

# Propuesta de valores en rutas desconocidas para el uso del algoritmo de Clarke – Wright y construcción de un algoritmo de ruteo. Caso practico

Jonatán Edward Rojas Polo, Mg.<sup>1</sup>, Alexia Cáceres Cansaya<sup>1</sup>, Cesar Stoll Quevedo, Dr.<sup>1</sup>, y Jackeline Alva Zelada<sup>1</sup>

<sup>1</sup>Pontificia Universidad Católica del Perú, Perú, jrojas@pucp.pe, alexia.caceres@pucp.pe, cstoll@pucp.pe, jalvaz@pucp.pe

*Resumen– Esta investigación surgió mediante la búsqueda de la Mejora en el abastecimiento y recolección de materia prima en una empresa agroindustrial. En el transcurso de dicha investigación central, desviamos la atención en la optimización del ruteo de vehículos usando el algoritmo de Clarke y Wright, específicamente en los valores de ahorros generados entre cada par de puntos (sitios), el problema surge cuando no tenemos la distancia entre dos sitios y se asume como distancia un valor muy grande,  $M$ , dado que es un problema de minimización de la distancia a recorrer, en la factibilidad de la solución óptima no se tomará dicho valor.*

*La propuesta de esta investigación aborda a redes (grafos) que no presenten un ciclo Hamiltonianos, o rutas desconocidas. Se usó dos tipos de artificios. El primero es un artificioso en base al algoritmo de Dijkstra en las rutas desconocidas, lo cual implica de que existe una probabilidad de circular más de una vez en un lugar, no obstante ello es permitido para no caer en la Infactibilidad al hallar la solución óptima. El segundo artificioso se centra en colocar un valor grande, no obstante aquí si se discrimina con la cercanía entre los sitios, es decir para las distancias no existentes se coloca  $M(n-1)$ , siendo  $M$  un valor grande y  $n$  el mínimo número de sitios intermedios que existen en los dos sitios de interés. Finalmente se validó los dos tipos de artificios y se obtuvo un algoritmo más amigable y eficiente para Rutear vehículos.*

*Palabras claves - VRP, Clark & Wright, Dijkstra, Ciclos no Hamiltonianos.*

*Abstract– This research came about through the search for Improvement in the supply and collection of raw material in an Agrobusiness company. In the course of this central research, we turned our attention to the optimization of vehicle routing using the Clarke and Wright algorithm, specifically in the savings values generated between each pair of points (sites), the problem arises when we do not have the distance Between two sites and assuming as a distance a very large value,  $M$ , since it is a problem of minimization of the distance to travel, in the feasibility of the optimal solution will not take that value. The proposal of this research approaches to networks (graphs) that do not present a Hamiltonian cycle, or unknown routes. Two types of artifices were used. The first is an artificioso based on the Dijkstra algorithm in unknown paths, which implies that there is a probability of traveling more than once in one place, however this is allowed not to fall into Infractibility when finding the optimal solution.*

Digital Object Identifier (DOI): <http://dx.doi.org/10.18687/LACCEI2017.1.1.453>  
ISBN: 978-0-9993443-0-9  
ISSN: 2414-6390

*The second artificioso focuses on placing a large value, however here if it is discriminated with the proximity between the sites, ie for non existent distances is placed  $M(n-1)$ , where  $M$  is a large value and  $n$  the minimum number of Intermediate sites that exist in the two sites of interest. Finally the two types of devices were validated and a more friendly and efficient algorithm was obtained for Routing vehicles.*

*Keywords - VRP, Clark & Wright, Dijkstra, Hamiltonian Cycles.*

## I. INTRODUCCIÓN

Durante los últimos años el problema de distribución ha ido tomando un rol muy importante en la logística, ya que el costo de distribución están alrededor del 20% al 30% del costo total de los productos [1], por lo cual es fundamental usar técnicas de optimización matemática.

Buscando métodos para la solución de este problema, se acudió a herramientas de la teoría de redes, tales como los caminos Hamiltonianos, que se caracterizan porque conectan las diferentes aristas de un grafo sin pasar por un mismo arco más de una vez, y conectando todos los puntos; además del uso de heurísticas de ruteo de vehículos (VRP). Lo propuesto fue identificar todas las variables relevantes para luego elaborar el grafo de red representativo, diseñar un algoritmo ideal para resolver este caso considerando los principios de caminos Hamiltonianos [2].

## II. ESTADO DEL ARTE

Para esta investigación se recopiló literatura científica del algoritmo Clarke y Wright, VRP (Vehicle Routing Problem), y el TSP (Traveling Salesperson Problem).

*Algoritmo de Clarke y Wright (El algoritmo de ahorros)*

El algoritmo fue propuesto por G. Clarke y J.W Wright [3], considera posible la asignación de los camiones de carga. Donde cada punto de clientes estará vinculado a otros dos puntos, uno de los cuales será el depósito  $P_0$ , y los otros serán los dos puntos  $P_y$  y  $P_z$  a los cuales se les considera los puntos vinculados a  $P_y$  que serán  $P_{y\pm 1}$  y a  $P_z$  que serán  $P_{z\pm 1}$ . El efecto de agrupar  $P_y$  y  $P_z$  será calculado. Además el algoritmo supone que  $P_y$  y  $P_z$  están en diferentes rutas desde  $P_0$ . Si estuviesen en

la misma ruta se aplicarán las mismas consideraciones, excepto que uno de los siguientes cuatro casos no estará permitido.

La figura 1.a muestra las posiciones de  $P_y$  y  $P_z$  en una posible asignación. La figura 1.b, 1.c, 1.d y 1.e muestran las cuatro descomposiciones posibles de estas rutas causada por unir a  $P_y$  y  $P_z$ . Estas consisten en la ruptura de los vínculos  $P_{y-1}P_y$  o  $P_yP_{y+1}$  con la ruptura de los vínculos  $P_{z-1}P_z$  o  $P_zP_{z+1}$ .

Las distancias ‘ahorradas’ para cada una de estas cuatro distintas agrupaciones son:

- Ahorro 1.b:  $d_{y,y+1} - d_{0,y+1} + d_{z,z+1} - d_{0,z+1} - d_{y,z}$
- Ahorro 1.c:  $d_{y-1,y} - d_{0,y-1} + d_{z,z+1} - d_{0,z+1} - d_{y,z}$
- Ahorro 1.d:  $d_{y,y+1} - d_{0,y+1} + d_{z,z-1} - d_{0,z-1} - d_{y,z}$
- Ahorro 1.e:  $d_{y-1,y} - d_{0,y-1} + d_{z,z-1} - d_{0,z-1} - d_{y,z}$

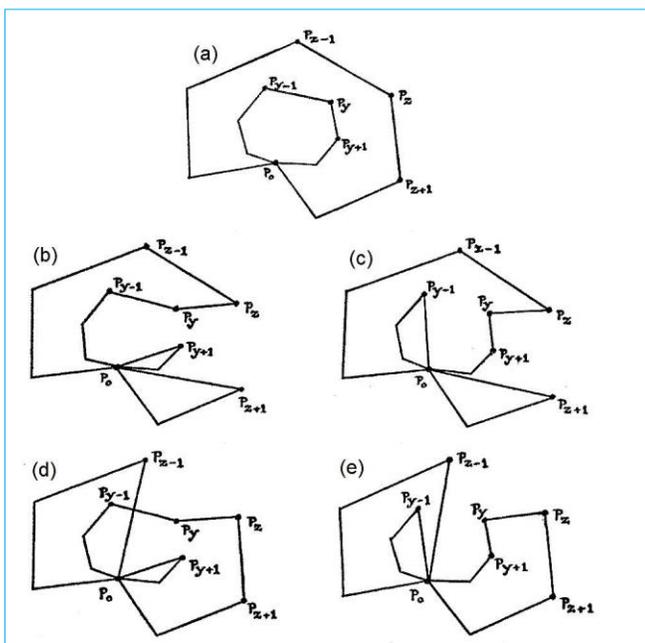


Fig. 1 Conexión entre el depósito y dos clientes  
Fuente: G. Clarke y J.W Wright (1962)

Estos cuatro tipos de ahorros se calculan para cada par de clientes. Una vez que los clientes están agrupados, el ahorro será recalculado. En programación lineal es equivalente a asignar dos precios duales a un cliente, el precio dual para el cliente  $P_y$  será  $d_{y-1,y} - d_{0,y-1}$  y  $d_{y,y+1} - d_{0,y+1}$ . Luego si un enlace se rompe, el precio dual será la celda que causa la ruptura, dado que este valor de la celda es un valor máximo, cada vez que un punto esté enlazado a otros dos (no el depósito  $P_0$ ), todos los valores de las celdas serán negativos, y este punto no será considerado para enlazar.

Finalmente como resultado de esto, los únicos vínculos que se separarán serán los de los puntos vinculados a  $P_0$ , si estos estuvieron involucrados en la vinculación de la celda (y: z), el ahorro será  $d_{0,y} + d_{0,z} - d_{y,z}$ .

Una manera más simple de entender este algoritmo es explicado por Flores [4], el cual menciona que si existe una ruta *depósito - i - depósito* y una ruta *depósito - j - depósito*, al combinarse formarán una nueva *depósito - i - j - depósito* como se muestra en la figura 2, generando así un ahorro por cada unión de éstas. El ahorro es calculado mediante el costo de la suma de las rutas independientes menos el costo de la ruta combinada, el ahorro es denotado por  $S_{ij}$ .

$$S_{ij} = [(C_{oi} + C_{io}) + (C_{oj} + C_{jo})] - [C_{oi} + C_{ij} + C_{jo}]$$

$$S_{ij} = C_{oi} + C_{oj} - C_{ij}$$

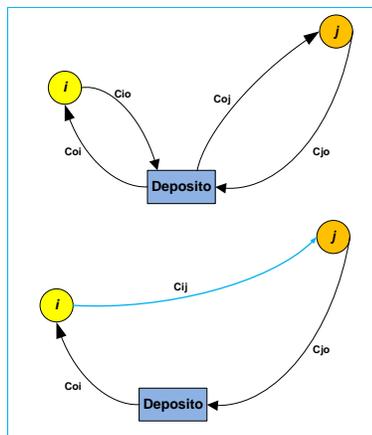


Fig. 2 Rutas, antes y después de combinarse

### Ciclos Hamiltonianos

A inicios del siglo XIX, el matemático Irlandés William Rowan Hamilton (1805 - 1865) diseñó el juego del dodecaedro regular, que luego fue vendido a un distribuidor de juegos en Londres en el año de 1859 por 25 libras, este juego consistía en una lámina de cartón con agujeros, tal como se muestra en la Figura 3, donde los vértices del dodecaedro regular se etiquetan con los nombres de las ciudades, cuya tarea era encontrar un recorrido circular a lo largo de los bordes del dodecaedro visitando cada ciudad exactamente una vez [5].



Fig. 3 El juego del dodecaedro - ICOSIAN  
Fuente: Parada (2009)

**Teorema 1:** Para un número entero  $n \geq 4$ , existe un ciclo Hamiltoniano cíclico de  $K_n - I$  si y sólo si  $n \equiv 2, 4$  (residuo 8) y  $n \neq 2p^\alpha$  donde  $p$  es un número primo e impar y  $\alpha \geq 1$  [6].

## VRP (Vehicle Routing Problem)

El VRP, problema de ruteo de vehículos (Vehicle Routing Problem, o VRP), es considerado un problema de gran importancia y un paradigma en la literatura especializada [7]. Este tipo de problema plantea a un almacén central que cuenta con una flota de vehículos y debe atender a un conjunto de clientes geográficamente distribuidos [8], tal como vemos en la figura 4.

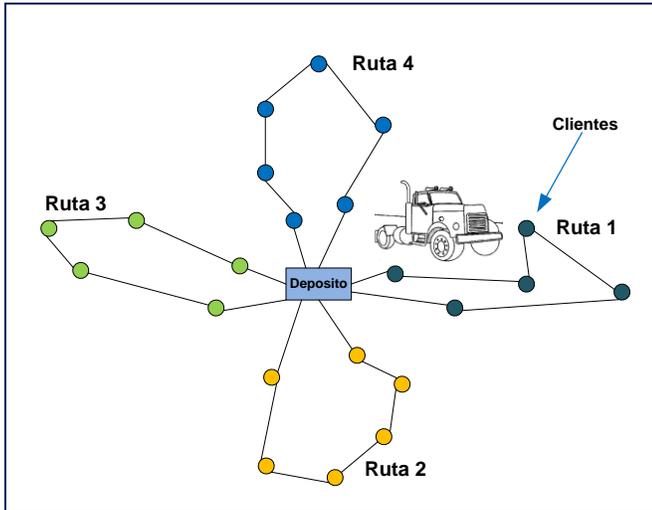


Fig 4. Clúster -VRP.

Por lo tanto es de vital importancia el diseño óptimo de las rutas de entrega que se originan y terminan en dicho depósito, sujeta a restricciones y cuyo objetivo es minimizar el costo de transporte. Dichas restricciones básicas son las siguientes:

- La ciudad  $i$  debe ser visitada solo una vez por un único vehículo.
- La ruta comienza y termina en el depósito.

Este algoritmo es aplicable a cualquier sistema en donde se requiera trasladar objetos de ciertos lugares a otros, se presenta una estructura que consiste en una flota de vehículos que deben visitar determinados clientes máximo una vez, partiendo y retornando a un lugar dado [8].

Dependiendo de la situación presentada en la vida cotidiana, el VRP puede tener muchas variaciones. En la literatura científica, Dantzig y Ramser fueron los primeros autores en 1959, cuando estudiaron la aplicación real en la distribución de gasolina para estaciones de carburante. Con esto podremos minimizar la distancia total recorrida y el tiempo de espera [9].

## TSP (Traveling Salesperson Problem)

El Problema del Agente viajero (TSP - “travel salesman problem”) es uno de los problemas más famosos e importantes

en todos los tipos de optimización combinatoria. Se ha estudiado a fondo durante más de 60 años, y hay una abundancia de literatura sobre el tema [10].

Este problema comenzó a tomar importancia en Europa, en el siglo XVII, donde era común proveer de productos a las ciudades, pueblos y aldeas en las zonas rurales alejadas de las ciudades principales, mediante un agente de ventas que salía de las oficinas centrales a su recorrido predeterminado y regresaba con su lista de pedidos, este recorrido es igual a la distancia total que resulta de la suma de las distancias entre localidades vecinas. Para las oficinas centrales era muy importante que su agente viajero visitara todos las ciudades de su tarea con distancia mínima. Hasta que en 1832, Voigt publica un manual para el agente viajero, llamado “El punto principal de la ruta del agente viajero consiste en visitar tantos lugares como sea posible sin tocar un lugar dos veces”, y finalmente este problema se denominó “el problema del agente viajero” por el matemático austriaco Karl Menger en un coloquio en Viena en 1930, [11].

$$x_{ij} \begin{cases} 1; & \text{si se va del punto } i \text{ al punto } j \\ 0; & \text{no se va del punto } i \text{ al punto } j \end{cases}$$

$$\min z = \sum_i \sum_j c_{ij} x_{ij}$$

$$s. a \quad \sum_{i=1}^{i=N} x_{ij} = 1 \quad (\text{para } j = 1, 2, \dots, N) \quad (1)$$

$$\sum_{j=1}^{j=N} x_{ij} = 1 \quad (\text{para } i = 1, 2, \dots, N) \quad (2)$$

$$u_i - u_j + N x_{ij} \leq N - 1 \\ (\text{para } i \neq j; i = 2, 3, \dots, N; j = 2, 3, \dots, N) \quad (3)$$

$$\text{ Toda } x_{ij} = 0 \text{ o } 1; \text{ toda } u_j \geq 0$$

Las restricciones (1) y (2) aplican para los modelos, estas indican que de cada ciudad (nodo  $n$ ) se salga exactamente hacia un solo destino y que cada destino tenga una sola llegada; sin embargo el TSP requiere una tercera restricción (3) que obligue a que toda la ruta sea recorrida mediante un solo camino y no dos o más caminos que cubran la ciudad de manera separada [12], [13], [14], [15] y [16].

## III. DESCRIPCIÓN DEL CASO

Se ha tomado como red de validación del artefacto para el algoritmo de Clarke y Wright, la red propuesta por Rojas & Silva [2], la cual posee 30 nodos, donde existe un camino central, con 6 nodos principales en el camino central, y nodos de acopio dentro del campo de cultivo. El ruteo se realiza diariamente con dos unidades de transporte. En la figura 5 se muestra 5 puntos de acopio. ¿Cómo podemos Rutear de manera óptima?

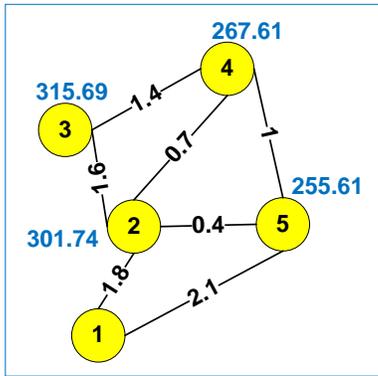


Fig 5. Clúster de 5 nodos

Analizando la red de la figura 5, podemos observar que presenta un ciclo Hamiltoniano, no obstante no se encuentra las rutas (1,3), (1,4) y (3,5). Posteriormente calculamos la matriz de Costos: distancia entre cada par de nodos, en las rutas no existentes el arco  $A_{ij}$  tendrá una distancia igual a 10 o M (valor grande).

**Tabla I.** Matriz de costos

|   |     |     |     |     |
|---|-----|-----|-----|-----|
|   | 2   | 3   | 4   | 5   |
| 1 | 1.8 | 10  | 10  | 2.1 |
| 2 |     | 1.6 | 0.7 | 0.4 |
|   |     | 3   | 1.4 | 10  |
|   |     |     | 4   | 1   |

Desarrollamos una asignación inicial para cada destino.

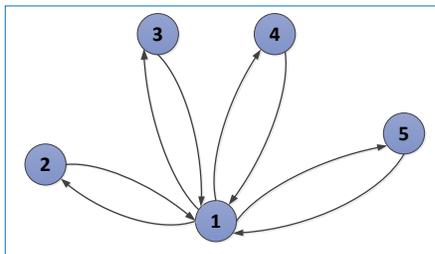


Fig 6. Asignación inicial.

Calculamos la matriz de ahorros: definida como la ganancia para cada par de nodos.

$$S_{ij} = C_{oi} + C_{oj} - C_{ij}$$

**Tabla II.** Matriz de ahorros.

|   |   |      |      |     |
|---|---|------|------|-----|
|   | 2 | 3    | 4    | 5   |
| T | 2 | 2    | 2    | 2   |
|   | 2 | 10.2 | 11.1 | 3.2 |
|   |   | 3    | 1.4  | 10  |
|   |   |      | 4    | 1   |

Iterando con el valor T:

La secuencia obtenida es: 1, 2, 3, 4, 5 y 1

En la figura 6 se muestra 30 puntos de acopio. ¿Cómo podemos Rutear de manera óptima?

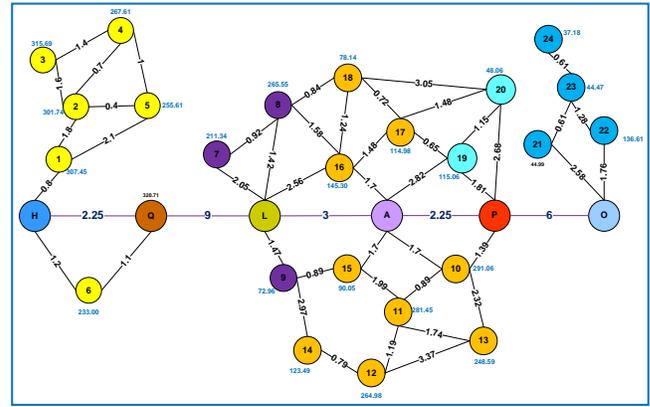


Fig 7. Clúster de 30 nodos

Analizando la red de la figura 7, podemos observar que no contamos con las todas las distancias. En el siguiente punto se describe los artificios desarrollados en el algoritmo de Clarke y Wright para poder rutear la red de la figura 7.

#### IV. DESCRIPCIÓN DEL MODELO PROPUESTO

Antes de comenzar a usar los artificios para las distancias faltantes en el uso de algoritmo de Clarke y Wright, se recomienda analizar si la red presenta un ciclo Hamiltoniano. Para realizar el ruteo por el algoritmo de Clarke y Wright, es necesario tener las distancias (arcos) entre los pares de nodos para armar la matriz de costos, además es necesario tener las distancias desde el nodo de origen hacia los demás puntos, con la finalidad de calcular la matriz de ahorros. No obstante, es ahí donde se vuelve compleja la resolución por este método. Además la red de la figura 7 presenta puentes, por lo cual se dificulta aún más la resolución de ésta. Entonces para facilitar el ruteo de la red de acopio, se procede a eliminar los puentes, para lo cual se identificara el par de nodos que contienen al arco puente, y se duplica cada uno de este par de nodos, con todos los arcos que entran y salen de él, tal como se muestra en la figura 8. También se normará que todo arco puente que contenga en uno de sus extremos a un nodo de grado 1, dicho nodo será incluido en el nodo predecesor, es decir se acumulará los parámetros de interés.

##### Eliminación de Nodos de Grado 1 (RP)

##### Inicializar

```

Sij = 0
Para i = 1 hasta N-1
  Para j = 2 hasta N
    Leer arco = dij
    Si existe arco D[i][j]
      Entonces Sij = Sij + 1
    De lo contrario D[i][j]= distancia
  artificio
  Fin
Fin
Para i = 1 hasta N-1
  Para j = 2 hasta N
    Si Sij < 2
      Entonces
        C N [i] = C N [i] + C N [j]
      Fin
    Fin
  Fin
Fin

```



- B) Variables con las que trabaja el algoritmo
- V: es el número de nodos a planificar su ruta.
  - Cmax[i]: es la cantidad máxima proyectada de un nodo [i].
  - Cmin[i]: es la cantidad mínima proyectada de nodo [i].
  - Moda[i]: es la cantidad promedio de un parámetro de interés que se encuentra en un nodo [i].
  - Es[i]: es la capacidad esperada de un parámetro que brinda el nodo [i].
  - Cesp[i]: es la capacidad esperada de materia prima que brinda el punto de acopio [i].
  - M[i,j]: es la matriz que contiene todas las distancias existentes entre los puntos de acopio.
  - Etiqueta[i]: es el valor de la distancia más corta desde un nodo x, hacia los demás nodos.
  - **M.Dijkstra[x,j]:** es el valor calculado por el algoritmo de Dijkstra, con la finalidad de tener **una matriz completa de distancias**.
  - M.Ahorros[i,j]: esta matriz representa la distancia ahorrada por un vehículo al recorrer desde el origen hacia los nodos i y j.
  - V.Ruta[cadena]: este arreglo almacena los nodos de un arco que se encuentran en la secuencia de ruta mejorada, su finalidad es evitar los sub-ciclos.
  - dt: es una variable usada como un temporal el cual ira acumulando la distancia que ejercerá la ruta obtenida.

C) Secuencia lógica del algoritmo:

- Paso 1: Se ingresa el número de nodos V, luego su proyección de capacidad máxima (Cmax[i]) y capacidad mínima (Cmin[i]), y el rendimiento frecuente (moda[i]). En el orden que se introdujo los datos de proyección de campo se calculara la capacidad esperada Es[i], de acuerdo a la fórmula de la distribución PERT.
- Paso 2: El algoritmo evalúa si la cantidad esperada Es[i] es menor a la cantidad mínima Cmin[i], cuya lógica es que la cantidad esperada Es[i] no debe ser menor que la cantidad mínima Cmin[i], esta decisión se representa en la cantidad esperada a reportar, Cesp[i]. Esto se observa en la figura 6.1.
- Paso 3: Luego se introduce las distancias conocidas entre los nodos, esta matriz estará denotada en M[i,j]. Esta inicial matriz de distancia se guardará en una matriz temporal M.ini[i,j] para realizar cálculos posteriores.
- Paso 4: Posteriormente se calculará la matriz completa de distancias entre los nodos, para lo cual será necesario aplicar el algoritmo de Dijkstra en los caminos no existentes en la matriz inicial M[i,j].
- Paso 5: Para calcular la matriz de Dijkstra, partimos desde un nodo el cual será etiquetado con una distancia de cero,

hacia los demás nodos a los cuales se les colocará una etiqueta inicial Etiqueta[i] de un valor grande (1000). Luego se analiza si el valor de la ruta desde el nodo “x” hacia los demás nodos existentes. Si esta distancia existe, se analiza si el valor de la distancia del nodo inicial más la distancia del arco entre los nodos adyacentes es menor a la etiqueta del nodo adyacente, si cumple esta restricción este valor se registra en la matriz de Dijkstra, M.Dijkstra[x,j]. Este paso 5 se realiza para todos los nodos (el valor de “x” abarca desde 1 hasta V).

- Paso 6: Otro paso importante es calcular la matriz de ahorros, la cual se calcula a partir de la matriz de Dijkstra, M.Dijkstra[x,j], se calcula al sumar la distancia del origen hasta el punto i y la distancia desde el origen hasta el nodo j menos la distancia directa entre los nodos “i” y “j”, además en la primera fila de esta matriz se inicia con grado 2 para todos los nodos. Estos datos se almacenan en la matriz M.Ahorros[i,j].
- Paso 7: Ahora se procede a calcular una adecuada secuencia de ruta que abarque a todos los nodos de la red de acopio. La lógica para este paso consiste en seleccionar el mayor valor de la matriz de ahorros (máx. M.Ahorros[i,j]), una vez obtenido este valor se verifica si el nodo i y el nodo j son de grado mayor que cero además no debe estar en la solución V.Ruta[cadena], si son mayores que cero y no se encuentran en la solución V.Ruta[cadena] se disminuye en un grado a ambos nodos y a este valor máximo se le multiplica por menos uno con la finalidad de que este número ya no vuelva a ser el número mayor y estos nodos se almacenaran en la ruta V.Ruta[cadena]. Luego se vuelve a verificar si alguno de estos nodos llegaron a tener grado cero, si llegaron a cero se elimina toda la fila y columna correspondiente ha dicho nodo. Este paso se repite hasta que se unan todos los nodos o hasta que la primera fila de la matriz de ahorro tenga como grado de los nodos el valor de cero, M.Ahorros[1,j] = 0. Seguidamente se imprime la secuencia de la ruta, V.Ruta[cadena], y su respectiva distancia, dt.

## V. CONCLUSIONES

1. Se muestra la viabilidad de un algoritmo para completar los datos faltantes en el cálculo del algoritmo de Clark y Wright.
2. El uso del algoritmo propuesto mejorará los índices de productividad, con respecto a la distancia del ruteo y al tiempo de ejecución de la asignación de la secuencia.
  - Ahorro de distancia en 36.21%
  - Ahorro en el tiempo de abastecimiento de 36.21%
3. El uso del algoritmo propuesto con el apoyo de la interface construida incrementará la productividad del ingeniero planificador de rutas, permitiéndole reducir el tiempo en

87.41%. Cabe mencionar que esto es factible hasta un tamaño de 40 nodos. Mayor que ello se recomienda hacerlo mediante una interface.

#### REFERENCIAS

- [1] Ballou, R (2004). Logística: administración de la cadena de suministro. Pearson education.
- [2] Rojas, J. & Silva, W (2014). Mejora en el abastecimiento y recolección de materia prima, aplicando heurísticas de Ruteo de Vehículos. 12<sup>th</sup> Latin American and Caribbean Conference for Engineering and Technology.
- [3] Clarke, G. & Wright, J. (1962). Scheduling of vehicles from A central Depot to a number of delivery points. Cooperative Wholesale Society. Operations Research 12 (1964), pp 568 – 581.
- [4] Flores, M. A. (2004). Optimización en la entrega de productos para una cadena de abastecimientos. Tesis De Ingeniería Industrial publicada, Universidad Nacional Mayor de San Marcos, Lima, Perú.
- [5] Parada, Z. (2009). El Rincón del Profesor. El Agente viajero. El modelo del agente viajero y Aplicaciones. Recuperado el 31 de Mayo del 2010. <http://laberintos.itam.mx/files/343.pdf>
- [6] Jordon, H. & Morris, J. (2007). Cyclic hamiltonian cycle systems of the complete graph minus a 1-factor. Discrete Mathematics 308 (2008) 2440 – 2449.
- [7] Barán, B. & Hermosilla, A. (2004). Comparación de un Sistema de Colonias de Hormigas y una Estrategia Evolutiva para el Problema del Ruteo de Vehículos con Ventanas de Tiempo en un Contexto Multiobjetivo. Universidad Nacional de Asunción, San Lorenzo, Paraguay.
- [8] Mester, D. & Braysy, O. & Dullaert, W. (2005). A multi-parametric evolution strategies algorithm for vehicle routing problems. ScienceDirect - Expert Systems with Applications, 32 (2007), pp 508 – 517
- [9] Olivera, A. (2004). Heurísticas para Problemas de Ruteo de Vehículos. Facultad de Ingeniería, Universidad de la República, Montevideo. : Uruguay.
- [10] Cordone, R. & Calvo, R. (2001). A Heuristic for the Vehicle Routing Problem with Time Windows. Journal of Heuristics vol N° 7. pp.107-129.
- [11] Jungnickel, D. (2008). Graphs, Networks and algorithms. Berlin: Springer.
- [12] Fleszar, K. & Osman, I.H. & Hindi, K.S. (2009). A variable neighborhood search algorithm for the open vehicle routing problem. European Journal of Operational Research. Vol N° 195 pp.803-809.
- [13] Dantzig, G. & Fulkerson, D. & Johnson, M. (1954). Solution of a large-scale traveling-salesman problem. The Rand Corporation, pp 393 - 410. California.
- [14] Dantzig, G. & Fulkerson, D. & Johnson, M. (1958). On a linear-programming, combinatorial Approach to the traveling-salesman Problem. The Rand Corporation, pp 58 - 66. California.
- [15] Dantzig, G. B. & Ramser, J. H. (1959). The Truck Dispatching Problem. The Rand Corporation. Management Science - Vol 6, pp 80 – 91. California.
- [16] Gillett, B. & Miller, L. (1971). A Heuristic Algorithm for the Vehicle-Dispatch Problem. Operations Research 22 (1974), pp 340 -349.