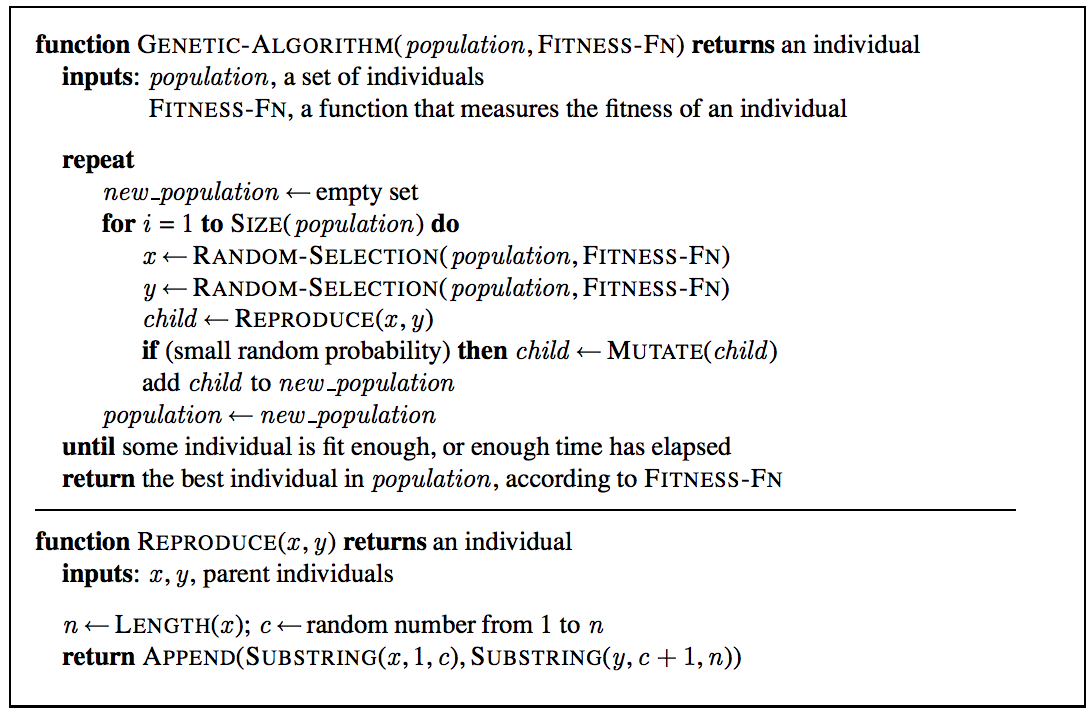
CS 420 Project Report

Approach:

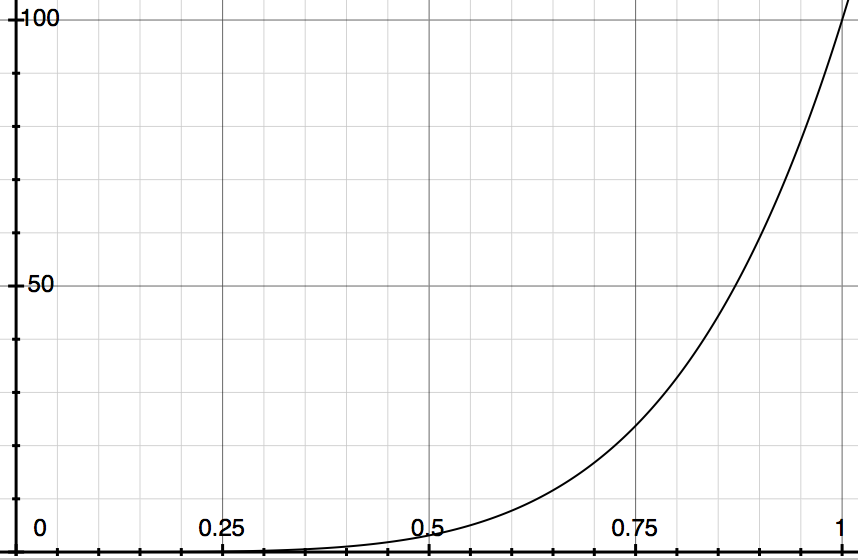
My approach for the random restart hill climbing method was to first create a randomly initialized array where the indices represented the columns and the values represented the rows of where a queen was located on an nxn chess board and calculate the number of non attacking pairs of queens for it. Then I would then calculate the number of non attacking pairs of queens on the board if one queen was moved and stored that value in a 2D array at the potential movement. The 2D array represented the children of the parent node where a -1 value represented the current configuration and a positive value represented the children and from it choose the best value and use that configuration as our next node to expand. If a local max was reached then a random restart would be conducted and repeat the algorithm again.

My approach for the Genetic algorithm was simple I used the pseudo code from the book except I sorted my population by number of non attacking pairs of queens where higher values are placed in front and lower ones in the back.



In my random select method I chose randomly from the population using the formula index = populationSize\*x^(selectionfactor), where x is a random value between 0 and 1 not including 1 and selection factor determines the probability of a good parent being chosen.

Below is a graph of 100\*x^(5)



For the graph above it shows that the likely hood of an index of 25 and below is roughly 75% of the time and 25% for any index higher. The formula ensure that the probability of the best configurations being chosen is high. Then if a solution of no attacking queens was found it would end otherwise on the final generation the best configuration in the population will be out putted.

Analysis:

After running each algorithm 1000 times it seems that on average for 21 queens problem Genetic algorithm out performs Random restart hill climbing in both time and accuracy. Genetic algorithm found a perfect solution 100% of the time where as Random restart hill climbing was around 40%. But if given more opportunities to restart Random restart hill climbing will converge towards 100% accuracy.

Sample Output For Random Restart Hill Climbing And Genetic Algorithm For 1 nQueen Problems:

Vikas Jetalpuria\_Vikas\_420p2 $ java N\_Queen\_Driver

N: 21

Random Restart Hill Climb 1 iteration

Initial Configuration:

[3, 2, 8, 17, 9, 10, 13, 15, 17, 20, 16, 3, 14, 12, 16, 5, 12, 13, 4, 15, 2]

Number of Non Attacking Pairs of Queens: 189

Number of Restarts: 25

Final Configuration:

[7, 1, 18, 17, 2, 9, 18, 12, 0, 11, 11, 2, 11, 10, 1, 8, 10, 14, 6, 18, 7]

Number of Non Attacking Pairs of Queens: 190

Genetic Algorithm for 1 iteration

initial Population Size: 100

Chance of Mutation: 0.05

Max number of Generations allowed: 10000

Final Configuration:

[10, 13, 17, 9, 7, 1, 19, 6, 15, 0, 18, 20, 5, 11, 16, 12, 2, 4, 8, 3, 14]

Number of Non Attacking Pairs of Queens: 210

Number of Generations Produced Before Finding Solution: 2078

Number of Childern Mutated: 436231

Vikas Jetalpuria\_Vikas\_420p2 $ java N\_Queen\_Driver

N: 21

Random Restart Hill Climb 1 iteration

Initial Configuration:

[20, 12, 18, 4, 19, 3, 0, 8, 15, 20, 3, 12, 12, 4, 9, 15, 3, 11, 18, 14, 12]

Number of Non Attacking Pairs of Queens: 180

Number of Restarts: 25

Final Configuration:

[4, 15, 10, 3, 0, 14, 4, 10, 9, 15, 2, 6, 1, 9, 2, 18, 3, 20, 13, 16, 3]

Number of Non Attacking Pairs of Queens: 190

Genetic Algorithm for 1 iteration

initial Population Size: 100

Chance of Mutation: 0.05

Max number of Generations allowed: 10000

Final Configuration:

[14, 2, 10, 19, 17, 12, 3, 6, 0, 7, 13, 4, 16, 18, 20, 15, 11, 8, 1, 5, 9]

Number of Non Attacking Pairs of Queens: 210

Number of Generations Produced Before Finding Solution: 107

Number of Childern Mutated: 22691

Vikas Jetalpuria\_Vikas\_420p2 $ java N\_Queen\_Driver

N: 21

Random Restart Hill Climb 1 iteration

Initial Configuration:

[19, 3, 4, 7, 19, 2, 19, 6, 3, 6, 15, 10, 14, 4, 2, 12, 4, 8, 7, 16, 19]

Number of Non Attacking Pairs of Queens: 177

Number of Restarts: 6

Final Configuration:

[5, 11, 16, 1, 4, 8, 14, 3, 20, 2, 17, 15, 9, 12, 0, 7, 18, 6, 19, 13, 10]

Number of Non Attacking Pairs of Queens: 210

Genetic Algorithm for 1 iteration

initial Population Size: 100

Chance of Mutation: 0.05

Max number of Generations allowed: 10000

Final Configuration:

[19, 6, 8, 1, 5, 16, 9, 17, 15, 3, 18, 7, 4, 13, 11, 14, 20, 0, 2, 12, 10]

Number of Non Attacking Pairs of Queens: 210

Number of Generations Produced Before Finding Solution: 524

Number of Childern Mutated: 110006

Sample Output For Random Restart Hill Climbing And Genetic Algorithm For 1000 nQueen Problems:

Vikas Jetalpuria\_Vikas\_420p2 $ java N\_Queen\_Driver

N: 21

Random Restart Hill Climb 1000 iterations

Total Solved: 404

Total Times Ran: 1000

Percent that were solved under 25 Restarts: 40.400000000000006

Average Time Taken (nanoSeconds): 3.7195381648E7

Genetic Algorithm for 1000 iterations

Total Solved: 1000

Total Times Ran: 1000

Percent that were solved: 100.0

Average Time Taken (nanoSeconds): 3.21475053556E8