A Comparison of Optimisation Methods in “Creature” Generation

Literature Review

Thomas Gardiner – S14112250

BSc (Hons) Computer Science

# Table of Contents

[Table of Contents i](#_Toc469652601)

[Introduction 1](#_Toc469652602)

[Optimisation Methods 1](#_Toc469652603)

[Genetic Algorithms 1](#_Toc469652604)

[Simulated Annealing 2](#_Toc469652605)

[Particle Swarm Optimisation 2](#_Toc469652606)

[Others 3](#_Toc469652607)

[Creature Generation 4](#_Toc469652608)

[Two-Dimensional 4](#_Toc469652609)

[Three-Dimensional 5](#_Toc469652610)

[Creature Evolution Evaluation Methods 5](#_Toc469652611)

[Conclusion 6](#_Toc469652612)

[References 8](#_Toc469652613)

[Appendix 11](#_Toc469652614)

# Introduction

In this section, existing literature will be reviewed to discover different optimisation methods (OM), comparing different aspects of each to determine the most suitable and practical OMs.

Literature defining a “creature”, and how one goes about generating them in both two and three dimensions (3D) will also be reviewed and discussed in order to find the final method to be used.

# Optimisation Methods

Three OMs will be compared for consideration of their use in the final product. Where possible, their effectiveness in comparison to other OMs will be noted.

## Genetic Algorithms

Commonly known as the most popular optimisation method, genetic algorithms are a heuristic search method that combine “genes” and “chromosomes” to find an optimal solution (Gonçalves and Resende, 2011).

In order to determine the source of a gas emission in a given area, a comparison was made between four different OMs and hybrids of each of them. Ma et al. (2013) found that the genetic algorithm method was one of the two highest scoring OMs, although however was the slowest to calculate evolutions.

However, while using both a non-dominated sorting genetic algorithm and particle swarm optimisation (PSO) to search for the optimal parameters in microwave components, Sedenka and Raida (2010) found that the genetic algorithm performed much worse than its rival. One benefit of the genetic algorithm was that it required less parameters to set.

This suggests that genetic algorithms may have potential to evolve creatures to their greatest potential using little configuration, however may take a long time to process the calculations, which goes against the aim of this project.

## Simulated Annealing

Simulated annealing (SA) is the simulation of the controlled cooling of glass and metals, and as the temperature slowly decreases to its minimum it is more restricted in exploring surrounding areas, eventually finding a global optimum.

Haridass et al. (2014) found that large improvements were made using SA over the human calculated equivalent. As much as a 34% decrease in delivery waiting times were made after using SA to calculate truck delivery routes.

Adzhar and Salleh (2014) also found success using SA in an attempt to reduce the number of layers required on a circuit board. No explicit comparison result was specified other than that it resulted in higher connections per layer and therefore fewer layers.

These papers promise positive results using SA, however with a slowly decreasing temperature results may take longer to compute than genetic algorithms, which again also goes against the aim of this project. Although due to the high performance of each resulting generation of creatures, SA is a likely candidate for comparison.

## Particle Swarm Optimisation

PSO is modelled after swarms of animals, such as flocks of birds, where each bird is a particle. Each particle has a direction, velocity, and fitness level, and the global best particle is stored in a vector to keep track of the improvements (Nedic et al., 2014; Sedenka and Raida, 2010).

In a comparison against Box’s Complex OM, Mendes et al. (2016) tested particle swarm optimisation to determine the best values possible for the geometry of a switched reluctance generator. They found that PSO often took around two times longer to calculate the optimal values, with an extreme case being 18 times slower. However, PSO performed many fewer iterations of optimisation in comparison to Box’s complex.

Zotes and Peñas (2012) used PSO to find an interplanetary route between different planets in our solar system using gravitational assistance. It was compared against the Hohmann transfer, a transfer used to get a spacecraft between two different planet’s orbits (Bacon, 2014). It was found that the total mission time would be 5,501 days using PSO, in comparison to 2,223 days from the Hohmann transfer. However, the total change in velocity was much less in comparison, with 14.08 km/s difference used by the Hohmann transfer, whereas PSO had a difference of 6.41 km/s. This is likely to use much less fuel, as a result of the main point of the paper, which could be regarded as a success.

In contrast to both SA and genetic algorithms, PSO appears to be generally inefficient and relatively complex to implement in comparison to the other OMs reviewed. This suggests that its inclusion into the prospective software is rather unlikely.

## Others

While the three previously mentioned optimisation methods are frequently used in academic papers, there are many others available that could potentially benefit future research.

For example, the Hooke and Jeeves method of optimisation involves both exploratory moves – searching each side around a base point – and then following patterned moves only if it finds a more successful fitness value after exploring (Nedic et al., 2014).

McCandless and Gregg (2011) set out to increase the speed of virtual machines and code interpreters by changing the order in which process operations are listed. Hill climbing was one OM used among others. Hill climbing searches the local space around it and only accepts better solutions, therefore it is greedy and will terminate at any optimal value, regardless of whether it is the global maximum. As a result, hill climbing did not benefit the research being made.

# Creature Generation

In order to demonstrate how OMs differ, a method of generating creatures must be chosen. These papers discuss how they generate creatures and move them about in a virtual space.

## Two-Dimensional

Sisnett (2015) created two dimensional creatures, each made up of multiple squares. Each square contains genetic information about the creature, in terms of how it moves and its appearance. The creatures may also contain actuators to help them move and sensors to detect other creatures and obstacles around them. These creatures were then tasked with moving as far as possible in a specified number of steps (or iterations), to visit as many different locations as possible, and the smallest number of steps to another side of the virtual space. The results of these tests determines the fitness value for each creature, which is then evolved within a genetic algorithm.

A system using spring-based mechanics was developed by Sanders et al. (2003). The creature is made up of springs that are affected by various inputs from a controller that keep track of its velocity and what parts of the creature are touching the ground. Each spring has its own time based delays so that they do not all move at the same time. These creatures are tasked with travelling the further in a given set time, which determines their fitness value, and so are also evolved in order to get the best creature. However, a reoccurring problem meant the creatures were changed to evolve based on a niche to ensure diversity.

## Three-Dimensional

Lehman and Stanley (2011) generated creatures using differently shaped cuboids. The creature starts with an origin cuboid where others are added to it using a node based system, so the creature ends up with a hierarchical structure and can therefore have joints and repetitions. Creatures generated using this system were tested novelty, fitness, global competition and local competition. Those tested and evolved for fitness and global competition resulted in a body made up of a single cuboid, with a strong correlation between height and mass, however those evolved for novelty and local competition had almost no correlation at all.

Similarly, Lessin and Risi (2015) procedurally generated creatures using segments and muscles within a node-based system. Each creature has a brain using a similar node system, where each node has an input value and an output value between 0 and 1, which can be sent to other nodes in the brain. The muscles act as springs between a randomly chosen point on two different segments and is also given two nodes in the creature’s brain to co-operate with other muscles. The creatures were not only evolved to perform their best, but were also tested against human controllers, using the keyboard to control different muscles in each creature. It was found that the evolved creature was more successful than the human after every single test.

# Creature Evolution Evaluation Methods

In order to best represent the results collected, different evaluation methods must be considered for use in this project. This is most likely to include quantitative sets of data in the form of graphs or tables considering the nature of the topic, however the following papers regarding the evolution of creatures have been discussed in order to form the final decision.

Pilat and Jacob (2010) created 3D creatures based on a node and neuron system that were able to follow a light source, and were evolved using genetic algorithms. To display the performance at the end of each generation, they used a multi-layered line graph to show every creature’s best and worst performance, with a thicker line to show the average of each, suggesting the use of a quantitative evaluation method. However, the performance of each creature could also be categorised depending on which neuron it used to detect the angle and distance (and combination of both) of the light.

­­In comparison, Lassabe et al. (2006) developed a system to generate not only 3D creatures, but also insects and plants, which are evolved through an evolutionary algorithm. However, when determining the success or failure of their evolution, it was only stated that the experiment was successful after 3 hours. The lack of graphs or tables present to validate their results suggests the results are qualitative.

Similarly to the majority of other 3D creature generation papers discussed in this review, Miconi and Channon (2006) generated creatures using cuboids attached to one another. They were split into two populations and tasked with fighting against each other in “tournaments”. The 50% of the generated population that succeeded against the other half the most amount of times during a single generation claimed victory and was evolved and competed in a tournament once more. This was repeated for 50 generations. The results of each tournament were plotted on a graph showing which population succeeded, however the results were lengthy and were described in detail about what took place, so not only was it quantitative, but also qualitative.

# Conclusion

As a comparison of OMs, at least two different methods should be chosen. While genetic algorithms are a widely known choice that has supporting data to prove its efficiency and performance, it is not often compared directly against other OMs. Despite hill climbing being an unlikely candidate against more powerful OMs, it will also be measured to see how it compares in contrast to them. Finally, simulated annealing will also be tested, as a supposed reliable alternative to both genetic algorithms and hill climbing.

Generating and rendering two dimensional creatures could be considerably faster than three dimensional shapes, as well as calculating friction, collisions, and movement vectors between shapes. In addition to the many calculations required by the OMs, this may be a good choice to ensure there is not excessive waiting between generations. Also, two dimensional creatures are restricted to one axis of horizontal movement, so a better representation of the travelled distance can be displayed to the screen. However, some aspects of the three dimensional creature controllers could be implemented, such as the node-based brain system used by Lessin and Risi (2015).

It was found that quite often the results of a creature’s evolution can be plotted to a graph with relative ease, as each creature should return a fitness value depending on how it performs on the task it is given. This suggests that the results of this project should be largely presented as quantitative. However, it may be very useful to categorise specific creatures, or a “species” of creature to describe the process they go through, or if one type is especially interesting.

# References

Adzhar, N. and Salleh, S. (2014) Simulated Annealing Technique for Routing in a Rectangular Mesh Network, *Modelling and Simulation in Engineering*, 2014, p. 46.

Bacon, W. (2014) *Hohmann Transfers*, [Online],  
Available at: <http://jwilson.coe.uga.edu/EMAT6680Fa05/Bacon/hohmanntransfers.html>   
[Accessed 14 December 2016].

Gonçalves, J.F. and Resende, M.G.C. (2011) Biased random-key genetic algorithms for combinatorial optimization, *Journal of Heuristics*, 17(5), pp. 487-525.

Haridass, K., Valenzuela, J., Yucekaya, A.D. and McDonald, T. (2014) Scheduling a log transport system using simulated annealing, *Information Sciences*, 264, pp. 302 - 316.

Lassabe, N., Luga, H. and Duthen, Y. (2006) Evolving Creatures in Virtual Ecosystems, International Conference on Artificial Reality and Telexistence, Hangzhou, China, pp. 11-20.

Lehman, J. and Stanley, K.O. (2011) Evolving a Diversity of Virtual Creatures Through Novelty Search and Local Competition, GECCO '11, Dublin, pp. 211-218.

Lessin, D. and Risi, S. (2015) Darwin's Avatars: A Novel Combination of Gameplay and Procedural Content Generation, GECCO '15, Madrid, pp. 329-336.

Ma, D., Deng, J. and Zhang, Z. (2013) Comparison and improvements of optimization methods for gas emission source identification, *Atmospheric Environment*, 81, pp. 188-198.

McCandless, J. and Gregg, D. (2011) Optimizing Interpreters by Tuning Opcode Orderings on Virtual Machines for Modern Architectures: Or: How I Learned to Stop Worrying and Love Hill Climbing, Proceedings of the 9th International Conference on Principles and Practice of Programming in Java, Kongens Lyngby, Denmark, pp. 161-170.

Mendes, R.P.G., Calado, M.R.A. and Mariano, S.J.P.S. (2016) Particle swarm and Box׳s complex optimization methods to design linear tubular switched reluctance generators for wave energy conversion, *Swarm and Evolutionary Computation* , 28, pp. 29 - 41.

Miconi, T. and Channon, A. (2006) Analysing Co-evolution Among Artificial 3D Creatures, Artificial Evolution: 7th International Conference, Evolution Artificielle, Lille, France, pp. 167-178.

Nedic, V., Cvetanovic, S., Despotovic, D., Despotovic, M. and Babic, S. (2014) Data mining with various optimization methods, *Expert Systems with Applications*, 41(8), pp. 3993-3999.

Pilat, M.L. and Jacob, C. (2010) Evolution of Vision Capabilities in Embodied Virtual Creatures, Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, Portland, Oregon, USA, pp. 95-102.

Sanders, M., Lobb, R. and Riddle, P. (2003) Evolving Controllers for Virtual Creature Locomotion, Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, Melbourne, Australia, pp. 255-256.

Sedenka, V. and Raida, Z. (2010) Critical Comparison of Multi-objective Optimization Methods: Genetic Algorithms versus Swarm Intelligence, *Radioengineering*, 19(3), pp. 369-377.

Sisnett, R. (2015) A flexible and simplified 2D environment for evolving autonomous virtual creatures, 2015 7th International Joint Conference on Computational Intelligence (IJCCI), Portugal, pp. 306-312.

Tan, J., Gu, Y., Turk, G. and Liu, C.K. (2011) Articulated Swimming Creatures, *ACM Transactions on Graphics*, 30(4), pp. 58:1-58:12.

Zotes, F.A. and Peñas, M.S. (2012) Particle swarm optimisation of interplanetary trajectories from Earth to Jupiter and Saturn, *Engineering Applications of Artificial Intelligence*, 25(1), pp. 189-199.

# Appendix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Title** | **Topics** | **Keywords** | **Year** | **Authors** | **Summary** |
| A Flexible and Simplified 2D Environment for Evolving Autonomous  Virtual Creatures | Creature Generation (2D) | Artificial Life, Genetic Algorithms, Neural Networks, Evolutionary Computing. | 2015 | Ricardo Sisnett | Two dimensional creatures are generated using squares that hold sensors and actuators, information about its surroundings and movement. |
| Evolving Controllers for Virtual Creature Locomotion | Creature Generation (2D) | Evolutionary algorithms,  virtual creatures,  mass-spring  systems,  locomotion controllers. | 2003 | Sanders et al. | Two dimensional creatures are generated using a spring-based system, each with an internal clock and a main “brain” controlling each one. |
| Evolving a Diversity of Creatures through Novelty Search and Local Competition | Creature Generation (3D) | Virtual creatures, Natural evolution, Artificial life, Novelty  Search. | 2011 | Lehman and Stanley | Three dimensional creatures generated using cuboids and a node-based system for both competition and novelty. |
| Darwin’s Avatars: A Novel Combination of Gameplay and Procedural Content Generation | Creature Generation (3D) | Evolved virtual creatures, artificial life, muscle drives, physics-based character animation, procedural content generation. | 2015 | Lessin and Risi | Three dimensional creatures generated with segments and muscles, evolved to travel a distance as fast as possible, and then also tested against human controllers. |
| Scheduling a log transport system using simulated annealing | Optimisation Methods | Log transport, Simulated annealing, Deterministic simulation, Vehicle routing. | 2014 | Haridass et al. | Simulated annealing was used to help improve a large scale system of picking up and delivering trucks of logs. |
| Simulated Annealing Technique for Routing in a Rectangular  Mesh Network | Optimisation Methods | Simulated annealing,  Network routing,  Travelling Salesman. | 2014 | Adzhar and Salleh | Simulated annealing was used in an attempt to reduce the number of layers required for circuit boards than the human calculated attempt. |
| Comparison and improvements of optimization methods for gas emission source identification | Optimisation Methods | Gas leakage, Source identification, Optimization methods, Neural network, Atmosphere dispersion | 2013 | Ma et al. | The source of gas leaks are found by using genetic algorithms, simulated annealing, pattern searching, and other optimisation methods. The results are then compared. |
| Particle swarm optimisation of Interplanetary trajectories from Earth to Jupiter and Saturn | Optimisation Methods | Swarm algorithms,  PSO,  Interplanetary trajectory,  Optimisation  Space mission. | 2012 | Zotes and Peñas | The most fuel efficient route is found by using particle swarm optimisation and compared against the human calculated equivalent. |
| Particle swarm and Box's complex optimization methods to design linear tubular switched reluctance generators for wave  energy conversion | Optimisation Methods | Particle swarm optimization,  Box's Complex Method,  Multi-objective design,  Wave energy conversion,  Linear switched reluctance generators | 2016 | Mendes et al. | The most efficient solution of a generator’s layout is found using particle swarm optimisation. |
| Critical Comparison of Multi-objective Optimization Methods: Genetic Algorithms versus Swarm Intelligence | Optimisation methods | Multi-objective optimization,  Binary genetic algorithm,  Particle swarm optimization,  Pareto front, Finite  element method. | 2010 | Sedenka and Raida | Genetic algorithms are tested against particle swarm optimisation on microwaves. |
| Optimizing Interpreters by Tuning Opcode Orderings on Virtual Machines for Modern Architectures | Optimisation Methods | Virtual machines,  Lua,  Python,  Locality,  Branch prediction | 2011 | McCandless and Gregg | The speed of virtual machines are increased by using the hill climbing algorithm. |
| Data mining with various optimization methods | Optimisation Methods | Traffic noise,  Artificial intelligence,  Genetic algorithm,  Hooke and Jeeves,  Simulated annealing,  Particle swarm optimization. | 2014 | Nedic et al. | Four different optimisation methods are compared against each other and the results are compared. |
| Evolution of Vision Capabilities in Embodied Virtual Creatures | Creature Evolution Evaluation Methods | Evolution,  Artificial Life,  Virtual Creatures,  Genetic Algorithms,  Vision. | 2010 | Pilat and Jacob | 3D creatures are generated and evolved to follow a light source. |
| Evolving Creatures in Virtual Ecosystems | Creature Evolution Evaluation Methods |  | 2006 | Lassabe et al. | Creatures, insects and plants are generated and evolved to find the most successful creature. |
| Analysing Co-evolution Among Artificial 3D Creatures | Creature Evolution Evaluation Methods |  | 2006 | Miconi and Channon | 3D creatures are generated and split into two populations, then fought against each other in tournaments. The results are then discussed. |
|  |  |  |  |  |  |