CONCORDIA UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE AND SOFTWARE ENGINEERING

COMP 6651/3: Algorithm Design Techniques

Fall 2019

Project report due:
Part I - **October 4, 2019 before noon**
Part II - **October 27, 2019 before midnight**
Part III - **November 24, 2019 before midnight**
Part IV - **November 28, 2019 before midnight**
(electronic submission only)

Term Project

WARNINGS

- Parts I and II of the project are 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

Algorithm 1: Greedy_BRP1: Bike Repositioning Algorithm

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

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)
$n^{\text{V}} = 100$
$\text{CAP} = 10$
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

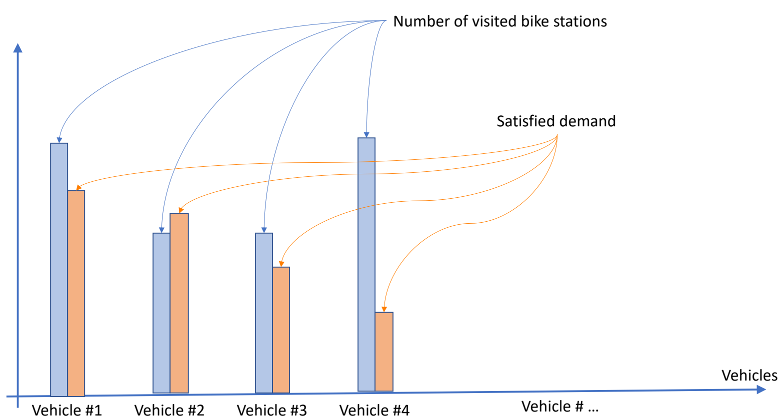

Figure 1: Demand Decrease and Number of Vehicles

Grading Scheme

Detail	Max points
Data generator	10
Greedy BRP heuristic	20
Complexity analysis of greedy heuristic	10
10 data sets	10
Summarize output of heuristic	20
Critical analysis of heuristic	10
Program	20

Part II: Better Data Sets and an Improved Greedy Heuristic

The goal of Part II is to improve on the generation of data sets and the design of a greedy heuristic.

4.1 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.

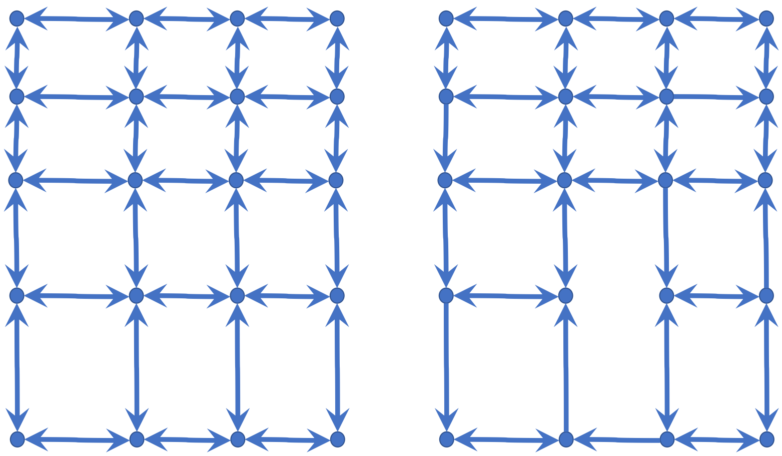

Figure 2: Generate city graph with a grid

The city graph will be generated randomly, using a grid as represented in Figure 2. Then, we will randomly eliminate some arcs in order to introduce one-way streets, while checking that the resulting grid remains connected, i.e., for any two pairs of nodes v, v' (i.e., intersections), there is always to go from v to v', and to v' to v. Next step is to generate a given number of bike stations along the streets (at random in between the two endpoints of a link) and then insert them in the city graph, see Figure 3. In other words, bike stations correspond to additional points, different from the street intersections. Lastly, we introduce the location of the depot and connect it as well to the city graph.

Question 1. Write a detailed algorithm in order to describe such a data set generation.

Question 2. Perform a detailed complexity analysis of the proposed algorithm.

Question 3. Generate 10 different data sets with the following parameters. Use the same input file format as in Part I.

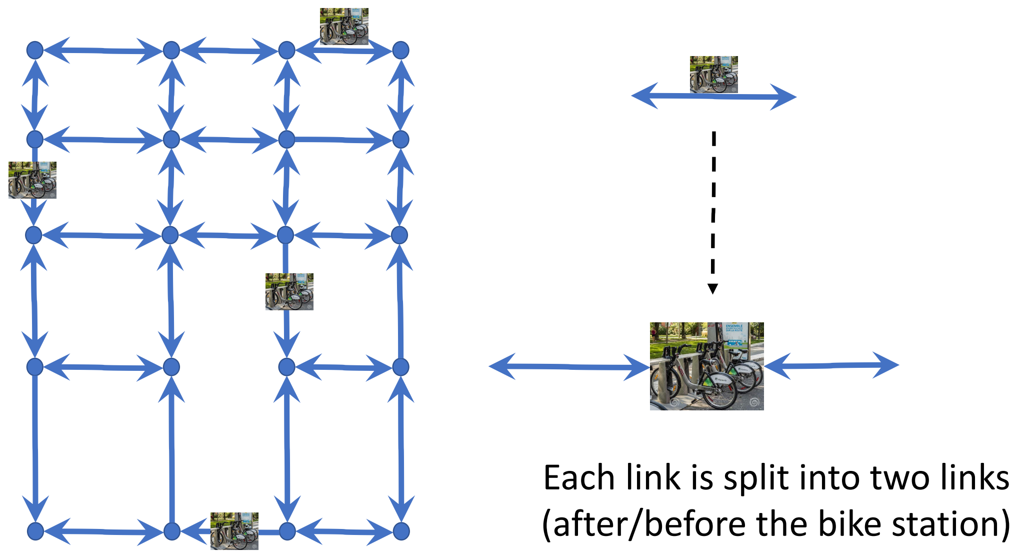

Figure 3: Add the bike stations

- Grid 10×12, i.e., 10 squares $\times$ 12 squares $\rightsquigarrow$ 11 intersections $\times$ 13 intersections. It also corresponds to $11 \times 12 + 10 \times 13 = 132 + 130 = 262$ edges, i.e., up to 524 arcs.
- Travel times of the arcs, at random in a set $[1, a]$. Explain how to select a.
- Overall set of arcs: choose a number of arcs so that close to 40% of streets are one-way streets ($\leftrightarrow$ $\equiv 2$ arcs, $\leftarrow$ $\equiv 1$ arc)
- 90 bike stations, same capacity as in Part I
- Capacity of the vehicles: 10, as in Part I
- Number of bikes distributed over the set of bike stations: estimate a number such that *(i)* it is possible to satisfy the demand without ideally adding too many bikes from the depot; *(ii)* about 50% of the bike stations are deficit ones (i.e., do nit satisfy the minimum number of bikes)

4.2 Design and implementation of a greedy heuristic

We assume that it is allowed to return some bike to the depot, while it should be avoided as much as possible. At the beginning of the day (i.e., before defining the set of tours), we assume that the depot contains no bike. During the construction of the tours, if the depot has some bike, the vehicle much leave with those bikes.

We assume that from each station, the time to go from the depot to the station, and the time to return to the depot from that station is shorter than 8 hours.

Algorithm 2: Greedy_BRP2: Bike Repositioning Algorithm

initialization;
for *every bike station* **do**
    Compute one shortest path from each bike station to the depot
end
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

Question 4. Write a detailed greedy algorithm in order to describe all the details of algorithm Greedy_BRP2. It must clarify how each station is selected along the tour.

Question 5. Rather that computing a single shortest path, assume you want to compute all shortest paths from a given bike station, and then use the shortest path with the largest number of bike stations. Write the details of the corresponding algorithm. Call Greedy_BRP3 the resulting algorithm for repositioning the bikes. What is the complexity of algorithm Greedy_BRP3?

Question 6. Implement algorithm Greedy_BRP2.

Question 7. Provide the results of algorithms Greedy_BRP1 (Part I) and Greedy_BRP2 using the same diagrams as for algorithm Greedy_BRP1 in Part I, i.e, Table 1 and Figure 1 should contain the average results over your 10 data sets (generated in Part II). Therefore, it means providing 2 tables and 2 figures for summarizing the results.

5 What you need to submit

1. A report with your answers to Questions 1 to 7
2. Your 10 data sets (Input/Output). Clearly indicate the numbers you used for generating your data sets: number of arcs for the city graph, number of bikes, parameter a for the transit times.
3. Your programs (with some comments)
4. Provide a critical comparison of the two heuristics.

6 Grading Scheme

Detail	Max points
Data generator	10
Complexity analysis of data generator's algorithm	10
10 data sets	10
Detailed pseudo code of greedy heuristic BRP2	20
Complexity analysis of BRP3	15
Compare BRP2 vs BRP3	15
Implementation of BRP2	10
Summarize results	10

Part III: A Different Type of Greedy Heuristic

The goal of Part III is to design a different type of greedy heuristic, in which you first partition the set of bikes, and then build a tour on each subset of bikes.

We will label a bike station as an EXCESS one if its number of bikes is larger than 3/4 of its capacity. We will label a bike station as a DEFICIT one if its number of bikes is smaller than 1/4 of its capacity. Consequently, the demand of a bike station is satisfied if its number of bikes is in between 1/4 and 3/4 of its capacity.

6.1 Proposed heuristic: Greedy_BRP3

1. Estimate the number n^{VEH} of vehicles **(Algorithm 1)**
2. Partition the set of bike stations into n^{VEH} cluster sets. **(Algorithm 2)**
3. Define a tour that maximizes the number of bike stations with a satisfied demand for each cluster set. **(Algorithm 3)**
4. If you cannot find a feasible solution (i.e., a solution that allows re-positioning the bikes as to satisfy the demand with 8h hour tours), return to Step 1 in order to adjust the number of vehicles.

Question 1. Write a detailed greedy algorithm for each step of Greedy_BRP3

Question 2. Is there an interest of keeping all the bike stations in Step 2 of Greedy_BRP3, i.e., even those with a satisfied demand? Justify your answer.

Question 3. Perform the complexity analysis of your three algorithms.

Question 4. Implement Greedy_BRP3 and compare its performance with Greedy_BRP2 of Part II.

Question 5. Provide the results of Heuristic_BRP3 using the same diagrams as for algorithm Greedy_BRP1 in Part I, i.e, Table 1 and Figure 1 should contain the average results over your 10 data sets (generated in Part II). Therefore, it means providing 2 tables and 2 figures for summarizing the results. Use the dat set generated in Part II. You can use them without modifying them, or you can regenerate new data sets with a capacity of the bike stations equal to 12.

7 What you need to submit

1. A report with your answers to Questions 1 to 5 of Part III. Clearly indicate whether you sue the same data sets as in Part II, or whether you regenerated new data sets.
2. Your programs (with some comments)

3. Provide a critical comparison of heuristics Greedy_BRP2 and Greedy_BRP3

Part IV: State of the Art

Question 1. Read the following paper and summarize it in a set of 10 slides.
S.C. Hoa and W.Y. Szeto, Hybrid large neighborhood search for the static multi-vehicle bike-repositioning problem, *Transportation Research*, Part B 95 (2017) 340363.
Please no cut and paste from the paper: you need to use your own words.

Question 2. Can you identify one weak point in the paper of Hoa and Szeto (2017)?

Question 3. Can you identify one strong point in the paper of Hoa and Szeto (2017)?

Question 4. What are your own comments on the numerical results?