UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 700 Proposal

12017800, Marius Riekert (supervisor), Prof. Mardé Helbig (supervisor)
May 1, 2017
Department of Computer Science
University of Pretoria

Deriving Functions for Pareto Optimal Fronts Using Genetic Programming

KEYWORDS

genetic programming, expression trees, pareto optimal fronts

ABSTRACT

Genetic Programming is a specialized form of Genetic Algorithms that evolve trees. This paper will attempt to evolve expression trees, which is an NAry tree that represents a mathematical equation, in order to find a function that describes a given set of points in some space. The points will be the result of a Multi Objective Optimization Algorithm (MOOA) which lay on the so called pareto optimal front. The pareto optimal front is a curve in a multi dimensional space that describes the boundary where a single objective in set of objectives cannot improve more without sacrificing the optimal value of the other objectives. The algorithm, in this paper, will thus find the mathematical function that describes a pareto optimal front after a MOOA has solved a problem. This will assist in finding other points on the pareto optimal front that was not discovered by the MOOA.

Deriving Functions for Pareto Optimal Fronts Using Genetic Programming

120 178 00*, Riekert, M†(supervisor) and Prof. Helbig, M‡(supervisor)
Department of Computer Science
University of Pretoria
Email: *12017800@tuks.co.za, †mriekert@cs.up.ac.za, ‡mhelbig@cs.up.ac.za

***Index Terms*—genetic programming, expression trees, pareto optimal fronts**

***Abstract*—Genetic Programming is a specialized form of Genetic Algorithms that evolve trees. This paper will attempt to evolve expression trees, which is an NAry tree that represents a mathematical equation, in order to find a function that describes a given set of points in some space. The points will be the result of a Multi Objective Optimization Algorithm (MOOA) which lay on the so called pareto optimal front. The pareto optimal front is a curve in a multi dimensional space that describes the boundary where a single objective in set of objectives cannot improve more without sacrificing the optimal value of the other objectives. The algorithm, in this paper, will thus find the mathematical function that describes a pareto optimal front after a MOOA has solved a problem. This will assist in finding other points on the pareto optimal front that was not discovered by the MOOA.**

I. Introduction

Optimization is a field in computer science that deals with finding the best possible value that satisfies some conditions. The best value can either be the maximum or the minimum. Using the simple parabolic equation $f(x) = (x+1)^2+1$ as an example, the optimum value would be $x = -1$ (if we are using a minimization function), since the smallest value for f is 1, and $f(-1) = 1$.

Real world problems often do not contain a single objective that needs to be satisfied, but is rather described by multiple conflicting objectives (Konak et al., 2006). For instance the price of a car vs comfort. Choosing a cheaper car results in a decrease in comfort and vice versa. These two objectives are in conflict since an optimization in one results in a less optimal value in the other.

These problems are usually solved by Artificial Intelligence (AI). A popular and useful AI for finding a set of optimal solutions (a set of points where one objective cannot be optimized more without sacrificing the other objective), called a Pareto Optimal Front (POF), is Genetic Algorithms (GA) (Deb and Sundar, 2006). Since this algorithm, as well as Particle Swarm Optimization (PSO), has a set of individuals that can traverse the search space independently from each other (Konak et al., 2006). This means that each individual, or group of individuals, can find a different solutions along the POF (Konak et al., 2006).

A Pareto Optimal Point (POP) is, according to Marler and Arora (2004), a point in the search space where there does not exist another point which dominate it. A point, $\mathbf{x^*}$, is dominant if and only if (iff) $\forall f \in \mathbf{F}, \forall \mathbf{x} \in \mathbf{X}, \exists g \in \mathbf{F}$ such that $f(\mathbf{x^*}) \leq f(\mathbf{x})$ and $g(\mathbf{x^*}) < g(\mathbf{x})$, where $\mathbf{F}$ is the set of optimization functions. A complete set of these POPs is then collectively called the POF.

Figure 1 shows an example of a POF.

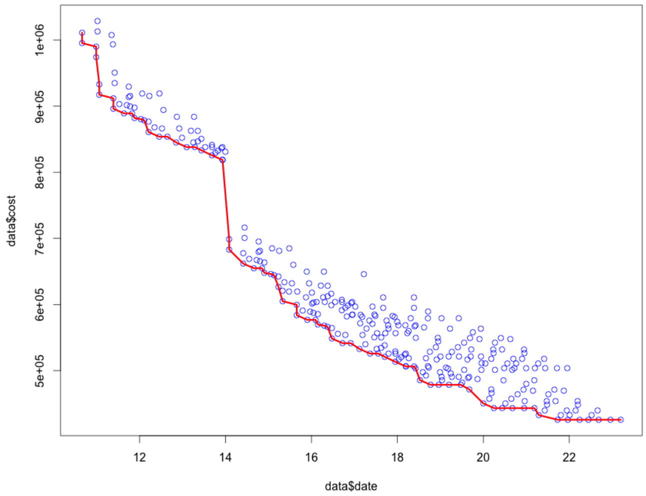

Figure 1. Pareto Optimal Front Example (https://i.stack.imgur.com/5KLU2.png)

The algorithms that solve the Multi Objective Optimization Problems (MOOP) often produces an incomplete set of POPs rather than a mathematical formula that represents the entire POF. The objective of this research is thus to find a mathematical formula that describes the points on the POF. This attempted by using Genetic Programming (GP) to evolve expression trees. This expression tree can then be used to find other points on the POF that was not found by the optimization algorithm.

II. Problem Statement

The main objective of this research will be to find a method of using GP to evolve functions for a POF, based on a set of

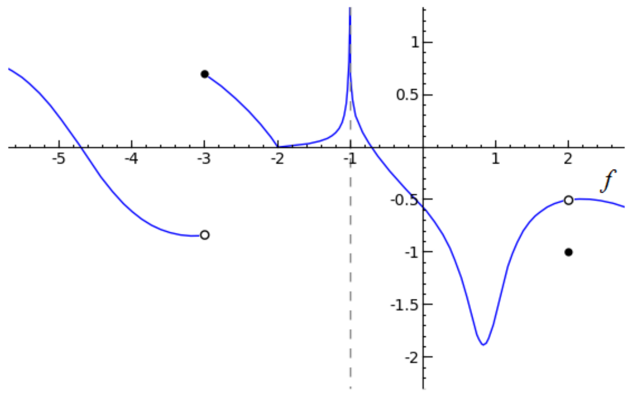

Figure 2. Conditional Equation (http://www.sagemath.org/calctut/calctut-pix/onesided09.png)

given points.

The first milestone of this project would be to determine the representation of the genotypes of the individuals. Also the crossover and mutation operations that will be used to evolve the individuals.

The next milestone will be to find a function that accurately describes the POF for two objectives. This paper will not look into finding functions for problems with more than two objectives.

The last milestone is to find a function for the POF that can include conditional equations.

III. Literature Survey

Even though some tools exist to find equations from provided points (like GeneticSharp), there are clear gaps in the research done in this field. Especially when focusing on the multi objective and POF part of this problem. The existing tools are also not suited well for solutions where a POF is not continuous, or contain conditional functions like the example in equation 1 or figure 2.

$$f(x) = \begin{cases} \frac{x^2-x}{x}, & \text{if } x \geq 1 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Ferreira (2002) elaborated on the Gene Expression Programming (GEP) algorithm in order to find the function that describes a given set of data points in a multi dimensional search space. These "function finding algorithms" will be carried over to the MOOP field and applied to the discovery of POFs based on provided POPs.

Ferreira (2002) uses arrays to represent the trees. This poses a problem for problems that have long equations and thus require large trees. In In the book *Genetic Programming: an introduction* by Banzhaf et al., it is suggested to have a tree of unbounded sizes. The size of the tree can then be regulated by adding a penalty to larger trees. Another technique is to only grow the tree as soon as no better solution can be found (Banzhaf et al., 1998).

Engelbrecht (2007) discusses GP, which is a specialized version of GAs for evolving trees. Here he discusses the various techniques for evolving tree structures, including the various mutation and cross over techniques used in GAs. Koza (1992) had a very large contribution to GP by applying the concept to optimization and search problems.

Expression trees used in GP consists of two node types. The first is function nodes, or non-terminal nodes, and the second is leaf nodes, or terminal nodes (Engelbrecht, 2007). Function nodes refers to mathematical functions, like addition, subtraction, multiplication, etc. Each of these require one or more input parameters. They thus have a subtree. Leaf nodes consist of variables and constants, like x_i or 10. These nodes do not have subtree.

IV. Methodology

A variety of research methods will be used to conduct the research. These include a literature survey, the development of an algorithm, the conduction of experiments, and arguments.

A. *Literature Survey*

In order to assess the gap in available research in this field and in order to gain a better, more in depth understanding of the problem at hand a literature survey was required. The result of this survey can be seen in section III.

B. *Algorithms*

Since the main purpose of this paper is to provide a new algorithmic approach to find mathematical functions for points in a 2D space, the found algorithm will act as the core of the research process.

C. *Experiments*

The effectiveness of this algorithm will be determined by how well and consistently the algorithm can find equations for a wide variety of POPs. The difficulty of the equations will also be used as a measurement of the algorithm's performance. This will be used to determine what crossover and mutation operators suit the problem the best.

D. *Arguments*

In order to further support the results obtained in the experiments an argumentative approach will be used to solidify the findings.

E. *Extended Literature Review*

This will not be used as it has been found that the area that this research paper will address has very little research done.

F. Models

Although some models will be used to explain the structure of a genotype of an individual, it will not be used as one of the main research method since this paper will be more focused on the algorithm that will be required for solving the problem.

V. Project Plan

A. Literature Survey

During the first part of the project the various individual components will be the main focus. Gaining additional knowledge of different approaches to solve these components will prove valuable when the research has to tie all the components together.

The results from the literature survey done in this proposal will be used as a basis and will be expanded on a bit more.

B. Design and Planning

The next phase will involve the design of the program. This will be used to simplify the implementation phase by finding potential problems ahead of time. Some important things to discuss here is:

- Representation of a genotype within the GA.
- Various crossover techniques that should be tested.
- Various mutation techniques that should be tested.

C. Implementation

This phase will deal with the actual development of the algorithm. During this phase there will be several sub-phases which was discussed in section II. A brief overview of this is as follows:

- Develop the GA to find a function for a two objective MOOP.
- Modify the algorithm to be able to accommodate conditional expressions.

D. Experimentation

Run the algorithm for various different sets of input data and observe the outcome. Should the results not be satisfactory, adopt the crossover and mutation techniques to find the best configuration for these.

E. Documentation and Conclusion

Document all the experiment results and attempt to find a solution to the original problem by discussing and arguing about the findings.

F. Submit to Supervisors and Make Changes

The final stage will be to have the supervisors read through the paper and suggest any improvements that can be made.

G. Schedule

The schedule that will be followed can be seen in table I.

Task	Start Date	End Date	Days Allocated
Literature Survey	13/03/2017	27/03/2017	14
Research Proposal	21/04/2017	08/05/2017	17
Design and Planning	09/05/2017	29/05/2017	21
Implementation	30/05/2017	09/09/2017	109
Testing	10/09/2017	23/09/2017	14
Experiments	24/09/2017	21/10/2017	28
Finalization of Paper	22/10/2017	05/10/2017	14
Supervisor Review	06/11/2017	14/11/2017	15
Make Changes	15/11/2017	23/11/2017	7
Submission	24/11/2017	24/11/2017	1

Table I
The schedule that will be followed.

References

Wolfgang Banzhaf, Peter Nordin, Robert E Keller, and Frank D Francone. *Genetic programming: an introduction*, volume 1. Morgan Kaufmann San Francisco, 1998.

Kalyanmoy Deb and J Sundar. Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 635–642. ACM, 2006.

Andries P Engelbrecht. *Computational intelligence: an introduction*. John Wiley & Sons, 2007.

Candida Ferreira. Gene expression programming in problem solving. In *Soft computing and industry*, pages 635–653. Springer, 2002.

Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

John R Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.

R Timothy Marler and Jasbir S Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.