This document outlines a plan for optimizing an evolutionary algorithm (EA) designed to search for good solutions to IC floorplanning problems.

# 1 Problem Scope

**Objective:**

> *Design an EA that searches for IC floorplans with minimal area and wire length. Analyze the EA and identify those attributes that produce optimal search performance in terms of progression rate, escape from local optima, and final solution quality.*

The test problems will initially be some of the MCNC benchmarks, but other problems will be investigated as needed. Two problems are of interest: (1) hard module only placement and (2) hard module placement with one soft module.

The analysis of the EA is going to be somewhat difficult because conventional means (e.g., correlation in the fitness landscape) cannot be used. This is because conventional techniques work only on *direct encoded* solutions whereas the IC layout work done so far uses sequence pairs which are *indirect encoded* solutions. An example will illustrate the difference.

Suppose you want to find $x$ and $y$ values that maximize some given function $f(x, y)$. For simplicity assume $x$ and $y$ are integers in the range [0,255] (which means $x$ and $y$ are encoded as 8-bit binary strings).

We can form a 3-dimensional fitness landscape with $x$ in one dimension, $y$ in a second dimension and $f(x, y)$ the third dimension. The EA then explores the fitness landscape searching for the highest "peak". Each point in the $xy$-plane has 4 "neighbors". That is, the point $(x, y)$ has neighbors $(x - 1, y)$, $(x + 1, y)$, $(x, y - 1)$, and $(x, y + 1)$. We could investigate how rugged the landscape is by visiting neighboring points and comparing fitness values; rugged landscapes have radically different fitness values between neighboring points. Notice each solution is encoded as a 16-bit binary string (8-bits for $x$ and 8-bits for $y$). This is a direct encoding because, given an $x$ and $y$ value, you know exactly where to go on the fitness landscape and you know all of the neighboring solutions.

The IC work we have done to date uses sequence pairs. These are only an indirect encoding because you can't look at the sequence pair and then go to a point on a fitness landscape. These sequence pairs are actually data for a set of instructions that gives the module placements. In other words, you can't look at sequence pair and immediately identify its neighbors. Hence, there is no obvious way to test for ruggedness in the landscape.

So why is ruggedness in the fitness landscape important? The structure in the landscape provides valuable clues about the properties of the an effective search operator[1]. Consider a parent solution in a generation of the EA. We perturb this parent with a reproduction operator to create an offspring solution. If the perturbation was not large, then the offspring is physically near to the parent in the fitness landscape. This is exactly what we want if the landscape is rugged. On the other hand, if the landscape is smooth, neighboring solutions have the same relative fitness so the offspring can be much further away from its parent. Hence, the reproduction operation can make a large perturbation to the parent.

The bottom line is this: we are going to have to develop a new way of analyzing a fitness landscape for an indirect encoding. I am unaware of any work in this area[2].

One thing we should decide in the very near future is whether or not we want this problem formulated as a multi-objective optimization problem.

[1] The reproduction operators in an EA like mutation and crossover are the search operators.

[2] One reason is indirect encodings are rare. The only other one that comes to mind is homology studies of protein structure (structure is inferred from the peptide sequence)

## 2 Work Plan

Task 1: Get the EA up and running for an instance of an MCNC benchmark (probably a 48 module size).

> No attempt will be made initially to fine-tune the algorithm. This initial work involves hard modules only so any Lagrangian relaxation can be stripped out.

Task 2: Develop a fitness landscape analysis method for indirect encodings.

> I would recommend we have a student sign up for say 2 credits of ECE 505 (Reading & Conference) to do an extensive literature search. Once this is done a student could take some ECE 501 (Research) or ECE 506 (Special Project) credits to run some tests once we formulate some ideas.

Task 3: Exploit the fitness landscape analysis to fine-tune a set of reproduction operators for the EA.

> We should also investigate augmenting the EA with local search capability. It will be interesting to see if Lamarkian evolution or the Baldwin effect will improve the search quality.

Task 4: See how all of the above applies to a soft module problem.

> I'm guessing the fitness landscape analysis will apply, but I'm not positive. If it does change we would have to create a new set of reproduction operators. We should probably retain Lagrangian Relaxation.

## 3 Timeline

We can start Task 1 as soon as a student is identified. The student could come up to speed on the code and the problem definition. I could give them enough of a background on evolutionary computation to get them going.

Task 2 should probably wait until Spring 2005 so the student can sign up for not only my ECE 559 (Genetic Algorithms) but ECE 505 credits as well.

The other tasks should probably wait until after ECE 559 has been taken. Fitness landscape analysis isn't covered until late in the quarter.