Transportation of company workers for the reduction of traffic and pollution

FELIPE RÍOS LÓPEZ
UNIVERSIDAD EAFIT
FRIOSL@EAFIT.EDU.CO

SANTIAGO GIL ZAPATA
UNIVERSIDAD EAFIT
SGILZ@EAFIT.EDU.CO

TEACHER: MAURICIO TORO
DATA STRUCTURES AND ALGORITHMS 2

ABSTRACT:

The problem we had to solve is how could we reduce the traffic and the number of cars that go to a same place in a way that the drivers take another driver so then they could save the environment, meet new people, save money and also the most important requisite: without extending too much the time they usually take to get to that place. Some approaches to this problem are in example the delivery business that need to get the most efficient route to reduce the time and cost going to a certain ubication.

We can conclude that certainly this problem needs to be solved really in not much time because in the example mentioned before, if it takes too long to get the route, that could cause loses of money for the (empresa) or even a demand for not (cumplir) the expectations of the consumer.

B)

KEYWORDS

QUADTREE, SHORTEST PATH, MOBILITY EFFICIENCY.

ACM KEYWORDS

OBJECT ORIENTED PROGRAMMING, SOFTWARE, PROGRAM VERIFICATION, COMPUTER APPLICATIONS: GENERAL, COMPUTERS IN OTHER SYSTEMS.

ABSTRACT:

In this first delivery we searched four similar problems to the final Project with its respective solution, to have a guide and an idea of the objective of it. The Project consists of designing an algorithm to improve traffic through the strategy of shared vehicles, so we must find the minimum number of private vehicles in which all workers of a company can go to work picking up others, because if all the people enter at the same time to work, it generates too much traffic, and on the other hand, this could contribute to greatly improve the economy of the owners of the vehicles since they would spend less on gasoline, contribute to the environment and could meet their teammates.

These problems were found:

Colombian microenterprises cannot supply their business when they do not count with a routing system because of the lack of resources.

Finding the minimum path with reduced graphs.

The best way a person can choose their professional career.

Finding the most motivated workers of a company.

Description of the problems and solutions:

2.1 Supplying business

It was found that Colombian microenterprises are not able to supply themselves. Also, generally they use as transport bikes, motorcycles and those are not enough to get to the place of all the clients they have and that's why they end up losing time and clients.

Solution:

National University done an adaptation of SDVRPTW by heuristics.

SDVRPTW consist in:

Given

-A limited number of vehicles with capacity Q located in a central point.

- A set of clients demanding a product.

Get the routes that:

Can satisfy the demand of clients

Starting and ending in the central point

Not exceeding the capacity of the vehicle

Every route must accomplish the limit time.

The total distance must be minimized.

1. Generate factible initial solutions construction routes inserting the most near client to the farthers.
2. Realize a searching by the initial solution
3. Post optimization

In the initial algorithm, for the insertion of a new client in a route it is evaluated that the vehicle can get back to the central point in the limit time and if a client was already attended and his demand has not been served, the quantity of products to distribute in the process should satisfy totally the demand. All of this to use the maximum capacity of the vehicles and satisfy the clients with no more than 2 vehicles at time.

Constructed those routes the post optimization combines randomly each position of the route searching a better solution to minimice the total distance.

2.2 Reduced graphs

Generally, the solution to these problems is found using algorithms like Djikstra or A*, but they are useless for big graphs due to the large amount of time they take to respond. There exist many approaches to the solution but they have errors. Even though, we can take advantage of this errors to make our own solution, the problem is that minimizing the graphs means losing information, and that means it does not grant optimum solutions.

Description of the solution:

In various problems that are solved using graph's theory, it is common the need of the transformation of a graph, this means, a reduction if the new graph has less quantity of vertex.

Gi=({Vi},{}) is a graph where Vi belongs to Gi
Gj = (Vj, Ej), is a graph
Vin and Vout are 2 sets of information where the edge orientation is defined We must rewrite the writing of the graph in the following way, with "in" and "out" (see table 3.1)
Algorithm:
1. Input: G = (Vr, Er, f, R) R is a set of writing rules
2. If the entry is an equivalence relation
3. Obtain partition of P (V / RE). /End yes
4. Construct the set of vertices Vr of the reduced graph Gr from P.

5. Construct set of Er edges where Vm belonging to the partition, be minimum.
6. For all partitions:
7. If the | minimum |> 1, write a rewrite rule to substitute the vertex / Endpoint subparagraph for
8. Create the reduced graph Gr: = (Vr, Er, f, Rr)
9. For all partition
10. If the minimum> 1, update (G, Rr [ai], Gi, Rr [minimum], Gj, fr) ./ End to.
11. Return Gr.

2.3 Choosing a career or work.

You need to know which is the job in which it is more feasible for a person to fit easily in order to choose a career in a safe way, machine learning has a great advance in these territories of people and human resources management, but which Is the best way to store this data? so that they are totally personalized and effective in all people, who may look for a stable job and have a degree or simply

dedicate themselves to what they are good at, such as art, music or some other activities, it takes more than simple human management . Description of the solution:
The learning experience platform algorithm that requires Big Data and artificial intelligence that can label the grouped and decentralized content in an efficient way and then recommend training courses taking into account the profile of a person, their work experience and interests; the result must be a hierarchical range of courses that will surely interest you the most.
The tool detects the positions that may be interesting or aspects in which the person needs to improve based on the person and historical data of other people who occupied the same position, with this incentive to retain the worker. Something like what vocational tests do but with many improvements and extra options that the algorithm can learn by itself.
2.4 Motivation of workers
It is a common mistake to believe that all people are motivated by the same actions, such as thinking that retribution is the main motivation of all, so you need to find a way to know how to motivate each worker to improve the profits of a company.
Description of the solution:
We need a 4-step algorithm to define motivation.
1. Choose what employees and managers consider to be motivating, grouping them into blocks of management, development, communication, compensation, self-motivation, direction, monetary compensation, ethics, teamwork.
2. Each employee evaluates the factors according to their perception, with linguistic labels or intervals to have a more or less precise scale.
3. Design an ideal profile; leaders create an employee profile according to company policy

4. Fit between data and employees; order from lowest to highest employees that are most assimilated to that profile.

TABLAS:

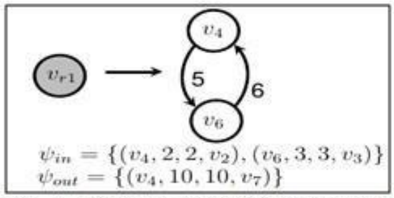
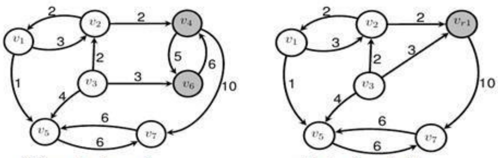


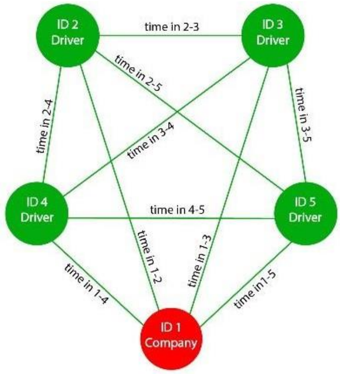
Fig. 1. Ejemplo de regla de reescritura



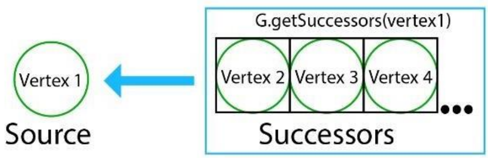
a) Ejemplo de grafo b) Grafo reducido
Fig. 2. Ejemplos de grafo

3. Referencias

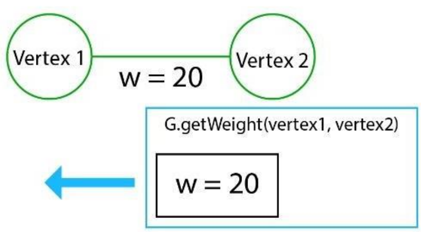
https://revistas.unal.edu.co/index.php/dyna/article/view/46104/53861 (1)
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59362017000100004 (2)
http://www.josemanuelmartin.com/2018/03/la-experiencia-netflix/
https://retina.elpais.com/retina/2018/01/15/talento/1516032354_866124.html (3)
https://www.observatoriorh.com/motivacion-y-compromiso/40234.html (4)



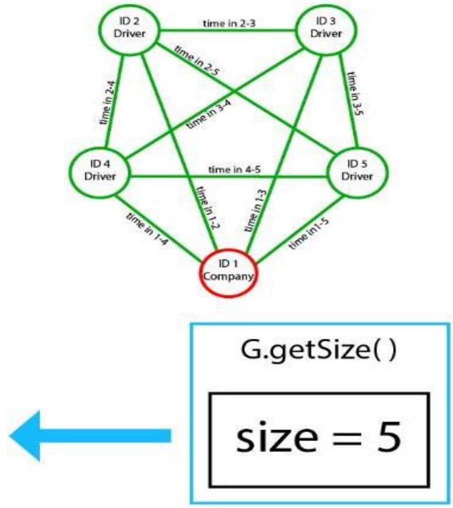
getSuccessors(v1)



getWeight(v1, v2)



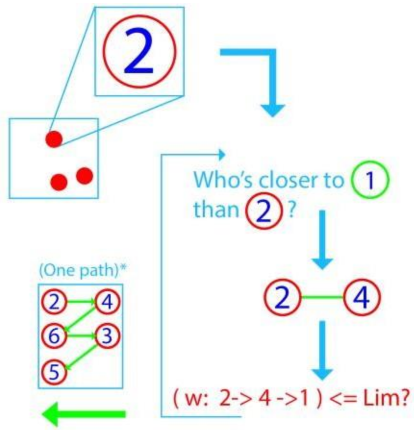
getSize( )



Being n the number of vertices that the graph has

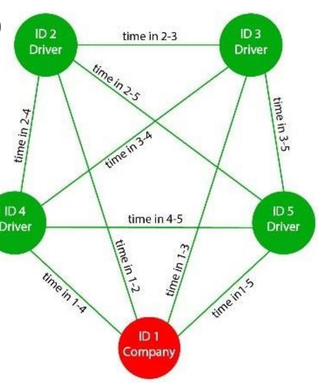
Method	Comp
Add arc	O(1)
Get successors	O(n)
Get weight	O(1)
Get size	O(1)

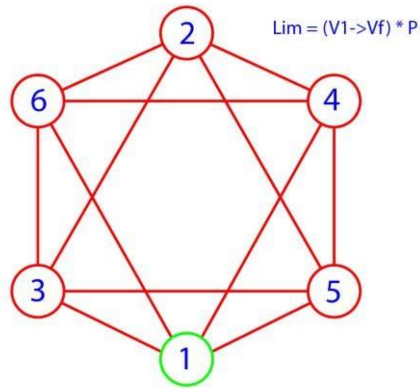
For the case of assignation of cars, we noticed that it is a problem that can be developed with the use of complete graphs not directed where every vertex represents a house of a person. The time that takes the person to ger other vertex is the weight of the edges which also connect to all the other vertex. For this implementation of the graph we decided to make an adjacency matrix, because if it is a complete graph, there exists an arc and a weight with a pair of vertex that we can save in that matrix



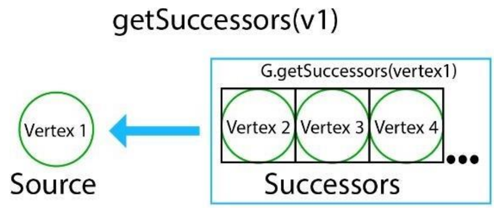
J) First we need to analize an especific área where a person can be taken to work by another person without exceeding the number of passengers. And so, that is why we calculate the proximity of the locations in order to avoid comparing those which are too far from each other, keeping the others and trying to get the most efficient path to work of those workers.

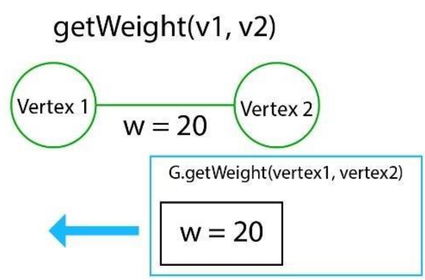
C)

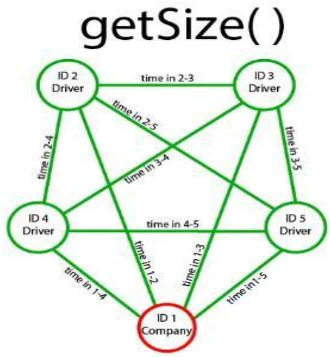


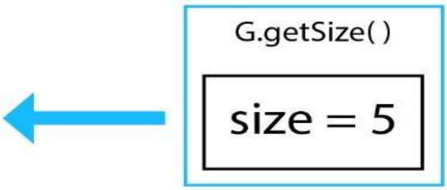


D)









e) Structure complexity

Being n the number of vertices that the graph has

Method	Comp
Add arc	O(1)
Get successors	O(n)
Get weight	O(1)
Get size	O(1)

f) We basically updated our algorithm because we were not really taking into account the given coordinates, they were unnecessary. Everything is the same except for that.

g) Explication of the algorithm:

First we need to understand that the data we have is a complete graph, where every vertex has a path to all the other vertex. Then what we did with the data was to organize the drivers (vertex) in a vector by the closest from destiny point (in our case EAFIT university) to the farthest to EAFIT, but the exception here is that in the first place of the vector we have EAFIT. After that, we also organize in the same way in another vector (of vectors) containing the closest successors to the driver.

The rest are simple calculations with P (time that the driver takes * P) that the driver is minded to give, the restriction of the maximum passengers that a car can take to the central point (in this case, 5) and how far are the other drivers from the driver that will probably take them to work.

h) Complexity analysis:

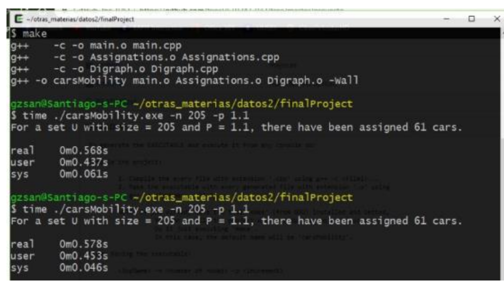
Worst of the cases: $O(n)^2$

Average= O/(n logn)

Best: n^2

i) we thought that it was easier to make it that way because it was not that really complex to make and turns out to be effective for a massive amount of cars, due to the O(n) in the worst of the cases for all the algorithm

j) Time: 0.456 seconds



RAM memory consumption: 3.4 MB

For the structure:

The graph uses a matrix NxN N is the number of cars. That means that in memory there are stored N^2 int values.

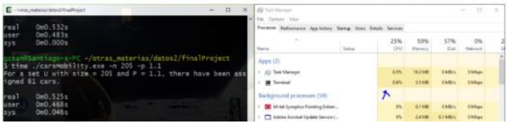
Taking into account that an int consumes 32 bits and the biggest dataset has 205 values, the data structure complexity: O(N^2)

In the biggest dataset= (32*205)^2 = 43.033.600 bits

With an average RAM of 4 GB,it is 0.000012% of the total memory.

Algorithm complexity: O(N): The cars are only storaged once...

In the biggest dataset= (32*205) = 6560 bits



k) The basic concept for this algorithm is to organize in a list the cars from the nearest to the farthest to the company (EAFIT), then, we have a list inside that list that contains the successors, sorted again in the same way. After that the calculations are made and it returns the size of the new array containing the drivers in a txt file.

We did not have an actual first solution, but the most accurate approach to our problem was using reduced graphs where the size of the graph will be less as we take the passengers.

In the second solution we had an idea on how to do the algorithm, but we did not have a solution, we just made the data structure, but now with the algorithm done, we are satisfied because it is as fast as we planned it to be and the cars where reduced nearly 30% of the total, and that's a really good improvement.

In the future we could implement graphics where the paths could actually be seen in real time or an interface more readable for a non-programmer user.

l) This project was made possible by ANDI and SAPIENCIA for supporting our studies.

m) cybergraphy

https://revistas.unal.edu.co/index.php/dyna/article/view/46104/53861 (1)
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59362017000100004 (2)
http://www.josemanuelmartin.com/2018/03/la-experiencia-netflix/
https://retina.elpais.com/retina/2018/01/15/talento/1516032354_866124.html (3)
https://www.observatoriorh.com/motivacion-y-compromiso/40234.html (4)