CONCORDIA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

COMP 6651/3: Algorithm Design Techniques

Fall 2019

Project report due:
October 4, 2019 before noon
(electronic submission only)

Term Project

WARNINGS

- Part I of the project is individual
- All data and parameters have to be read from an (or several) input file(s), so that it is easy to change them. No hard coding is allowed for any of the input parameters. Sample files can be found on Moodle: please use the same format.
- Notations should be kept similar to the ones used in the description of the project.
- Submission should be made with **single .zip file**
- Each project submission must be submitted with the signed originality form
- Possible choices for the programming language: Java, C++ or Python
- Programs need to be well written and structured, for instance, no program made of a single function. You need to write programs with good programming practice rules.

1 Background of the project

Bike sharing systems offer a mobility service in which public bicycles are available for shared use. These bicycles are located at stations that are displayed across an urban area. The users of the system can take a bicycle from a station, use it for a journey, leave it into a station (not necessarily the one of departure), and then pay according to the time of usage.

These systems are an important instrument used by public administrations to obtain a more sustainable mobility, decrease traffic and pollution caused by car transportation, and solve the so-called last mile problem related to proximity travels.

Stations are made of different slots, each of which hosts a single bicycle. Stations are connected to the Internet and display in real time the occupation status of each slot. In this way users can easily check where it is possible to pick up or drop a bicycle. The usage of the system is monitored continuously, and the collected information is used to improve the level of service.

The daily operating costs include maintenance, insurance, possibly website hosting and electricity, and, most important, the cost due to the **redistribution of bikes among the stations**. Indeed, during the day some stations are typically full and others are empty. A commonly adopted rule for rebalancing is to keep each station only partly occupied, i.e., there should always be in a station some slots occupied by bicycles, to allow users to pick them up, and some free slots, to allow users to drop a bicycle at the end of their journey. Let us suppose that a desired level of occupation is present in the early morning in a given bike station, then the number of bikes may change drastically during the day from the desired level because of the users' travel behavior. This happens typically in cities characterized by a hilly territory, where users take a bike from a station located at the top of a hill, leave it at the bottom and then take the journey back with different means of transportation. It is also common for cities located in flat areas, where some stations have large inflows or outflows at different times of the day.

Repositioning is usually done by means of capacitated vehicles based on a central depot that pick up bicycles from stations where the level of occupation is too high and deliver them to stations where the level is too low. Usually a buffer of bicycles is kept at the depot, and used to allow a more flexible redistribution. The resulting optimization problem of deciding how to route the vehicles so as to perform the redistribution at minimum cost is known in the literature as the Bike sharing Rebalancing Problem (BRP), and will be the focus of the term project for COMP 6651, Fall 2019.

2 Formal Statement of the Bike Rebalancing Problem (BRP)

2.1 Notations

V^X	Set of street intersections
V^S	Set of bike stations
v_{D}	Depot
n^{BIKES}	Size of the bike fleet
n^{BS}	Number of bike stations
$n^{\text{AVAIL\_BIKES}}$	Overall number of available bikes in the bike stations
n^{V}	Number of vehicles
CAP	Capacity of the bike station and the vehicles: we assume bike stations and vehicles have the same capacity
D_v	Demand (or equivalently number of available bikes) at bike station v
δ	Duration of a worker shift (i.e., vehicle driver)
δ_{ij}	Travel time from node v_i to v_j (assume it includes bike loading/unloading and that it is independent of the number of bikes to be loaded/unloaded)

2.2 BRP Statement

We assume the city to be described by a city graph $G = (V, L)$ where V is a node set made of *(i)* street intersections, *(ii)* bikes stations and *(iii)* a depot, and L a set of streets connecting the nodes. Consequently,

$$V = V^X \cup V^{\text{S}} \cup \{v^{\text{D}}\},$$

where V^X is the set of street intersections, V^{S} is the set of bike stations and v^{D} is the node associated to the depot.

Assume we have a fleet of n^{BIKES} bikes. Some of them are in use, i.e., $n^{\text{BIKES}} - n^{\text{AVAIL\_BIKES}}$, while others ($n^{\text{AVAIL\_BIKES}}$) are in bike stations.

Capacitated vehicles can be used to relocate the bikes. The tour of a vehicle cannot last more than δ hours, taking into account that each tour starts and ends at the depot. Note that each tour corresponds to a cycle in the undirected graph (or a circuit in the directed city graph), as the vehicle driver starts and ends its shift at the depot.

We would like therefore to reposition the bikes, using the minimum number of vehicles in order that each bike station is half full, i.e., demand $D_v \geq .5 \times \text{CAP}$.

Part I: Data Sets and a Greedy Heuristic

3 Objective of Part I: Data Sets and a First Greedy Algorithm

The goal of the project, Part I, is to build some data sets and a first greedy heuristic.

3.1 Design and implementation of some data sets

Objective is to design and implement a data generator with the following INPUT parameters and OUTPUT data sets.

The city graph will be generated randomly, with a fixed number of edges ($|L|/2$), all equally probable, and each edge being replaced by two arcs, one in each direction (or, in other words, no one-way street). You need to make sure that the city graph is connected.

Assuming that the duration δ of a worker shift (i.e., drivers of the vehicles) is 8 hours, specify the range for the δ_{ij} values so that vehicle tours include the visit of at least $n^{\text{VISIT\_S}} = 4$ bike stations on average.

3.2 Greedy Heuristic

```
initialization;
while demand is not satisfied do
    Consider one vehicle r starting and returning to the depot;
    Define a tour for the vehicle in order to maximize the demand it will take
      care of it;
    Update the demand as the vehicle moves assuming the bike station
      demand remains stable
end
```

Algorithm 1: Greedy_BRP: Bike Repositioning Algorithm

4 What is required

1. A detailed description of your algorithms for both the data generator and the Greedy_BRP heuristic
2. A detailed complexity analysis of your greedy heuristic. Clearly indicate the definition of the set of parameters you use to express the time complexity
3. A set of 10 data sets generated with the following parameters:
   $n^{\text{AVAIL\_BIKES}} = 60$ bikes randomly distributed over the set of bike stations

$n^{\text{BS}} = 10$
At least 3 stations are deficit stations, i.e., stations with a number of available bikes in $[0, 4]$
$|V^X| = 20$ (number of street intersections)
70 edges in the city graph (each edge leads to two links, one in each direction)
$n^{\text{V}} = 100$
$\textsc{cap} = 10$
Objective: $D_v \geq 5$ (or 50%) for all $v \in V^{\text{S}}$

4. The output of your Greedy_BRP heuristic, with a summary of your results in Table 1 and Figure 1. Specify the characteristics of the computer used for your experiments.

5. Provide a critical analysis of the greedy algorithm: what are its key drawbacks?

Data Sets	# required vehicles	Computational times (sec.)
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Table 1: Performance of the Greedy Heuristic

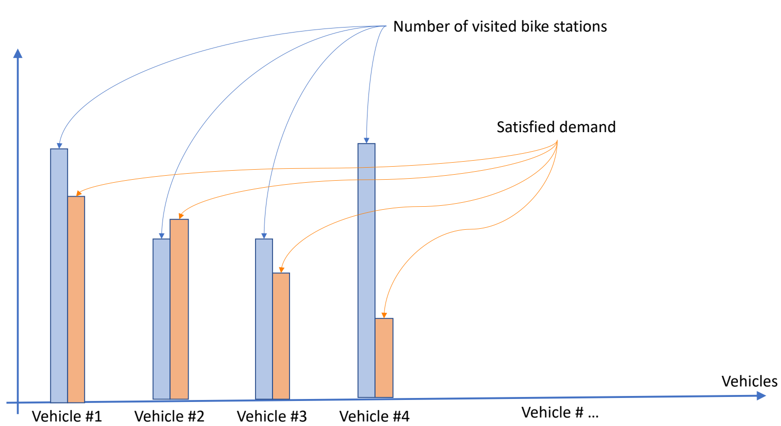

Figure 1: Demand Decrease and Number of Vehicles