a sequential genetic algorithm (SGA) and a strictly isolated island model, i.e., no migration. The SGA executed the same algorithm as the subpopulation processes, but on a population of 450 individuals.

In the experiments, the SGA was set to perform the same number of recombinations per generation as the island model does over all subpopulations, namely, *(number of subpopulations)* × *(offspring per subpopulation)*. The SGA and island-model configurations were run the same total number of generations. Thus, we ensure a fair method (with regard to the total number of solutions generated) to compare our parallel approach with a sequential genetic algorithm.

The fundamental baseline was a derived measure based on the best-known objective measure for each problem instance. Remember that for the objective function, smaller values indicate higher

quality solutions. The derived measure is referred to as Δ and is calculated as indicated in the following. Let R_{BK} be the objective measure of the best-known solution and let R_{SGA} be the best-seen result on a particular run of SGA. Then,

$$\delta_{SGA} = \frac{R_{SGA} - R_{BK}}{R_{BK}} \tag{C6.3.2}$$

is a relative (to the best-known) difference for a single run of SGA. The δ_{SGA} were averaged over five runs for each benchmark and over the five benchmarks to yield Δ_{SGA}.

In figures C6.3.3 through C6.3.7, this Δ_{SGA} is shown as a 100% bar in the leftmost position in the plot. Similar Δ's were obtained for the various island-model configurations and shown as percentages of Δ_{SGA}. Thus, if a particular island-model configuration is shown with a 70% bar, then the average relative difference for that configuration is 30% better than SGA's average relative difference to the best-known result.

This derived measure was used in order to combine comparisons across problem instances with disparate objective function value ranges. In addition, the measure establishes a baseline through both best-known and SGA results. We remind the reader that for each benchmark problem this island-model system evolved a solution equal to or better than any previously published system.

Number of migrants and epoch lengths

We investigated the influence of different epoch lengths (number of generations between migration) for different numbers of migrants (number of individuals sent to each of the four neighbors). The migrants were chosen randomly, with each migrant allowed to be sent only once. Figure C6.3.3 shows that the sequential approach was outperformed by all parallel variations (when averaged over all considered benchmarks). Note that the set of parallel configurations included the version with no migration, i.e., the strictly isolated island model (shown in figure C6.3.3 as "0 migrants"). Thus, the splitting of the total population size into independent subpopulations already increased the probability that at least one of these subpopulations would evolve toward a better result (given at least a "critical mass" at each subpopulation).

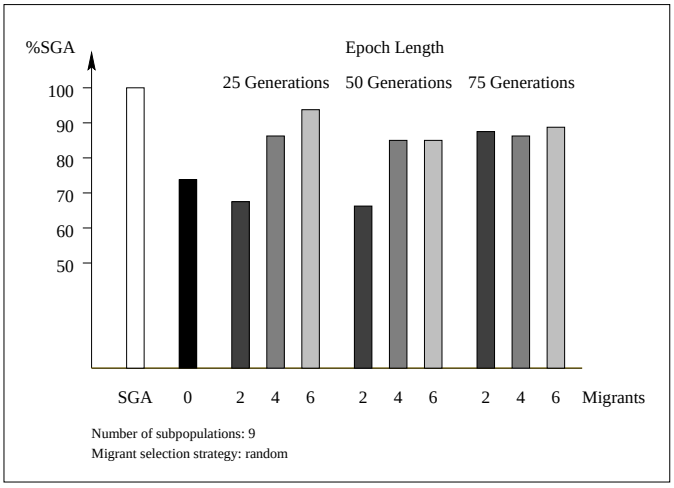

Figure C6.3.3. Comparison of results on the benchmark suite with different numbers of migrants and epoch lengths. Each bar is a Δ value for a different configuration. A Δ value is an average relative difference from the best-known solution as normalized to the Δ_{SGA} value, so the SGA bar is always 100%. Thus, the lower the bar, the better the average result of the particular configuration.

Figure C6.3.3 also shows that a *limited* migration between the subpopulation further enhanced the advantage of a parallel genetic algorithm. Two migrants to each neighbor with an epoch length of 50 generations are seen to be the best parameters when averaged over all problem instances.

On the one hand, more migrants or too short epoch lengths are counterproductive to the idea of disjointly and parallel evolving subpopulations. The resulting intermixing diminishs the genetic diversity between the subpopulations by "pulling" them all into the same part of the search space, thereby approaching the behavior of a single-population genetic algorithm. On the other hand, insufficient migration (epoch length 75 generations) simulates the isolated parallel approach (zero migrants) — the genetic richness of the neighboring subpopulations does not have enough chance to spread out.

Figure C6.3.4 shows this behavior in the context of individual subpopulations, that is, it presents the convergence behavior of the best individuals in each of the parallel evolving subpopulations on a specific problem instance (channel "Joo6_13"). It clearly indicates the importance of migration to avoid premature stagnation by infusing new genetic material into a stagnating subpopulation. The "stabilizing" effect of migration is also evident in the reduced variation among the best objective values gained in five independent runs, as shown in the righthand plot of figure C6.3.4.

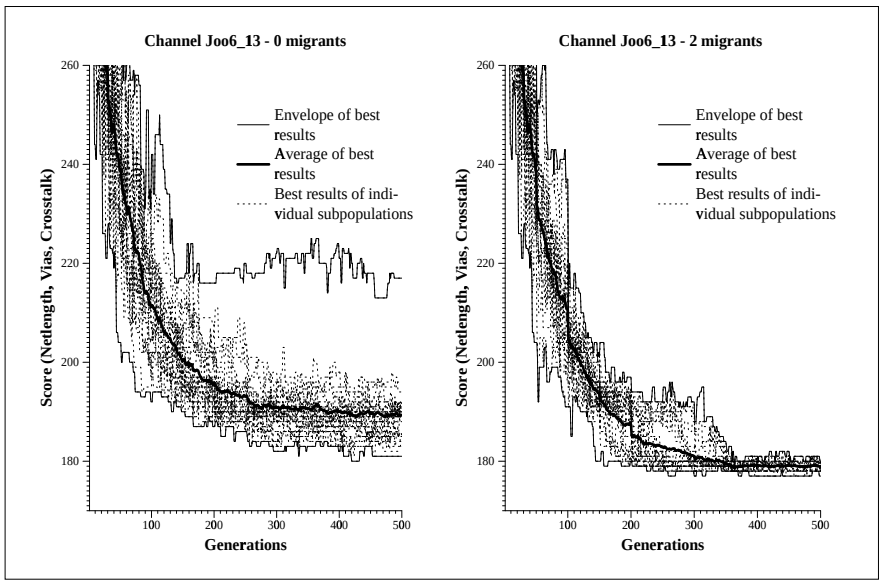

Figure C6.3.4. Comparison of the convergence of the best solutions in the individual, parallel evolving subpopulations. Plotted are five runs with nine subpopulations, i.e., 45 runs, in isolation (left) and with two migrants (right). (Note: the envelope for the plot on the left looks unusual due to an outlier subpopulation.)

Variable epoch lengths

The theory of punctuated equilibria is based on two main ideas: (1) an isolated subpopulation in a constant environment will stabilize over time with little motivation for further development, and (2) continued evolution can be obtained by introducing new individuals from other, also stagnating subpopulations. However, all known computation models that are based on this theory use a fixed number of generations between migration. Thus, they do not exactly duplicate the model that migration occurs only *after* a stage of equilibrium has been reached within a subpopulation.

The algorithm was modified to investigate the importance of this characteristic. Rather than having a fixed number of generations between migrations, a stop criterion was introduced that took effect when stagnation in the convergence behavior within a subpopulation had been reached. After some experimentation with different models, we defined a suitable stop criterion to be: 25 generations with no improvement in the best individual within a subpopulation.

To ensure a fair comparison, we kept the overall number of generations the same as in all other experiments. This lead to varying numbers of epochs between the parallel evolving subpopulations (due to different epoch lengths) and resulted in longer overall completion time.

The results achieved with this variable epoch length are shown in figure C6.3.5. The results suggest that a slight improvement compared with a fixed epoch length can be achieved by this method. However, it is important to note that this comparison is made with a fixed epoch length that has been shown to be the most suitable after numerous experiments (see figure C6.3.3). Thus, the important attributes to notice are that the variable-epoch-length configuration frees the user from finding a suitable epoch length and that it gave more consistent results over the various migration settings.

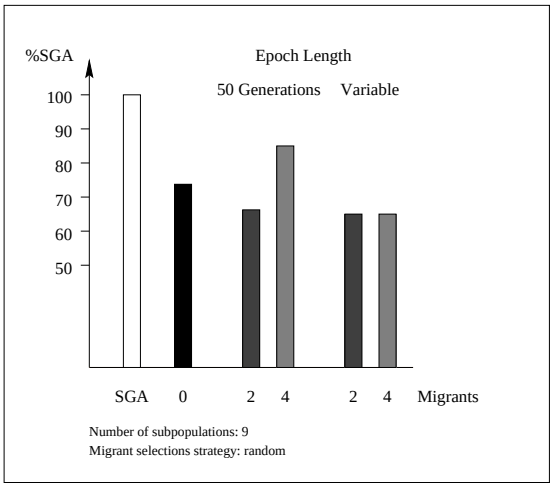

Figure C6.3.5. Comparison of results on the benchmark suite with fixed and variable epoch lengths. Variable length epochs were terminated after 25 generations of no improvement of the best individual within the subpopulation. Each bar is a Δ value.

Different migrant selection strategies

The influence of the quality of the migrants on the routing results was investigated using three migrant selection strategies: "Random" (migrants were chosen randomly with uniform distribution among the entire subpopulation), "Top 50%" (migrants were chosen randomly among the individuals with a fitness above the median fitness of the subpopulation), and "Best" (only the best individuals of the subpopulation migrated). The migrants were sent in a random order to the four neighbors.

As figure C6.3.6 indicates, we cannot find any improvement in the obtained results by using migrants with better quality. On the contrary, selecting better (or the best) individuals to migrate lead to a faster convergence — the final results are not as good as those achieved with a less elitist selection strategy. According to our observations, this is due to the dominance of the migrants having their (locally good) genetic material reach all the subpopulations, thus leading the subpopulation searches into the same part of the search space concurrently.

Different number of subpopulations

To compare the influence of the number of subpopulations, the size of the subpopulations were kept constant and the number of subpopulations increased from $N = 9$ to $N = 16$ and $N = 25$ (still connected in a torus). Accordingly, the population size and the number of recombinations of the SGA were increased to maintain a fair comparison. The resulting plots for SGA vs. 16- and 25-subpopulations (with $n = 50$) are qualitatively similar to the SGA vs. 9-subpopulations comparison of figure C6.3.3. For problems of the difficulty of these benchmark routing problems one should expect the SGA with slightly larger populations to do better than the small population SGA's. That expectation was indeed born out in these experiments. The important observation is that the island-model performance also increased, thus the relative performance advantage of the island model was maintained.

In an interesting variation on this experiment, the total population size, M, was held constant

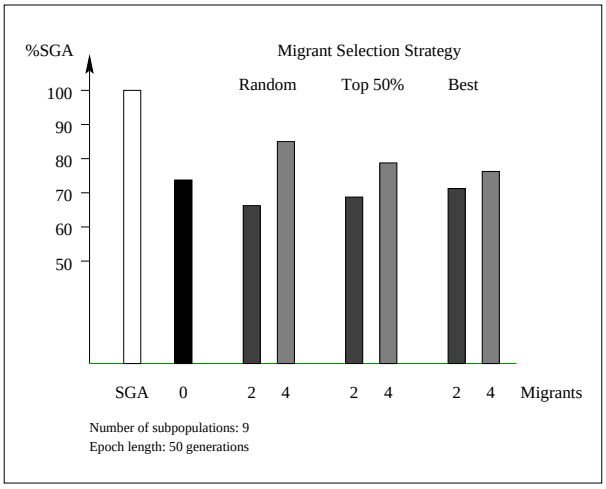

Figure C6.3.6. Comparison of results on the benchmark suite with different migrant selection strategies. Each bar is a Δ value.

while increasing the number of subpopulations (and so reducing the subpopulation sizes). Holding M near 450 and increasing N to 16 yielded $n = 28$, while increasing N to 25 yielded $n = 18$.

The results presented in figure C6.3.7 show that subpopulation size is an important factor. The figure clearly indicates that (for this average measure) partitioning a population into subpopulations yielded (for $N = 9$) results better than SGA, then yielded progressively worse results as N was increased (and n was decreased). We contend that this progression was due to the subpopulation size, n, going below "critical mass" for the specific benchmark problem instances. Remember, the plotted values are aggregate measures. When we looked at the component values for each problem instance we found further evidence for our contention. The evidence was that for the simpler benchmarks the $N = 25$ island model still had extremely good performance, in fact, equalled the best-known solution repeatedly. For the other, more complex benchmarks, the $N = 25$ island model performed very poorly, thus dragging the average measure well below the SGA performance level. Thus, the advantage of more varied evolving subpopulations can be obtained by increasing N only if n remains above "critical mass". This critical value for n is dependent on the complexity of the problem instance being solved.

C6.3.6 Final remarks and conclusions

In general it is difficult to compare sequential and parallel algorithms, particularly for stochastic processes such as genetic algorithms. As mentioned at the beginning of our discussion, the comparison is often according to overall time-to-completion, i.e., wall-clock time. We contend that the island model constitutes a different evolutionary algorithm, not just a faster implementation of the sequential genetic algorithm, and one that yields qualitatively better results. Here we have argued for this contention from the point of view of biological evolution and in the context of a difficult VLSI design application.

Several of the experimental-design decisions we made for the application experiments merit reiteration here. First, the application is an important problem with an extensive literature of heuristic systems that "solve" the problem. Our derived baseline measure incorporated the best-known objective values from this literature (see (C6.3.2)). For this particular VLSI design problem, most of the heuristic systems are deterministic, thus we have aggregate values for the various GA's versus single values for the deterministic systems. Our measure does not directly account for this but we have provided an indication of the variation associated with the set of runs for the island model.

Second, our comparisons to a sequential genetic algorithm (SGA) are based on "best-seen" objective values, not CPU time or time-to-completion. In order to make these comparisons fair, we

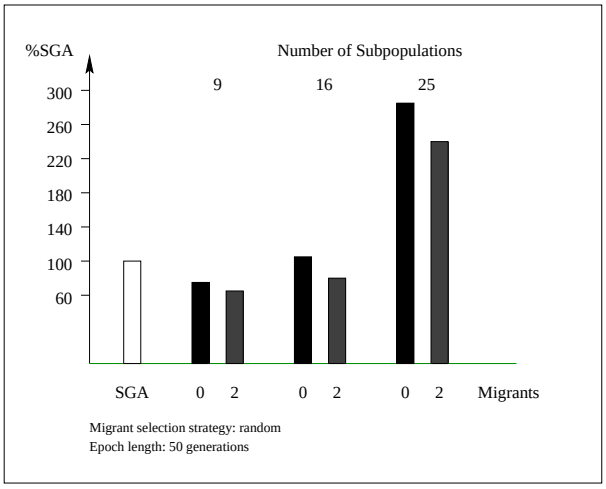

Figure C6.3.7. Comparison of results on benchmark suite with different numbers of subpopulations. The size of the total population, M, is kept constant at 450 individuals. Since $M = N * n$, the increase in N, the number of subpopulations, requires a reduction in n, the size of each subpopulation.

have endeavored to hold the computational resources constant and make them consistent across the SGA and the various configurations of the island model. This was done by fixing the total number of recombinations, i.e., the number of applications of the crossover operator, which relates directly to the total number of objective function evaluations.

Using the number of recombinations, as opposed to CPU time, for example, allows us to ignore properly implementation details for subsidiary processes, e.g., sorting or insertion into a sorted list (Garey and Johnson 1979). Note, a CPU-time measure can "cut both ways" between the serial and parallel versions. On the one hand, if a subsidiary process has a small initial constant but poor performance as the data structure size increases, then the island model has an advantage simply through the partition of the population. On the other hand, if a subsidiary process is increasingly efficient for larger data structures but has a large initial constant, then the SGA has the advantage, again, simply through the partition of the population. These are examples of the subtle ways by which time-based comparisons confound primary search effort with irrelevant implementation details.

Third, our comparisons have ignored the cost of communication. This is generally appropriate for island models because the isolated computation time for each subpopulation is extremely large relative to the communication time for migration (any course-grained parallelization should have this attribute). For medium- and fine-grained parallel models communication is much more of a real concern. Ignoring communication time is also reflective of our interest in the evolutionary behavior of the models, not the raw speed of a particular implementation.

With all GA's, the evolutionary behavior depends heavily on the interplay between the problem complexity and population size. For SGA vs. island model comparisons this is particularly problematic because the island model has two population sizes: the total population size, M, and the subpopulation size, n. Which is the proper one to consider relative to the SGA population size? For our experiments, we have used the total population size. We believe this makes the versions more conformable and is most consistent with number of recombinations as the computation measure.

Further, for comparing stochastic process, such as GA's, the total number of trials important, particularly when the evaluation measure is based on best-seen results. The attentive reader will have noticed that the strictly isolated island model (no migration) often does better than the SGA. This might seem curious, since a single run of the isolated island model is just a set of N separate SGA's and ones with smaller population sizes. As long as the subpopulations size, n, is above what we are calling the "critical mass" level, the isolated island model has a statistical advantage over the single SGA (under our evaluation measure).

The measure gives this advantage because the best-seen is taken over each run. Thus, each run of the isolated island model gets N "samples" to determine its best-seen, while each SGA run gets only one "sample". [Shonkwiler (1993) calls these "IIP parallel" GA's and gives an analysis of "hitting time" expectation.] We consider the "samples" in this case to be evolutionary trajectories through the solution space. Now, note that in almost all cases allowing migration provides the island model with the means to derive even better results.

This application of the island model to detailed routing problems in VLSI circuit design has shown that a parallel genetic algorithm based on the theory of "punctuated equilibria" outperforms a sequential genetic algorithm. Furthermore, the results are qualitatively equal to or better than the best-known results from published channel and switchbox routers.

In investigating the parameters of the island model, the following conclusions have been reached.

- The island model consistently performs better than the sequential genetic algorithm given a consistent amount of computation.
- The size of a subpopulation, the total amount of immigration, i.e., the number of connected subpopulations times the number of migrants per neighbor, the epoch length and the complexity of the problem instance are interrelated quantities. The problem instance complexity determines a minimum population size for a viable evolutionary trajectory. The total amount of immigration must not be disruptive [our experiments indicate that more than 25% of the subpopulation size is disruptive]. And the epoch length must be long enough to allow exploitation of the infused genetic material. Within those constraints the island model will perform better with more subpopulations, even while holding the total population size and the total number of recombinations, i.e., amount of computation, constant.
- Variable epoch lengths determined via equilibrium measures within subpopulations achieve overall results slightly better than those obtained with (near-)optimized fixed epoch lengths. Though an equilibrium measure must be chosen, allowing variable epoch lengths frees the user from having to select that parameter value.
- Quality constraints on the migrants do not improve the overall behavior of the algorithm, on the contrary, quality requirements on the selection of the migrants increases the occurrence of premature stagnation.
- Given a sufficient number of individuals per subpopulation, the larger the number of parallel evolving subpopulations, the better the routing results. The complexity of the problem and the minimal subpopulation size have a direct correlation that must be taken into account when dividing a population into subpopulations.

Finally, we would like to return to an issue that we mentioned at the very beginning of our discussions, namely, the island-model formulation of the genetic algorithm is not simply a hardware accelerator of the single population genetic algorithm. The island model does map naturally to distributed-memory message-passing multiprocessors, so it is amenable to the speed-up in time-to-completion that such parallel architectures can provide. However, the formulation can improve the quality of solutions obtained even via sequential simulations of the island model. As supported by the "shifting balance" and "punctuated equilibria" theories of natural evolution, the emergent properties of the computation derive from the *concurrent* evolutionary trajectories of the subpopulations *interacting* through limited migration.

References

Acan A Ünver Z 1992 Switchbox routing by Simulated Annealing: SAR *Proc IEEE Int Symp on Circuits and Systems* **4** pp 1985-1988

Adamidis P 1994 *Review of Parallel Genetic Algorithms* (Aristotle Univ of Thessaloniki: Dept of ECE) Technical Report

Bailey J 1992 First we reshape our computers, then our computers reshape us: The broader intellectual impact of parallelism *Daedalus* pp 67–86

Barnes G H Brown R M Kato M Kuck D J Slotnick D L Stokes R A 1968 The ILLIAC IV computer *IEEE Trans Computer* **17** (8) 746–757

Cho T W Sam S Pyo J Heath R 1994 PARALLEX: A parallel approach to switchbox routing *IEEE Trans Computer-Aided Design* **13** (6) pp 684–693

Cohoon J P Hegde S U Martin W N and Richards D S 1987 Punctuated Equilibria: A parallel genetic algorithm, *Proc Second Int Conf on Genetic Algorithms* pp 148-154

Cohoon J P Hegde S U Martin W N and Richards D S 1991a Distributed genetic algorithms for the floorplan design problem *IEEE Trans Computer-Aided Design* **10** (4) 483–492

Cohoon J P and Heck P L 1988 BEAVER: A computational-geometry-based tool for switchbox routing *IEEE Trans Computer-Aided Design* **7** (6) pp 684–697

Cohoon J P Martin W N and Richards D S 1991b Genetic algorithms and Punctuated Equilibria in VLSI *Parallel Problem Solving from Nature* Schwefel H P and Männer R (eds) Lecture Notes in Computer Science **496** (Berlin, Germany: Springer Verlag) pp 134–144

Eldredge N 1989 *Macro-evolutionary Dynamics: Species, Niches, & Adaptive Peaks* (New York: McGraw-Hill)

Eldredge N Gould S J 1972 Punctuated Equilibria: An alternative to phyletic gradualism *Models of Paleobiology* Schopf T J M (ed) (San Francisco, CA: Freeman, Cooper and Co) pp 82–115

Forrest S (ed.) 1991 *Emergent Computation* (Cambridge, MA: MIT Press)

Fung L W 1976 *MPPC: A massively parallel processing computer* (Greenbelt, MD: Goddard Space Flight Center) Section Report

Garey M R and Johnson D S 1979 *Computers and Intractability: A guide to the theory of NP-completeness* (San Francisco, CA: Freeman)

Geraci, M Orlando P Sorbello F Vasallo G 1991 A genetic algorithm for the routing of VLSI circuits, *Proc Euro ASIC '91* pp 218–223

Gerez S H Herrmann O E 1989 Switchbox routing by stepwise reshaping *IEEE Trans Computer-Aided Design* **8** (12) pp 1350–1361

Goldberg D E 1989 *Genetic Algorithms in Search, Optimization, and Machine Learning* (Reading, MA: Addison-Wesley)

Grimshaw A S 1993 Easy-to-use object-oriented parallel programming with Mentat *IEEE Computer* **26** (5) 39–51

Gorges-Schleuter M 1990 Explicit parallelism of genetic algorithms through population structures *Parallel Problem Solving from Nature* Lecture Notes in Comp Sci **496** (Berlin, Germany: Springer-Verlag) pp 150–159

Holland J H 1975 *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI: University of Michigan Press)

Huxley J 1942 *Evolution: The Modern Synthesis* (New York, NY: Harper and Brothers)

Joobbani R 1986 *An Artificial Intelligence Approach to VLSI Routing* (Boston, MA: Kluwer Academic Pub)

Lienig J 1996 A parallel genetic algorithm for two detailed routing problems *IEEE Int Symp on Circuits and Systems* pp 508–511

Lienig J Thulasiraman K 1994 A genetic algorithm for channel routing in VLSI circuits *Evolutionary Computation* **1** (4) pp 293–311

Lin Y-L Hsu Y-C and Tsai F-S 1989 SILK: A simulated evolution router *IEEE Trans Computer-Aided Design* **8** (10) 1108–1114

Mentat Homepage: "http://www.cs.virginia.edu/~mentat/"

Mühlenbein H 1989 Parallel genetic algorithms, populations genetics and combinatorial optimization *Proc Third Int Conf on Genetic Algorithms* pp 416–421

Mühlenbein H Gorges-Schleuter M Krämer O 1987 New solutions to the mapping problem of parallel systems — the evolution approach *Parallel Computing* **6** pp 269–279

Pettey C B Leuze M R Grefenstette J J 1987 A parallel genetic algorithm *Proc Second Int Conf on Genetic Algorithms* pp 155-161

Radcliffe N J 1991 Forma analysis and random respectful recombination *Proc Fourth Int Conf on Genetic Algorithms* pp 222-229

Seitz C L 1985 The Cosmic Cube *Communications of the ACM* **28** 22-33

Shonkwiler R 1993 Parallel genetic algorithms *Fifth Int Conf on Genetic Algorithms* pp 199–205

Slotnick D Borck W McReynolds R 1962 The SOLOMON computer *Proc Fall Joint Computer Conf* (AFIPS) pp 97–107

Tanese R 1987 Parallel genetic algorithms for a hypercube *Proc Second Int Conf on Genetic Algorithms* pp 177-183

Wright S 1932 The roles of mutation, inbreeding, crossbreeding and selection in evolution *Proceedings of the Sixth Int Congress of Genetics* **1** pp 356–366

Wright S 1964 Stochastic processes in evolution *Stochastic Models in Medicine and Biology* Gurland J (ed) (Madison WI: University of Wisconsin Press) pp 199–241

Wright S 1982 Character change, speciation, and the higher taxa *Evolution* **36** (3) pp 427–443

Wulf W A Bell C G 1972 C.mmp – A multi-mini-processor *Proc Fall Joint Conf* (AFIPS) pp 765–777

Yoshimura T Kuh E S 1982 Efficient algorithms for channel routing *IEEE Trans Computer-Aided Design* **1** (1) pp 25–35

