Tuesday, 12 July 2022

Graded Assignment 1 - Genetic Algorithms

Solving the Eight Queens Problem

Introduction

This report provides a discussion of the method used and results of applying a genetic algorithm to solve the Eight Queens problem.

Method

The genetic algorithm I have written follows a similar basic structure to the one detailed by Farhan, Tareq and Awad (2015) in their study, and is as follows:

Population generation - Firstly a random population is generated, made up of individual ‘states’. These states are representations of each Queen’s position on a given board.

Fitness - Each state is denoted a fitness level which is representative of how close the state is to a being goal state.

Parent selection - From the population certain states will be selected to become parents, with a bias towards those states with a higher fitness level. I decided to use a stochastic universal sampling method for parent selection, which is where we calculate the probabilities of each individual (individual fitness / population fitness), take a number of equally spaced ‘pointers’ equating to the number of parents we want to sample, across this range randomly shuffle the population, then begin to step through the population adding up the probabilities. If a pointer lies in range of a probability then this individual is taken as a parent (Pencheva, Atanassov and Shannon, 2009).

Crossover and mutation - From these parents states we can apply a crossover operator to generate new child states. I decided to implement a single point crossover, which involves generating a random index to be the cross over point and splitting up the individual state at this point, and then combining these sub-states with each other to create new combinations of children (Kora and Yadlapalli, 2017). We also apply a mutation operator to randomly change a few individual states, which involves swapping out a number in the state for a randomly generated one.

New population selection - The top states from the previous population as well as the new children are then selected to form the new population. The process then continues until the goal state is found.

Results

Following the method detailed above I was able to produce an algorithm that successfully solves the Eight Queens problem, within the noted time limit of 30s.

The table below shows the min, max and average time it took the algorithm to produce a goal state in a sample of 10 and 100 iterations.

Number of iterations	Min	Max	Avg	Med
10	2.7	16.0	11.5	12.4
100	2.3	79.3	13.8	12.0

Algorithm running time (s)

Optimisation

In order to optimise my algorithm, I decided to investigate alternative methods of parent selection. The results of a study conducted by Oberoi and Rylander (2004), find that using a Tournament selection style method performs the best overall, which is further supported by a similar study conducted by Shukla, Pandey and Mehrotra (2015). I decided to add an additional function to my algorithm that selects parents using this alternative method, where a random small sample is taken of the population to be entered into the ‘tournament’, and the individual with the highest fitness from the sample is selected to be a parent. This process is repeated for the number of parents you would want to extract. Results using this method have been included in the table below.

Number of iterations	Min	Max	Avg	Med
10	1.5	26.8	6.7	3.7
100	0.8	216.3	9.7	4.7

Algorithm running time (s) post optimisation

Looking at the values from the both tables we can clearly see that the average time it takes to produce a solution has reduced following optimisation, and the median value also reflects that conclusion as well, with the original algorithm finding a solution in a median time of 12s and the optimised in 4.7s. The optimised algorithm has a higher max value though, which is something that could be investigated as part of future work, as it suggests the algorithm sometimes struggles to converge, it could be interesting to look at a hybrid model of these methods instead.

References

Farhan, A. S., Tareq, W. Z. and Awad, F. H. (2015). Solving N Queen Problem using Genetic Algorithm. International Journal of Computer Applications, 122(12), pp.11–14. doi:10.5120/21750-5005.

Kora, P. and Yadlapalli, P. (2017). Crossover Operators in Genetic Algorithms: A Review. International Journal of Computer Applications, 162(10), pp.34–36. doi:10.5120/ijca2017913370.

Oberoi, D. and Rylander, B. (2004). Determining the Best Parent Selection Method for a Genetic Algorithm through Varying Problem Sizes and Complexities. [online] Available at: http://gpbib.cs.ucl.ac.uk/gecco2004/LBP039.pdf [Accessed 12 Jul. 2022].

Pencheva, T., Atanassov, K. and Shannon, A. (2009). Modelling of a Stochastic Universal Sampling Selection Operator in Genetic Algorithms Using Generalized Nets. [online] Tenth Int. Workshop on Generalized Nets Sofia, pp.1–7. Available at: https://ifigenia.org/images/2/2c/IWGN-2009-01-07.pdf [Accessed 13 Jul. 2022].

Shukla, A., Pandey, H.M. and Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm. 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE). doi:10.1109/ablaze.2015.7154916.