

Visualising Ant Colony Optimisation

Version: 1.0 Draft
G400 Computer Science, CS39440

Author: Christopher Edwards
che16@aber.ac.uk

Supervisor: Dr. Neil MacParthalain
ncm@aber.ac.uk

A requirements specification for a Computer Science Major Project

Department of Computer Science
Aberystwyth University
Wales

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed description for the ‘Visualising Ant Colony Optimisation’ application. This document will cover the interface interactions and methods as well as providing definitions to important terminology. The document is primarily intended to be used as a reference point for the initial stages of development.

1.2 Scope

The ‘Visualising Ant Colony Optimisation’ application is a desktop application which is designed to demonstrate the behaviour of an underlying ACO algorithm given various algorithm parameters. These parameters will be defined by the user and can be modified at their convenience. The algorithms behaviour given these parameters will be visible and the users can make a clear assessment about how each of the parameters impacts the algorithms performance.

During the background research into this subject area there does not seem to be too many applications which offer ACO visualisations and there are even less which provide a ‘friendly’ environment which is simple and intuitive to use regardless of the users background knowledge in regards to the subject area.

The software will be deployed in an educational environment, and aims to provide a means for teaching ACO to students as part of an Artificial Intelligence course or for independent use as a self-learning exercise. As a result the software must cater for the majority of user groups to maximise its effectiveness. This means the software must be accessible on all major platforms and perform equally well on said platforms.

1.3 Definitions

Term	Definition
ACO	Ant Colony Optimisation
user	Anybody who is using or wishes to use the software
user group	A collective group of users representing different user needs.
interface	A graphical user interface
standalone	Operates independently of other hardware or software
agent	The entities which will be traversing the graph

2 Overview

This section will give an overview of the proposed application. This section will also expand on the expected user groups and functionality required by said groups. The constraints

2.1 Product Descriptive

The software application will be standalone and does not need to communicate with another system or application, because of this there is no need for any form of network connection to be present in order to use the application to its maximum potential.

The application will communicate with the Operating System on the host machine in order to enable the save and load functionalities through simple file input and output. However the users access to the host machines file system will be restricted by the fact that the saving and loading will be restricted to the users home directory preventing the overwriting of important documents.

The application itself will not take up too many system resources even if a large problem is being handled. This allows the users on a system or network to run the application without it impacting the performance of other services. Given that the target audience is educational establishments this is especially important as many teaching fellows have multiple applications

running during a lesson or lecture and a negative impact on their system could reduce the amount of information taught during said session.

2.2 Product functionality

The users will be able to view a world which represents the Agents and the nodes in the graph otherwise known as the world. The state of this world will be directly related to the algorithm parameters specified by the users using the interface provided. There are several parameters which can be modified by the user, each of which will have a different impact of the state of the word and the algorithms behaviour.

The parameters which can be modified will be clearly labelled and will be obviously editable. As these parameters will be user defined there will be strict error checking measures in place to catch any illegal values before they can cause problems for the algorithm, and in addition each parameter will have a range of legal limits applied to them. This will prevent the users from entering values of the incorrect type (String when the systems needs a double) and will also prevent values from outside of the specified range being accepted. When a complication or error arises there will be a simple error message presented to the user informing them of both the error and why it occurred which should enable the user to resolve what they did incorrectly.

2.3 User Groups and Characteristics

There will be three main user groups associated with this application. Each of these user groups will interact with the application in a different manner but the main purpose and result will remain the same.

Teachers/Teaching fellows will use the application with the underlying knowledge already in hand. They will be mainly be using the applications to visually portray ideas and will have expectations in regards to what to expect for a solution and will have some idea how the parameters impact the final result.

Students will use the application with some background knowledge of the underlying concepts but will still use the application in an experimental manner and may have little expectations or understanding of how changing certain parameters impacts the final result.

People new to the subject area will use the application with potentially no idea about the underlying concepts. There will be measures in place to explain the underlying metrics and give an insight into what the application is actually doing. Given that they have less knowledge of this subject area that the other two user groups mentioned above, they will still be able to achieve the same results and levels of functionality. The application will cater for all users regardless of prior knowledge.

2.4 Constraints

As the application is standalone is reduces the amount of constraints which it becomes subject to. The main constraint which the application is associated with is the dimensions of the users display. The interface has to house a lot of elements in order to produce a simple and effective environment, thus it takes up quite a lot of screen real estate. However in modern times the amount of space required for the application to perform as expected is far from unreasonable and the application will be developed with this in mind.

The algorithms execution time directly proportional to the user defined parameters, more specifically the number of agents and the number of nodes in the graph. The more agents and nodes the larger the execution time and resource requirements will be. The application will be developed with this in mind as there will be a constraint on how much memory and system resources the application should use. There is no expectation on the user to have a superfast high end machine therefore the application will be designed to accommodate a standard machine for these modern times and correct limits will be placed on these user parameters.

The application does write the host systems file store so there must be adequate room to do so, however the files that will be written are simple text files which will not take up a lot of room on the host machines disk. Depending on the users machine this could still be a constraint. The responsibility and handling for this will belong to the users machine.

2.5 Assumptions

It will be assumed that the user will have the correct drivers installed and their machine will be able to handle the algorithms execution. The applications algorithms are not too resource intensive, therefore this is a reasonable assumption given the modern era and the advancement of computer technology.

It will be assumed that users will meet the minimum display requirements thus no dynamic resizing of the interface based on the users display dimensions will be performed. This significantly reduces complexity.

Another assumption is that every user will have some experience of using similar software and the interface will be familiar and therefore will be easy to use and navigate. The interface will use traditional methods such as simple buttons, text boxes and drop down menus to provide the user access to certain functionality.