# An Introduction to Evolutionary Computing in Design Search and Optimization

## 1. Introduction

- *Design Search and Optimization*
  · Life is becoming more computationally complex and in the field of engineering we are presented with an increasing number of computationally complex problems to solve.
  Ex) Ourselves, designing aircraft wings, trying to schedule some process in a factory or simply trying to fit curves to data
- *The Application of Evolutionary computing (EC) Algorithms*
  · The whole point to evolutionary computing is the creation of designs, particularly in the domain of complex problems and environments.
- *Problem Solving Environment (PSE)*
  · Designing complex things requires us to put lots of things together.
  · Some type of search and optimization routine with resource management
- *Search and Optimization*
  · Optimization is about trying to walk up a hill and hopefully, through effectively searching the landscape, walking up the highest hill (Figure 1).
  · How do we know we are at the top of the hill and how do we know it is the right hill?
- *Resources*
  · Computing power dominates this process.
- *Current Techniques*
  · Some sort of hill-climbing optimization and parallel computation is common.
  · Evolutionary search is not generally in everyday use.
- *Representation*
  · People represent their designs with numbers, so the method of encoding between the two is important.
  · An engineer also has a selection of models which can be used which range in sophistication and cost.

## 2. A Brief Overview of Evolutionary Computing

- *History*
  · In 1960s, the introduction of ideas and techniques such as GA, ES, and EP
  · All these EC techniques use the same metaphor of mapping problem solving onto a simple model of evolution:

| EVOLUTION | | PROBLEM SOLVING |
|---|---|---|
| Individual | ↔ | Candidate Solution |
| Fitness | ↔ | Quality |
| Environment | ↔ | Problem |

- *Overview*
  - We have a population which we want to evolve from time step $t$ to time step $t+1$ by selection, recombination and mutation.
  - Introducing diversity in the designs through mutation and recombination, and then exploit this exploration through selection of better solutions
  - Through the correct choice of operators, we achieve a balance between exploration and exploitation and our solutions steadily get better.
  - There tends to be a large diversity at the beginning and very little towards the end.
- *Some General Points about the Advantages of EC Techniques*
  - Widely applicable
  - Low development and application cost
  - Easily incorporated into other methods
  - Solutions are interpretable
  - Can be run interactively and allows incorporation of user-proposed solutions
  - Provide many alternative solutions
- *Several Disadvantages*
  - No guarantee for optimal solution within finite time
  - Weak theoretical basis
  - May need parameter tuning for good performance
  - Often computationally expensive and thus slow

## 3. Evolutionary Computing in Design Search and Optimization

- *Representation*
  - Deciding on a good representation is fundamental to the performance of EC techniques.
  - We can trade-off the ability of any representation to be compact against its ability to represent all possible designs.
  - We need a domain expert and an EC expert together.
- *Constraints and Multiple Objective*
  - Often the objective of a design are not completely defined.
  - An engineer will often have multiple objectives and in some sort of hierachy.
  - How do we handle a list of certain criteria which invalidate a particular design?
- *Mutation*
  - Making local steps in our landscape, that is, making small changes to our design and not leaping to a radically different solution
- *Recombination*
  - The combination of parts of one parent with the complementary parts from another
  - Recombination or crossover allows parents to pass on some of their characteristics to their children.
  - Recombination tends to be seen as more of an exploitation and mutation as more of exploration operator.

- *Niching*
  · We many want to impose some form of niching techniques to allow the population to divide into smaller sub-populations, each focused around a different part of the problem space.
- *Repeatability and Elitism*
  · EC techniques are stochastic and thus no two runs will necessarily produce the same results.
  · Do we just want to get one very good solution or do we want a good solution every time?
  · A good solution once found within the population may be lost later in the evolution.
  · To keep a record of the best solutions, elitism strategies ensure that these solutions stay in the population.
- *When to Stop?*
  · Without any prior knowledge, it is impossible to tell whether the best solution has been reached.
  · The decision most often comes down to one of time and computing resources.

## 4. An Example of EC Design

- *Boom Design (NASA Mission)*
  · The objective was to use EC techniques to design the geometry of the beam such that the vibrational noise does not travel through it.
  · Combining computationally expensive structural analysis routines with an optimization technique.
  · Population size of 300, 15 generations: the complete run took three weeks of computation using 11 parallel workstations.
  · Three decades of improvement in the vibration performance (Figure 5)

## 5. Conclusions

· Evolutionary computing algorithms have been shown to be of great value in design search and optimization.
· The concept of a problem solving environment clearly increases the utility of these algorithms but the problems of computational expense must be addressed.