

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

ESTUDIO COMPARATIVO DE LA APLICACIÓN DE HEURÍSTICAS AL PROBLEMA DE RUTEO DE VEHÍCULOS

Tesis para optar el Título de **Ingeniero Industrial**, que presenta el bachiller:

José Josymar Cadillo Paredes

ASESOR: Walter A. Silva Sotillo

Lima, Agosto del 2011

RESUMEN

El problema de ruteo de vehículos ha sido estudiado ya hace bastante tiempo; sin embargo, no se le ha dado la importancia que merece, a pesar de que es una de las dificultades más importantes de la logística, especialmente en el área de transportes. El presente estudio es un trabajo aplicativo de un problema de este tipo, el cual se busca resolver mediante la aplicación de modelos heurísticos y un modelo computacional para ampliar la gama de posibilidades de resolución. La situación actual en donde ocurre el problema ya mencionado es una empresa dedicada a la venta de productos de consumo masivo, la cual cuenta con 42 puntos de venta alrededor de la ciudad de Lima Metropolitana. En primer lugar realizamos el mapeo de dichos puntos para luego resolver el problema con diversas metodologías y aplicaciones computacionales. Una vez concluidas todas las propuestas se llega a tomar una decisión acerca de cuál sería el mejor modelo a seguir para la resolución de este VRP (*Vehicule Routing Problem*) a partir de las distancias recorridas por el transporte hacia cada uno de los puntos de la red. En un futuro lo que se busca es elaborar un *Traveling Management System* que se ajuste a todas las necesidades de empresas con este tipo de problemas, con lo que de esta manera podrían reducir costos operativos y optimizar la cadena de abastecimiento.

ÍNDICE DE TABLAS

Tabla 3-1. Matriz de distancias entre puntos <i>cluster A</i> – Propuesta 1	26
Tabla 3-2. Costos del origen al punto <i>cluster A</i> – Propuesta 1	26
Tabla 3-3. Matriz de ahorros <i>cluster A</i> – Propuesta 1	27
Tabla 3-4. Matriz de distancias entre puntos <i>cluster C2</i> – Propuesta 2	32
Tabla 3-5. Costos del origen al punto <i>cluster C2</i> – Propuesta 2	32
Tabla 3-6. Matriz de ahorros <i>cluster C2</i> – Propuesta 2	33
Tabla 3-7. Resultados Propuesta: Cercanía de Puntos	46
Tabla 3-8. Resultados Propuesta: Método del Barrido	49
Tabla 4-1. Cuadro resumen del factor de linealidad	53
Tabla 4-2. Cuadro resumen del dimensionamiento real	57
Tabla 4-3. Descripción de Propuestas	58
Tabla 4-4. Cuadro resumen de distancias ideales	59
Tabla 4-5. Evaluación Cuantitativa de Propuestas	60
Tabla 4-6. Resultados finales de los modelos	61



INDICE DE GRÁFICOS

Gráfico 1.1. Una solución formada por 2 sub-tours.....	6
Gráfico 1.2. Dos rutas antes y después de ser unidas.....	13
Gráfico 3.1. Mapeo de clientes	24
Gráfico 3.2. Formación de <i>clusters</i> propuesta 1	25
Gráfico 3.3. Ejemplo 1 - método de ahorros	27
Gráfico 3.4. Ejemplo 1 - después de la 1era interacción.....	28
Gráfico 3.5. Ejemplo 1- después de la 2da interacción.....	28
Gráfico 3.6. Ejemplo 1 - después de la 3era interacción.....	29
Gráfico 3.7. Ejemplo 1 - después de la 4ta interacción.....	29
Gráfico 3.8. Ejemplo 1 - Ruta óptima	30
Gráfico 3.9. Ruta óptima – Propuesta 1	30
Gráfico 3.10. Formación de <i>clusters</i> – Propuesta 2	31
Gráfico 3.11. Ejemplo 2 – método de ahorros.....	33
Gráfico 3.12. Ejemplo 2 – después de la 1era interacción.....	34
Gráfico 3.13. Ejemplo 2 – después de la 2da interacción	34
Gráfico 3.14. Ejemplo 2 – después de la 3era interacción.....	35
Gráfico 3.15. Ejemplo 2 – después de la 4ta interacción	35
Gráfico 3.16. Ejemplo 2 – después de la 5ta interacción	36
Gráfico 3.17. Ejemplo 2- Ruta óptima	36
Gráfico 3.18. Ruta óptima – Propuesta 2.....	37
Gráfico 3.19. Gran ruta – Propuesta 3.....	39
Gráfico 3.20. Formación de <i>clusters</i> – Propuesta 3.....	39
Gráfico 3.21. Formación de <i>clusters</i> incluyendo gran ruta – Propuesta 3.....	40
Gráfico 3.22. Formulación <i>cluster A</i> - LINDO.....	42
Gráfico 3.23. Reporte <i>cluster A</i> - LINDO.....	43
Gráfico 3.24. Programación lineal <i>cluster B</i> -LINDO.....	44
Gráfico 3.25. Reporte <i>cluster B</i> - LINDO.....	45
Gráfico 3.26. Programación lineal <i>cluster 3</i> - LINDO	48
Gráfico 3.27. Reporte <i>cluster 3</i> - LINDO	48
Gráfico 3.28. Ventana de ingreso de datos.....	50
Gráfico 3.29. Resultado del método Hamiltoniano.....	51
Gráfico 4.1. Ruta real <i>cluster A</i> – Propuesta 1.....	54
Gráfico 4.2. Ruta real <i>cluster 5</i> – Propuesta 2	55
Gráfico 4.3. Ruta real <i>cluster 4</i> – Propuesta 3	55
Gráfico 4.4. Ruta real <i>cluster C</i> – Propuesta 4.....	56
Gráfico 4.5. Ruta real <i>cluster 3</i> – Propuesta 5	56

ÍNDICE GENERAL

CAPÍTULO 1. MARCO TEÓRICO.....	1
1.1. Programación Lineal.....	1
1.1.1. Simbología.....	2
1.1.2. Modelo de Asignación	3
1.2. Problemas de Ruteo de Vehículos (VRP)	4
1.2.1. El Problema de los m Agentes Viajero (m-TSP).....	6
1.2.2. El Problema con Flota Heterogénea (FSMVRP).....	9
1.2.3. El problema con Ventanas de Tiempo (VRPTW).....	11
1.3. Heurísticas Tradicionales para el VRP	12
1.3.1. Heurística de Clark Wright.....	12
1.3.2. Algoritmo de Pétalos.....	14
1.3.3. Cluster First – Route second	15
1.3.3.1. Algoritmo de Barrido o <i>Sweep</i>	15
1.3.3.2. Heurística de Fisher y Jaikumar	16
1.3.4. Route First – Cluster second	17
CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SITUACIÓN ACTUAL ...	19
2.1. Descripción de la situación actual	19
2.2. Descripción de la problemática.....	20
2.2.1. Supuestos y limitaciones	21
2.3. Desempeño del sistema actual.....	22
2.3.1. Indicadores	22
CAPÍTULO 3. DESARROLLO DE MODELOS	23
3.1. Mapeo de clientes	23
3.2. Propuestas de Ruteo.....	25
3.2.1. Propuesta 1: Por cercanía de Puntos	25
3.2.2. Propuesta 2: Método del Barrido	31
3.2.3. Propuesta 3: <i>Route First – Cluster Second</i> (Gran Ruta).....	38
3.2.4. Propuesta 4: Modelo Matemático por Cercanía de Puntos ...	41
3.2.5. Propuesta 5: Modelo Matemático por el método del Barrido .	46
3.2.6. Propuesta 6: Modelo Computacional.....	49
CAPÍTULO 4. EVALUACIÓN DE RESULTADOS	52
4.1 Factor de Linealidad	52
4.2 Dimensionamiento Real	54
4.3 Diferencias y similitudes entre modelos	57
4.4 Evaluación cuantitativa de las propuestas.....	59
4.5 Selección del mejor modelo.....	61
CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES	62
5.1 Conclusiones	62
5.2 Recomendaciones.....	63
REFERENCIAS BIBLIOGRÁFICAS.....	64

CAPÍTULO 1. MARCO TEÓRICO

1.1. Programación Lineal

En los últimos años se ha ido buscando la simplicidad de las cosas o simplemente la solución a problemas que parecen ser muy complicados, es por ello que dentro de la investigación de operaciones nos encontramos con una solución a muchos problemas tanto de la vida cotidiana como problemas que poseen un alto nivel de complejidad.

Podríamos decir que la programación lineal (PL) se ha convertido en una herramienta para resolver problemas de optimización; sin embargo, para aclarar este concepto se conoce a la programación lineal según Arreola (2003) como... “una técnica matemática que se utiliza para la solución de diferentes tipos de problemas, tanto teóricos como prácticos, en diversas áreas del conocimiento. El éxito en su aplicación a problemas reales, sofisticados y complejos es avalado por una gran cantidad de instituciones de bienes y servicios en muchos países”.

La programación lineal consiste básicamente en la construcción, solución y análisis del modelo lineal de un problema dado. Se entiende por modelo lineal al cual está integrado sólo por funciones lineales.

A continuación definiremos los parámetros que se cumplen para plantear una programación lineal.

1.1.1. Simbología

Para la formulación de problemas de programación lineal (PL) se procede de la siguiente manera, según Winston (2006), consta de tres partes:

1. La **función objetivo** o también llamada la función lineal la cual está compuesta por las variables de decisión (por ejemplo: X_1, X_2, \dots, X_n). Esta función se puede maximizar o minimizar.
2. Un conjunto de **restricciones**, conformadas por una igualdad lineal o desigualdad lineal, que limita los valores que podrían asumir las variables de decisión.
3. Las **restricciones de signo** o también conocido como el rango de existencia, que consiste en restringir todas las variables X_j que sean mayores o iguales a cero.

Matemáticamente tenemos la siguiente formulación:

Variables de Decisión:

$$x_j, \quad j = 1, 2, \dots, n$$

Función Objetivo:

$$\text{Maximizar o Minimizar } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Restricciones:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \{ \leq, =, \geq \} b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \{ \leq, =, \geq \} b_2$$

.....

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \{ \leq, =, \geq \} b_m$$

Rango de existencia

$$x_j \geq 0, \quad j = 1, 2, \dots, n$$

En esta formulación propuesta observamos que dado un conjunto de “m” desigualdades lineales o ecuaciones lineales, con “n” variables, se requiere hallar valores no negativo de éstas variables que satisfagan las restricciones y maximicen o minimicen la función objetivo, cumpliendo además con el rango de existencia.

1.1.2. Modelo de Asignación

Dentro de la programación lineal encontramos muchos casos especiales para cada tipo problema que se nos pueda presentar, y una de ellos es el problema de asignación el cual analizaremos a continuación, debido a que es el que utilizaremos más adelante para formular los modelos tanto heurísticos como metaheurísticos.

Para el presente caso de estudio trataremos el problema de asignación como un modelo de distribución que específicamente trata de asignar un número de orígenes (individuo, vehículos, etc.) a un mismo número de destinos (clientes, tareas, etc.) para que de esta manera se optimice el coste o tiempo, es por ello que la función objetivo sería de minimización.

A continuación analizaremos la formulación de un problema de asignación presentada por Ríos (1996):

Variables de decisión:

$$x_{ij} = \begin{cases} 1 & \text{si el individuo } i \text{ se asigna a la tarea } j \\ 0 & \text{en otro caso.} \end{cases}$$

Función Objetivo:

$$\text{Min } Z = \sum_{i=1}^m \sum_{j=1}^m c_{ij}x_{ij}$$

Restricciones:

$$\sum_{j=1}^m x_{ij} = 1 \quad i = 1, \dots, m$$

$$\sum_{i=1}^m x_{ij} = 1 \quad i = 1, \dots, m$$

Rango de existencia:

$$x_{ij} = 0, 1 \text{ para todo } i, j$$

La formulación descrita arriba se entiende como un caso dado en el que m individuos que hay que asignar a m tareas, siendo c_{ij} , el costo de asignar al individuo i la tarea j , se trata de determinar una asignación de tal manera que costo total sea mínimo.

1.2. Problemas de Ruteo de Vehículos (VRP)

En general un problema de ruteo de vehículos consiste, en que dado un conjunto de clientes y depósitos o almacenes distribuidos geográficamente se encuentran dispersos, para lo cual se requiere una flota de vehículos con los que se llegará a estos clientes desde los depósitos tomando en cuenta las rutas que minimicen los costos de partir de un depósito y llegar al mismo habiendo visitados a los clientes.

Aunque el problema de ruteo de vehículos pueda tener muchas variaciones, podemos reducirlas a unos cuantos tipos básicos como observaremos a continuación:

1.1.1. El Problema del Agente Viajero (TSP)

El problema del Agente Viajero es uno de los problemas mejor estudiados en la optimización combinatoria en la clase de complejidad *NP - Hard* (o *NP - "Complejo"*) y además ha sido intensamente utilizado como un banco de pruebas para experimentar nuevas ideas algorítmicas.

También conocido como TSP por sus siglas en inglés (*Travelling Salesman Problem*) tiene sus inicios en los años 1930 en el estudio de la Universidad de Princeton llamado "*48 States Problem*" presentado por Hassler Whitney¹, quien planteaba encontrar la ruta (tour) más corta para el agente viajero comenzando desde una ciudad (estado) dada, visitando cada una de las ciudades (estados) de un grupo específico. Luego regresando al punto original de partida.

A partir del estudio realizado por Hassler podemos definir hoy en día al TSP como la visita de un solo vehículo a todos los clientes en una sola ruta y a un costo mínimo, y de esta manera encontrar una aplicabilidad a nuestro contexto actual. Entonces el problema puede formularse como:

$$\text{mín } Z = \sum_{(i,j) \in E} c_{ij} x_{ij}$$

s.a.

$$\sum_{j \in \Delta^+(i)} x_{ij} = 1 \quad \forall i \in V$$

$$\sum_{i \in \Delta^-(j)} x_{ij} = 1 \quad \forall j \in V$$

$$\sum_{i \in S, j \in \Delta^-(j) \setminus S} x_{ij} \geq 1 \quad \forall S \subset V$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E$$

¹ Matemático Estadounidense (1907 – 1989), quien en base a sus estudios planteo la teoría de grafos.

Formulación propuesta por Dantzig según Applegate (2006), quien presenta en la función objetivo el hecho de minimizar el costo total (C_{ij}) de la solución, el cual es representado por la suma de todos los costos de los arcos utilizados (x_{ij}). Además las primeras dos restricciones aseguran que cada nodo tenga tanto una entrada como una salida. Por último, la formulación presenta una restricción más la cual se asegura de que no ocurran sub-tours; es decir, que no se formen grupos por separado como observamos en el gráfico 1.1.

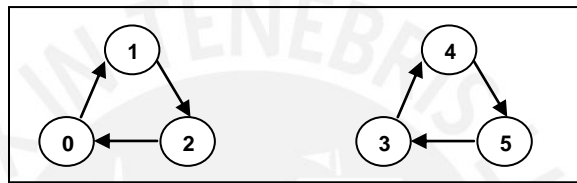


Gráfico 1.1. Una solución formada por 2 sub-tours

1.2.1. El Problema de los m Agentes Viajero (m-TSP)

A diferencia del problema del Agente Viajero, en este caso se tiene un depósito (almacén) y “m” números de vehículos, en donde el principal objetivo es elaborar exactamente “m” rutas para cada vehículo, de modo que cada cliente sea visitado una vez por uno de los vehículos. Cada ruta debe comenzar y finalizar en el depósito y puede contener a lo mucho p clientes. A continuación observaremos la formulación propuesta por Miller et al. (1960):

$$\text{Min} \quad Z = \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (1.1)$$

s. a.

$$\sum_{j \in \Delta^+(0)} x_{0j} = m \quad (1.2)$$

$$\sum_{j \in \Delta^+(i)} x_{ij} = 1 \quad \forall i \in V \setminus \{0\} \quad (1.3)$$

$$\sum_{i \in \Delta^-(j)} x_{ij} = 1 \quad \forall j \in V \setminus \{0\} \quad (1.4)$$

$$u_i - u_j + px_{ij} \leq p - 1 \quad \forall (i, j) \in E, i \neq 0, j \neq 0 \quad (1.5)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E \quad (1.6)$$

$$u_j \geq 0 \quad \forall i \in V \setminus \{0\} \quad (1.7)$$

El modelo es similar al segundo modelo presentado para el TSP. La restricción 1.2 indica que exactamente m vehículos salen del depósito y las 1.3 y 1.4 aseguran que cada cliente es un nodo intermedio en exactamente una ruta. Finalmente, con 1.5 se eliminan los sub-tours y se impone que en cada ruta no haya más de p clientes.

En el caso que $p = n$ (es decir, cuando la cantidad de clientes por ruta no está acotada) el m-TSP puede formularse como un TSP con m copias del depósito tales que la distancia entre ellas es infinita. Las soluciones a ese TSP no utilizaran arcos que conectan dos copias del depósito y por lo tanto, pueden ser interpretadas como soluciones del m-TSP.

1.1.2. El Problema con Capacidades (VRP o CVRP)

El VRP es una extensión del m-TSP en la cual cada cliente $i \in V \setminus \{0\}$ tiene asociada una demanda d_i y cada vehículo tiene una capacidad C (la flota es homogénea). En este problema la cantidad de rutas no es fijada de antemano como en el TSP y en el m-TSP.

Para un conjunto de clientes S , $d(S) = \sum_{i \in S} d_i$ es su demanda total y $r(S)$ indica la mínima cantidad de vehículos necesarios servirlos a todos. En la formulación conocida con el nombre de *flujo de vehículos de dos índices*, se utilizan las variables binarias x_{ij} para determinar si el arco (i, j) se utiliza o no en la solución. El problema se formula de la siguiente manera:

$$\text{Min } Z = \sum_{(i,j) \in E} c_{ij} x_{ij} \quad (1.8)$$

s. a.

$$\sum_{j \in \Delta^+(0)} x_{0j} = m \quad (1.9)$$

$$\sum_{i \in \Delta^-(0)} x_{i0} = m \quad (1.10)$$

$$\sum_{j \in \Delta^+(i)} x_{ij} = 1 \quad \forall i \in V \setminus \{0\} \quad (1.11)$$

$$\sum_{i \in \Delta^-(j)} x_{ij} = 1 \quad \forall j \in V \setminus \{0\} \quad (1.12)$$

$$\sum_{i \in S, j \in \Delta^+(i) \setminus S} x_{ij} \geq r(S) \quad \forall S \subset V \setminus \{0\} \quad (1.13)$$

$$m \geq 1 \quad (1.14)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E \quad (1.15)$$

La función objetivo 1.8 es el costo total de la solución. Las restricciones 1.9 y 1.10 indican que m es la cantidad de vehículos utilizados en la solución y que todos los vehículos que parten del depósito deben regresar. Las restricciones 1.11 y 1.12 aseguran que todo cliente es un nodo intermedio de alguna ruta.

Finalmente, la restricción 1.13 actúa como restricción de eliminación de sub-tours y a la vez impone que la demanda total de los clientes visitados por un vehículo no puede superar la capacidad C .

Determinar el valor de $r(S)$ requiere la resolución del siguiente problema:

$$r(S) = \text{Min} \sum_{k \in K} y^k \quad (1.16)$$

s. a.

$$\sum_{i \in S} d_i x_{ik} \leq C y_k \quad \forall k \in K \quad (1.17)$$

$$\sum_{k \in K} x_{ik} = 1 \quad \forall i \in S \quad (1.18)$$

$$x_{ik} \in \{0,1\} \quad \forall i \in S, k \in K \quad (1.19)$$

$$y_k \in \{0,1\} \quad \forall k \in K \quad (1.20)$$

Donde K es un conjunto con suficientes vehículos para satisfacer la demanda (por ejemplo, n). Este problema, según Martello (1990) es conocido como *Bin Packing Problem* (BPP). Una cota inferior para la cantidad de vehículos está dada por el valor óptimo de la relajación lineal del BPP, que es $\left\lceil \frac{d(S)}{C} \right\rceil$.

La formulación es válida incluso cuando se sustituye $r(S)$ por la cota inferior $\left\lceil \frac{d(S)}{C} \right\rceil$.

1.2.2. El Problema con Flota Heterogénea (FSMVRP)

En los problemas con flota heterogénea los costos y capacidades de los vehículos varían, existiendo un conjunto $T = \{1, \dots, |T|\}$ de tipos de vehículo. La capacidad de los vehículos $k \in T$ es q^k y su costo fijo (si lo tuvieran) es f^k . Los costos y tiempos de viaje para cada tipo de vehículo son c_{ij}^k y t_{ij}^k respectivamente. Se asume que los índices de los vehículos están ordenados en forma creciente por capacidad (es decir, $q^{k_1} \leq q^{k_2}$ para $k_1, k_2 \in T, k_1 < k_2$).

En la siguiente formulación de *flujo de vehículos de tres índices*, planteado por Golden et. al. (1984), se agrega un índice para discriminar entre los tipos de vehículos.

$$\text{mín } Z = \sum_{k \in T} f^k \sum_{j \in \Delta^+(0)} x_q^k + \sum_{k \in T} \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k \quad (1.21)$$

s. a.

$$\sum_{k \in T} \sum_{i \in \Delta^+(j)} x_{ij}^k = 1 \quad \forall j \in V \setminus \{0\} \quad (1.22)$$

$$\sum_{j \in \Delta^+(i)} x_{ij}^k - \sum_{j \in \Delta^-(i)} x_{ij}^k = 0 \quad \forall i \in V, \forall k \in T \quad (1.23)$$

$$r_0 = 0 \quad (1.24)$$

$$r_i - r_j \geq (d_j + q^{[T]}) \sum_{k \in T} x_{ij}^k - q^{[T]} \quad \forall i \in V \setminus \{0\}, \forall j \in \Delta^+(i) \quad (1.25)$$

$$r_i \leq \sum_{k \in T} \sum_{i \in \Delta^-(j)} q_k x_{ij}^k \quad \forall j \in V \setminus \{0\} \quad (1.26)$$

$$x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in E, \forall k \in T \quad (1.27)$$

$$r_j \geq 0 \quad \forall j \in V \quad (1.28)$$

Las variables binarias x_{ij}^k indican si el arco (i,j) es utilizado por el vehículo k y las variables r_i positivas indican la carga acumulada en la ruta correspondiente hasta el nodo i (inclusive).

La función objetivo (1.21) mide el costo total de la solución incluyendo costos fijos y variables. Las restricciones 1.22 establecen que todo cliente debe ser visitado por algún vehículo. En 1.23 se indica que si un vehículo de tipo k visita al nodo i , entonces un vehículo del mismo tipo debe abandonarlo.

Las restricciones 1.24 y 1.25 fijan los valores de las variables r_i y actúan como restricciones de eliminación de sub-tours, mientras que la capacidad de los vehículos se impone en 1.26. En esta formulación se asume que la cantidad de vehículos de cada tipo es ilimitada

1.2.3. El problema con Ventanas de Tiempo (VRPTW)

El problema de Ruteo de Vehículos con Ventanas de Tiempo también conocido como VRPTW (por sus siglas en inglés *The Vehicle Routing Problems with Time Windows*), es una generalización del VRP sumando la complejidad de los tiempos de entrega admisibles, o ventanas de tiempo. En estos problemas el servicio de un cliente, teniendo en cuenta la entrega de los bienes o servicios, puede comenzar dentro del tiempo definido por los tiempos más tempranos y más tardes cuando el cliente va a permitir que empiece el servicio.

Para este caso asumimos que los límites de la ventana de tiempos son $[e_i, l_i]$, utilizando los nodos 0 y $n + 1$ para representar al depósito y el conjunto K para representar a los vehículos (no a los tipos de vehículos como en la sección anterior), entonces según Cordeau et. al. (1999), el problema se formula para una flota de vehículos posiblemente heterogénea, de la siguiente manera:

$$\text{Min} \sum_{k \in K} \sum_{(i,j) \in E} c_{ij}^k x_{ij}^k \quad (1.29)$$

s. a.

$$\sum_{k \in K} \sum_{j \in \Delta^-(i)} x_{ij}^k = 1 \quad \forall i \in V \setminus \{0, n + 1\} \quad (1.30)$$

$$\sum_{j \in \Delta^+(0)} x_{0j}^k = 1 \quad \forall k \in K \quad (1.31)$$

$$\sum_{j \in \Delta^+(i)} x_{ij}^k - \sum_{j \in \Delta^-(i)} x_{ij}^k = 0 \quad \forall k \in K, i \in V \setminus \{0, n+1\} \quad (1.32)$$

$$\sum_{i \in V \setminus \{0, n+1\}} d_i \sum_{j \in \Delta^+(i)} x_{ji}^k \leq q^k \quad \forall k \in K \quad (1.33)$$

$$y_j^k - y_i^k \geq s_i + t_{ij}^k - M(1 - x_{ji}^k) \quad \forall i, j \in V \setminus \{0, n+1\}, k \in K \quad (1.34)$$

$$e_i \leq y_i^k \leq l_i \quad \forall i \in V \setminus \{0, n+1\}, k \in K \quad (1.35)$$

$$x_{ji}^k \in \{0, 1\} \quad \forall (i, j) \in E, k \in K \quad (1.36)$$

$$y_i^k \geq 0 \quad \forall i \in V \setminus \{0, n+1\}, k \in K \quad (1.37)$$

Las variables x_{ij}^k indican si el arco (i, j) es recorrido por el vehículo k . Las variables y_i^k indican la hora de arribo al cliente i cuando es visitado por el vehículo k (si el cliente no es visitado por dicho vehículo el valor de la variable no tiene significado). La función objetivo (1.29) es el costo total de las rutas. La restricción 1.30 indica que todos los clientes deben ser visitados. Las restricciones 1.31 y 1.32 determinan que cada vehículo $k \in K$ recorre un camino de 0 a $n+1$. La capacidad de cada vehículo es impuesta en 1.33. Siendo M una constante lo suficientemente grande, la restricción 1.34 asegura que si un vehículo k viaja de i a j , no puede llegar a j antes que $y_i + s_i + t_{ij}^k$, y actúan además como restricciones de eliminación de sub-tours. Finalmente, los límites de las ventanas de tiempo son impuestos en 1.35.

1.3. Heurísticas Tradicionales para el VRP

1.3.1. Heurística de Clark Wright

Es la heurística clásica más significativa para el VRP. Esta heurística es un procedimiento simple que realiza una exploración limitada del espacio de búsqueda y da una solución de calidad más o menos aceptable en tiempo de

cálculo moderado. Las soluciones luego pueden ser mejoradas con los algoritmos de mejora del TSP (como 2-opt).

Si en una solución dos rutas diferentes $(1, \dots, i, 1)$ y $(1, j, \dots, 1)$ pueden ser combinadas formando una nueva ruta $(1, \dots, i, j, \dots, 1)$ como se muestra en el gráfico 1.2, el ahorro (en distancia) obtenido por dicha unión es:

$$s_{ij} = c_{i1} + c_{1j} - c_{ij}$$

Pues en la nueva solución los arcos $(i, 1)$ y $(1, j)$ no serían utilizados y se agregaría el arco (i, j) . Se parte de una solución inicial en la cual todos los clientes son servidos por un solo vehículo.

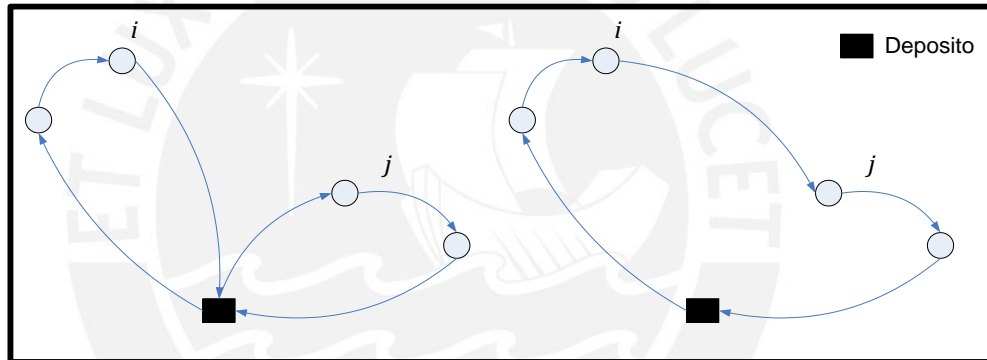


Gráfico 1.2. Dos rutas antes y después de ser unidas
Fuente. Clarke y Wright (1964)

- **Paso 1** (inicialización) para cada cliente i construir la ruta $(1, i, 1)$.
- **Paso 2** (cálculo de ahorros) calcular s_{ij} para cada par de clientes i y j .
- **Paso 3** (mejor unión). Sea $s_{i^*j^*} = \max. s_{ij}$, donde el máximo se toma entre los ahorros que no han sido considerados aún. Sean r_{i^*} y r_{j^*} las rutas que contienen a los clientes i^* y j^* respectivamente. Si i^* es el último cliente de r_{i^*} y j^* es el primer cliente de r_{j^*} y la combinación de r_{i^*} y r_{j^*} es factible, combinarlas.

- **Paso 4** Eliminar $S_{i^*j^*}$ de futuras consideraciones. Si quedan ahorros por examinar ir a 3, si no terminar.

1.3.2. Algoritmo de Pétalos

Supongamos que se dispone de un conjunto de rutas R , de modo que cada ruta $r \in R$ es factible, pero cada cliente es visitado por varias de las rutas. El problema de seleccionar un subconjunto de R de costo mínimo que visite exactamente una vez a cada cliente puede formularse como un *Set Partitioning Problem* (SPP):

$$\text{Min} \sum_{k \in R} a_{ik} x_k = 1 \quad (1.38)$$

s. a.

$$\sum_{k \in R} a_{ik} x_k = 1 \quad \forall i \in V \setminus \{0\} \quad (1.39)$$

$$x_{ik} \in \{0, 1\} \quad \forall k \in S \quad (1.40)$$

Donde a_{ik} vale 1 si el cliente i es visitado por la ruta r_k y 0 si no y donde c_k es el costo de la ruta r_k . La variable x_k indica si la ruta r_k es seleccionada o no en la solución final. Esta formulación se debe a Balinski y Quandt (1964). En el caso extremo de que R contenga todas las posibles rutas factibles, solucionar el SPP es equivalente a resolver el problema en forma exacta. Como la cantidad de rutas factibles es, en el caso general, exponencial en la cantidad de clientes, se suele generar solamente un subconjunto de formado por “buenas” rutas.

Cada columna del SPP representa una ruta de R . Cuando en toda columna los ceros aparecen de forma consecutiva, el problema verifica la propiedad de Columnas Circulares y el SPP correspondiente puede ser resuelto en tiempo polinomial. Trasladada al problema la propiedad establece que, para

determinado ordenamiento de los clientes del problema, el conjunto de clientes visitado por cada ruta forma un intervalo (que en algunos casos tiene forma de pétalo). Diversas técnicas han sido propuestas para generar “buenos” conjuntos de rutas que verifiquen la propiedad llamadas 1-pétalos y 2-pétalos.

1.3.3. Cluster First – Route second

Los métodos asignar primero y rutear después (*cluster first - route second*) proceden en dos fases. Primero se busca generar grupos de clientes, también llamados *clusters*, que estarían en una misma ruta en la solución final. Luego, para cada *cluster* se crea una ruta que visite a todos sus clientes. Las restricciones de capacidad son consideradas en la primera etapa, asegurando que la demanda total de cada *cluster* no supere la capacidad del vehículo. Por lo tanto, construir las rutas para cada *cluster* es un TSP que, dependiendo de la cantidad de clientes en el *cluster*, se puede resolver la forma exacta o aproximada.

1.3.3.1. Algoritmo de Barrido o Sweep

En la heurística de barrido planteada por Gillet y Miller (1974), los *clusters* se forman girando una semirrecta con origen en el depósito e incorporando los clientes “barridos” por dicha semirrecta hasta que se viole la restricción de capacidad. Cada *cluster* es luego ruteado resolviendo un TSP de forma exacta o aproximada.

Este algoritmo puede aplicarse en problemas planos, es decir, en los que cada nodo se corresponde con un punto en el plano y las distancias entre ellos se definen como la distancia euclídea. Se supone que cada cliente i está dado por sus coordenadas polares (ρ_i, θ_i) en un sistema que tiene al depósito como origen.

Paso 1 (inicialización).

Escoger un vehículo no usado k

Paso 2 (selección).

Ordenar los clientes según θ_i de manera creciente. Si dos clientes tienen igual valor de θ_i , colocar primero el de menor valor p , y asignar los clientes a la ruta hasta que se llene la capacidad del camión o no se extienda la longitud de la ruta. Si todos los clientes pertenecen a algún *cluster*, y no se pueden agregar más nodos a la ruta por las restricciones ya dichas, ir a 3.

Paso 3 (optimización).

Optimizar las rutas utilizando un algoritmo que resuelva el TSP para cada *cluster*.

1.3.3.2. Heurística de Fisher y Jaikumar

Fisher y Jaikumar (1981) proponen generar los *clusters* resolviendo un Problema de Asignación Generalizada (GAP) sobre los clientes. Primero se fijan K clientes semilla s_k con $k = 1, \dots, K$ sobre la base de los cuales se construirían los *clusters*.

En una segunda fase, se decide qué clientes asignar a cada uno de los *clusters* de modo de no violar la capacidad del vehículo, resolviendo un GAP que se define a continuación:

$$\min \sum_{k=1}^K \sum d_{ik} x_{ik} \quad (1.41)$$

s.a.

$$\sum_{k=1}^K x_{ik} = 1 \quad \forall i \in V \setminus \{0\} \quad (1.42)$$

$$\sum_{i \in V \setminus \{0\}}^K q_{ik} x_{ik} \leq Q \quad \forall k = 1, \dots, K \quad (1.43)$$

$$X_{iK} \in \{0,1\} \quad \forall i \in V \setminus \{0\}, \forall k = 1, \dots, K \quad (1.44)$$

Las variables x_{ik} indican si el cliente i es asignado al *cluster* k . El objetivo es minimizar el costo total de la asignación, como se indica en 1.41. La restricción 1.42 indica que cada cliente es asignado a exactamente un *cluster*. En la desigualdad 1.43 se impone que la demanda de los clientes en un mismo *cluster* no puede superar la capacidad del vehículo. En conclusión, para la formulación de este método tenemos los siguientes pasos:

- **Paso 1** (inicialización).

Formar K *clusters* e inicializar cada uno con un cliente s_k ($k = 1, \dots, K$).

- **Paso 2** (asignación).

Resolver el Problema de Asignación Generalizada (GAP) para decidir a qué *cluster* es asignado cada cliente.

- **Paso 3** (ruteo).

Para cada *cluster*, resolver un TSP con sus clientes

1.3.4. Route First – Cluster second

Algunas de las heurísticas clásicas para el problema de ruteo de vehículos están basadas en el principio de agrupar primero y rutear después, como es el caso del algoritmo de Barrido visto anteriormente. Por otro lado, el principio de rutear primero y agrupar después ha sido pocas veces usado. Beasley (1983) sugirió algunas heurísticas basadas en este principio, pero sin resultados numéricos.

En estos métodos también se procede en dos fases. Primero se calcula una ruta que visita a todos los clientes resolviendo un TSP. En general esta ruta no respeta las restricciones del problema y se divide en varias rutas, cada una de las cuales sí es factible.

Dada $r = (0, v_1, \dots, v_n, 0)$, la solución del TSP obtenida en la primera fase, se determina la mejor partición de r que respete la capacidad del vehículo. Este problema se puede formular como el de hallar un camino mínimo en un grafo dirigido y acíclico. Para ello, se construye un grafo $G = (X, V, W)$ donde $X = \{0, v_1, \dots, v_n\}$.

Los arcos del G conectan todo par de clientes v_i y v_j con $i < j$ y tales que la demanda total de los clientes v_{i+1}, \dots, v_j no supera la capacidad del vehículo:

$V = \{(v_i, v_j) \mid i < j, \sum_{k=i+1}^j d_{v_k} \leq Q\}$. Cada arco (v_i, v_j) se pondera con el costo de la ruta $(0, v_{i+1}, \dots, v_j, 0)$, es decir

$$w(v_i, v_j) = c_{0, v_{i+1}} + c_{v_j, 0} + \sum_{k=i+1}^{j-1} c_{v_k, v_{k+1}} \quad (1.45)$$

Un arco (v_i, v_j) representa la ruta $(0, v_{i+1}, \dots, v_j, 0)$. Cada camino de 0 a v_n en G representa una posible partición de la ruta r en rutas que respetan las restricciones de demanda. Por lo tanto, el camino de costo mínimo entre 0 y v_n representa la partición de costo mínimo de la ruta original en rutas que respetan la restricción de capacidad. Como el grafo es acíclico (sólo hay arcos (v_i, v_j) con $i < j$), puede utilizarse el Algoritmo de Dijkstra para hallar dicho camino.

CAPÍTULO 2. DESCRIPCIÓN Y ANÁLISIS DE LA SITUACIÓN ACTUAL

2.1. Descripción de la situación actual

La presente investigación se realizará en base a una empresa que pertenece al rubro de *retail* la cual tiene dentro sus actividades principales la venta de productos de consumo masivo. En este contexto se analizará como dicha empresa distribuye sus productos desde su almacén principal hacia sus principales puntos de venta, todos ubicados en la ciudad de Lima Metropolitana.

Se tomara en cuenta sólo 42 puntos de distribución de la empresa, y un almacén principal de donde se proveerá todos los productos de consumo masivo. En el Anexo 1, se muestran todas las direcciones de los puntos mencionados así también como del almacén principal. Por otro lado, es importante para el desarrollo de la investigación, conocer el sistema de gestión de almacenes o *Warehouse Management System* (WMS), ya que este sistema se representa a través de un ciclo de almacenamiento el cual incluye a la gestión de despacho, que es en donde se concentra una parte del estudio.

Según el WMS de la empresa en la operación de despacho se utiliza Radiofrecuencia, se cargan las entregas relacionadas a las paletas según la tienda que se va a despachar y el sistema genera la ruta. El ordenamiento en el despacho es respetando el sistema FIFO, es decir, lo primero que entra es lo primera que sale. Para dicha carga de la unidad se colocan de manera organizada los productos.

Una vez terminada la carga de la mercadería, pasamos al transporte de la misma hacia los diferentes puntos ya mencionados, el medio de transporte que utilizan es un vehículo o transporte articulado, es decir, una unidad de tracción que está separada de la unidad de carga. El transporte se realiza agrupando distritos de acuerdo a la cercanía entre las tiendas. Y finalmente, la descarga de la mercadería la cual se asume que se da en su totalidad, se entiende, que si el transporte llega a la tienda descarga toda la mercadería correspondiente a dicha tienda.

2.2. Descripción de la problemática

La problemática actual en la empresa que analizaremos es, principalmente, que no existe un estudio del ruteo de los vehículos que realizan el transporte de la mercadería desde el almacén principal hacia las tiendas o puntos de distribución. Por lo tanto, la ruta a seguir es según el criterio del conductor o de un analista, quien toma en cuenta la proximidad entre los distritos en donde están ubicadas las tiendas que le han sido designados por las solicitudes de pedido. Entonces el problema en si sería desde el despacho ya que en el momento de cargar las unidades de transporte se decide hacia donde se dirige el vehículo; es decir, las tiendas en las que dejará la mercadería.

El hecho de no elegir de la manera más óptima la ruta a seguir implica que se incurra en costos excesivos de distribución, debido al combustible que consume el vehículo y a que el tiempo de entrega es muy elevado, entonces retrasa la apertura de las tiendas. Además, existe la posibilidad de no entregar toda la mercadería que se requiere en un día, esto dejaría sin stock a las tiendas, lo cual perjudicaría a las ventas de los productos ausentes.

2.2.1. Supuestos y limitaciones

A continuación se mostrarán los supuestos que debemos tomar en cuenta para la realización del estudio:

- Los pedidos de acuerdo a los requerimientos de cada punto de venta o distribución serán entregados completos. Se asume que no existen mermas o pérdidas en el traslado de las mismas.
- En el momento del transporte de la mercadería, el vehículo estará sujeto al tráfico que pueda existir en la ruta haya sido asignado. Sin embargo, no lo estará a posibles cierres de calles temporales que está siendo un común problema actualmente.
- La empresa cuenta con 8 vehículos para el transporte de la mercadería. Además los vehículos pueden o no llenar el total de sus capacidades de transporte.
- Las ventanas de tiempo dependerán de los clientes; es decir, del horario de atención que tengan cada uno de los puntos de distribución de donde se está dejando la mercadería.
- Trabajaremos un problema de flota homogénea, es decir, los vehículos utilizados poseen los mismos costos y la misma capacidad como mencionamos anteriormente.
- Tomaremos en cuenta que todas las rutas o calles que pertenecen a la misma son hábiles para el tipo de transporte que utilizamos.

2.3. Desempeño del sistema actual

El sistema debe estar medido bajo ciertos indicadores para obtener la eficiencia del sistema, así también como la utilización de los recursos, como combustible y capacidad por ejemplo.

2.3.1. Indicadores

A continuación observaremos los indicadores a tomar en cuenta para nuestra investigación, con el propósito de que mediante estos se pueda elegir más adelante la mejor solución luego de la comparación de métodos:

- **Distancia recorrida:**
Distancia recorrida por cada *cluster* entre la distancia recorrida total por cada transporte.
- **Coeficiente de cumplimiento:**
Se mide como el porcentaje de veces que el transportista entregó la mercadería a tiempo en fecha, hora, cantidad, conservación de la misma, para esto medimos Eficacia de Traslado.
- **Factor de uso:**
Es el tiempo que la unidad de transporte se encuentra en movimiento dividido entre el tiempo total (tiempo en movimiento + tiempo parada).
- **Frecuencia de entrega:**
Indica la cantidad de veces que se puede realizar entregas en un periodo de tiempo dado, para esto medimos los Pedidos a Tiempo.

CAPÍTULO 3. DESARROLLO DE MODELOS

En el presente capítulo el objetivo es plantear las propuestas para el problema de ruteo de vehículos que se da en esta empresa de acuerdo a los puntos tocados en el marco teórico, además presentar el desarrollo de estas mediante las metodologías explicadas en dicho capítulo; sin embargo, se presentará una propuesta obtenida mediante experiencias de los propios transportistas. Por lo tanto, antes de presentar las propuestas debemos localizar los puntos sobre los cuales se trabajará, así como uno de los principales factores que afectará a nuestras sugerencias como lo es el tráfico.

3.1. Mapeo de clientes

En la actualidad existen muchas compañías que ofrecen servicios de mapeo de clientes para poder localizar fuerzas de ventas o también para el tema que estamos tratando como es el ruteo de vehículos, muchas de estas empresas utilizan *software* desarrollados por ellos mismos o terceros.

Para realizar el mapeo de los clientes utilizamos la herramienta de *Google Maps* en donde ubicamos las direcciones de 42 clientes o centros de distribución (*Anexo 1*), los cuales se ubican en Lima Metropolitana, abarcando la mayoría de los distritos que esta representa como por ejemplo Comas, Independencia, Los Olivos, San Miguel, El Callao, San Isidro, etc.

Como se muestra a continuación en el gráfico 3-1 tenemos tres tipos de marcadores uno azul el cual representa al almacén principal, amarillo un tipo de cliente 1 y rojo un tipo de cliente 2.

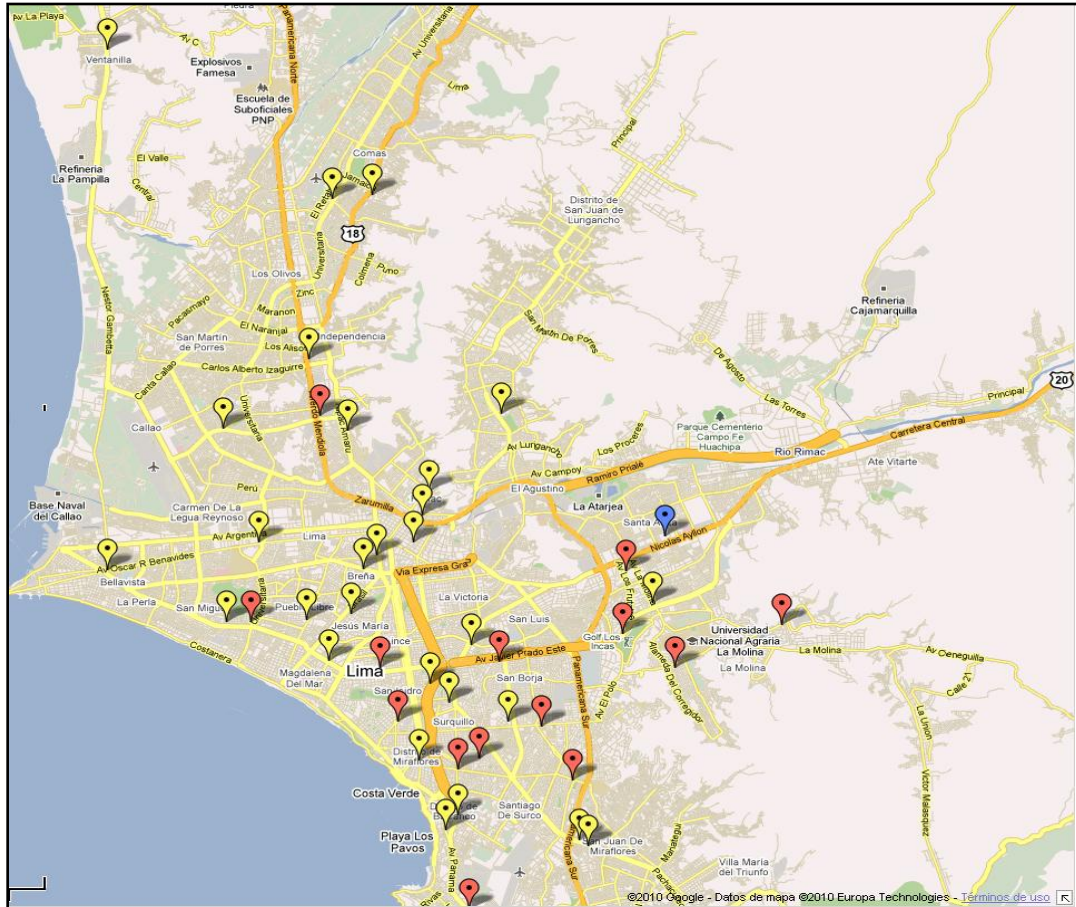


Gráfico 3.1. Mapeo de clientes

Fuente: Google Maps (2011)

Los clientes se diferencian en tipo 1 y tipo 2 debido a que estos poseen distintas ventanas de tiempo con respecto a la recepción de mercadería, entonces esta distinción nos ayudará a identificar los clientes en los que debemos dar prioridad en caso estemos sujetos al tiempo.

A partir del mapeo de clientes que observamos en el gráfico 3.1 se realizarán todas las propuestas que plantearemos en el presente estudio, tomando como base este *mapping* se intentará solucionar el problema de ruteo de vehículos.

3.2. Propuestas de Ruteo

En esta sección plantearan y detallaran las propuestas del presente estudio, las cuales se basan en los principios expuestos en el marco teórico descrito en un comienzo. Para cada propuesta se realizará una descripción paso a paso de cómo es que llegamos a la solución.

3.2.1. Propuesta 1: Por cercanía de Puntos

La primera propuesta se basa en el método asignar primero y rutear después, ya que estamos formando los grupos de acuerdo a que tan cerca se encuentran los puntos entre ellos y luego se creará una ruta que visite a todos estos puntos o clientes mediante el método de los ahorros. A continuación en el gráfico 3.2 observamos la formación de los *clusters*:

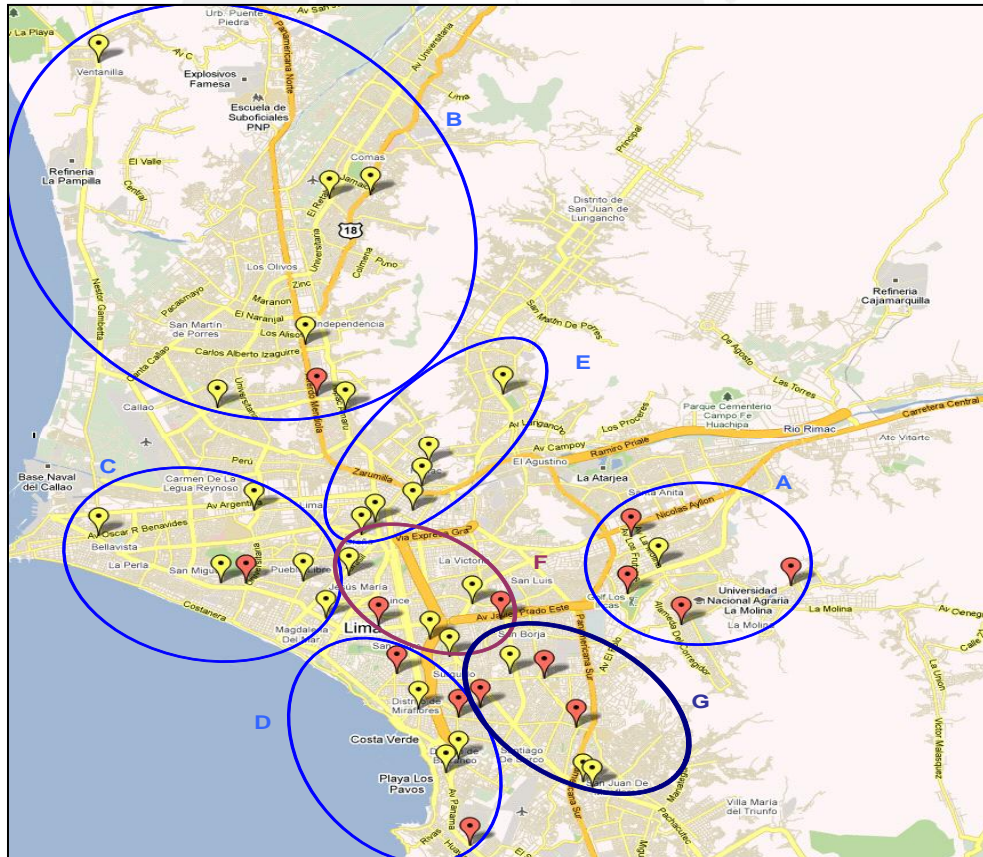


Gráfico 3.2. Formación de *clusters* propuesta 1

Fuente: Google Maps (2011)

En el gráfico 3.2 mostramos los siete (7) grupos que se han formado para esta primera propuesta tomando en cuenta la cercanía entre ellos. Cada grupo está conformado por seis (6) puntos excepto por los *clusters* A y B, lo cuales poseen cinco (5) y siete (7) respectivamente. Para esta primera propuesta y para la siguiente se tomará en cuenta sólo siete (7) puntos por grupo como máximo, esto debido a la capacidad del tipo de transporte que utilizan en la empresa.

Ahora pasaremos a describir como hemos aplicado el método del ahorro, también conocido como el algoritmo de Clarke y Wright, para encontrar la ruta que visite a todos los puntos dentro de un *cluster* específico, tomaremos al *cluster* A como ejemplo para nuestra descripción.

Nota: todas las distancias están dadas en kilómetros.

1. Construcción de matriz de distancias entre los puntos del *cluster*.

Tabla 3-1. Matriz de distancias entre puntos *cluster* A – Propuesta 1

PUNTOS	1	2	4	3	5
1		1.46	3.07	4.04	5.08
2	1.46		1.53	3.16	4.37
4	3.07	1.53		2.98	4.91
3	4.04	3.16	2.98		4.04
5	5.08	4.37	4.91	4.04	

Elaboración propia

2. Definición de los costos (distancias) de hacer un viaje desde el origen (almacén) hasta cada punto del grupo.

Tabla 3-2. Costos del origen al punto *cluster* A – Propuesta 1

COSTO DE ORIGEN	PUNTOS	C_{0j}
j	1	3.70
	2	5.02
	3	9.85
	4	7.82
	5	10.30

Elaboración propia

3. Elaboración de la matriz de ahorros.

Para el cálculo de ahorro utilizamos la siguiente fórmula:

$$S_{ij} = C_{0i} + C_{0j} - C_{ij}$$

Tabla 3-3. Matriz de ahorros *cluster* A – Propuesta 1

AHORROS		j				
	S _{ij}	1	2	4	3	5
i	1		7.26	8.45	9.51	8.92
	2	7.26		11.32	11.71	10.95
	4	8.45	11.32		14.69	13.21
	3	9.51	11.71	14.69		16.11
	5	8.92	10.95	13.21	16.11	
T		2	2	2	2	2
		2	2	2	1	1
		2	2	1	0	1
		2	1	0	0	1
		1	1	0	0	0

Elaboración propia

En la tabla 3.3 observamos que algunas celdas rellenas de amarillo y otras de verde, esto es porque las de amarillo son los ahorros considerados para la elaboración de nuestra ruta; mientras que, las de color verde significan que esos ahorros no cumplen con alguna de las restricciones, es por ello que no se les toma en cuenta.

4. Construcción del modelo de la ruta seleccionada.

- Modelo inicial.

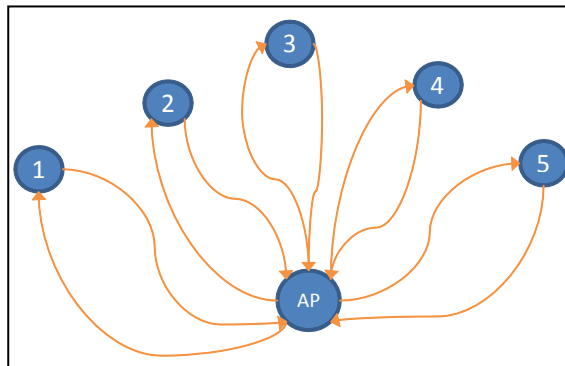


Gráfico 3.3. Ejemplo 1 - método de ahorros

- Primera interacción:

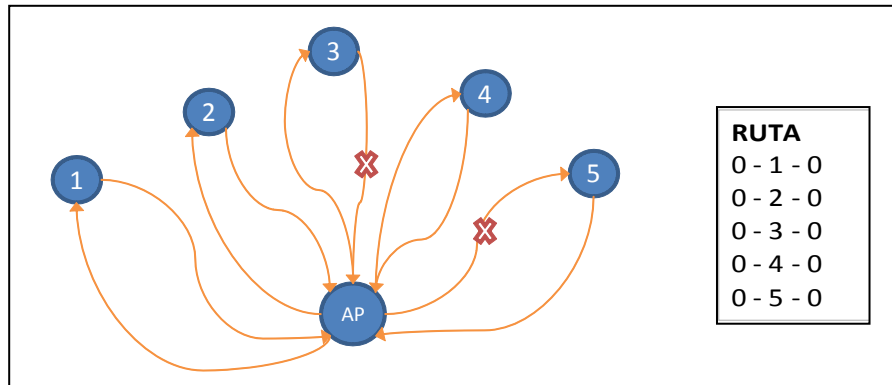


Gráfico 3.4. Ejemplo 1 - después de la 1era interacción

- Segunda interacción: S_{3-5}

Para esta interacción se ha tomado en cuenta el mayor ahorro, el cual fue, según la tabla 3.4, de 16.11 que se dio entre los puntos 3 y 5.

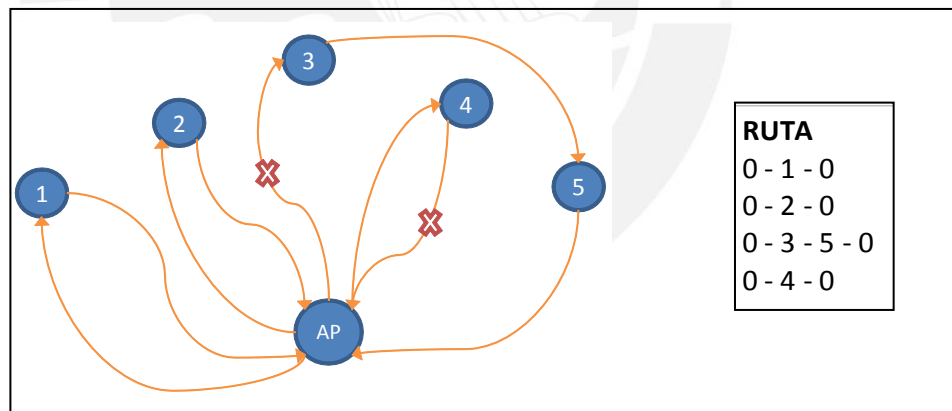


Gráfico 3.5. Ejemplo 1- después de la 2da interacción

- Tercera interacción: S_{4-3}

Para esta interacción se ha tomado en cuenta el siguiente mayor ahorro posible que cumpla con las condiciones, el cual fue de 14.69 que se dio entre los punto 4 y 3.

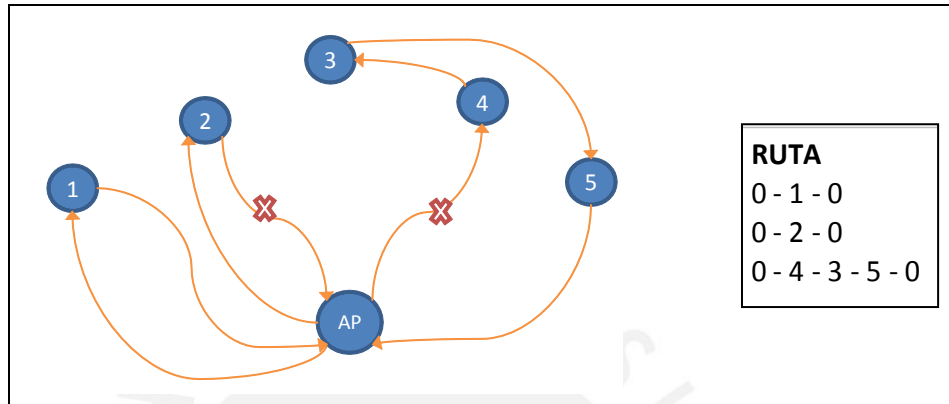


Gráfico 3.6. Ejemplo 1 - después de la 3era interacción

- Cuarta interacción: S_{2-4}

Para esta interacción se ha tomado en cuenta el siguiente mayor ahorro posible que cumpla con las condiciones, el cual fue de 11.32 que se dio entre los punto 2 y 4.

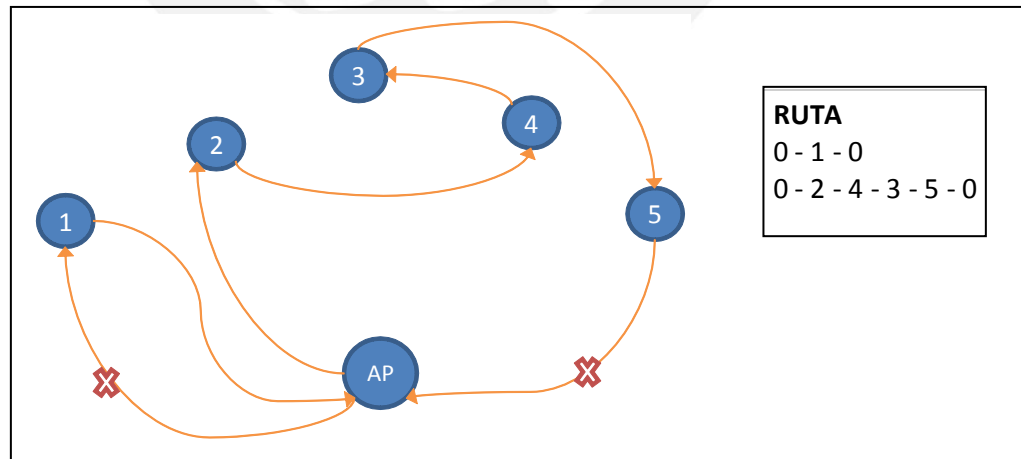


Gráfico 3.7. Ejemplo 1 - después de la 4ta interacción

- Quinta interacción: S_{1-5}

Para esta última interacción se ha tomado en cuenta el siguiente mayor ahorro posible que cumpla con las condiciones, el cual fue de 8.92 que se dio entre los punto 1 y 5. Esta interacción nos da la ruta óptima para la segunda propuesta.

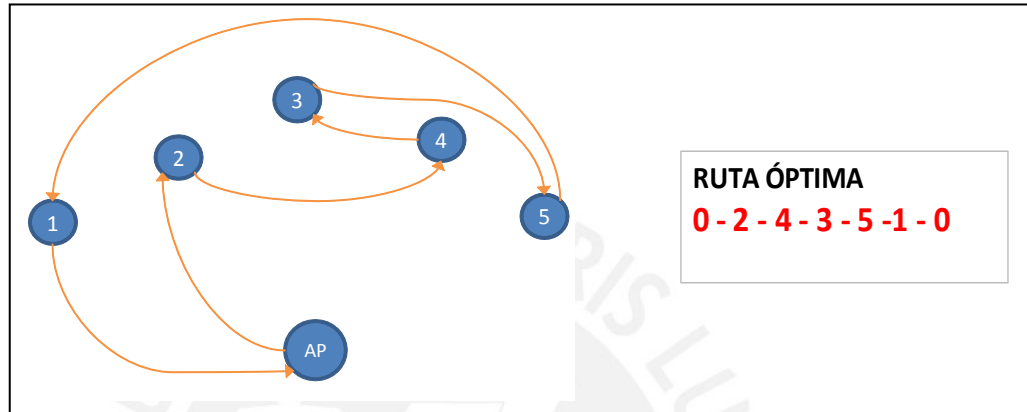


Gráfico 3.8. Ejemplo 1 - Ruta óptima

En el gráfico 3.9 presentamos la ruta óptima en el plano de la ciudad de Lima, después de elaborar el método de los ahorros para esta propuesta. La distancia total recorrida sería de **17.98 km** (sin considerar las calles ni el tráfico).

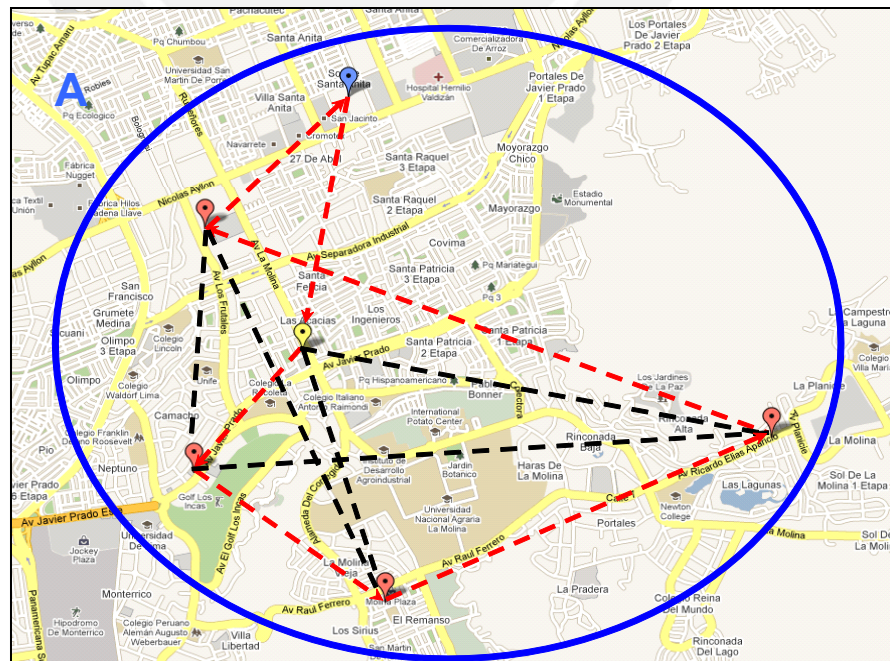


Gráfico 3.9. Ruta óptima – Propuesta 1

Fuente: Google Maps (2011)

3.2.2. Propuesta 2: Método del Barrido

La segunda propuesta también se basa en el método asignar primero y rutear después, pero esta vez se creará una ruta que visite a todos los puntos mediante el método del barrido o también conocido como el algoritmo de Gillet y Miller. Para la formación de *clusters* se han realizado los siguientes pasos:

- Asumimos las coordenadas (r_i, θ_i) , en donde r_i será el almacén y θ_i el ángulo que forman los clientes con este.
- Se formaron siete (7) *clusters* conformados por seis (6) puntos, comenzando con los clientes que poseen el menor ángulo con respecto del almacén, así hasta completar los siete (7) grupos.

A continuación en el gráfico 3.10 observamos el resultado de la aplicación del método de barrido:

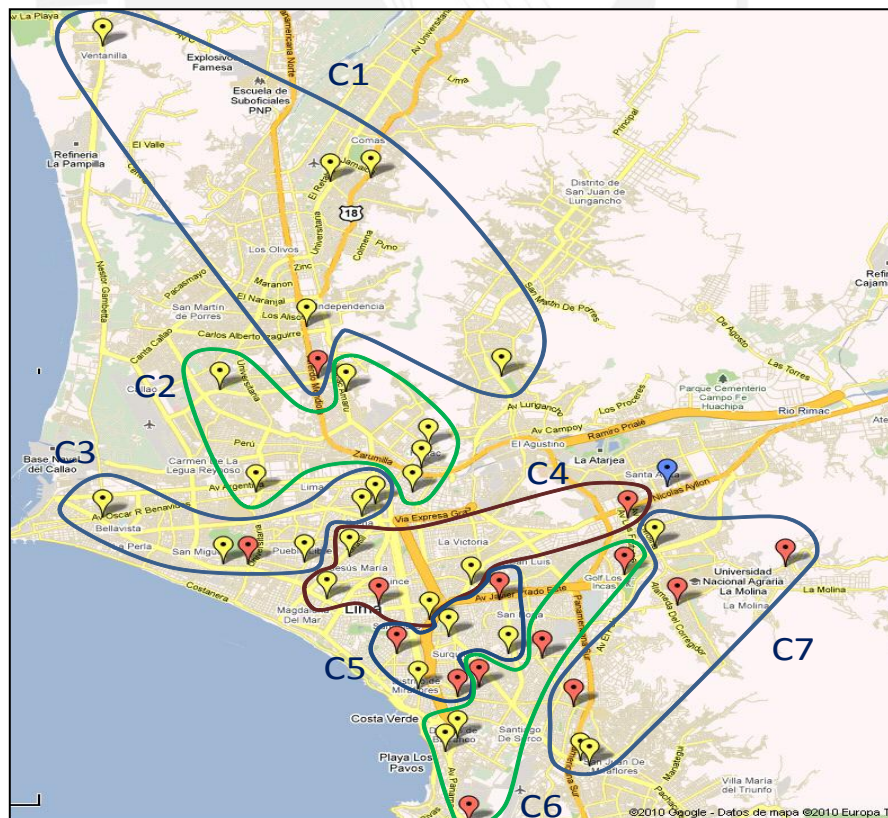


Gráfico 3.10. Formación de *clusters* – Propuesta 2

Fuente: Google Maps (2011)

Ahora nos toca explicar cómo hemos efectuado el algoritmo de Clarke y Wright para esta propuesta, en donde, también, buscamos encontrar una ruta que visite a todos los puntos dentro de un grupo dado, en esta oportunidad analizaremos el *cluster C2* para mostrar el detalle de la aplicación.

Nota: todas las distancias están dadas en kilómetros.

1. Construcción de matriz de distancias entre los puntos del *cluster*.

Tabla 3-4. Matriz de distancias entre puntos *cluster C2* – Propuesta 2

	33	34	35	36	37	40
33		3.44	3.99	4.70	5.10	4.18
34			1.01	2.01	6.08	7.26
35				1.04	5.55	7.44
36					5.20	7.69
37						4.44
40						

Elaboración propia

2. Definición de los costos (distancias) de hacer un viaje desde el origen (almacén) hasta cada punto del grupo.

Tabla 3-5. Costos del origen al punto *cluster C2* – Propuesta 2

COSTO DE ORIGEN	PUNTOS	C_{oj}
<i>j</i>	33	22.72
	34	16.16
	35	16.38
	36	16.90
	37	27.29
	40	30.80

Elaboración propia

3. Elaboración de la matriz de ahorros.

Para el cálculo de ahorro utilizamos la siguiente fórmula:

$$S_{ij} = C_{0i} + C_{0j} - C_{ij}$$

Tabla 3-6. Matriz de ahorros *cluster* C2 – Propuesta 2

AHORROS		<i>j</i>					
<i>Sij</i>		33	34	35	36	37	40
<i>i</i>	33		35.44	35.12	34.92	44.92	49.34
	34			31.54	31.04	37.37	39.70
	35				32.24	38.13	39.75
	36					38.99	40.01
	37						53.66
	40						
T		2	2	2	2	2	2
		2	2	2	2	1	1
		1	2	2	2	1	0
		1	2	2	1	0	0
		0	1	2	1	0	0
		0	1	1	0	0	0

Elaboración propia

En la tabla 3-6 observamos que algunas celdas rellenas de amarillo y otras de verde, esto es porque las de amarillo son los ahorros considerados para la elaboración de nuestra ruta; mientras que, las de color verde significan que esos ahorros no cumplen con alguna de las restricciones, es por ello que no se les toma en cuenta.

4. Construcción del modelo de la ruta seleccionada.

Modelo inicial.

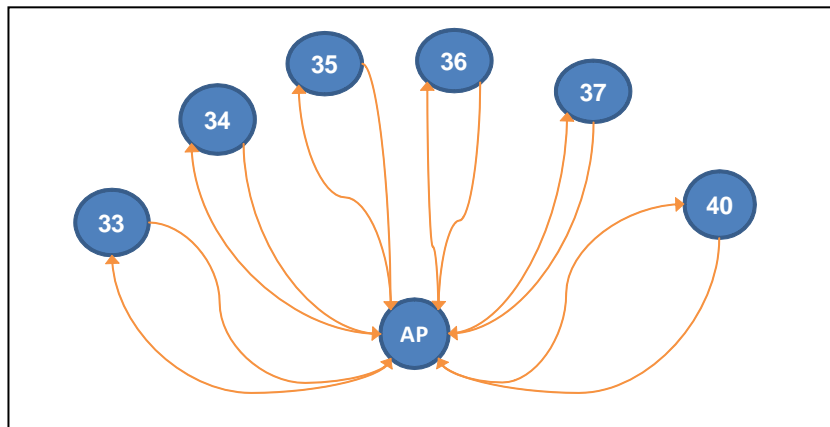


Gráfico 3.11. Ejemplo 2 – método de ahorros

Primera interacción:

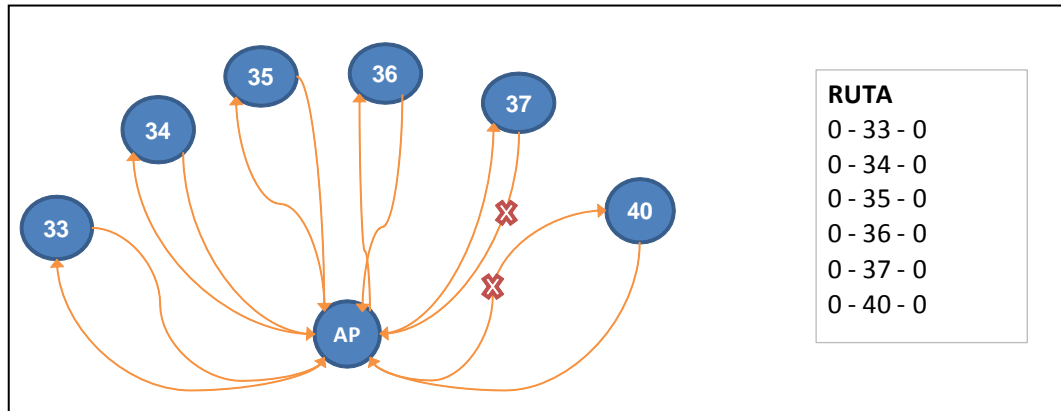


Gráfico 3.12. Ejemplo 2 – después de la 1era interacción

Segunda interacción: S_{37-40}

Para esta interacción se ha tomado en cuenta el mayor ahorro, el cual fue de 53.66 que se dio entre los punto 37 y 40.

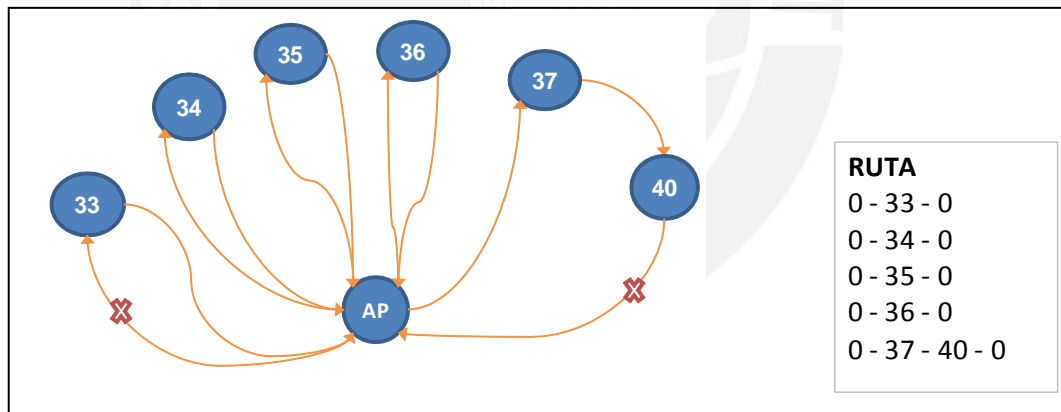


Gráfico 3.13. Ejemplo 2 – después de la 2da interacción

Tercera interacción: S_{40-33}

Para esta interacción se ha tomado en cuenta el siguiente mayor ahorro posible que cumpla con las condiciones, el cual fue de 49.34 que se dio entre los punto 40 y 37.

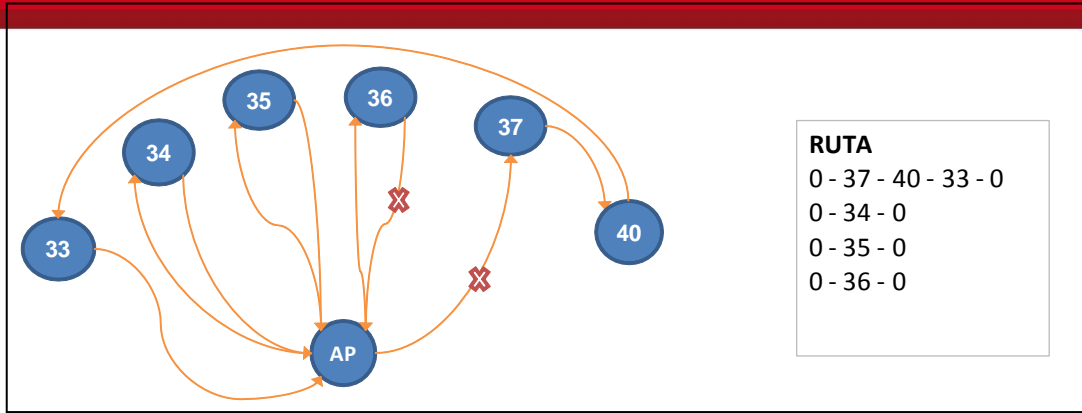


Gráfico 3.14. Ejemplo 2 – después de la 3era interacción

Cuarta interacción: S_{36-37}

Para esta interacción se ha tomado en cuenta el siguiente mayor ahorro posible que cumpla con las condiciones, el cual fue de 38.99 que se dio entre los punto 36 y 37.

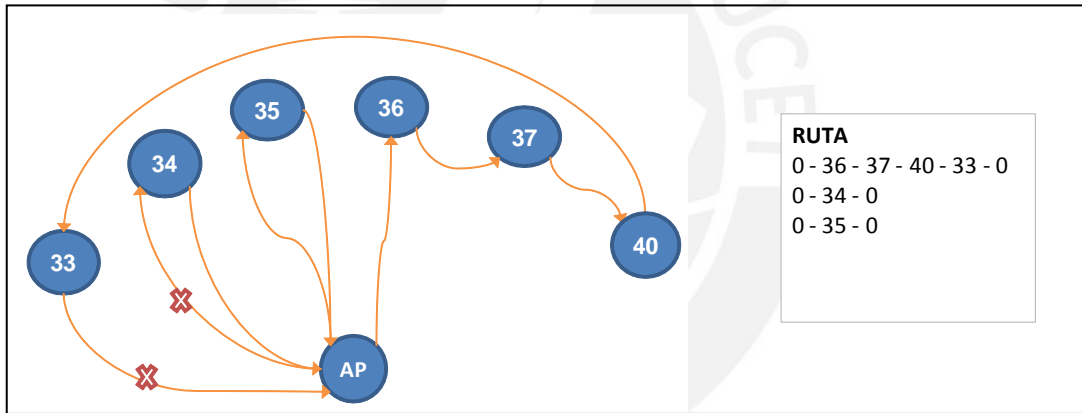


Gráfico 3.15. Ejemplo 2 – después de la 4ta interacción

Quinta interacción: S_{33-34}

Para esta interacción se ha tomado en cuenta el siguiente mayor ahorro posible que cumpla con las condiciones, el cual fue de 35.44 que se dio entre los punto 33 y 34.

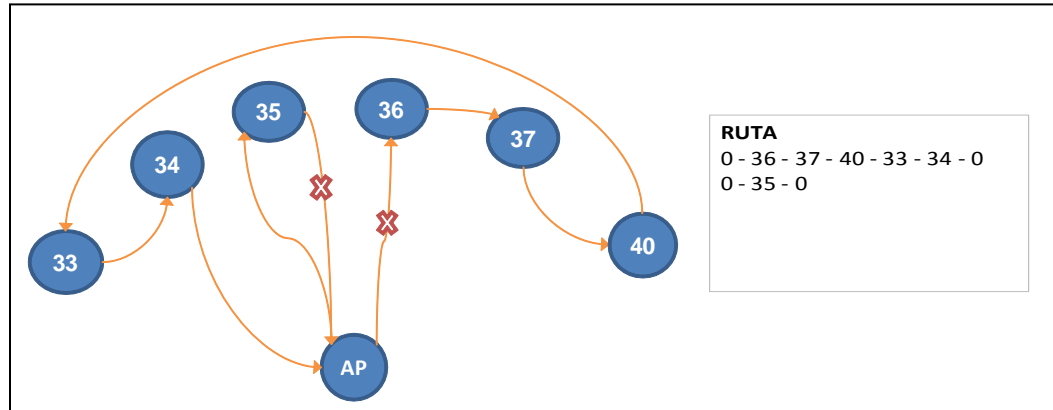


Gráfico 3.16. Ejemplo 2 – después de la 5ta interacción

Sexta interacción: S_{35-36}

Para esta última interacción se ha tomado en cuenta el siguiente mayor ahorro posible que cumpla con las condiciones, el cual fue de 32.24 que se dio entre los punto 35 y 36. Esta interacción nos da la ruta óptima para la segunda propuesta.

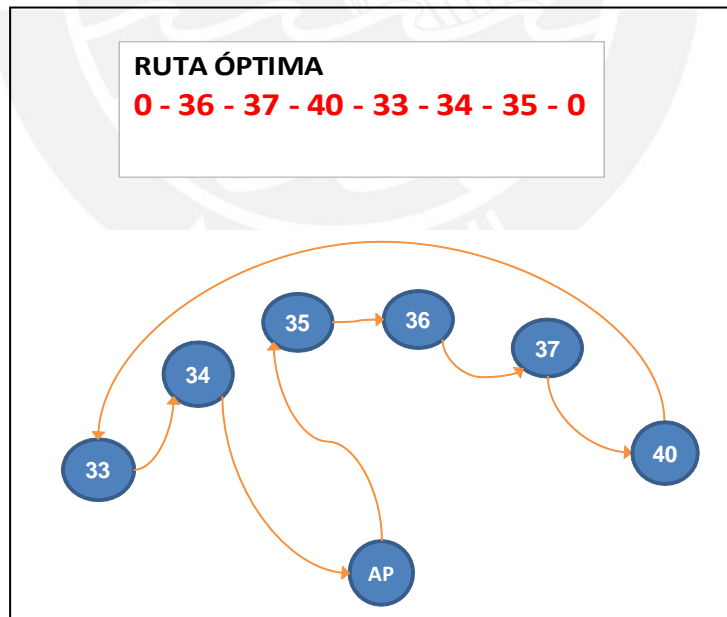


Gráfico 3.17. Ejemplo 2- Ruta óptima

En el gráfico 3.18 observamos la ruta óptima en el plano de la ciudad de Lima, una vez elaborado el algoritmo de Clark y Wright para esta propuesta. La distancia total recorrida sería de **34.58 km** (sin considerar las calles ni el tráfico).

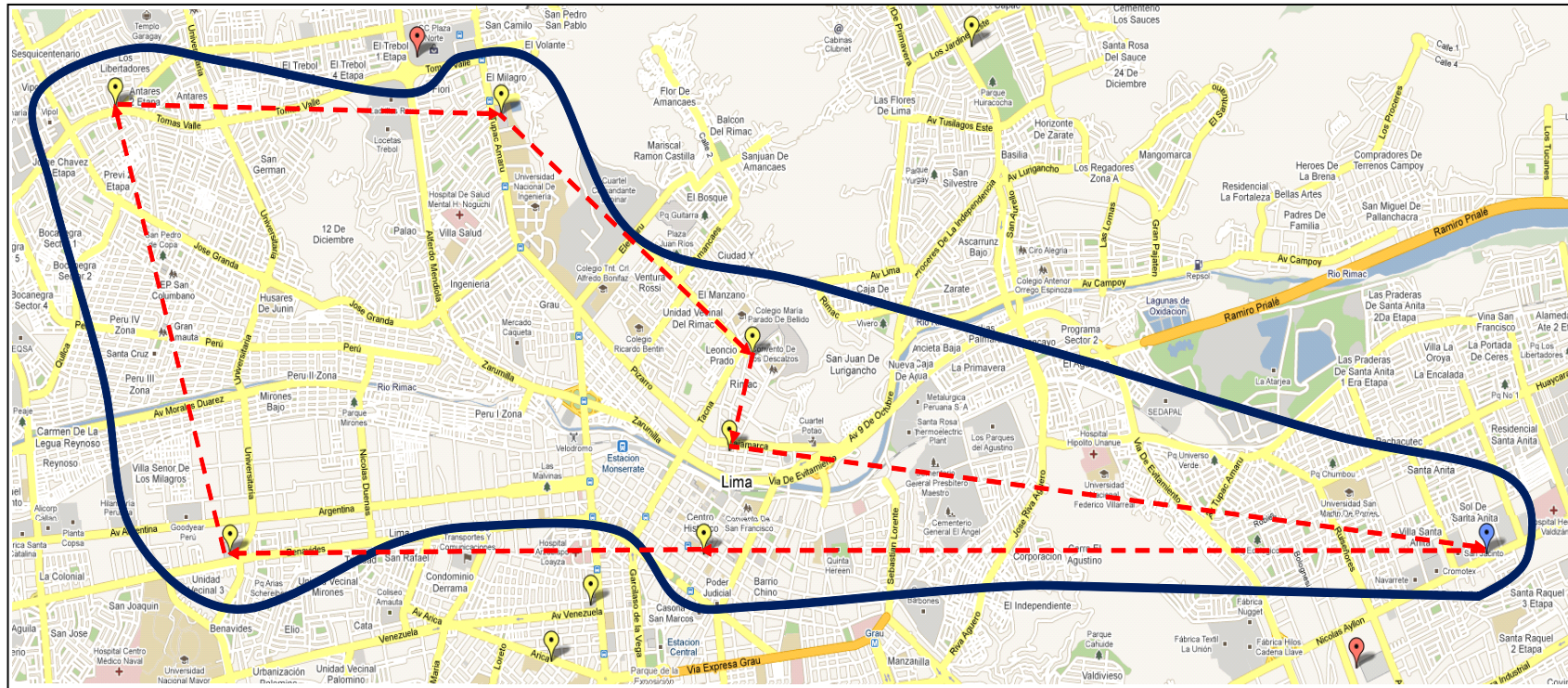


Gráfico 3.18. Ruta óptima – Propuesta 2

Fuente: Google Maps (2011)

3.2.3. Propuesta 3: *Route First – Cluster Second* (Gran Ruta)

A continuación describiremos la propuesta basada en el método de rutear primero y asignar después, en donde, mediante el método del ahorro se encontró la gran ruta para los cuarenta y dos puntos a tratar, a partir de los resultados obtenidos se formaron siete *clusters*, debido a que se tiene como restricción la capacidad del camión que solo puede abastecer como máximo hasta seis puntos en una sola ruta, excepto casos especiales como lo vimos en la propuesta 1.

En la propuestas anteriores primero se formaban los *clusters* y luego se aplicaba el método de Clark y Wright para encontrar la solución óptima en cada *cluster*, ahora haremos el procedimiento inverso. En primer lugar, se encontrará la ruta óptima para todos los puntos como si se tratara de un *cluster* con cuarenta y dos puntos y ya no con seis. Como hemos explicado para las anteriores propuestas tenemos que seguir cada uno de los pasos para resolver el método de ahorros, debido a la extensión de los puntos (42) el procedimiento se puede observar en el anexo electrónico llamado: “*Distancias_Propuesta_3*”.

Para encontrar la ruta óptima primero hallamos las distancias desde las tiendas hacia el almacén principal. Luego calculamos las distancias entre cada uno de los cuarenta y dos puntos llegando a la matriz final de distancias, con ello realizamos la matriz de ahorro aplicando la siguiente fórmula:

$$S_{ij} = C_{0i} + C_{0j} - C_{ij}$$

Una vez aplicado el método del ahorro obtenemos la siguiente gran ruta:

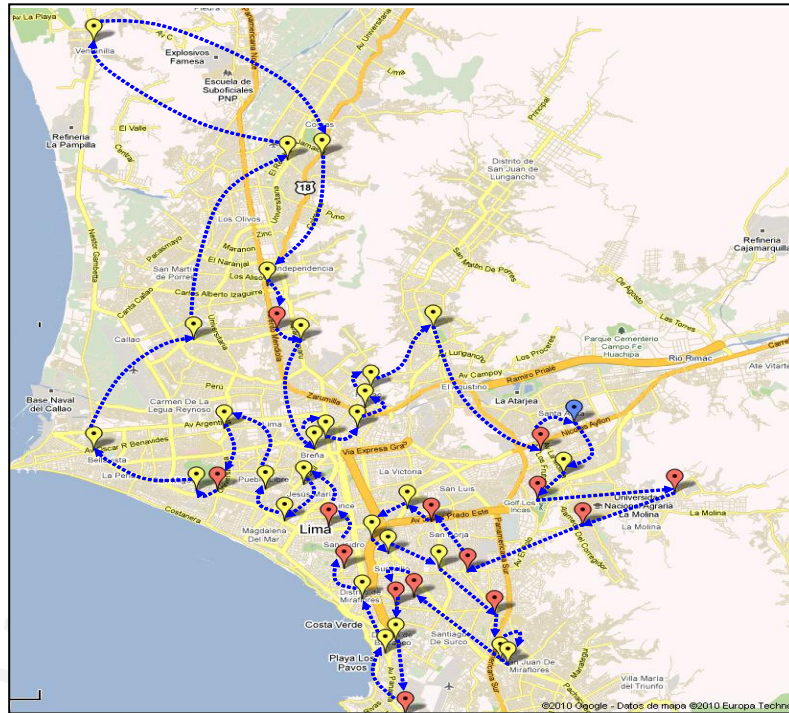


Gráfico 3.19. Gran ruta – Propuesta 3

Fuente: Google Maps (2011)

Luego pasamos a la formación de *clusters* tomando como restricción el abastecimiento máximo de sólo seis tiendas:

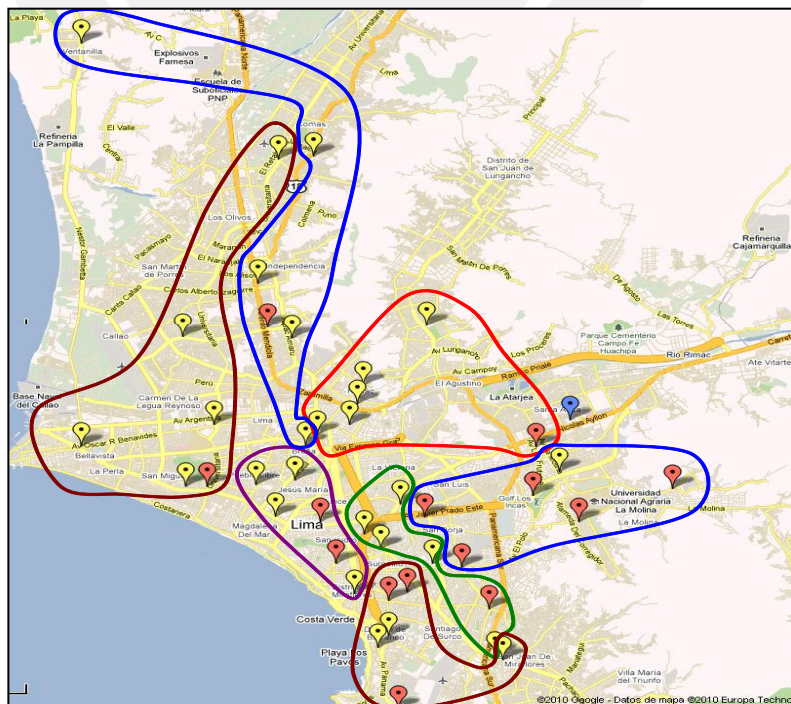


Gráfico 3.20. Formación de *clusters* – Propuesta 3

Fuente: Google Maps (2011)

Finalmente, en el gráfico 3.21 observamos la ruta que se debe seguir desde el almacén principal hacia las tiendas para cada *cluster* formado en la presente propuesta:

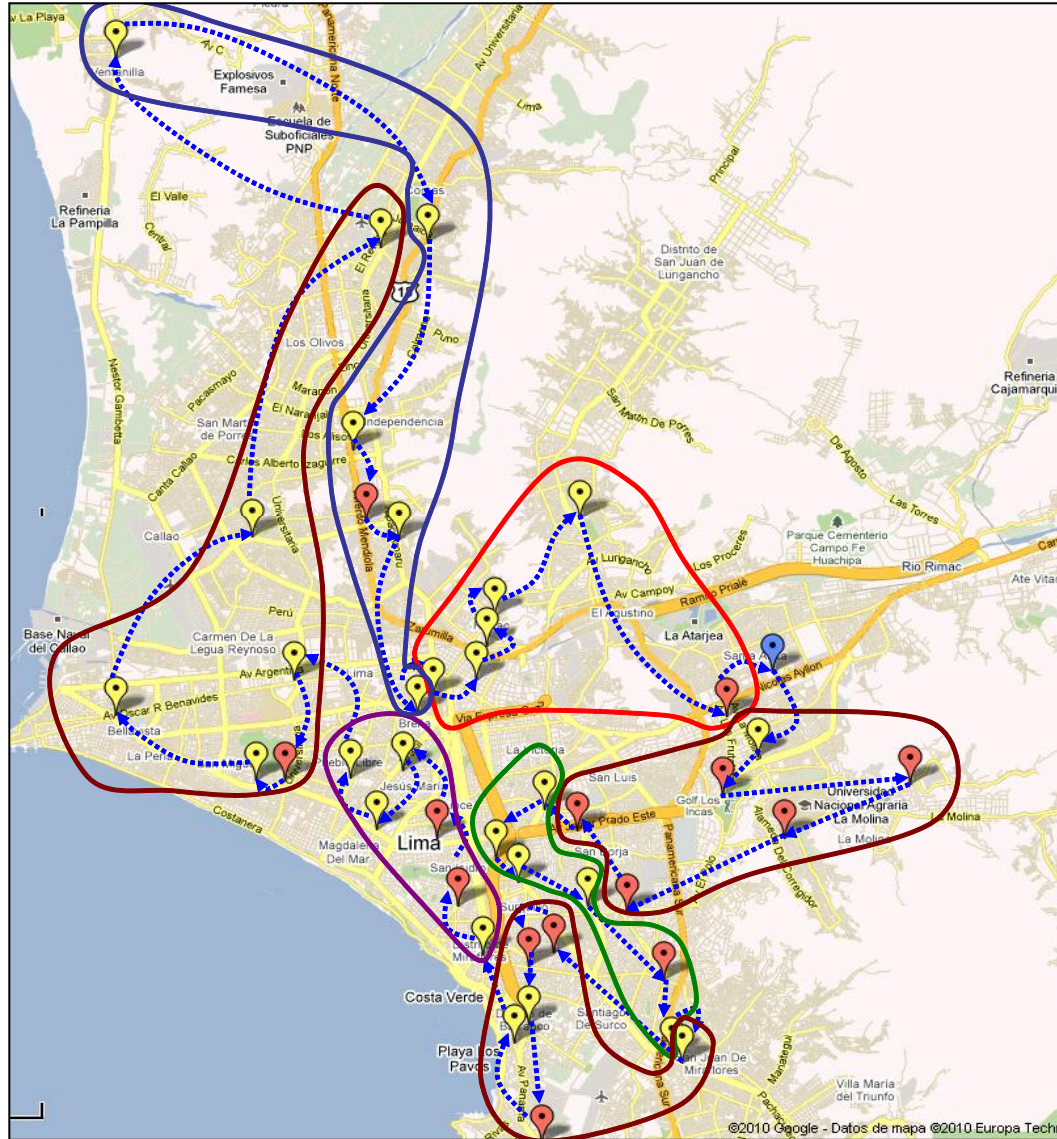


Gráfico 3.21. Formación de *clusters* incluyendo gran ruta – Propuesta 3
Fuente: Google Maps (2011)

3.2.4. Propuesta 4: Modelo Matemático por Cercanía de Puntos

En la presenta propuesta utilizaremos el método aplicado en la propuesta uno para la formación de *clusters*; es decir, partiremos del hecho que ya se han formado los *clusters* por cercanía de puntos. La diferencia es que esta vez ya no lo resolveremos por el método del ahorro, sino mediante el modelo matemático que plantea el problema del agente viajero TSP (por sus siglas en inglés *Travelling Salesman Problem*).

A continuación plantearemos dos tipos de problemas que se pueden presentar al momento de realizar la programación lineal, se tomaran como ejemplos los *clusters* A y B formados por cercanía de puntos, para ello tenemos los siguientes PL:

CLUSTER A:

Variables de decisión:

$$\left\{ \begin{array}{l} x_{ij} = \text{Decisión de ir o no desde el punto } i \text{ hacia el punto } j \\ \quad i: 0, 1, 2, 3, 4, 5 \quad j: 0, 1, 2, 3, 4, 5 \\ 0 = \text{ caso contrario} \end{array} \right.$$

Función objetivo:

$$\begin{aligned} \min \quad Z = & 1.85x_{01} + 2.51x_{02} + 4.92x_{03} + 3.91x_{04} + 5.15x_{05} + \\ & 1.46x_{12} + 4.04x_{13} + 3.07x_{14} + 5.08x_{15} + \\ & 3.16x_{23} + 1.53x_{24} + 4.37x_{25} + \\ & 2.98x_{34} + \\ & 4.04x_{35} + \\ & 4.91x_{45} \end{aligned}$$

Sujeto a:

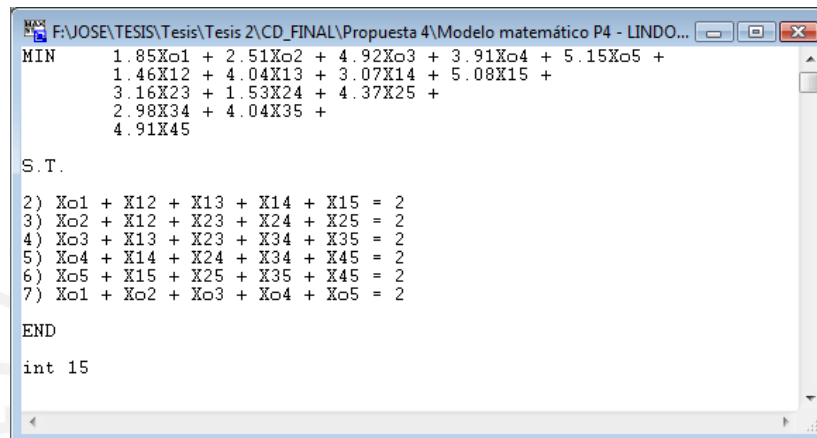
$$\begin{aligned} x_{01} + x_{02} + x_{03} + x_{04} + x_{05} &= 2 \\ x_{01} + x_{12} + x_{13} + x_{14} + x_{15} &= 2 \\ x_{02} + x_{12} + x_{23} + x_{24} + x_{25} &= 2 \\ x_{03} + x_{13} + x_{23} + x_{34} + x_{35} &= 2 \\ x_{04} + x_{14} + x_{24} + x_{34} + x_{45} &= 2 \\ x_{05} + x_{15} + x_{25} + x_{35} + x_{45} &= 2 \end{aligned}$$

Rango de existencia:

$$x_{ij} = 0 \text{ ó } 1$$

En este caso no hay necesidad de colocar la restricción que evita la presencia de sub-tours, por lo tanto, basta con la programación lineal planteada para resolver el problema. Utilizamos el software LINDO, el cual es una herramienta de optimización.

En el gráfico 3.22 observamos la formulación en LINDO, la cual nos dará a forma de reporte los puntos que debemos tomar en cuenta para escoger la ruta óptima.



```

F:\JOSE\TESIS\Tesis 2\CD_FINAL\Propuesta 4\Modelo matemático P4 - LINDO...
MIN    1.85X01 + 2.51X02 + 4.92X03 + 3.91X04 + 5.15X05 +
        1.46X12 + 4.04X13 + 3.07X14 + 5.08X15 +
        3.16X23 + 1.53X24 + 4.37X25 +
        2.98X34 + 4.04X35 +
        4.91X45

S. T.

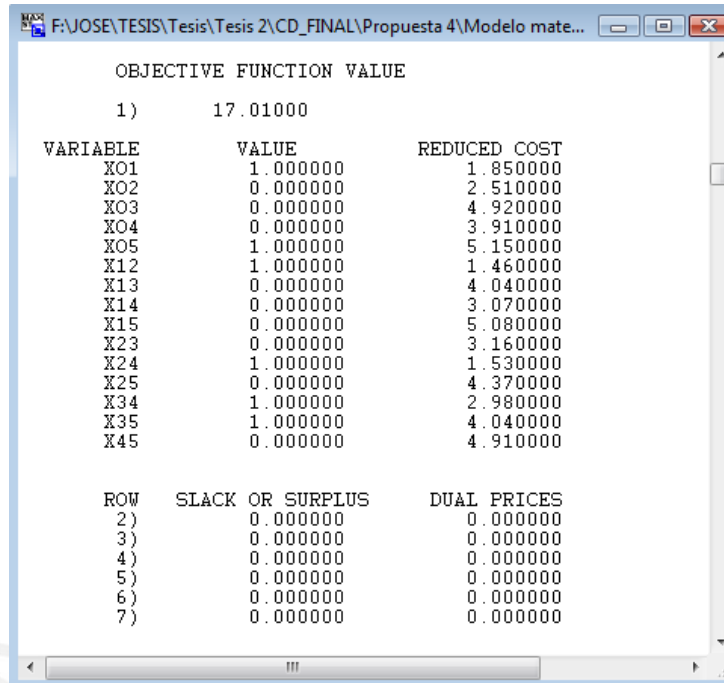
2) X01 + X12 + X13 + X14 + X15 = 2
3) X02 + X12 + X23 + X24 + X25 = 2
4) X03 + X13 + X23 + X34 + X35 = 2
5) X04 + X14 + X24 + X34 + X45 = 2
6) X05 + X15 + X25 + X35 + X45 = 2
7) X01 + X02 + X03 + X04 + X05 = 2

END

int 15
  
```

Gráfico 3.22. Formulación *cluster A* - LINDO

Luego el software LINDO resuelve el problema presentando la solución en un reporte, en el que indica el valor de la función objetivo, además detalla los valores de cada una de las variables de decisión, para nuestro caso sólo serán 1 y 0, ya que nuestro modelo es un problema de asignación. En el gráfico 3.23 observamos el reporte que nos brinda el LINDO, en el cual se interpreta que la ruta elegida para el *cluster A* sería: **O-1-2-4-3-5-O**, recorriendo una distancia de **17.01 Km**.



OBJECTIVE FUNCTION VALUE

1) 17.01000

VARIABLE	VALUE	REDUCED COST
X01	1.000000	1.850000
X02	0.000000	2.510000
X03	0.000000	4.920000
X04	0.000000	3.910000
X05	1.000000	5.150000
X12	1.000000	1.460000
X13	0.000000	4.040000
X14	0.000000	3.070000
X15	0.000000	5.080000
X23	0.000000	3.160000
X24	1.000000	1.530000
X25	0.000000	4.370000
X34	1.000000	2.980000
X35	1.000000	4.040000
X45	0.000000	4.910000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	0.000000	0.000000
6)	0.000000	0.000000
7)	0.000000	0.000000

Gráfico 3.23. Reporte cluster A - LINDO

Ahora analizaremos un caso en el que si es necesario incluir la restricción para evitar sub-tours, como lo fue para el cluster B, presentamos la programación lineal:

CLUSTER B:

Variables de decisión

$$\left\{ \begin{array}{l} x_{ij} = \text{Decisión de ir o no desde el punto } i \text{ hacia el punto } j \\ \quad i: 0, 29, 30, 31, 32, 33, 40, 42 \quad j: 0, 29, 30, 31, 32, 33, 40, 42 \\ 0 = \text{caso contrario} \end{array} \right.$$

Función objetivo

$$\begin{aligned} \min Z = & 11.36x_{033} + 12.42x_{032} + 13.68x_{031} + 15.40x_{040} + \\ & 16.90x_{030} + 26.17x_{042} + 16.17x_{029} + 1.07x_{3332} + \\ & 3.01x_{3331} + 4.22x_{3340} + 16.99x_{3342} + 8.74x_{3330} + \\ & 8.95x_{3329} + 2.22x_{3231} + 3.32x_{3240} + 16.09x_{3242} + \\ & 8.18x_{3230} + 8.51x_{3239} + 3.94x_{3140} + 14.13x_{3142} + \\ & 6.12x_{3130} + 6.44x_{3129} + 15.39x_{4042} + 9.40x_{4030} + \\ & 10.14x_{4029} + 9.80x_{4230} + 10.75x_{4229} + 1.40x_{3029} \end{aligned}$$

Restricciones

$$\begin{aligned}
 X_{033} + X_{032} + X_{031} + X_{040} + X_{042} + X_{030} + X_{29} &= 2 \\
 X_{033} + X_{3332} + X_{3331} + X_{3340} + X_{3342} + X_{3330} + X_{3329} &= 2 \\
 X_{032} + X_{3332} + X_{3231} + X_{3240} + X_{3242} + X_{3230} + X_{3229} &= 2 \\
 X_{031} + X_{3331} + X_{3231} + X_{3140} + X_{3142} + X_{3130} + X_{3129} &= 2 \\
 X_{040} + X_{3340} + X_{3240} + X_{3140} + X_{4042} + X_{4030} + X_{4029} &= 2 \\
 X_{042} + X_{3342} + X_{3242} + X_{3142} + X_{4042} + X_{4230} + X_{4229} &= 2 \\
 X_{030} + X_{3330} + X_{3230} + X_{3130} + X_{4030} + X_{4230} + X_{3029} &= 2 \\
 X_{029} + X_{3329} + X_{3229} + X_{3129} + X_{4029} + X_{4229} + X_{3029} &= 2 \\
 X_{029} + X_{042} + X_{030} + X_{3329} + X_{3342} + X_{3330} + X_{3229} + X_{3242} + X_{3230} + \\
 X_{4029} + X_{4042} + X_{4030} + X_{3129} + X_{3142} + X_{3130} &\geq 1
 \end{aligned}$$

Rango de existencia

$$x_{ij} = 0 \text{ ó } 1$$

De la misma manera se ingresa la programación lineal planteada al software LINDO, para la resolución del mismo, nótese la diferencia en la última restricción, para llegar a esta primero se corrió el modelo matemático sin considerar la posibilidad de sub-tour como lo hicimos para el *cluster* A, en el reporte que nos arrojó LINDO la solución óptima incluía un sub-tour, es por ello que se agregó la última restricción para evitar este tipo de complicaciones. Finalmente se ingresó el modelo matemático al software de la siguiente forma:

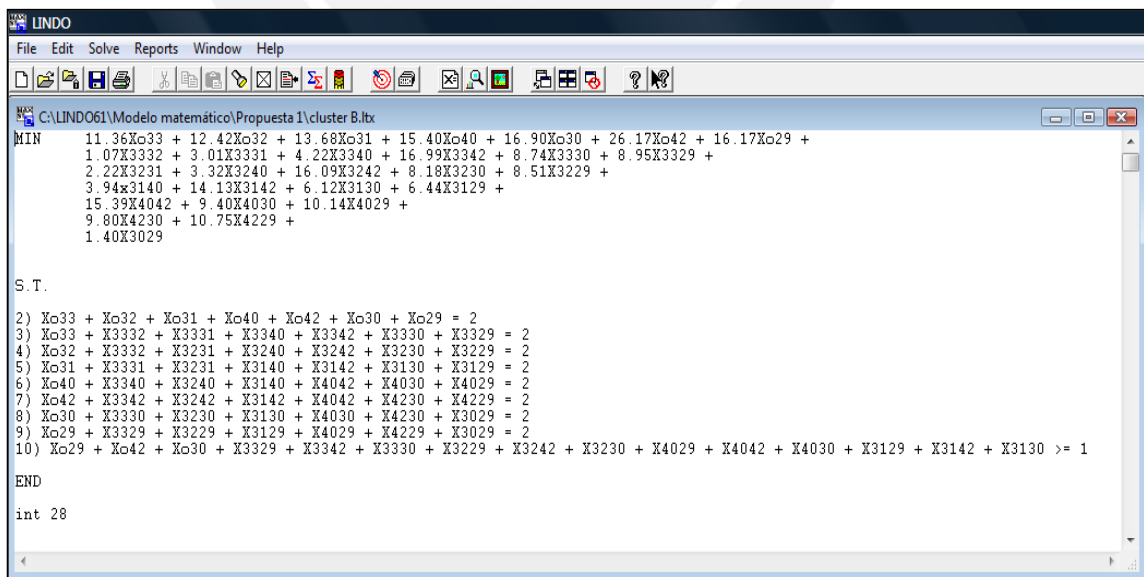


Gráfico 3.24. Programación lineal *cluster* B -LINDO

Acto seguido, el LINDO nos presenta el resultado de la formulación vista en el gráfico 3.24 en el siguiente reporte:

OBJECTIVE FUNCTION VALUE

1) 61.19000

VARIABLE	VALUE	REDUCED COST
X033	1.000000	11.360000
X032	0.000000	12.420000
X031	0.000000	13.680000
X040	0.000000	15.400000
X030	0.000000	16.900000
X042	0.000000	26.170000
X029	1.000000	16.170000
X3332	1.000000	1.070000
X3331	0.000000	3.010000
X3340	0.000000	4.220000
X3342	0.000000	16.990000
X3330	0.000000	8.740000
X3329	0.000000	8.950000
X3231	0.000000	2.220000
X3240	1.000000	3.320000
X3242	0.000000	16.090000
X3230	0.000000	8.180000
X3229	0.000000	8.510000
X3140	1.000000	3.940000
X3142	1.000000	14.130000
X3130	0.000000	6.120000
X3129	0.000000	6.440000
X4042	0.000000	15.390000
X4030	0.000000	9.400000
X4029	0.000000	10.140000
X4230	1.000000	9.800000
X4229	0.000000	10.750000
X3029	1.000000	1.400000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	0.000000	0.000000
6)	0.000000	0.000000
7)	0.000000	0.000000
8)	0.000000	0.000000
9)	0.000000	0.000000
10)	1.000000	0.000000

NO. ITERATIONS= 77
BRANCHES= 6 DETERM. = 1.000E 0

Gráfico 3.25. Reporte *cluster B* - LINDO

La interpretación del reporte nos indica que la ruta óptima para el *cluster B*, según el modelo de la programación lineal, es la siguiente: **O-33-32-40-31-42-30-29-O**, recorriendo una distancia de **61.19 Km**.

El mismo procedimiento es seguido para los demás clusters, tanto para la propuesta 1 (por cercanía de punto) como para la propuesta 2 (método del barrido), lo que finalmente nos lleva a la siguiente tabla de resultados:

Tabla 3-7. Resultados Propuesta: Cercanía de Puntos

CLUSTERS	DISTANCIA KM
CLUSTER A: O-1-2-4-3-5-O	17.01
CLUSTER B: O-33-32-40-31-42-30-29-O	61.19
CLUSTER C: O-21-22-24-23-41-37-O	39.9
CLUSTER D: O-15-18-13-12-14-17-O	35.16
CLUSTER E: O-36-39-38-35-34-28-O	25.13
CLUSTER F: O-19-20-25-26-27-9-O	26.44
CLUSTER G: O-6-8-16-10-11-7-O	28.18
TOTAL	233.01

Elaboración propia

3.2.5. Propuesta 5: Modelo Matemático por el método del Barrido

En la presenta propuesta utilizaremos el método aplicado en la propuesta dos para la formación de *clusters*; es decir, partiremos del hecho que ya se han formado los *clusters* por el método del barrido. La diferencia es que esta vez ya no lo resolveremos por el método del ahorro, sino mediante el modelo matemático que plantea el problema del agente viajero TSP (por sus siglas en inglés *Travelling Salesman Problem*).

A continuación plantearemos el tipo de problema más complejo que se puede presentar al momento de realizar una programación lineal de este tipo; es decir es necesario incluir restricciones para evitar sub-tours. Se tomara como ejemplo el cluster 3 formado por el método del barrido, para ello tenemos el siguiente PL:

CLUSTER 3:

Variables de decisión

$$\left\{ \begin{array}{l} x_{ij} = \text{Decisión de ir o no desde el punto } i \text{ hacia el punto } j \\ \quad i: 0, 22, 23, 24, 38, 39, 41 \quad j: 0, 22, 23, 24, 38, 39, 41 \\ 0 = \text{ caso contrario} \end{array} \right.$$

Función objetivo

$$\begin{aligned} \min Z = & 12.42x_{022} + 15.13x_{023} + 14.27x_{024} + 9.70x_{038} + 18.78x_{041} + \\ & 2.67x_{2223} + 1.92x_{2224} + 3.43x_{2238} + 2.72x_{2239} + 6.95x_{2241} + \\ & 0.88x_{2324} + 5.66x_{2338} + 5.04x_{2339} + 4.39x_{3941} + \\ & 4.94x_{2438} + 4.31x_{2439} + 5.15x_{2441} + \\ & 0.70x_{3839} + 9.09x_{3841} + \\ & 8.63x_{3941} \end{aligned}$$

Restricciones

$$\begin{aligned} x_{022} + x_{023} + x_{024} + x_{038} + x_{039} + x_{041} &= 2 \\ x_{022} + x_{2223} + x_{2224} + x_{2238} + x_{2239} + x_{2241} &= 2 \\ x_{023} + x_{2223} + x_{2324} + x_{2338} + x_{2339} + x_{2341} &= 2 \\ x_{024} + x_{2224} + x_{2324} + x_{2438} + x_{2439} + x_{2441} &= 2 \\ x_{038} + x_{2238} + x_{2338} + x_{2438} + x_{3839} + x_{3841} &= 2 \\ x_{039} + x_{2239} + x_{2339} + x_{2439} + x_{3839} + x_{3941} &= 2 \\ x_{041} + x_{2241} + x_{2341} + x_{2441} + x_{3841} + x_{3941} &= 2 \\ x_{022} + x_{023} + x_{024} + x_{041} + x_{2238} + x_{2338} + x_{2438} + x_{3841} + x_{2239} + \\ x_{2339} + x_{2439} + x_{3941} &\geq 1 \\ x_{023} + x_{024} + x_{041} + x_{2223} + x_{2224} + x_{2241} + x_{2338} + x_{2438} + x_{3841} + \\ x_{2339} + x_{2439} + x_{3941} &\geq 1 \end{aligned}$$

Rango de existencia

$$x_{ij} = 0 \text{ ó } 1$$

De la misma manera se ingresa la programación lineal planteada al software LINDO como vemos en el gráfico 3.26, para la resolución del mismo, nótese la diferencia en las últimas restricciones, para llegar a estas primero se corrió el modelo matemático sin considerar la posibilidad de sub-tour como lo hicimos para el *cluster A*, en el reporte que nos arrojó LINDO la solución óptima incluía un sub-tour, es por ello que se agregó estas últimas restricciones para evitar este tipo de complicaciones. Finalmente se ingresó el modelo matemático al software LINDO el cual nos presenta el siguiente reporte:

```

F:\JOSE\TESIS\Tesis 2\CD_FINAL\Propuesta 5\Modelo matemático P5- LINDO\cluster 3_P2.lbx
MIN    12.42X022 + 15.13X023 + 14.27X024 + 9.70X038 + 10.19X039 + 18.78X041 +
        2.67X2223 + 1.92X2224 + 3.43X2238 + 2.72X2239 + 6.95X2241 +
        0.88X2324 + 5.66X2338 + 5.04X2339 + 4.39X3941 +
        4.94X2438 + 4.31X2439 + 5.15X2441 +
        0.70X3839 + 9.09X3841 +
        8.63X3941

S. T.

2) X022 + X023 + X024 + X038 + X039 + X041 = 2
3) X022 + X2223 + X2224 + X2238 + X2239 + X2241 = 2
4) X023 + X2223 + X2324 + X2338 + X2339 + X2341 = 2
5) X024 + X2224 + X2324 + X2438 + X2439 + X2441 = 2
6) X038 + X2238 + X2338 + X2438 + X3839 + X3841 = 2
7) X039 + X2239 + X2339 + X2439 + X3839 + X3941 = 2
8) X041 + X2241 + X2341 + X2441 + X3841 + X3941 = 2
9) X022 + X023 + X024 + X041 + X2238 + X2338 + X2438 + X3841 + X2239 + X2339 + X2439 + X3941 >= 1
10) X023 + X024 + X041 + X2223 + X2224 + X2241 + X2338 + X2438 + X3841 + X2339 + X2439 + X3941 >= 1

END

int 21
    
```

Gráfico 3.26. Programación lineal cluster 3 - LINDO

SOLUCIÓN CLUSTER 3: Bloc de notas

Archivo Edición Formato Ver Ayuda

LAST INTEGER SOLUTION IS THE BEST FOUND
RE-INSTALLING BEST SOLUTION...

OBJECTIVE FUNCTION VALUE

1) 34.50000

VARIABLE	VALUE	REDUCED COST
X022	0.000000	12.420000
X023	0.000000	15.130000
X024	0.000000	14.270000
X038	1.000000	9.700000
X039	1.000000	10.190000
X041	0.000000	18.780001
X2223	0.000000	2.670000
X2224	1.000000	1.920000
X2238	0.000000	3.430000
X2239	1.000000	2.720000
X2241	0.000000	6.950000
X2324	1.000000	0.880000
X2338	0.000000	5.660000
X2339	0.000000	5.040000
X3941	0.000000	13.020000
X2438	0.000000	4.940000
X2439	0.000000	4.310000
X2441	0.000000	5.150000
X3839	0.000000	0.700000
X3841	1.000000	9.090000
X2341	1.000000	0.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	0.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	0.000000	0.000000
6)	0.000000	0.000000
7)	0.000000	0.000000
8)	0.000000	0.000000
9)	1.000000	0.000000
10)	1.000000	0.000000

NO. ITERATIONS= 127
BRANCHES= 8 DETERM. = 1.000E 0

Gráfico 3.27. Reporte cluster 3 - LINDO

La interpretación del reporte nos indica que la ruta óptima para el *cluster* 3, según el modelo de la programación lineal, es la siguiente:

O-39-22-24-23-41-38-O, recorriendo una distancia de **34.50 Km**.

El mismo procedimiento es seguido para los demás clusters para la propuesta 2 (método del barrido), lo que finalmente nos lleva a la siguiente tabla de resultados:

Tabla 3-8. Resultados Propuesta: Método del Barrido

CLUSTERS	DISTANCIA KM
CLUSTER 1: O-28-29-30-42-31-32-O	56.42
CLUSTER 2: O-34-33-40-37-36-35-O	34.57
CLUSTER 3: O-39-22-24-23-41-38-O	34.5
CLUSTER 4: O-1-19-26-25-21-20-O	26.21
CLUSTER 5: O-8-17-18-15-27-9-O	25.4
CLUSTER 6: O-4-6-12-13-14-16-O	32.07
CLUSTER 7: O-2-7-10-11-3-5-O	28.72
TOTAL	237.89

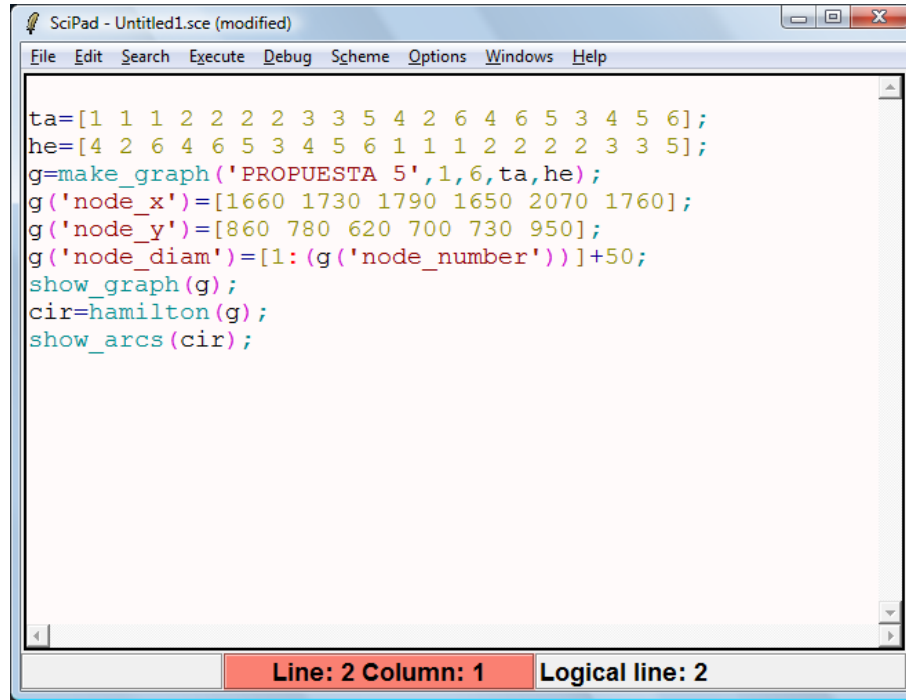
Elaboración propia

Más adelante en la evaluación de resultados analizaremos las todas las propuestas con las que se llegara a escoger la mejor de ellas.

3.2.6. Propuesta 6: Modelo Computacional

Finalmente, tenemos la propuesta computacional, para la cual también se buscará resolver el problema de ruteo de vehículos, pero esta vez mediante el software Scilab en la versión 4.1. Este software nos proporciona una serie de herramientas para la resolución de problemas tipo VRP; sin embargo, se ha escogido la función de Hamilton, lo que hace la función es encontrar un circuito hamiltoneano (si es que existe) de un gráfico dirigido g que para nuestro caso está conformado por las 42 tiendas.

Los *input* necesario para compilar el programa son las coordenadas de los 42 puntos más el almacén, además los posibles arcos que existe como variable de unión. En el anexo electrónico “Coordenadas” dentro de la carpeta “Propuesta 5” observamos el cálculo realizado para obtener los datos de entradas del programa. En el gráfico 3.26 se muestra cómo es que los datos deben ser ingresados al programa para que este los ejecute.



```

SciPad - Untitled1.sce (modified)
File Edit Search Execute Debug Scheme Options Windows Help

ta=[1 1 1 2 2 2 2 3 3 5 4 2 6 4 6 5 3 4 5 6];
he=[4 2 6 4 6 5 3 4 5 6 1 1 1 2 2 2 2 3 3 5];
g=make_graph('PROPUESTA 5',1,6,ta,he);
g('node_x')=[1660 1730 1790 1650 2070 1760];
g('node_y')=[860 780 620 700 730 950];
g('node_diam')=[1:(g('node_number'))]+50;
show_graph(g);
cir=hamilton(g);
show_arcs(cir);

Line: 2 Column: 1 Logical line: 2
  
```

Gráfico 3.28. Ventana de ingreso de datos

Fuente: Scilab 4.1

Los datos ingresados corresponden a los 5 primeros puntos y el almacén, entonces con esta información cuando se ejecuta el programa nos arroja el siguiente resultado:

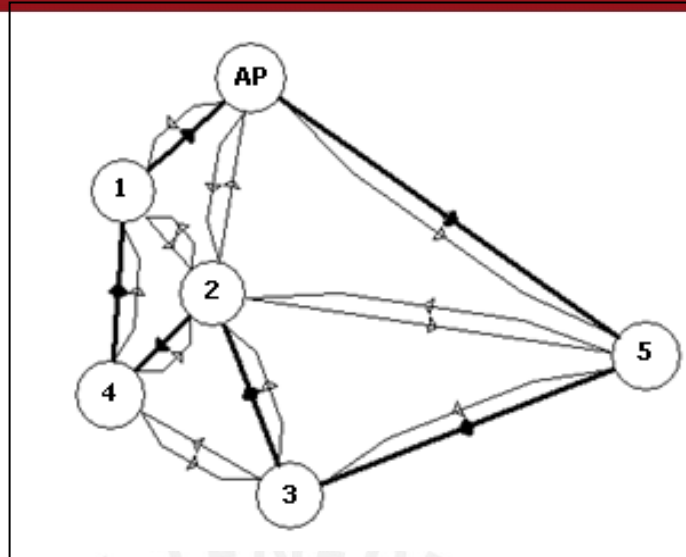


Gráfico 3.29. Resultado del método Hamiltoniano

Fuente: Scilab 4.1

En el gráfico 3.27 apreciamos el resultado obtenido a través del Scilab 4.1, en cual podemos identificar que la ruta propuesta por el software sería la siguiente: **O-5-3-2-4-1-O**.

Finalmente, en el anexo 2 podemos observar el resultado final del modelo computacional para las 42 tiendas en total en donde apreciamos el circuito hamiltoniano propuesta.

CAPÍTULO 4. EVALUACIÓN DE RESULTADOS

En el presente capítulo se realizará la evaluación de todos los resultados obtenidos a partir de las propuestas planteadas en el capítulo anterior. En primer lugar, comenzaremos determinar el factor de linealidad, el cual nos indicará la proporción existente entre la distancia ideal, ya calculada para cada una de las propuestas, y la distancia real. Luego realizaremos el análisis de dimensionamiento real de las propuestas mostrando algunos ejemplos con las rutas reales que seguiría el transportista para llegar a las tiendas de un *cluster* determinado. Además, compararemos las propuestas mostrando las diferencias y similitudes encontradas y, finalmente, seleccionaremos la mejor propuesta.

4.1 Factor de Linealidad

En la presente investigación hemos utilizado o nos hemos basado específicamente en las distancias ideales; es decir, no se han tomado en cuenta las calles por donde pasaría el transporte ni el sentido de estas; por tanto, no hemos considerado las distancias reales necesarias para llegar de una tienda a otra, es aquí donde entra a tallar lo que llamamos el factor de linealidad.

El factor de linealidad se puede definir como el coeficiente por el cual es multiplicada la distancia ideal para obtener la distancia real que el transporte recorrería si considera las calles y el sentido de la ruta propuesta. Entonces el factor se halla de la siguiente manera:

$$\text{factor de linealidad} = \frac{d_{\text{real}}}{d_{\text{ideal}}}$$

En donde tenemos:

d_{real} : Distancia real recorrida por la ruta propuesta

d_{ideal} : Distancia ideal recorrida por la ruta propuesta

El objetivo principal del el cálculo del factor de linealidad es obtener un valor aproximado de la distancia real para poder medir cierto indicadores como el combustible por *cluster* o el tiempo total de abastecimiento, bajo el supuesto de una velocidad constante del transporte.

A continuación presentamos en la tabla 4-1 el cuadro resumen de los cálculos del factor de linealidad:

Tabla 4-1. Cuadro resumen del factor de linealidad

RESUMEN	Media	Desviación Estándar
Propuesta 1	1.407	0.146
Propuesta 2	1.410	0.171
Propuesta 3	1.330	0.199
Propuesta 4	1.402	0.155
Propuesta 5	1.398	0.051
Propuesta 6	1.351	0.057
Promedio Total	1.383	0.130

Elaboración propia

En conclusión, para dar un ejemplo podríamos decir si es que la distancia ideal de un punto a otro es 10 Km, la distancia tomando en cuenta las calles o carreteras sería aproximadamente 13.83 Km. con una variación de 130 m.

El detalle de todos los cálculos hechos para llegar a los resultados planteados en la tabla 4-1 se pueden apreciar en el anexo electrónico “Factor de linealidad”.

4.2 Dimensionamiento Real

Como mencionamos en el capítulo anterior, obtener las distancias reales es un factor clave, ya que sin estas no podríamos elegir la mejor propuesta, además los modelos planteados no se ajustarían a la realidad, y por ende los indicadores de medición serían erróneos o fuera de un proceso real.

El dimensionamiento real lo hemos hallado con la ayuda de una herramienta de Google: *Google Maps*, la cual nos brinda la opción de unir dos o más puntos tomando en cuenta las calles y el sentido de las mismas y como resultado nos proporciona la distancia de una ruta en kilómetros.

En el punto anterior mostramos el cuadro resumen para cada una de las propuestas, ahora presentaremos un ejemplo de cada propuesta, para su realización escogeremos un *cluster* de cada uno.

En el gráfico 4.1 observamos la ruta real para el *cluster A* de la propuesta por cercanía de puntos, es decir, las calles por donde pasaría el transporte, la cual tiene una distancia aproximada de **23.5 km**:

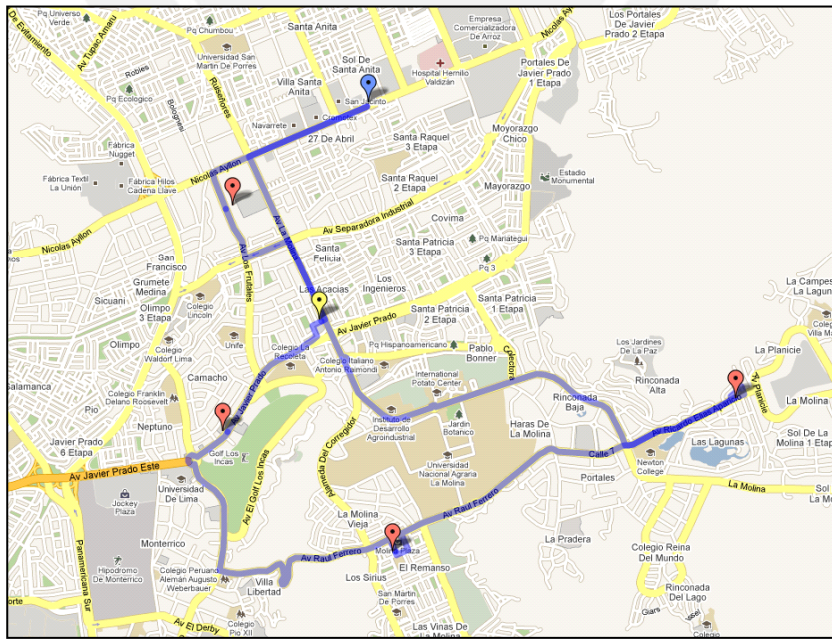


Gráfico 4.1. Ruta real *cluster A* – Propuesta 1
Fuente: Google Maps (2011)

En el gráfico 4.2 observamos la ruta real para el *cluster 5* de la propuesta del método del barrido:

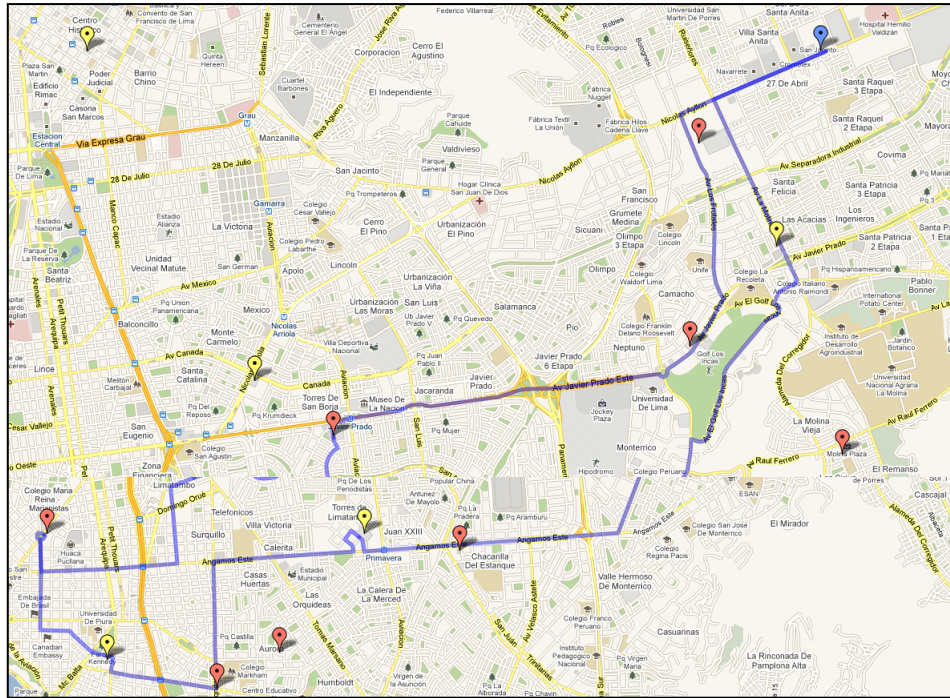


Gráfico 4.2. Ruta real *cluster 5* – Propuesta 2
Fuente: Google Maps (2011)

En el gráfico 4.3 observamos la ruta real para el *cluster 4* de la propuesta de rutear primero y asignar después:

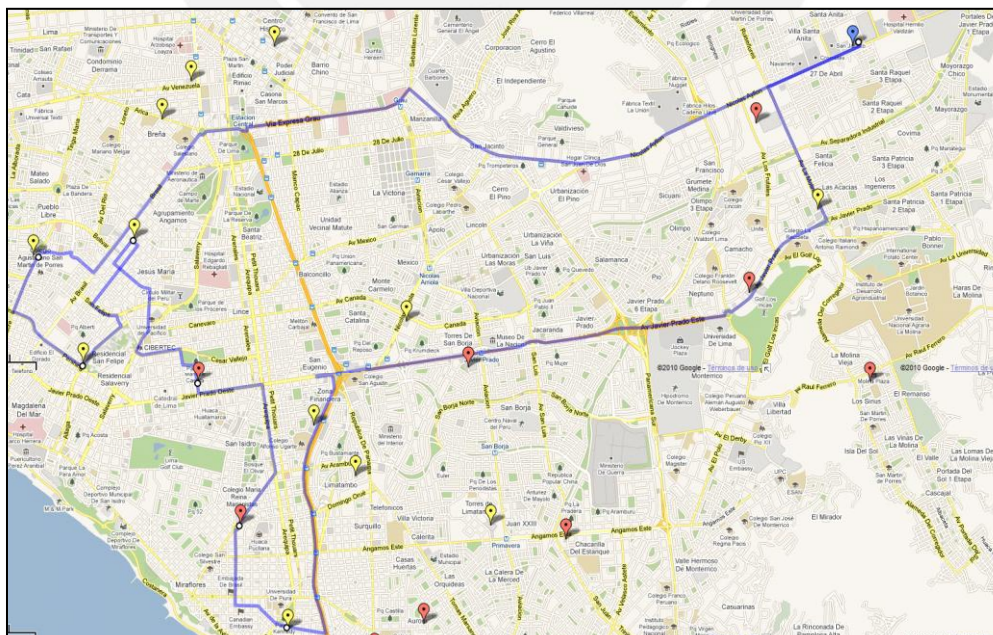


Gráfico 4.3. Ruta real *cluster 4* – Propuesta 3
Fuente: Google Maps (2011)

En el gráfico 4.4 observamos la ruta real para el *cluster C* de la propuesta del modelo matemático:

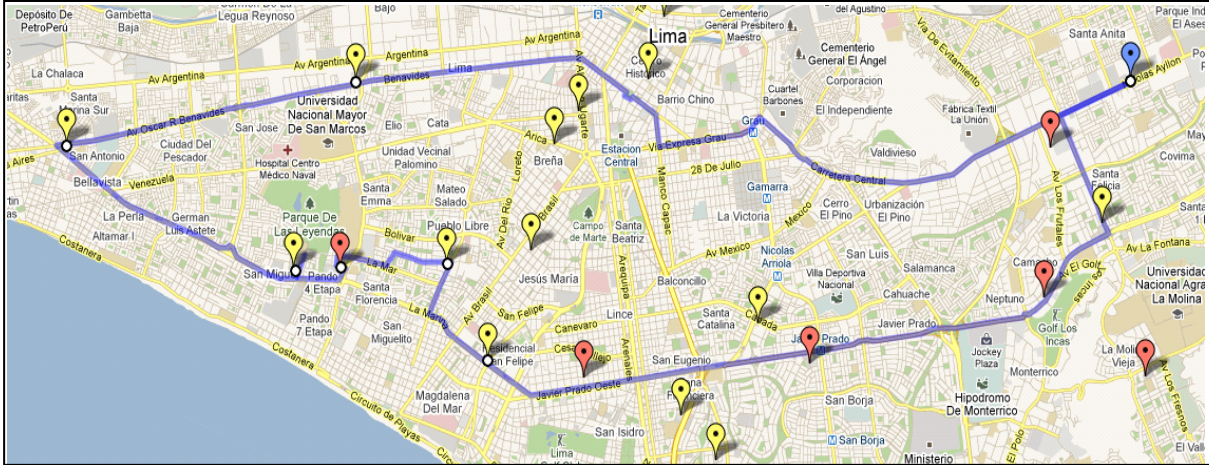


Gráfico 4.4. Ruta real *cluster C* – Propuesta 4
Fuente: Google Maps (2011)

En el gráfico 4.5 observamos la ruta real para el *cluster 3* de la propuesta del modelo computacional desarrollado con circuitos hamiltonianos:

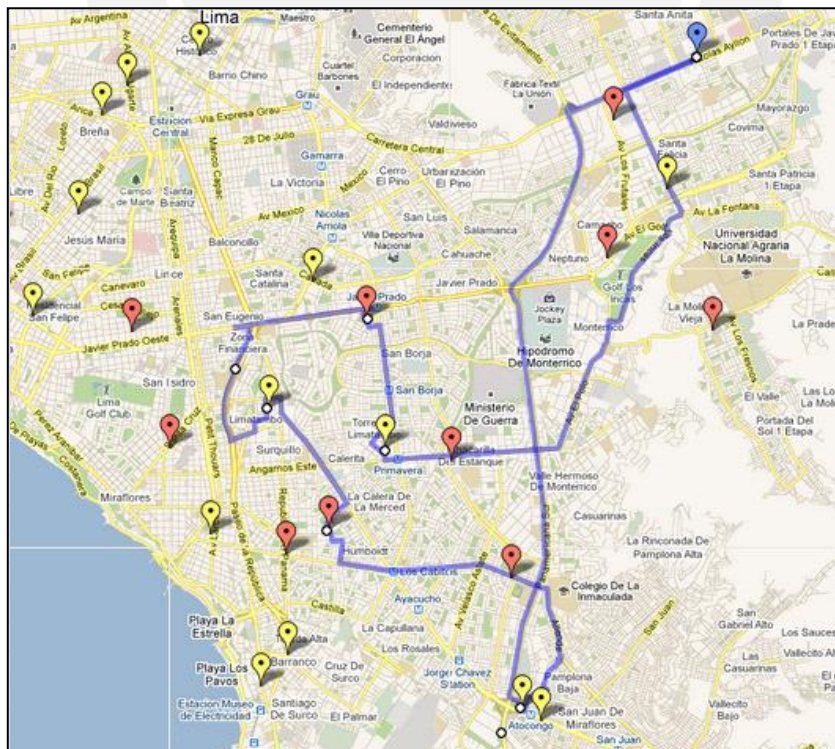


Gráfico 4.5. Ruta real *cluster 3* – Propuesta 5
Fuente: Google Maps (2011)

Ahora presentamos el cuadro resumen final de las distancias reales para cada propuesta, la tabla 4-2 muestra el total de distancia (Km) recorrida por el transporte.

Tabla 4-2. Cuadro resumen del dimensionamiento real

PROPUESTA	DISTANCIA KM
PROPUESTA 1	345.30
PROPUESTA 2	355.40
PROPUESTA 3	362.59
PROPUESTA 4	333.70
PROPUESTA 5	334.54
PROPUESTA 6	388.70

Elaboración propia

Finalmente, en el anexo electrónico “Evaluación de Resultados” podemos encontrar el detalle de las distancias para cada propuesta, en este archivo se encontrará la distancia real de cada cluster.

4.3 Diferencias y similitudes entre modelos

Las propuestas planteadas en el presente estudio tienen diferencias y similitudes muy precisas, ya que el problema de ruteo de vehículos para nuestro caso está dividido en tres partes: rutear, asignar, y el método de solución. También debemos analizar el orden en que se va a rutear y asignar, por dicha razón empezaremos a encontrar algunas similitudes y diferencias mostradas al momento de plantear los modelos de solución.

Primero Asignar:

Las propuestas por cercanía de puntos (1), el método del barrido (2) y la del modelo matemático (4), en primer lugar armar los *clusters* para luego llegar una solución con un método determinado, el cual analizaremos más adelante.

Primero Rutear:

Las propuestas de la gran ruta (3) y el modelo computacional (5) plantean primero encontrar la ruta óptima para luego pasar a la formación de *clusters* basándose en su ruteo inicial.

Método de solución.

Las propuestas por cercanía de puntos, método del barrido y la gran ruta plantean solución el problema por el método de ahorros, mientras que el modelo matemático plantea justamente una programación lineal para llegar a la solución óptima. Además, el modelo computacional propone llegar a un resultado mediante la utilización de un software.

A continuación en la tabla 4-3 observamos una breve descripción de cada una de las propuestas, la cual nos especifica claramente en que se basa cada modelo:

Tabla 4-3. Descripción de Propuestas

Propuesta	Descripción
PROPUESTA 4	Modelo matemático - Cercanía de pts
PROPUESTA 5	Modelo matemático - Método del barrido
PROPUESTA 1	Cercanía de puntos - Método del ahorro
PROPUESTA 2	Método del barrido - Método del ahorro
PROPUESTA 3	Gran ruta - Método del ahorro
PROPUESTA 6	Modelo computacional

Elaboración propia

4.4 Evaluación cuantitativa de las propuestas

En la evaluación cuantitativa de los modelos presentaremos las distancias ideales para cada una, y luego analizaremos si es que existe alguna diferencia con las distancias reales que expondremos al final.

Las distancias ideales como mencionamos no toman en cuenta las calles ni las carreteras ni el sentido de estas, por ende resultan ser menores que las distancias reales, es por ello que previamente hallamos el factor de linealidad. En la tabla 4-4 observamos la distancia total ideal recorrida por cada modelo, además cabe resaltar que la para distancias ideales la propuesta 4.1 es la que menor distancia total recorre con **233.01 Km.**

Tabla 4-4. Cuadro resumen de distancias ideales

PROPUESTA	DISTANCIA KM
PROPUESTA 1	239.95
PROPUESTA 2	247.21
PROPUESTA 3	267.43
PROPUESTA 4	233.01
PROPUESTA 5	237.89
PROPUESTA 6	286.74

Elaboración propia

Ahora realizaremos la comparación con la tabla 4-2 en donde presentamos las distancias reales recorridas por cada propuesta, para esto analizaremos las diferencias que existen entre las distancias real e ideales, así como la variación en porcentaje de incremento.

Tabla 4-5. Evaluación Cuantitativa de Propuestas

PROPUESTA	IDEAL KM	REAL KM	DIFERENCIA KM	VAR %
PROPUESTA 1	239.95	345.30	105.35	43.90%
PROPUESTA 2	247.21	355.40	108.19	43.76%
PROPUESTA 3	267.43	362.59	95.16	35.58%
PROPUESTA 4	233.01	333.70	100.69	43.21%
PROPUESTA 5	237.89	334.54	96.65	40.63%
PROPUESTA 6	286.74	388.70	101.96	35.56%

Elaboración propia

En la tabla 4-5 podemos observar algunos detalles resaltantes:

- El que mayor incremento tiene cuando se calcula la distancia real es la propuesta 1 (por cercanía de puntos). A pesar de ello no es la mayor distancia total recorrida.
- El que menos incremento tiene cuando se calcula la distancia real es la propuesta 3 (Gran ruta). Sin embargo no es la menor distancia total recorrida.

4.5 Selección del mejor modelo

Finalmente llegamos a la selección del mejor modelo, para ello hemos realizado el análisis cuantitativo previo, el cual nos arrojó el siguiente resultado:

Tabla 4-6. Resultados finales de los modelos

Puesto	Propuesta	Descripción	Km Total
1	PROPUESTA 4	Modelo matemático - Cercanía de ptos	333.70
2	PROPUESTA 5	Modelo matemático - Método del barrido	334.54
3	PROPUESTA 1	Cercanía de puntos - Método del ahorro	345.30
4	PROPUESTA 2	Método del barrido - Método del ahorro	355.40
5	PROPUESTA 3	Gran ruta - Método del ahorro	362.59
6	PROPUESTA 6	Modelo computacional	388.70

Elaboración propia

La tabla 4-6 nos muestra que la mejor propuesta es la del **modelo matemático**, realizado a partir de una formación de *cluster por cercanía de puntos (propuesta 4)*, el cual recorre una distancia real de **333.70 Km**, siendo así la menor distancia y por ende para nuestro estudio la óptima.

CAPÍTULO 5. CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- El mejor modelo para nuestro estudio y que debería ser elegido es la propuesta 4, la cual realiza la formación de *clusters* a través de la cercanía de los puntos y resuelve los mismos mediante una programación lineal aplicada que en este caso fue la del "Agente Viajero", es elegida debido a que recorre la menor distancia no sólo ideal sino también real, lo cual lo hace nuestra mejor opción. Las distancias ideales y reales fueron de **239.95** y **345.30** kilómetros respectivamente.
- La selección de la mejor propuesta se realiza a partir de las distancias reales recorridas por el transporte para cada una de nuestras propuestas, es decir, se escoge la mejor propuesta en base a la menor cantidad de kilómetros tal que todos los transportes salgan del centro de distribución y se visiten todos los puntos analizados. Para esto se ha utilizado el factor de linealidad que según el análisis fue de **1.383** con una desviación estándar de **0.130**.
- El ruteo por las calles y las direcciones de las mismas influyen directamente en el resultado, ya que esto hace que la distancia total recorrida incremente o disminuya. Es por ello que la solución no permite que un vehículo vaya por una calle en contra del sentido original o que atraviese parques o divisiones que no estén asfaltadas.
- Todas las propuestas han sido planteadas según la capacidad de abastecimiento de la empresa de transporte, ya que esta puede como máximo abastecer a **06** tiendas, es por ello que los modelos poseen 06 puntos. Excepto para el *cluster* B formador por cercanía de puntos, debido a que según la experiencia de los transportistas en base a la demanda si se podrían atender todos los puntos que este *cluster* involucra. Esta restricción también se debe a las ventanas de tiempo de atención que poseen las tiendas y el tiempo que se demoran en atender los transportes.

5.2 Recomendaciones

- Sería útil calcular los tiempos totales, desde que el transporte llega al Almacén para la carga hasta que despacha la mercadería en la última tienda asignada, además de calcular los tiempos promedio de carga y descarga en las tiendas. Este un factor clave, ya que puede ayudar a aumentar o reducir los puntos máximos de visita por cada transporte, haciendo que el modelo se ajuste mucho más a la realidad.
- La resolución de un problema de ruteo de vehículos involucra muchas variables a tomar en cuenta, por tanto se debe marginar el estudio a tan sólo las más relevantes y a las cuales se tenga acceso, caso contrario el estudio sería poco práctico y engorroso. Para nuestro caso hemos tomado como variable fundamental la distancia recorrida real basada en un límite de puntos que se pueden visitar como máximo. Otro caso de estudio podría analizar en base a capacidades del medio de transporte como por ejemplo metros cúbicos o kilogramos.
- Con este tipo de estudio también denominado como Travelling Management System muchas empresas pequeñas podrían reducir costos operativos ya que la distribución de sus productos sería óptima, además la inversión necesaria para implementar este tipo de proyectos no es considerable.
- El análisis podría servir como base para la elaboración de un algoritmo necesario en la configuración de un software que sirva como mejora en el proceso de ruteo para una empresa que realiza su propia distribución o para una empresa que se desempeña como operador logístico.

REFERENCIAS BIBLIOGRÁFICAS

- RÍOS, Sixto
1996 “El modelo de la Asignación”. *Investigación operativa: programación lineal y aplicaciones*. Madrid: Centro de Estudios Ramón Areces, pp. 209-210.
- APPLGATE, David
2006 “Dantzig, Fulkerson, and Johnson”. *The traveling salesman problem: a computational study*. New Jersey: Princeton University Press. pp. 81 - 93.
- MARTELLO, S. y TOTH, P.
1990 Knapsack problems: algorithms and computer implementations. John Wiley and Sons
- CLARKE, G. y WRIGHT, W.:
1964 “Scheduling of vehicles from a central depot to a number of delivery points”. *Operations Research* 12. pp 568–581
- CARREÑO SOLÍS, Adolfo
2009 *Logística Industrial*. Material de enseñanza. Lima: Pontificia Universidad Católica del Perú, Sección Ingeniería Industrial
- OLIVERA, Alfredo.
2004 *Heurísticas para Problemas de Ruteo de Vehículos*. Montevideo: Instituto de Computación, Facultad de Ingeniería, Universidad de la República.
- DANTZIG G.B. y RAMSER J.H.
1959 *The Truck Dispatching Problem*, Management Science, vol. 6, pp 80-91.
- MUNICIPALIDAD DE LIMA
2009 *Plan de desvío vehicular*. s/l: Gobierno del Perú, Municipalidad de Lima. Consulta: 06 de abril de 2010.
<<http://www.munlima.gob.pe/Mapas/ListaMapa.aspx>>
- ARREOLA, J., ARREOLA, A.
2003 “Programación Lineal”. *Programación lineal: una introducción a la toma de decisiones*. México: Thomson.
- WINSTON, Wayne
2006 “Programación Lineal”. *Investigación de operaciones: aplicaciones y algoritmos*. México: Thomson.

- BALINSKI, M. y QUANDT, R.
1964 "On an Integer Program for a Delivery Problem". *Operations Research*. Vol. 12, pp. 300-304
- GILLET, B. y MILLER, L.
1974 "A Heuristic Algorithm for the Vehicle Despatch Problem". *Operations Research* 22, 340-9
- FISHER, M. y JAIKUMAR, R.
1981 A generalized assignment heuristic for the vehicle routing problem. *Networks* 11 pp. 109–124

