MA Draft

Ema Skottova

June 24, 2020

Contents

1 Introduction **2**

2 Complete Problem Statement **3**

3 Theoretical Background **5**
3.1 Simulated Annealing 5
3.2 Genetic Algorithms 5

4 Methods **6**
4.1 General Solution 6
4.1.1 Parameters 6
4.2 Evaluation 7

5 Process **8**
5.1 Input Generator 8
5.2 Input/Output 8
5.3 General Solution V1 8
5.4 Simulated Annealing 8
5.5 General Solution V2 8
5.6 Genetic Algorithm 9
5.7 Improvements General Solution 9

6 Results **10**

Chapter 1

Introduction

In the first round of the Swiss Olympiad in Informatics there is always a creativity task, which does not have an optimal solution, but there are many approaches to reach an acceptable solution. In the fall of 2019, the task was to optimize a train network. During the first round I only submitted a very simple solution, which worked for a special case of small networks. The goal of this paper is to compare the ability of two optimization methods to solve this task. The chosen methods are simulated annealing and a genetic algorithm, both of which will be explained in the theory chapter.

Chapter 2

Complete Problem Statement

As already stated in the introduction, the goal of this paper is to compare two optimization methods. In order to do this, one first needs a problem to optimize. For this paper, the creativity task SOIway of the first round of the Swiss Olympiad in Informatics 2019/2020 was chosen. The task is as follows:

The program is given constants and a list of events, then it has to print actions.

Constants:

- minimal amount of time to change a line
- maximal number of passengers allowed on a train
- maximal number of passengers allowed at a station

Events (with their time of appearance):

- Appearance of a station (coordinates of the station and the type of the station)
- Appearance of a passenger (starting station and type of end station)
- Additional train
- Additional train line

Actions:

- Setting/Changing a line (stations that the line will cover)
- Adding a train to a line (line and the station where it starts)
- Boarding a passenger

- Unboarding a passenger

The program has failed if it produces invalid output. If the program does not fail, the process ends if (a) All passengers have arrived at their final destination or (b) There are more passengers than allowed at a station

Chapter 3

Theoretical Background

3.1 Simulated Annealing

The general idea is that at the beginning, the program will take larger steps in testing sets of parameter values and with time its step size decreases, and it only looks at options closer to options that were already good.

3.2 Genetic Algorithms

A genetic algorithm starts with a few 'individuals' (sets of parameter values) and the individuals multiply with 'mutations' (changes in parameter values) and after a few mutations only the best few survive. This process continues until a sufficiently good answer is found.

Chapter 4

Methods

A general solution, which depends on a set of parameters, is written first. Then, the two optimization methods will be used to find the best set of parameters.

4.1 General Solution

The general solution goes through each event and action and evaluates whether it should make an action. An action will be taken if the sum of the parameters which influence it is above 100.

4.1.1 Parameters

Picking up passenger

- Distance (time, stops, line changes in current train)
- Number of passengers on the train
- Number of passengers at the station
- Network capacity
- These values for the following train(s)/passenger(s)/station(s)

Letting passenger leave

- Correct station
- Number of passengers at the station
- Number of passengers on the train
- Distance (time, stops, line changes) of current line
- Distance (time, stops, line changes) of connecting lines

- Capacity of current line
- Capacity of neighboring lines
- Values for other trains/stations/passengers

Changing line for train

- Passengers per train on each line

Changing train line (time)

- Balance in current network
- Important trains on current line

New train line (route)

- Frequency of visits
- Number of passengers starting

4.2 Evaluation

The two methods will be compared the following way:

Method 1		Method 2		~~Result~~
Reason for end	Steps	Reason for end	Steps	
(b)	x	(b)	$< x$	Method 1 wins
(b)	x	(b)	x	Draw
(b)	x	(a)	y	Method 2 wins
(a)	x	(a)	$> x$	Method 1 wins
(a)	x	(a)	x	Draw

Chapter 5

Process

5.1 Input Generator

At first a program which generates random test data was written. This program's parameters were set to match a test data sample from the original competition.

5.2 Input/Output

Next, the first part of the main program was created. This part is responsible for reading the input and printing the output.

5.3 General Solution V1

After this, the general program which calculates a solution will be written. This version will only consider very simple parameters, which can be calculated easily. The goal of this version is to have a running program so that tests can be done on the program. In particular, this should make it possible to test first versions of the two methods.

5.4 Simulated Annealing

This algorithm will be implemented and tested using the first version of the general solution.

5.5 General Solution V2

After first tests with simulated annealing, improvements will be added to the general solution.

5.6 Genetic Algorithm

This algorithm will be implemented and tested. First comparisons between the algorithms can be made.

5.7 Improvements General Solution

In subsequent versions, missing parameters will be added and other necessary improvements will be made.

Chapter 6

Results