A* Graph Optimization Using Genetic Algorithms

Stuart Bowman

May 2, 2016

1 Introduction

A* is known to be the undisputed champion of path finding for start to finish traversals of graphs and 2D planar environments. A* pathfinding does however require a set of waypoints to analyze viable paths and traverse terrain. While it isnt unreasonable to manually create a set of waypoints for most applications or use cases, it would be ideal to automate the process of generating and connecting the waypoints needed to run A* on a given environment. Genetic algorithms may provide a vehicle for creating, modifying and evolving data in order to provide the optimized graphs needed to run A*.

Genetic algorithms, as defined by John McCall, are a heuristic search and optimization technique inspired by natural evolution. [2] In their broadest sense, they represent a small branch of artificial intelligence that exists to construct solutions for problems that require optimizing many different attributes or parameters simultaneously. They are particularly useful under contexts where solutions can be optimized through repeated generational selection and breeding of entities containing desirable traits. [1]

Genetic algorithms provide an excellent platform for rapidly building, evolving and mutating data to follow a series of metrics through heuristical analysis. The full process can be broken down into several steps: [1]

- Initialization of the base population to start the first generation
- Evaluation of each data members fitness relative to the goal
- Selection of the most fit individuals in the current generation to be used to breed the next generation. This selection should usually strive to direct the population towards a generation of individuals with preferred traits.
- Crossover represents the actual breeding phase. This typically involves borrowing a mix of traits from each of the parents to create a new child entity or entities to be evaluated in the next generation.
- Mutations are used to introduce small random or nondeterministic changes in a childs traits that does not reflect the parents attributes. This can be used to prevent populations from becoming overly homogeneous.
- Repeat the process, starting with the evaluation of the most recent generation.

[1]

Additionally, genetic algorithms offer the flexibility and functionality of being able to find an optimal solution in a reasonable amount of time. This means that rather than waiting unreasonable amounts of time for an optimal solution, a genetic algorithm can be stopped at any iteration when the desired attributes have met or exceeded a given threshold or local optimum. [1]

With the base concept and definition of genetic algorithms in mind, the next step is to decide how genetic algorithms will play into the iteration and optimization of waypoint selection and generation. One of the primary concerns with waypoint generation for use in A* is to reduce the search space as much as possible without losing valuable information. Having a smaller set of valuable waypoints will allow A* to run a quickly and produce desirable paths.

2 A* Waypoint Generation

3 Implementation and Tuning

4 Results

5 Citations

1. Jacobson L. (2012, Feb 12). Creating a Genetic Algorithm for Beginners [Blog]. Available: http://www.theprojectspot.com/tutorial-post/creating-a-genetic-algorithm-for-beginners/3

2. McCall, J.Genetic Algorithms for Modelling and Optimization Journal of Comp. and Applied Math, vol. 184, no. 1, pp. 205-222, Dec 2005