CS7641-- Randomized Optimization

# Introduction

The purpose of this project is to explore randomized optimization search. This assignment has 3 parts. The first part is about what are these 4 kinds of randomized optimization algorithms; in the second, I will apply these 4 algorithm into a neural network model which was created in assignment 1 using movies review dataset; the last part is discuss the advantages of different optimization algorithms used on different problem.

# Optimization Algorithms

Randomized Optimization algorithms are useful for find global maximum or minimum with continuous and discrete variables. This assignment is mainly about 4 Randomized Optimization algorithms which are Randomized Hill Climbing (RHC), Simulated Annealing (SA), Genetic Algorithm (GA) and MIMIC. All optimization problems result in this assignment were generated as part of the java code that was implemented using ABAGAIL.

## Randomized Hill Climbing

## Randomized Hill Climbing (RHC) search will start at a random state and move to the next state which has higher fitness, it stops moving when reaching the local or global optimum, and restart at random point, the purpose of random restarting is to leave current local optimum. However, randomized optimization algorithm is still weak at moving out of local optimum due to the uncertainty of restarting points.

## Simulated Annealing

Simulated Annealing (SA) models the physical process of heating a material and then slowly lowering the temperature to decrease defects, thus minimizing the system energy.

At each iteration of the simulated annealing algorithm, a new point is randomly generated. It will go to next point with a function of temperature and fitness. Comparing with Randomized Optimization, this algorithm can accept new points which has lower fitness function with lower probability and this avoids being trapped in local minimum. The movement will be more active with high temperature, with temperature gradually decreasing, it will search to converge to a minimum

## Genetic Algorithm

Genetic Algorithm (GA) is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. The genetic algorithm repeatedly modifies a population of individual solutions. At each step, the genetic algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution.

## MIMIC

Above 3 algorithms are all stating at a random point and move to the ending point depending more their metrics, there is no structure for searching the optimum; also, in most cases, they have an unclear probability distribution. To overcome these issue, MIMIC was created for directly model the probability distribution and successfully define the estimate of that distribution, this will effect the convey structure which is the structure about search of spaces.

# Neural Network with Optimization Algorithms

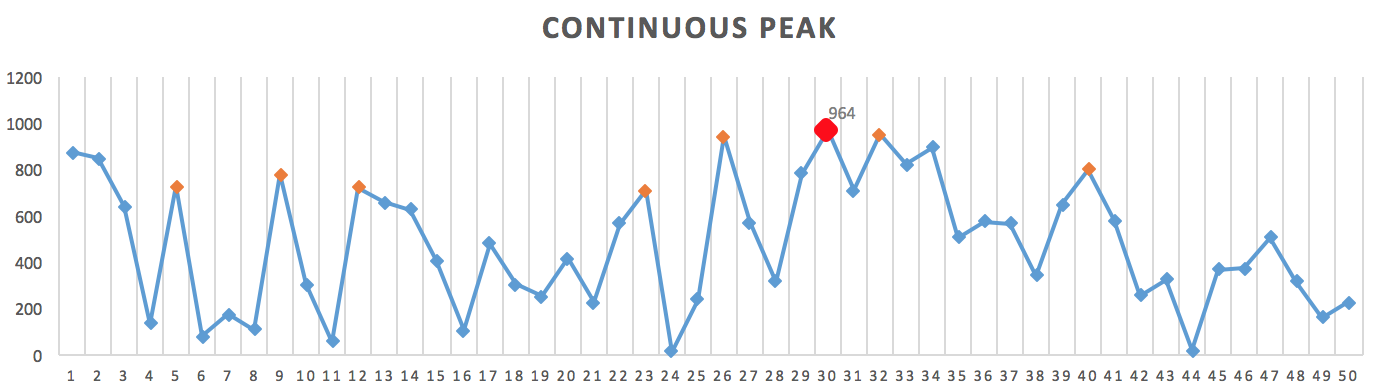
# Optimization Problems

I found below three optimization problems, which are Continues Peak, Travel Salesman Problem and Flipflop to demonstrate different optimizations functions. Below analysis will include problem introduction, Randomized Optimization method selection based on fitness and computational time.

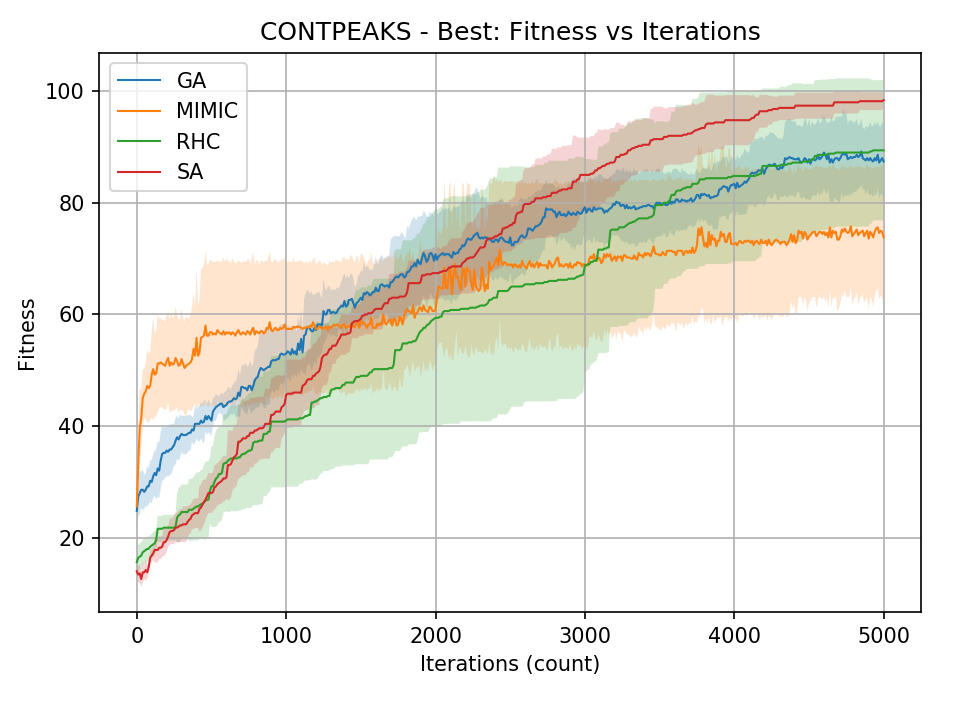
## Continuous Peaks Problem

### 3.1.1 Overview

The continuous peaks problem is to figure out the highest peak (global maxima, highlighted in red) versus the subsidiary peaks (local maxima, highlighted in orange). Below is an example for continuous peaks problem.

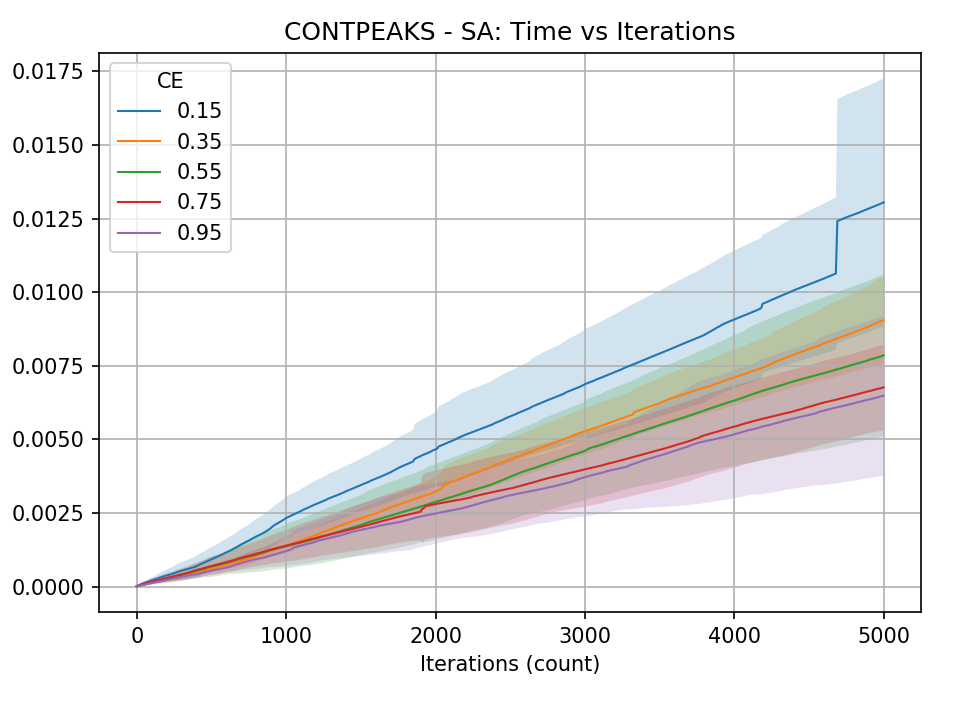
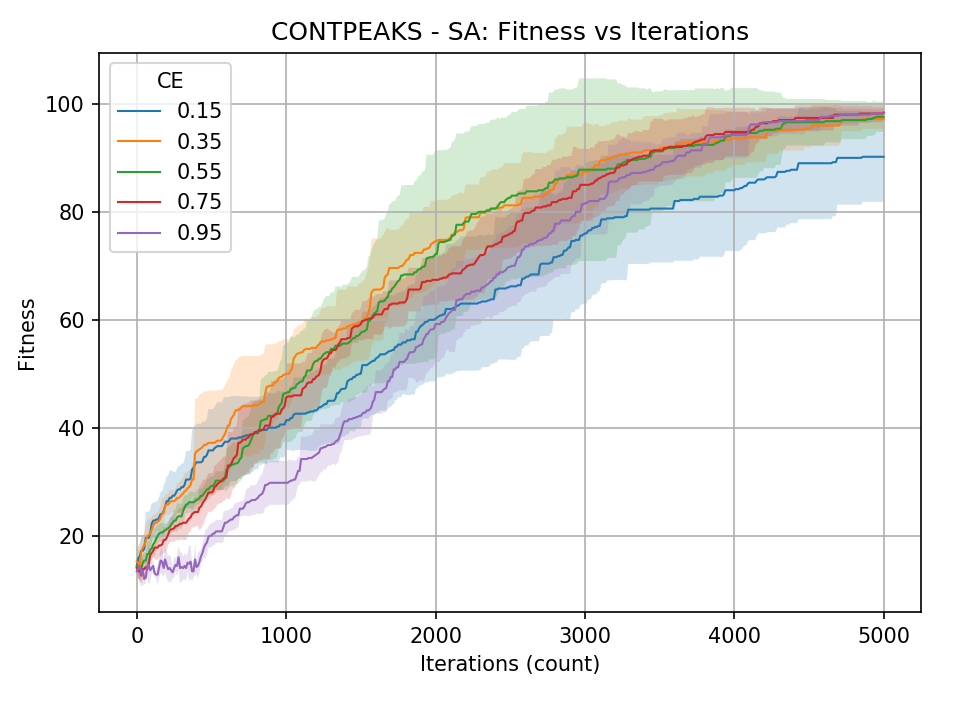


### 3.1.2 Optimization methods comparison

From the left line chart, we can see the begin of iterations (no. of iterations < 1000) shows that, MIMIC is a quick leaner which can obtain higher fitness in shorter time. While with the number of iteration increasing, MIMIC weakens. With 5000 iterations, the performance of SA is significantly better than other methods, RHC and GA are almost the same. This continuous peaks problem has showed strengths of SA; it’s performance is robustly increasing with decreasing margin.

Let’s dive deep into SA. The first chart below shows fitness versus no. of iteration using different cooling exponent parameters, all lines are converged at the end except for cooling exponent equals to 0.15. the slope of low cooling exponent is lower than low cooling exponent one, which show with higher cooling exponent, SA can obtain the optimal point with a higher pace, and it totally lost within extremely low no. of iterations (no. of iterations < 500).

In the perspective view of time, we can see from second chart that, the one with lowest cool exponent took the longest time (blue line), while the one with highest cool exponent took the shortest time (purple line), rest of them are close to each other.



As a conclusion, SA with 0.95 as cooling exponent is the best configuration, in other words, continuous peak problem has highlighted the advantage of Simulated Annealing.

## Flip-Flop

