Visualising Ant Colonoy Optimisation

Version: 1.0 Draft
G400 Computer Science, CS39440

Author: Christopher Edwards
che16@aber.ac.uk

Supervisor: Neil MacParthalain
ncm@aber.ac.uk

An outline project specification for a Computer Science Major Project

Department of Computer Science
Aberystwyth University
Wales
February 2, 2015

1 Project Description

The ultimate aim for the project is to produce an efficient and intuitive graphical user interface for Ant Colony Optimisation, allowing for non-dynamic modification of environmental variables and reflecting changes in the algorithms behaviour to the user. This project will only be classified as a success if this graphical user interfaces very simple to use. The project has strong potential to be used as a teaching aide for artificial intelligence, therefore the product must be simplistic and effective to maximise the potential user base, and cater for different levels of intelligence with respect to the subject.

During my research into the background of this project I found very few resources which visually represented Ant Colony Optimisation and its behaviours as well as allowing for user interaction . This is the main focus of the project, and is essential in achieving a worthwhile and successful project. The underlying algorithm will itself have some impact on the capabilities of this interface as using an existing implementation of the algorithm will mean there is far less flexibility in terms of algorithm modification and interaction when compared to a self-implemented approach.

At the end of the project there should be a suitable environment which allows a user of any background, with or without prior knowledge of the algorithm to modify the algorithms variables through a graphical user interface which visually displays the algorithms execution given the user-defined values. Therefore the main substance can be deducted to the graphical user interface and the interaction between itself and the underlying algorithm and architecture.

2 Proposed Tasks

2.1 Research

2.1.1 Language

The language choice for the algorithms implementation and graphical user interface is a very important decision. I will need to investigate which language(s) most suite the task at hand with respect to all aspects. I will also need to investigate the potential for external libraries to have a major influence in the language choice and final outcome as well as the design and implementation of the application code.

Design patterns should also be considered in conjunction with the choice of language, this regards the identification of potentially useful patterns and their implementation complexity given the merits of each language.

2.1.2 Background

Given that I am no expert in the topic I will need to research in depth the exact metrics and characteristics which will need to be modelled both visually and mathematically in order to completely my project to a professional level. This will involve studying what the algorithm is designed to do and how ants *pseudo-randomly* traverse the state space, as well what each of the environmental variables mean. As these variables are key to the applications purpose, the level of understanding is fundamental to the success of the project as validation and boundaries must be correctly placed.

2.2 User Interface

A large proportion of my resources needs to be correctly allocated to designing and implementing a suitable and effective user interface. In order to do this there will need to be several design iterations based upon research into existing systems of similar functionality as well as delving into research regarding user interaction methods and preferences. This will allow for the most suitable environment for the intended purpose.

This interface could be subject to potential intermittent reviews which will reflect the current effectiveness of the current design or in fact they could be used as a testing mechanism for new ideas. Ultimately discussions of some form will be needed from the end users of the product, removing bias out of the equation.

2.3 Algorithm

Another major resource consumer will be the implementation of the underlying algorithm. This will also include any considerations or design choices which will be subject to my understanding and any background research. Algorithm efficiency is key to the projects success so there must be suitable optimisation methods in place, which could include taking advantage of multi-threading on in fact re-using existing implementations of the algorithm such as the implementation provided by WEKA (Waikato Environment for Knowledge Analysis, provides open source java solutions), such implementations would be provided under the certain thus my application could therefore become subject to any license and its terms.

There is also the consideration of starting from scratch. Creating my own implementation the way I understand this will depend on the quality or suitability of existing solutions, and how complex they are in terms of interacting with them from a graphical user interface and visualising their execution. I will need to research and experiment with different approaches on a smaller scale and decide on the most appropriate method for production.

3 Proposed Deliverables

Fully Functional User Interface - A fully functional graphical front end will allow for user interaction and visualisation of the Ant Colony Optimisation algorithm. This interface will be simplistic, whilst maintaining a focus on usability and will encourage the user to experiment with different environmental variables for the algorithm.

Fully implemented Algorithm - There will be an underlying implementation of an Ant Colony Optimisation algorithm. The algorithm will be efficient and allow for its environmental variables to be easily modified with correct boundary conditions. These modifications will come from the user interface described above.

A Comprehensive Test Suite - There will be a set of unit tests that will sufficiently test the code logic for both the algorithm and the user interface. There will also be a series of black box tests which will be used to test the application's overall logic.

Proposed Project Specification - There will be a document detailing the proposed project functional requirements whilst also detailing any abstractions of the implementation such as any Unified Modelling Language (UML), Pseudo code or any details about the underlying algorithm(s).

Progress Report - Documentation detailing the current project situation given the current date. This will also state any changes that have been made to the proposed specification as well as any complications which have become apparent. Details of any proposed changes or potential complications in the future will also be flagged up in this document if they are in fact relevant at the time.

Intermittent Releases - Several versions will be released during the development process. Each version will be used to test the current implementation of features in order to catch any bugs or complications at their earliest stage reducing the complexity and the costs of resolving them at a later date. Each release will be a significant improvement on the previous edition there will not be a new release for every single feature that is implemented.

Final Report - A full report detailing the development process including any research as well as detailing any changes to the proposed specification, problems during development and any design or rationale will be produced. The document will collate all of the projects underlying ideas and concepts providing a detailed understanding of how the system functions and how the output is produced. The results of the projects testing phase will be present, as well as suggestions to further expansion opportunities and the potential significance of these.

References

[1] Eric Freemnan, Elisabeth Freeman, Bert Bates, Kathy Sierra. *Head First Design Patterns.* O'Reilly., Sebastopol, CA, first edition, sep 2004.

This book is used to give some background into the potential use of certain design patterns in the applications code. Considering the use of such patterns enabled the project to be much more maintainable in the future as well as making the design and development processes much simplier.

[2] Thomas Jungblut. "Ant Colony Optimization for TSP Problems" [Blog entry]. Available: http://codingwiththomas.blogspot.co.uk/2011/08/ant-colony-optimization-for-tsp.html, 30 Aug 2011. [Accessed: 26 Jan 2015].

This Blog posted by Thomas Jungblut covers multiple topics which are of high interest to the projects research. The information contained in the blog covers items like capitalising on multi-threading opportunities and how to efficiently do so. The underling calculations and evaluation procedures for each agent are also discussed. This is useful for modelling and design of the algorithm itself.

[3] Marco Dorigo et al.,. *Ant Colony Optimization and Swarm Intelligence.* Springer., Berlin, Germany, fifth edition, sep 2006. pp. 1-6.

This book is quite complex, possibly beyond the scope of this project. However the concepts are still relevant to my project research and development. The book itself is 250 pages in length, but the first few pages are sufficient for increasing background knowledge in regards to the projects problem.

[4] Weka 3: Data Mining with Open Source Machine Learning Software in Java. Weka 3: Data Mining Software in Java, [online] Jan 2015. Avaliable: http://www.cs.waikato.ac.nz/ml/index.html. Accessed: 26 Jan 2015.

WEKA provide an Ant Colony Optimisation library thus I have researched into what they provide and the potential usage of said libraries. However, any code provided by WEKA is subject to the GNU Public License if the project were to use such libraries then the project's source would also become subject to the GNU Public License terms. The use of these resources maybe limited to studying how thier implementations work and extracting concepts or ideas from those and developing a custom solution based off of the extracted knowledge.