

TRANSPORTE DE TRABAJADORES DE EMPRESAS PARA LA REDUCCIÓN DEL TRÁFICO Y CONTAMINACIÓN

FELIPE RÍOS LÓPEZ

UNIVERSIDAD EAFIT

FRIOSL@EAFIT.EDU.CO

PROFESOR: MAURICIO TORO

ESTRUCTURAS DE DATOS Y ALGORITMOS 2

SANTIAGO GIL

UNIVERSIDAD EAFIT

SGILZ@EAFIT.EDU.CO

A) KEYWORDS

QUADTREE, SHORTEST PATH, MOBILITY EFFICIENCY.

B) ACM KEYWORDS

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

C)

Resumen

En esta primera entrega buscamos 4 problemas parecidos al del Proyecto final con su respectiva solución, para tener una guía y una idea del objetivo del mismo. El Proyecto consiste en diseñar un algoritmo para mejorar el tráfico mediante la estrategia de vehículos compartidos, de modo que debemos encontrar el mínimo número de vehículos particulares para que todos los trabajadores de una empresa puedan ir a trabajar recogiendo a otros, debido a que generalmente todas las personas entran a una misma hora a trabajar, se generan muchos tacos, y por otro lado, esto podría contribuir a mejorar mucho la economía de los dueños de los vehículos ya que gastarían menos en gasolina, contribuirían al medio ambiente y podrían tener una charla agradable con sus compañeros.

1. INTRODUCCIÓN

Se encontraron los siguientes problemas:

Cómo las microempresas colombianas pueden abastecer sus negocios cuando no cuentan con un Sistema de enrutamiento de vehículos debido a falta de recursos.

Buscar un camino mínimo por medio de grafos reducidos para así mejorar su eficacia en el tiempo.

Buscar la forma más factible de que una persona pueda escoger su Carrera o puesto de trabajo.

Algoritmo para encontrar los empleados más motivados de una empresa.

2. DESCRIPCIÓN DE LOS PROBLEMAS Y SOLUCIONES

2.1 Abastecimiento de negocios en empresas con falta de recursos.

Se encontró que algunas microempresas colombianas, encargadas del abastecimiento de algunos negocios, que no cuentan con un sistema de enrutamiento de vehículos debido a recursos insuficientes, realizan este proceso manualmente.

Además, las microempresas por lo general usan vehículos pequeños como bicicletas y motos con tráiler, para los diferentes sectores a los que deben llegar. Cabe destacar que algunas veces los vehículos no alcanzan a cubrir a todos los clientes que se encuentran en la ruta. Por lo que terminan fraccionando los pedidos de los clientes en varias entregas. Las microempresas deberían ser capaces de fraccionar la demanda del cliente en varios vehículos con el propósito de satisfacer los requerimientos del sistema y adicionalmente, reducir el número de rutas y la distancia recorrida por los vehículos. Dichas microempresas desean maximizar el uso de transportes alternativos, reducir el impacto ambiental y social de sus actividades de distribución, mejorar sus debilidades en el ordenamiento territorial y mejorar el uso del espacio de almacenamiento, así como en unitarización de la carga, entre otras.

2.1.1 Descripción de la solución:

Para la solución de este problema, la Universidad Nacional de Colombia hizo una adaptación de SDVRPTW (Búsqueda de rutas de vehículos para entregas divididas en ventanas de tiempo).

Un SDVRPTW consiste en:

Dado:

- Un número ilimitado de vehículos homogéneos con capacidad Q localizados en un depósito central.
- Un conjunto de clientes con una demanda de productos (incluso superior a la capacidad de un vehículo).

Hallar las rutas que:

- Satisfagan la demanda de los clientes (la demanda puede ser fraccionada en varios vehículos, por lo que un cliente puede ser visitado más de una vez).
- Cada ruta empieza y termina en el depósito.
- Cada ruta no puede exceder la capacidad del vehículo asignado.
- Cada ruta debe cumplir las restricciones de ventanas de tiempo.
- La distancia total debe ser minimizada

La implementación de SDVRPTW implica tener una Heurística que cumpla con sus criterios. La heurística desarrollada en este caso contiene tres etapas:

- La primera etapa consiste en generar una serie de soluciones iniciales factibles construyendo rutas a través de la inserción del cliente más cercano al último cliente insertado en la ruta, ignorando el tiempo de espera en cada parada.
- En la segunda etapa realiza una búsqueda tabú a partir de la solución inicial para mejorar las rutas.
- En la tercera etapa se ejecuta un proceso de post-optimización.

En el algoritmo inicial, para la inserción de un nuevo cliente en una ruta en construcción, se evalúa que el vehículo pueda regresar al depósito dentro de la ventana de tiempo del depósito y que si un cliente ya fue atendido y tiene un remanente de demanda insatisfecha, la cantidad de productos a distribuir a dicho cliente en el proceso de inserción en la actual ruta en construcción debe satisfacer el remanente de demanda total de dicho cliente; todo esto con el ánimo de usar toda la capacidad disponible de los vehículos y satisfacer las necesidades de los clientes con no más de dos vehículos.

Construidas todas las rutas necesarias para satisfacer los requerimientos de los clientes, el algoritmo de post-optimización combina aleatoriamente cada posición de cada ruta buscando una mejor solución que minimice la distancia total recorrida en cada ruta y si la encuentra, almacena esa nueva secuencia.

Finalmente, el algoritmo propuesto asigna cada ruta creada a un vehículo teniendo en cuenta que el tiempo de servicio de cada ruta que se añada al vehículo no tome más tiempo que la ventana de tiempo del depósito. Las ventanas de tiempo de cada cliente visitado en la ruta añadida deben ser, añadiendo la siguiente ruta con el mínimo tiempo de espera. De esta manera se itera hasta que todas las rutas hayan sido asignadas a un vehículo.

2.2 Camino mínimo por medio de grafos reducidos:

Generalmente, la solución de estos problemas se busca por medio de algoritmos como Dijkstra o A^* , pero son poco útiles para grafos muy grandes debido a su largo tiempo de respuesta. Existen varios autores que han propuesto soluciones a ello pero dichas soluciones traen consigo mismas errores. Aunque de estos errores podemos sacar enfoques interesantes para nuestro provecho en la solución del mismo, pero por otro lado, existen algoritmos que permiten reducir un grafo al aplicar cualquier algoritmo sobre el mismo, pero su consecuencia es la pérdida de información, por lo que no se garantiza que se obtenga el resultado óptimo.

Descripción de la solución:

En varios problemas que se resuelven usando la teoría de grafos, se hace común la necesidad de transformar un grafo, es decir, una reducción si el nuevo grafo tiene una menor cantidad de vértices.

$G_i = (\{V_i\}, \{E_i\})$ un grafo donde V_i pertenece a V

$G_j = (V_j, E_j)$, es un grafo

V_{in} y V_{out} son 2 conjuntos de información donde se define la orientación de las aristas. Debemos reescribir la escritura del grafo de la siguiente manera, con "in" y "out" ([mirar tabla 3.1](#))

Algoritmo:

1. Entrada: $G = (V_r, E_r, f, R)$ R es un conjunto de reglas de escritura
2. Si la entrada es una relación de equivalencia
3. Obtener partición de $P(V/R)$. /Fin si
4. Construir el conjunto de vértices V_r del grafo reducido G_r a partir de P .

-
5. Construir conjunto de Aristas E_r donde V_m que pertenece a la partición, sea mínimo.
 6. Para toda partición:
 7. Si el $|mínimo| > 1$, escribir una regla de reescritura para sustituir por el subgrafo del vértice /Fin para
 8. Crear el grafo reducido $Gr := (V_r, E_r, f, R_r)$
 9. Para toda partición
 10. Si el mínimo > 1 , actualizar($G, R_r[ai], G_i, R_r[mínimo], G_j, fr$)./Fin para.
 11. Retornar Gr .

2.3 Escoger Carrera o trabajo

Se necesita saber cuál es el trabajo en el que es más factible que una persona pueda encajar fácilmente para así poder escoger una carrera de forma segura, el machine learning tiene un gran avance en estos territorios de gestión de personas y recursos humanos, pero ¿cuál es la mejor forma de almacenar esos datos? para que sean totalmente personalizados y efectivos en todas las personas, que puede que busquen un trabajo estable y tener un título o simplemente dedicarse a lo que son buenos, como el arte, la música o algunas otras actividades, se necesita más que simple gestión humana.

Descripción de la solución:

El algoritmo de learning experience platform que requiere de Big Data e inteligencia artificial que pueda etiquetar el contenido agrupable y descentralizado de forma eficaz para luego recomendar cursos de formación teniendo en cuenta el perfil de una persona, su experiencia laboral e intereses; el resultado debe ser un abanico jerarquizado de cursos que seguramente más le interesarán.

La herramienta detecta los puestos que pueden ser interesantes o aspectos en que la persona necesite mejorar basándose en la persona y en datos históricos de otras personas que ocuparon el mismo puesto, con esto se incentiva a retener al trabajador. Algo como lo que hacen los test vocacionales pero con muchas mejoras y opciones extra que puede aprender el algoritmo por sí mismo.

2.4 Motivación de los trabajadores

Es un error común creer que todas las personas se motive por las mismas acciones, como pensar que la retribución es la motivación principal de todos, por eso se necesita encontrar una forma de saber cómo motivar a cada trabajador para mejorar las ganancias de una empresa.

Descripción de la solución:

Necesitamos un algoritmo de 4 pasos para definir la motivación.

1. Escoger qué consideran los empleados y los directivos como motivador, agruparlos en bloques de gestión, Desarrollo, comunicación, compensación, automotivación, dirección, compensación monetaria, ética, trabajo en equipo.
2. Cada empleado evalúa los factores según su percepción, con etiquetas lingüísticas o intervalos para tener una escala más o menos precisa.
3. Diseñar un perfil ideal; los líderes crean un perfil de empleado acorde a la política de la empresa

4. Ajuste entre datos y empleados; ordenar de menor a mayor los empleados que más se asimilen a dicho perfil.

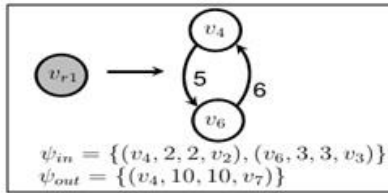
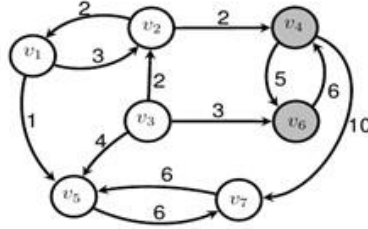


Fig. 1. Ejemplo de regla de reescritura

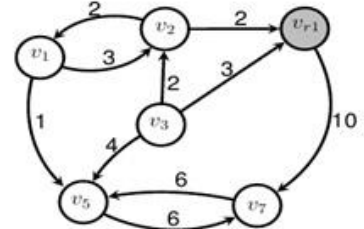
3. TABLAS

3.1



a) Ejemplo de grafo

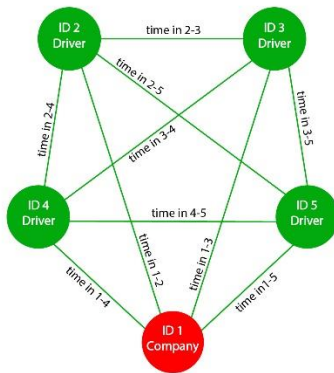
Fig. 2. Ejemplos de grafo



b) Grafo reducido

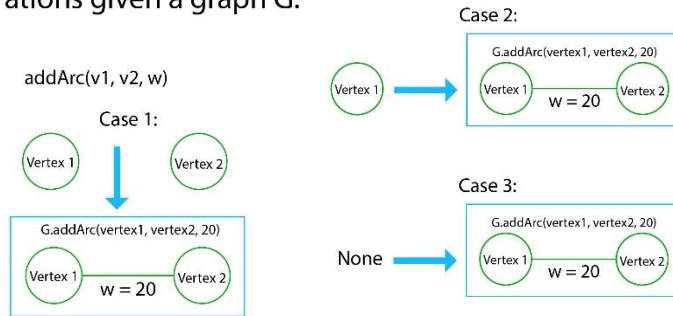
4. Referencias

- <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)

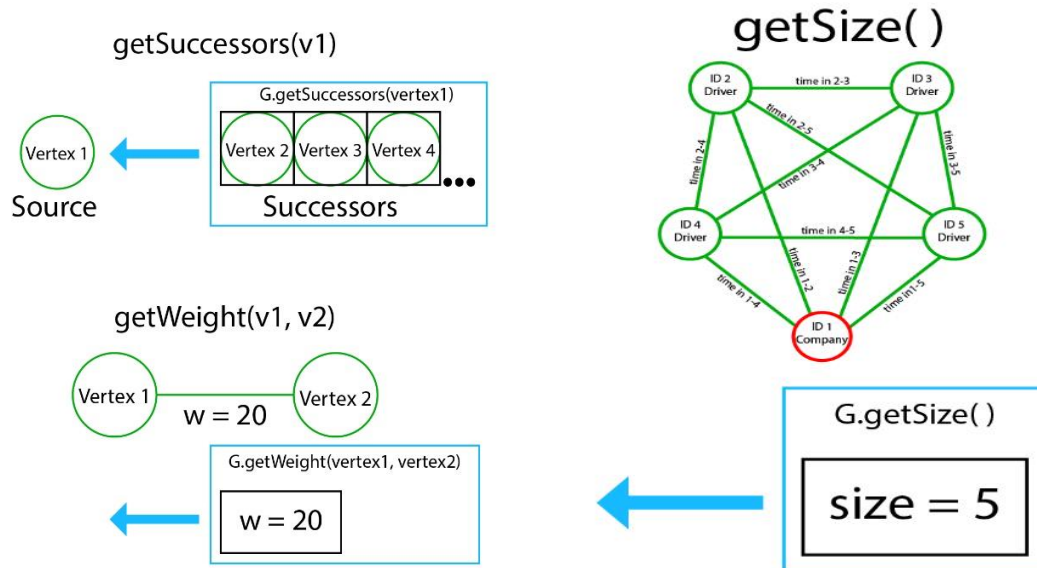


D)

Operations given a graph G:



E)



F)

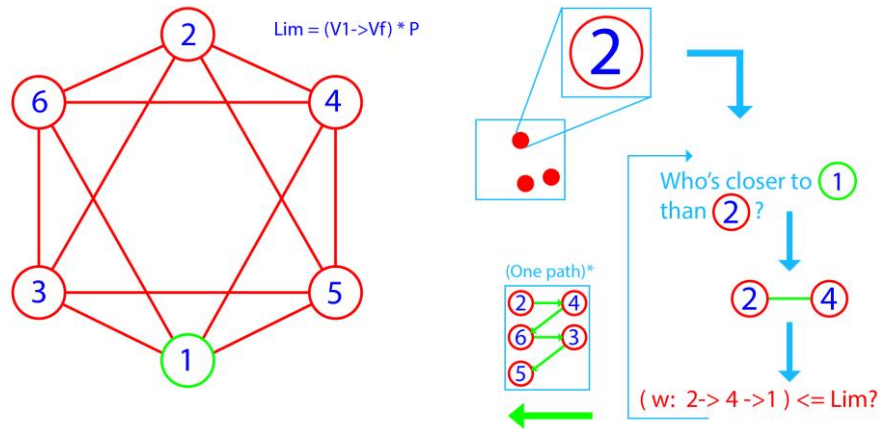
Being n the number of vertices that the graph has

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

G) For the case of assignation of cars, we noted that it is a problem that can be developed with the use of complete graphs not directed where every vertex represent a house of a person. The time that takes the person to get other vertex is the weight of the edges which also connect to all the vertex. For the 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.

H)



I)

rmZeros $O(1)$

getLine $O(s)$

makeline $O(s)$

extractk $O(s * \log s)$

$O_{\text{tot}} = O(s * \log s)$

J) First we need to analyze an specific area where a person can be taken to work by another person without exceeding the number of passengers. And so, that's 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.