Software Engineering Project (COS 301)

Software User Manual

Version 0.2, September 2016

Team:

Emilio Singh u14006512

Gerard van Wyk u14101263
Matheu Botha u14284104
Renton McIntyre u14312710

Overview

SwarmViz is a program that visualises how optimisation algorithms work.

Optimisation algorithms are a type of Artificial Intelligence that are used to automatically solve complex real-world problems in science, engineering, finance, and military research & development.

With SwarmViz anyone can gain an understanding of how these sophisticated problem-solving algorithms work by letting the user play around with trying different optimisation algorithms to solve different example problems.

Software Installation

There are currently 2 ways to install the program. Each will be detailed below.

Linux Installation
The base Linux installation has the following software requirements:

1) Qt
2) OpenGL
3) GLEW
4) SDL2
5) FreeType2
6) SDL2_image
7) SOIL

There is 1 downloadable zip file from our project site.

1) Src.zip is a file that will contain all of the source code necessary to compile the project. This will be used freshly compile the code.

Note that the code for the project can also be fetched from the Git repository via the link provided.

Once you have both of these, the installation process will then consist of navigating to the folder with containing the project makefile.
In a terminal opened in the folder, type the command “make run”, without the quotations. This will compile and prepare an executable for you.

Once this is done, the program will be compiled and executed and start up.

Software Usage

1. Run the SwarmViz executable, the following screen should appear:

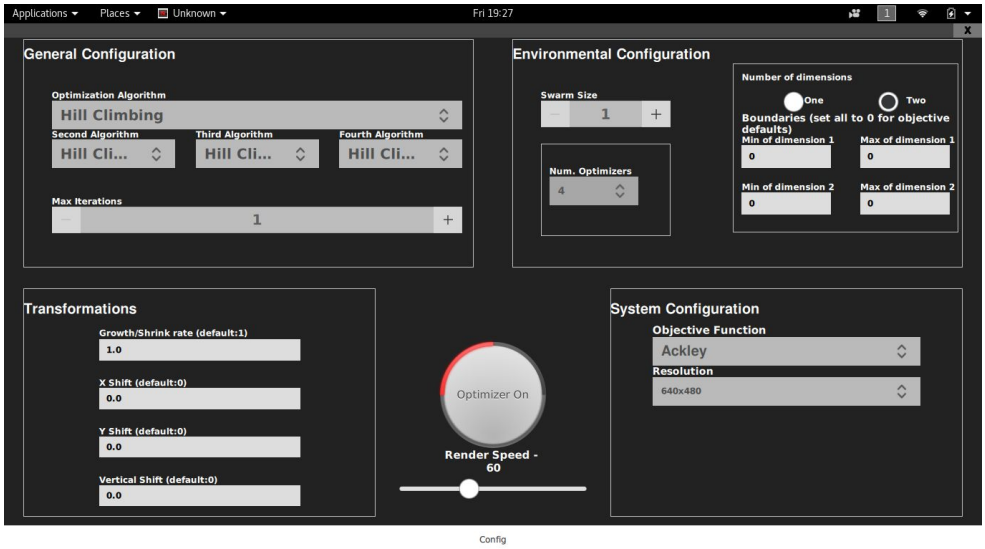
2. Use this page to edit your desired settings. These include:
   - **General Configuration**
     - Optimization Algorithm - Determines the optimizer that will attempt to solve the provided problem domain.
     - Max Iterations - The number of iterations of the optimization algorithm, before it stops and settles at a final location.
   - **Environmental Configuration**
     - Swarm Size - The number of particles (potential solutions to the problem) that attempt to solve. In algorithms such as basic Particle Swarm Optimization, where there is a “social” effect (particles positions’ affect each other), this can be a huge factor influencing the efficiency of the algorithm.
     - Number of Optimizers: Explained in next section.
     - Number of Dimensions - Determine whether optimizing on a 1D function mapped to a 2D landscape or a 2D function mapped to a 3D landscape (height is always fitness).
     - Boundaries - **[Optional]** If you would like to not use the default values for the landscape function, enter boundary values for

the dimension(s). If you would like to return to default values on further runs, set all values back to 0.

- **Transformations** - **[Minor]** These scale and shift values can alter the values upon which the algorithms function. Scale will rarely have a visible effect in the visualiser, due to the normalisation of graphical components. A shift will generally affect the bounds (and thus the landscape).
- **System configuration**
    - Objective choice - Determines the landscape that the optimization algorithm traverses. The landscape is created by mapping a mathematical function.
    - Resolution - Determines the size of the graphical window created. Choose a value that suits your display size and GPU capabilities.

3. *Choosing number of optimizers*

As mentioned before, the Environmental Configuration contains an option for the number of optimizers, allowing options 1, 2 or 4. When increased, the user is given the ability to select more optimization algorithms. The visualizer, when run, will display these optimizers on an identical landscape next to each other, thus allowing efficient comparison of the two. The screen will now look as follows:



4. When selecting Particle Swarm and its derivatives, further options appear under **System Configuration** (algorithm dependent). These include

- Social Coefficient
- Cognitive Coefficient
- Inertia Weight
- Success and Fail counts
- Maximum velocity
- Constriction Coefficient
- Neighbourhood Size

These values are what you should have fun fiddling with in order to make these optimizers perform differently. Enjoy!

5. You are now ready to click the “On” button and experience the magic!

6. After some problem-dependent delay, you should see a graphical window open and the animation process to begin. For example, you could see a screen like Figure A, slowly move to be like Figure B. Feel free to explore the environment in this time, observe behaviour and learn. :)

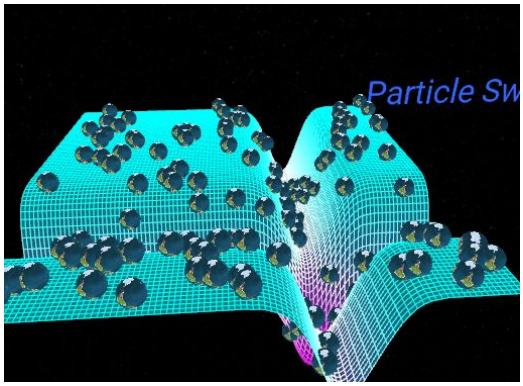


Figure A

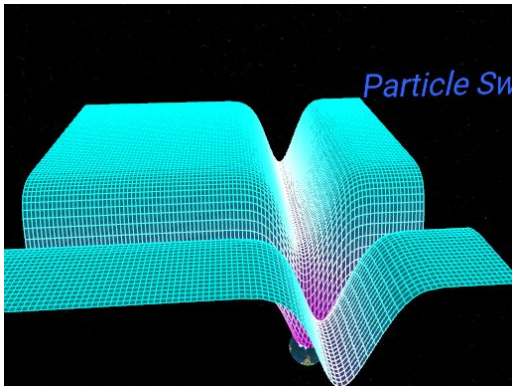


Figure B

7. When the vizualization is over, you can close that screen and continue to perform vizualizations from the User Interface.