Chapter 1

Introduction

Evolutionary Computation ("EC") is the study of computing techniques based on the guiding evolutionary principle of "survival of the fittest." Evolutionary Algorithms ("EAs") are powerful generic search algorithms capable of finding good solutions to complex problems. Some example areas in which EAs have been applied for problem solving and modeling include: optimization, automatic programming, machine learning, economics, immune systems, ecology, population genetics, evolution and learning, and social systems (see [Goldberg, 1989], [Ross and Corne, 1994], [Alander, 1995] and [Mitchell, 1996] for examples).

The problem with EC is that people find it difficult to understand the evolutionary search behaviour of their algorithms. Although searching the problem space by simulated evolution biases the search toward the better regions of the problem space, hundreds, if not thousands, of solutions are considered during a typical EA's execution. Summary statistics can be used to give an impression of the algorithm's evolution, such as the best, average and worst quality of the solutions contained in each population. However, at the beginning of this project there were no methods capable of supporting the EA user's comprehensive understanding of their algorithms' evolutionary search behaviour.

The primary objective of Software Visualization ("SV") is to facilitate peoples' understanding and effective use of computer software [Price et al., 1993]. This has been used successfully to illustrate the operation of programming languages [Eisenstadt and Brayshaw, 1987], [Reiss, 1990], [Lieberman and Fry, 1995], computer algorithms [Brown and Sedgewick, 1985], [Szasko, 1989], [Brown, 1991], and the effects of a program on a dataset [Moher, 1988], [Roman et al., 1992]. This
thesis explores how SV technology may be applied to support EC. This chapter introduces the two main themes of this work; Evolutionary Computation and Software Visualization, and explains the motivation, research approach, contributions and structure of this thesis.

1.1 Evolutionary Computation

*Evolutionary Computation* is a rapidly expanding area of artificial intelligence research, with more than twenty international events per year and at least half a dozen journals, over a thousand EC related papers are published per year [Schwefel and Kursawe, 1998].

Within EC there are three classes of EA; Evolutionary Programming, Evolution Strategies, and Genetic Algorithms. These classifications are based on the level in the hierarchy of evolution being modeled by the algorithm. Evolutionary Programming ("EP") models evolution as a process of adaptive species. Evolution Strategies ("ESs") models evolution as a process of the adaptive behaviour of individuals. Thirdly, Genetic Algorithms ("GAs") models evolution at the level of genetic chromosomes i.e. the basic instructions for making things.

EAs do not necessarily locate the optimal solution to a problem, the advantage of EAs is that they find “acceptably good” solutions to problems “acceptably quickly” [Beasley et al., 1993]. In their overview of GAs Beasley, Bull and Martin note that “where specialized techniques exist for solving particular problems, they are likely to out-perform GAs in both speed and accuracy of the final result” [Beasley et al., 1993, page 58]. It is in difficult areas where no such techniques exist that EAs should be applied. In these areas, the size of the problem space is such that an exhaustive search is impractical, and the structure of the problem space is such that traditional search algorithms are ineffective. EAs excel by striking a balance between the continued exploration of the problem space and the exploitation of the useful components held in the solutions discovered so far.

1.2 Thesis Motivation

The problem with EC is that EAs search large problem spaces by making gradual improvements to a set of possible solutions. There is no single point during the algorithm’s run that can be held responsible for the outcome, the solutions emerge during the course of the algorithm’s iterations.
This results in a fundamental credit assignment problem for EA users i.e. if good solutions are found what proportion of the credit should be attributed to the individual components of the algorithm's design?

This problem is further compounded by the fact that the users are unable to see the EA's search behaviour. EA users commonly examine how the quality of the solutions found by their algorithm changes over time using a graph of the population's fitness versus generation number. Although this graph illustrates the improvements in the quality of the solutions considered during the algorithm's run, it does not illustrate anything about the structure of the solutions being considered, or the regions of the search space being explored.

The aim of this project is to address this fundamental design problem by applying software visualization techniques to enable the user to examine the structure of the solutions being considered and the regions of the search space being explored. By enabling the user to see the search behaviour of their algorithms, they can then begin to attribute credit to the individual designs and judge the quality of each algorithm based on its exploration of the problem space.

1.3 Software Visualization

*Software Visualization* ("SV") has been defined as “the use of the crafts of typography, graphic design, animation and cinematography with modern human-computer interaction technology to facilitate the human understanding and effective use of computer software" [Price et al., 1993]. Visualization is specifically intended to enable the user to interact with, as well as observe, their data [McCormick et al., 1987]. A recent empirical evaluation of SV found that students who were able to control and interact with a variety of algorithm animations gained a better understanding of the algorithms' behaviour than those who could only passively observe the visualizations [Lawrence et al., 1994].

The application of visualization techniques to support peoples understanding of EAs has been receiving growing attention during the last few years; [Kapsalis and Smith, 1992], [Routen and Collins, 1993], [Chipperfield et al., 1994], [Nassershariif et al., 1994], [Dabs and Schoof, 1995], [Dybowski et al., 1996], [Harvey and Thompson, 1996], [Collins, 1997]
CHAPTER 1. INTRODUCTION

Motivating Problem:
- "Black box" approach to GA application
- User's need to see the GA's search behavior

Investigate the Problem:
- GA user survey
- Investigate available visualization support

Examine Possible Solutions:
- High dimensional mapping techniques
- Extendable frameworks

Development:
- Search space visualization techniques
- Henson EA visualization development framework

Implementation:
- Gonzo - GA search space visualization tool

Figure 1.1: An overview of the research approach taken in this project.

and [Shine and Eick, 1997]. By enabling EC users to observe and interact with EAs it is hoped that a better understanding of their behaviour will be achieved.

1.4 Research Approach

This section describes the research approach taken in this project. There were essentially five stages in this project. Figure 1.1 illustrates each stage; from the initial problem description, through the investigation of the problem, the examination of some possible solutions, and the development of specific solutions, to the implementation of an example (proof of concept) visualization tool.

At the start of this project the decision was taken to examine GAs as a specific case study of EC visualization. A case study approach was considered important for this project, in order to identify the unique visualization requirements of a specific EA, as well as the generic visualization requirements of EC. Although all three types of EA are based on a common metaphor, the slight differences between their use of that metaphor results in significant differences in their implementation and application.

As will be seen in the following chapter, GAs are significantly different to both ESs and EP, in that they emphasize genotypic rather than phenotypic transformations, i.e. the level in the evolutionary hierarchy at which GAs operate is very different to that of ESs or EP. However, the purpose of this thesis is to explore how visualization technology may be best applied to EC. Therefore, it is the generic approach to the provision of SV support for EC and the identification of generic EC, and specific GA, visualization techniques that are of importance. GA visualization was chosen as the specific EC domain in order to build on the existing body of work, see Section 4.1.

The motivating problem with GAs (as described in Section 1.2), is that GAs are difficult to apply because the user currently has no way of seeing the GA's exploration of the problem space. The
validity, importance and consequences of this problem were investigated through a GA user study. The study questionnaire was completed by nineteen GA users with a diverse range of reasons for using GAs. The responses to the questionnaire were used to establish a set of GA users' visualization requirements, and the contributions made by the available visualization support for fulfilling these requirements was examined.

Based on these results, a proposal was made to develop search space visualizations to support a user's understanding of the GA's search behaviour, and an extendable GA visualization framework with which GA users could develop their own visualizations. The development of the search space visualizations involved the investigation of a number of multivariate scaling techniques to produce two dimensional representations of the chromosomes in the GA's high dimensional search space. The "Henson" visualization framework was produced to support the development of GA visualization tools. Finally, an example GA search space visualization tool called "Gonzo" was implemented using Henson.

1.5 Thesis Contributions

This thesis makes the following contributions:

- Software Visualization

  1. "Henson," a framework for developing GA visualization tools.

- Evolutionary Computation

  1. A set of GA users' visualization requirements.

  2. The development of a set of high dimensional search space visualizations suitable for exploring a GA's search path.

  3. "Gonzo," a GA visualization tool for exploring the evolutionary search behaviour of GAs.

1.6 Thesis Overview

The objective of this project is to examine how SV techniques may be most effectively applied to support peoples understanding and use of EC. A visualization framework has been developed and
applied to produce a set of GA visualizations as a case study in EC visualization.

An overview of EC is given in Chapter 2. Chapter 3 presents the findings of a study carried out in order to identify the working practices of GA users, the difficulties they experience while applying GAs and their opinions regarding the potential use of visualization. The results of the study identify a set of GA users’ visualization requirements. The degree of support available from the existing work done on visualizing GAs is then explored in Chapter 4 along with some relevant contributions from the SV, information visualization, and human-computer interaction communities.

Chapter 5 discusses the design rationale used in this project. Specifically, it looks at the graphic design principles used, the advantages of using an extendable framework for developing GA visualization tools over a closed GA visualization system, and a series of visualization techniques for presenting the search path of a GA. Chapter 6 introduces “Henson,” a framework for developing GA visualization tools. Chapter 7 presents “Gonzo,” an example GA visualization tool implemented using the Henson framework, that supports the user’s perception of a GA’s search path. Chapter 8 critiques some of the work presented here and concludes this thesis with a summary of the contributions made and a speculative discussion of future work.