

*Proyecto de Grado*

---

Nuevas metaheurísticas  
basadas en la física para la  
resolución del TSP y del VRP

---

*Setiembre de 2003*

*Tutor:*

Ing. Omar Viera

*Estudiantes:*

Fabián Feijó

Rodrigo Suárez



Departamento de Investigación Operativa  
Instituto de Computación  
Facultad de Ingeniería  
Universidad de la República

<a href="#">Abstract</a> .....	4
<a href="#">1 Introducción</a> .....	5
<a href="#">2 Descripción del problema</a> .....	9
<a href="#">2.1 Definición</a> .....	9
<a href="#">2.2 A considerar en este proyecto</a> .....	14
<a href="#">3 Heurísticas y metaheurísticas conocidas</a> .....	16
<a href="#">3.1 Heurísticas Constructivas</a> .....	16
<a href="#">Heurísticas constructivas de Inserción</a> .....	17
<a href="#">3.2 Heurísticas de mejoramiento iterativo</a> .....	17
<a href="#">3.3 Heurísticas híbridas</a> .....	18
<a href="#">4 Propuestas para la resolución del TSP</a> .....	19
<a href="#">4.1 Centro de masas</a> .....	19
<a href="#">4.2 Heurísticas del centro de masas</a> .....	20
<a href="#">4.2.1 Criterios de selección del nodo inicial</a> .....	20
<a href="#">Criterios de inserción de nodos</a> .....	22
<a href="#">4.3 Criterios de selección de nodo y construcción de las heurísticas</a> .....	23
<a href="#">4.3.1 Selección uno: el más cercano al <math>CM_{NV}</math></a> .....	25
<a href="#">4.3.2 Selección dos: el más cercano al <math>CM_V</math></a> .....	28
<a href="#">4.3.3 Selección 3: el más cercano del <math>CM_V</math> y <math>CM_{NV}</math></a> .....	30
<a href="#">4.3.4 Selección 4: el más cercano al <math>CM_V</math> y <math>CM_{NV}</math></a> .....	33
<a href="#">4.3.5 Selección 5: el más cercano al punto medio entre <math>CM_V</math> y <math>CM_{NV}</math></a> .....	36
<a href="#">4.3.6 Selección 6: el más cercano al <math>CM_S</math></a> .....	39
<a href="#">4.3.7 Selección 7: el más lejano al <math>CM_{NV}</math></a> .....	41
<a href="#">4.3.8 Selección 8: el más lejano al <math>CM_V</math></a> .....	43
<a href="#">4.3.9 Selección 9: el más lejano al <math>CM_V</math> o <math>CM_{NV}</math></a> .....	46
<a href="#">4.3.10 Selección 10: el más lejano al <math>CM_V</math> y <math>CM_{NV}</math></a> .....	48
<a href="#">4.3.11 Selección 11: el más lejano al punto medio entre <math>CM_V</math> y <math>CM_{NV}</math></a> .....	52
<a href="#">4.3.12 Selección 12: el más lejano al <math>CM_S</math></a> .....	54
<a href="#">4.4 Análisis de resultados y conclusiones</a> .....	57
<a href="#">5 Propuestas para la resolución del VRP</a> .....	64
<a href="#">5.1 Metaheurísticas de fuerza de atracción</a> .....	64
<a href="#">5.1.1 Propuesta 1: Fuerza de atracción desde el depósito</a> .....	65
<a href="#">5.1.2 Propuesta 2: Fuerza de atracción desde cada nodo</a> .....	68
<a href="#">5.1.3 Propuesta 3: Fuerza de atracción desde el depósito "con vuelta"</a> .....	70
<a href="#">5.1.4 Propuesta 4: Fuerza de atracción desde cada nodo con "vuelta"</a> .....	75
<a href="#">5.1.5 Propuesta 5: Fuerza de atracción desde el primer nodo, variante "primero el de menor fuerza"</a> .....	77
<a href="#">5.1.6 Propuesta 6: Fuerza de atracción desde cada nodo, variante "primero el de menor fuerza"</a> .....	81
<a href="#">5.1.7 Propuesta 7: Fuerza de atracción desde el primer nodo con "vuelta", variante "primero el de menor fuerza"</a> .....	83
<a href="#">5.1.8 Propuesta 8: Fuerza de atracción desde cada nodo con "vuelta", variante "primero el de menor fuerza"</a> .....	86
<a href="#">5.2 Campo Eléctrico</a> .....	89
<a href="#">Análisis de resultados y conclusiones</a> .....	94
<a href="#">6 Verificación de los resultados</a> .....	99
<a href="#">6.1 Instancias usadas y condiciones de los experimentos</a> .....	99
<a href="#">7 Proceso de desarrollo</a> .....	101
<a href="#">7.1 Decisiones de herramientas</a> .....	101
<a href="#">7.2 Conocimiento de los Requerimientos</a> .....	101
<a href="#">Categorías de las funciones</a> .....	101
<a href="#">7.3 Funciones Básicas</a> .....	102

7.4	<a href="#">Casos de Uso: Descripción de procesos</a>	103
	<a href="#">Identificación de actores</a>	103
	<a href="#">Casos de Uso en alto nivel</a>	103
7.5	<a href="#">Diagramas de los casos de uso</a>	105
7.6	<a href="#">Casos de Uso expandidos</a>	105
7.7	<a href="#">Clasificación de los casos de uso</a>	110
7.8	<a href="#">Construcción del modelo conceptual</a>	111
	<a href="#">Conceptos</a>	111
	<a href="#">Modelo Conceptual</a>	112
7.9	<a href="#">Diagrama de Clases del Diseño</a>	113
	<a href="#">Diagrama</a>	113
7.10	<a href="#">Descripción e Implementación</a>	114
<b>8</b>	<a href="#">Conclusiones y trabajo futuro</a>	121
<b>9</b>	<a href="#">Apéndice: Estado del arte del TSP y VRP</a>	126
	<a href="#">Un poco de Historia</a>	127
	<a href="#">TSP</a>	129
	<a href="#">El Problema del Vendedor Viajero Múltiple</a>	141
	<a href="#">Problema de Ruteo de Vehículos</a>	142
<b>10</b>	<a href="#">Apéndice 2: Definiciones matemáticas</a>	148
	<a href="#">Aplicación a TSP</a>	150
<b>11</b>	<a href="#">Apéndice 3: Conceptos de física</a>	151
11.1	<a href="#">Estructura atómica y espectros atómicos</a>	151
11.2	<a href="#">Origen de las cargas eléctricas</a>	152
	<a href="#">Propiedades fundamentales de la carga:</a>	153
	<a href="#">Conductores, aisladores y semiconductores</a>	153
	<a href="#">El campo eléctrico</a>	157
11.3	<a href="#">La luz</a>	157
	<a href="#">Velocidad de la luz</a>	158
	<a href="#">Ondas y rayos</a>	159
	<a href="#">Glosario</a>	162
<b>12</b>	<a href="#">Apéndice 4: Manual de usuario Visual Tour</a>	164
12.1	<a href="#">Antes de comenzar</a>	164
12.2	<a href="#">Descripción General</a>	167
12.3	<a href="#">Operaciones</a>	169
<b>13</b>	<a href="#">Apéndice 5: Algoritmos Genéticos</a>	187
13.1	<a href="#">Introducción</a>	187
13.2	<a href="#">Algoritmos genéticos vs. evolutivos</a>	187
13.3	<a href="#">Descripción</a>	187
13.4	<a href="#">Aplicación de un Algoritmo genético</a>	188
13.4.1	<a href="#">Restricciones de dominio</a>	189
13.4.2	<a href="#">Equilibrio entre la exploración y la explotación de soluciones</a>	190
13.5	<a href="#">Aplicación de AG a TSP</a>	190
13.5.1	<a href="#">Construcción del algoritmo</a>	190
13.5.2	<a href="#">Restricciones de dominio</a>	191
<b>14</b>	<a href="#">Bibliografía</a>	193

## Abstract

El objetivo del proyecto es desarrollar nuevas metaheurísticas para la resolución del TSP (Travelling Salesman Problem, Problema del Vendedor Viajante) y del VRP (Vehicle Route Problem, Problema de Ruteo de Vehículos), basándonos en conceptos de la física y la microfísica y evaluar si los resultados ameritan una profundización en el tema.

Dado un conjunto de nodos, el TSP consiste en encontrar el camino de menor costo entre ellos de tal forma que todos sean visitados exactamente una vez. Para el VRP se considera que cada nodo tiene una demanda específica a satisfacer, que se cuenta con una flota de vehículos con capacidad de carga igual y acotada y que existe un nodo central. Bajo estas consideraciones, el VRP consiste en encontrar un conjunto de rutas de costo mínimo que, partiendo del nodo central, recorran todos los demás nodos (sin repetir ninguno) satisfaciendo las demandas correspondientes y teniendo en cuenta la capacidad de carga acotada de cada vehículo.

Para la resolución del TSP proponemos un conjunto de heurísticas utilizando el concepto de centro de masas y para la resolución del VRP proponemos metaheurísticas basadas en la fuerza electromagnética y en el campo eléctrico. Desarrollamos una herramienta, que llamamos Visual Tour para la ejecución y comparación de las técnicas conocidas estudiadas y las propuestas en este proyecto. Se obtuvieron resultados, del 11% en promedio y del 4% en el mejor caso por encima de la mejor solución conocida para la resolución del TSP. Y del 20% en promedio por encima de la mejor solución conocida para la resolución del VRP tomando en ambos casos como referencia la librería TSPLIB95 [TSPLIB].

En conclusión, recomendamos continuar desarrollando los estudios en lo que respecta a metaheurísticas tomando como base conceptos de la física. Hemos constatado que al considerar variantes en los criterios utilizados en las mismas se perciben considerables diferencias en los resultados obtenidos. Ahondando más en el análisis expuesto en este proyecto y considerando otros criterios derivados de los mismos creemos que se puede lograr mejorar aún más las soluciones obtenidas.

# 1 Introducción

Este trabajo fue propuesto como proyecto de grado para la carrera Ingeniería en Computación por el Departamento de Investigación Operativa. Este departamento pertenece al Instituto de Computación de la Facultad de Ingeniería de la Universidad de la República.

Distintos comportamientos de fenómenos de la naturaleza se han observado y estudiado con el objetivo de obtener ideas para la resolución de diversos problemas planteados en la historia de la ciencia. En particular, en el estudio del TSP (Travelling Salesman Problem, Problema del Vendedor Viajante) y del VRP (Vehicle Route Problem, Problema de Ruteo de Vehículos) se han desarrollado metaheurísticas basadas en el comportamiento de las hormigas [*TSPAC*] (ant colony), redes neuronales [*OPTRN*] y algoritmos genéticos [*IAG*] entre otros. La aplicación de dichas metaheurísticas a estos problemas ha sido exitosa. Esto significa que se han obtenido buenos resultados en comparación con las mejores soluciones conocidas, motivando la investigación en otros campos para la realización de nuevas metaheurísticas. El objetivo de este proyecto nace de la idea de explorar el campo de la física y de la microfísica para obtener ideas y tomar conceptos en los cuales se basen nuevas metaheurísticas para la resolución de los problemas mencionados. También poder evaluar los resultados obtenidos para descartar o profundizar en los métodos desarrollados.

Para describir en mayor detalle el objetivo de este proyecto definiremos lo que es una heurística y una metaheurística. Una heurística es un algoritmo que, a un bajo costo computacional en cuanto a la eficiencia, busca encontrar con alta probabilidad una solución cercana a la óptima. Una metaheurística es una heurística que basa su lógica en reglas o comportamientos observados en la naturaleza. La utilización de metaheurísticas para la resolución de los problemas planteados ha dado buenos resultados consiguiendo igualar e inclusive, en algunos casos, superar las soluciones provistas por algoritmos conocidos como por ejemplo en el caso de la metaheurística de la colonia de hormigas [*ACO*] con la cual se ha logrado superar las mejores soluciones conocidas planteadas por la TSPLIB. El TSP es un problema de naturaleza NP-completo [*DHMCE*] por lo que la resolución del mismo por un algoritmo exacto tiene la característica de necesitar un tiempo de ejecución que crece exponencialmente con el crecimiento lineal de la entrada (cantidad de ciudades en este caso). Con la utilización de metaheurísticas para la resolución del TSP, no solo se ha logrado obtener buenas soluciones, si no también una ejecución eficiente de los programas que las implementan. Es por esto, y por la facilidad de comprensión de la lógica planteada en las metaheurísticas (ya que se hace una analogía con algún fenómeno de la naturaleza) que se han difundido ampliamente. Los logros obtenidos con la utilización de diferentes metaheurísticas en muchos campos de la ciencia han hecho de esta área de investigación un campo interesante para plantearse como alternativa de estudio en la resolución de diferentes problemas. Si bien para la resolución del TSP y del VRP ya se han desarrollado distintas metaheurísticas, en este proyecto se plantea explorar otra área de la naturaleza (la física y la microfísica) para tomar ideas y desarrollar nuevas metaheurísticas. De los conceptos estudiados y analizados hemos utilizado los de centro de masas, fuerza electromagnética y campo magnético [*FIS1*] [*FIS2*]. Estos conceptos se adecuan al problema a resolver al plantearnos una analogía de los nodos como partículas de igual masa para la resolución del TSP y de partículas con diferente

carga para la resolución del VRP. La utilización de la idea del centro de masas se basa en la distribución de los nodos en el espacio y la analogía de la diferencia de carga se plantea con respecto a las diferentes demandas de cada nodo.

Dado un grafo, (por sus nodos, aristas, y costos asociados a cada arista) el TSP busca determinar un ciclo de costo mínimo que pase por todos los nodos del grafo, una sola vez. Visto mas formalmente, sea un grafo  $G=[N,A,C]$  donde  $N$  es el conjunto de nodos,  $A$  el conjunto de aristas y  $C=[c_{ij}]$  la matriz de costos, donde  $c_{ij}$  es el costo de desplazarse o la distancia que existe entre el nodo  $i$  y el nodo  $j$ , el TSP busca determinar el ciclo hamiltoniano en  $G$  de menor costo [MAT]. Se ha demostrado que este problema es de tipo NP-completo [DHMCE]. Esto implica que no existe un algoritmo exacto para el TSP que lo resuelva en orden polinomial. Es por esto que se busca resolverlo, no por un algoritmo exacto, sino utilizando una heurística o metaheurística que, en un tiempo razonable, obtenga cómo resultado una buena aproximación a la solución óptima.

Si se considera que cada nodo tiene una demanda específica a satisfacer, que se cuenta con una flota de vehículos con capacidad de carga acotada e igual para todos y que existe un nodo central, el VRP consiste en encontrar un conjunto de rutas de costo mínimo que, partiendo del nodo central, recorran todos los demás nodos (sin repetir ninguno) satisfaciendo las demandas correspondientes y teniendo en cuenta la capacidad de carga acotada de cada vehículo. Visto mas formalmente, dados un grafo  $G=[N,A,C]$ ,  $n_c$ , y  $K$ , donde  $N$  es el conjunto de nodos,  $n_c \in N$  es el nodo central,  $A$  el conjunto de aristas,  $C=[c_{ij}]$  la matriz de costos, donde  $c_{ij}$  es el costo de desplazarse o la distancia que existe entre los nodos  $i, j$ , y  $K$  la capacidad máxima de carga de los vehículos, el VRP busca determinar un conjunto de ciclos de costo mínimo que pasan por  $n_c$  y que cumplan con las siguientes restricciones:

- Todos los nodos (menos  $n_c$ ) son visitados exactamente una vez.
- La suma de las solicitudes de los nodos de cada ciclo no excede  $K$ .

Este problema, al igual que el TSP, es de naturaleza NP-completo [VRPNP], por lo que interesa encontrar métodos (metaheurísticas en este caso) que lo resuelvan en un tiempo razonable.

Si bien nos fue planteado no estudiar en profundidad métodos conocidos de resolución de los problemas para que esto no influyera en la creación de otros nuevos fue necesario estudiar algunos algoritmos, heurísticas y metaheurísticas ya desarrolladas para adentrarnos en el tema. Formular al menos un algoritmo exacto que brinde el resultado óptimo para estos problemas no es difícil ya que son problemas de búsqueda en un espacio de soluciones, y como tales admiten la resolución mediante un algoritmo enumerativo. Un algoritmo enumerativo recorre todas las posibles soluciones recordando cual fue el óptimo encontrado hasta el momento. Una vez finalizada la búsqueda, el mejor encontrado hasta el momento es el óptimo del problema. Como se mencionó antes, este algoritmo funciona, pero el orden de ejecución crece exponencialmente con la cantidad de nodos considerados. Para ejemplificar esto, basta con mencionar que en un grafo de 10 nodos existen 181.440 ciclos diferentes (si consideramos las aristas con el mismo costo hacia ambos lados, sino, existen 362.880) y en un grafo de 20 nodos existen más de  $6 \times 10^{16}$  circuitos diferentes. Se estudiaron tres clases de heurísticas: constructivas (vecino mas cercano, ahorro de Clark y Wright [GCWW] y de inserción), de mejoramiento iterativo (intercambio 2-opt [RSV] e híbridas (triangulación Delanauay, OCIDT) [TDEL]. También se estudiaron las metaheurísticas de ant colony y el conjunto de metaheurísticas "algoritmos genéticos".

Para la resolución del TSP y del VRP desarrollamos distintas heurísticas y metaheurísticas. Utilizando el concepto de centro de masas proponemos un conjunto de heurísticas constructivas para la resolución del TSP, que consideran distintos criterios de selección e inserción. Utilizando el concepto de fuerza electromagnética de las cargas eléctricas y considerando cada nodo como una carga de valor proporcional a su demanda se propuso un conjunto de metaheurísticas para la resolución del VRP. La analogía utilizada plantea que la fuerza de atracción experimentada por el vehículo sea proporcional a la demanda de los nodos e inversamente proporcional a la distancia que debe recorrer para llegar a ellos. Utilizando el concepto de campo magnético se propone una metaheurística para la resolución del VRP que utiliza la dirección del vector campo magnético para el criterio de selección.

Para verificar los resultados obtenidos, estudiar comportamientos y poder realizar comparaciones de las técnicas, se desarrollo la herramienta Visual Tour en la cual programamos algunos de los algoritmos conocidos y todos los propuestos por nosotros.

Los resultados obtenidos por las metaheurísticas propuestas las comparamos con:

- Los resultados obtenidos por heurísticas conocidas, estudiadas e implementadas.
- Las mejores soluciones conocidas hasta el momento, brindadas por la TSPLIB 95. [TSPLIB]

Hemos podido ver que el estudio de distintos fenómenos de la naturaleza (la física en este caso) es una guía útil en la creación de técnicas para resolver el TSP y el VRP mediante metaheurísticas. Obtuvimos diversos resultados al implementar los heurísticas y metaheurísticas propuestas, y si bien algunos no han dado los resultados esperados, otros sí al compararlos con la mejor solución expuesta en la librería de soluciones utilizada de referencia [TSPLIB]. La herramienta implementada Visual Tour es útil para la comparación de diferentes algoritmos y por la forma en que está diseñada es fácilmente escalable para agregar otros módulos de resolución. A nivel teórico y académico concluimos que el estudio de estos problemas es muy amplio, y que dentro de las heurísticas y metaheurísticas exploramos un área que, hasta ahora, no había sido explorada, o en caso de haber sido estudiada no ha sido difundida. Habiendo obtenido un promedio del 11% sobre la mejor solución conocida para el TSP y del 20% sobre la mejor solución conocida para el VRP creemos que vale la pena seguir profundizando en el área incurrida. En particular planteamos como trabajo futuro seguir refinando los criterios utilizados en las metaheurísticas propuestas con el objetivo de disminuir la varianza observada en los resultados obtenidos al contrastarlos con las soluciones de referencia. [TSPLIB]

Este informe está compuesto por 8 capítulos, que exponen en mayor grado de profundidad que lo expuesto hasta ahora, los diferentes aspectos cubiertos en el proyecto. Adjuntos a este informe, diferentes documentos amplían los temas estudiados en él, así como también brindan una extensión de otros aspectos que no comprenden el objetivo principal del proyecto.

En el Capítulo 2 se definen en profundidad los problemas de TSP y VRP, se detallan las distintas variantes de estos problemas y se especifica a cuales en especial nos hemos abocado en este trabajo. Esto define el marco de trabajo del proyecto, los requerimientos y los objetivos del mismo.

En el Capítulo 3 exponemos las heurísticas y metaheurísticas estudiadas para la resolución del TSP y VRP.

En el Capítulo 4 se exponen las heurísticas propuestas para la resolución del TSP, así como los resultados obtenidos al implementarlas, la comparación entre ellas y el análisis de resultados.

En el Capítulo 5 se exponen las metaheurísticas propuestas para la resolución del VRP, así como los resultados obtenidos al implementarlas, la comparación entre ellas y el análisis de resultados.

En el Capítulo 6 se describen los casos de prueba utilizados a si como la herramienta de verificación. También se especifica la plataforma en la cual se ejecutaron las pruebas.

En el Capítulo 7 se presenta un análisis funcional de los requerimientos para la implementación de la herramienta que nos permite representar el problema planteado y ejecutar las distintas soluciones. Esta herramienta tiene como objetivo poder testear y comparar las distintas técnicas propuestas con otras conocidas, ya sea a nivel de algoritmo en tiempo de ejecución o de la solución obtenida.

En el Capítulo 8 se presentan las conclusiones generales, tanto a nivel académico como a nivel del proyecto. Los distintos aspectos del proyecto a nivel teórico (las heurísticas), a nivel práctico (la herramienta implementada) y a nivel de objetivos logrados como proyecto de grado son presentados y analizados. También se presenta el trabajo futuro, que tomando como base lo estudiado se puede continuar para ampliar tanto a nivel teórico como a nivel de implementación. Otras áreas de la física y microfísica que se pueden estudiar para proponer meta heurísticas que resuelvan los problemas planteados y distintas variantes de estos.

Al final del informe se presentan los siguientes apéndices:

- Apéndice 1: Estado del arte TSP y VRP.
- Apéndice 2: Definiciones matemáticas.
- Apéndice 3: Conceptos de física.
- Apéndice 4: Manual de usuario Visual Tour.
- Apéndice 5: Algoritmos genéticos.
- Apéndice 6: Bibliografía.



## 2 Descripción del problema

### 2.1 Definición

#### TSP – Travelling Salesman Problem

El TSP consiste en que una persona debe visitar un conjunto de  $n$  ciudades, comenzando en una ciudad específica (elegida como origen), debiendo regresar a ella, luego de haber visitado todas las demás exactamente una vez. El objetivo es minimizar la longitud de la secuencia de visita de las ciudades. El costo de ir de una ciudad a otra está dado por un función objetivo.

Si los costos de las aristas son simétricos, o sea que el costo de viajar entre dos nodos no depende de la dirección del viaje, entonces el TSP es simétrico; de otra forma es asimétrico o dirigido.

La importancia del TSP dentro del mundo de problemas NP radica en que es utilizado como conjunto de pruebas para los nuevos algoritmos asociados a la resolución de esta clase de problemas. Su naturaleza NP Completo hace posible que se puedan resolver una gran familia de problemas equivalentes mediante transformaciones particulares para cada caso. Además, como el TSP presenta una gran cantidad de aplicaciones en el mundo real, se ha convertido en uno de los problemas más estudiados por la comunidad científica mundial.

Los métodos o estrategias óptimas para el TSP se basan en formulaciones de programación matemática. Por simplicidad en la siguiente formulación se asume que los costos son simétricos (o sea  $c_{ij} = c_{ji}$ ) y vamos a definir a  $c_{ii} = 0$  para  $i = 1, 2, \dots, n$ . El problema consiste en determinar un tour que pase por todos los nodos comenzando y finalizando en el nodo origen, el cual vamos a definir como el nodo 1, de tal manera que se recorra la mínima distancia o el menor costo. Se define entonces,

$$x_{ij} = \begin{cases} 1 & \text{si } \text{arc } \bar{ij} \text{ esta en el tour} \\ 0 & \text{si no esta en el tour} \end{cases}$$

Se puede encontrar un desarrollo mas profundo de esta formulación matemática en el Apéndice "Estado del Arte" y en [RSV] páginas 83 y 84.

Para formular el hecho de que solo debe haber un camino que entre y uno que salga por cada nodo, se presenta una formulación basada en la asignación, para ello se considera la matriz de decisión  $X = (x_{ij})$  de tal manera que solo un arco  $(i, j)$  salga del nodo  $i$ , y que exactamente un arco entre al nodo  $j$ . Esto implica una asignación para cada nodo con su respectivo nodo sucesor en el tour. Este requerimiento de asignación por si solo no determina que la matriz  $X$  corresponda a un tour, puesto que las asignaciones pueden resultar en subtours.

Para eliminar la posibilidad de que se formen subtours, se le impone algunas restricciones más para la selección de arcos de la matriz X. Por lo tanto el problema

planteado se formula de la siguiente manera: Minimizar  $\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

$$\text{Sujeto a } \left\{ \begin{array}{l} \sum_{i=1}^n x_{ij} = b_j = 1 \quad (j = 1, \dots, n) \quad \{\text{Implica que a cada nodo solo llegue un arco}\} \\ \sum_{j=1}^n x_{ij} = a_i = 1 \quad (i = 1, \dots, n) \quad \{\text{Implica que de cada nodo solo salga un arco}\} \\ X = (x_{ij}) \in S \\ x_{ij} = 0 \text{ o } 1 \quad (i, j = 1, \dots, n) \end{array} \right.$$

El conjunto S puede contener cualquier restricción que prohíba que soluciones con subtours satisfagan las restricciones previamente definidas. Estas restricciones son llamadas restricciones destructoras de subtours.

Por ejemplo:

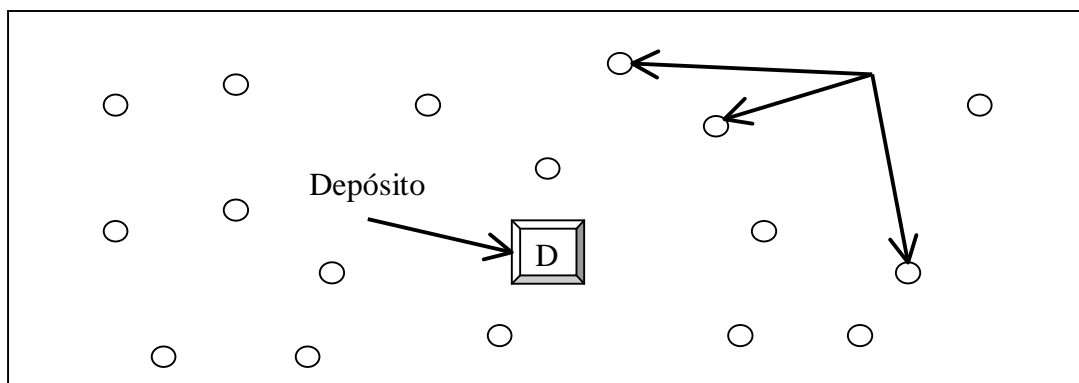
$$S = \left\{ (x_{ij}) : \sum_{i \in Q} \sum_{j \notin Q} x_{ij} \geq 1 \text{ para cada subconjunto } Q \text{ de } N \right\}$$

Esta restricción destructora de subtours dice que todo subconjunto Q de nodos debe estar conectado a los otros nodos del grafo en la solución X.

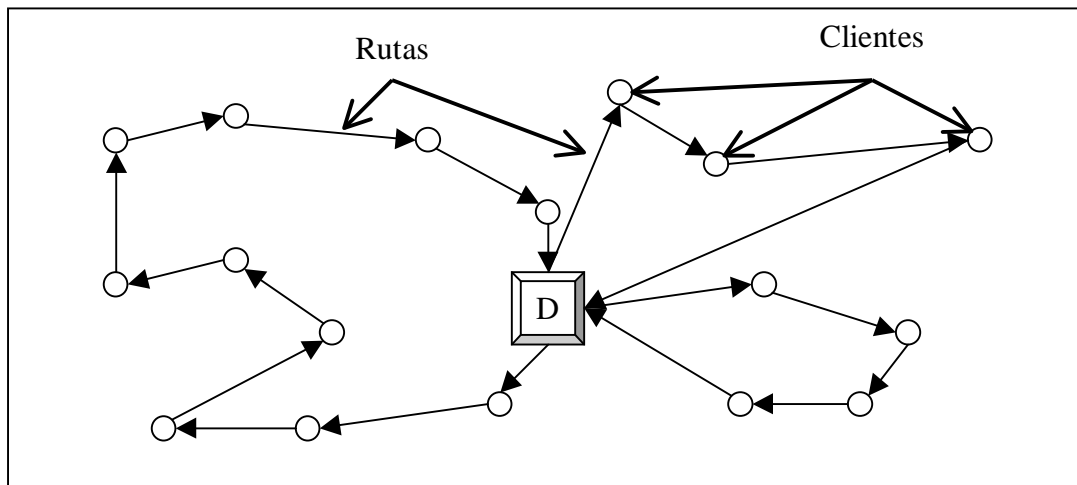
### VRP – Vehicle Routing Problem

Además de su relevancia práctica, el Problema de Ruteo de vehículos es un área de gran importancia para la Investigación Operativa. La siguiente definición de VRP fue tomada de las notas introductorias de J.E. Beasley tituladas "OR-Notes" (ver referencias).

Se considera la situación que presenta la figura de abajo, se tiene un depósito rodeado por un número finito de clientes que deben ser abastecidos. Esta situación se ilustra en la siguiente figura:



El administrador del depósito se encuentra con la tarea de diseñar y asignar rutas para realizar sus entregas utilizando una flota de vehículos los cuales dispone. Este problema de diseño de rutas se conoce como "Problema de ruteo de vehículos" (VRP).



Consecuentemente el Problema de Ruteo de Vehículos puede ser definido como el problema de asignar rutas a vehículos de reparto (de capacidades conocidas) los cuales operan a partir de un único depósito para proveer a un conjunto de clientes cuya localización y demanda son conocidas. Las rutas para los vehículos son diseñadas para minimizar ciertos objetivos tales como la distancia total recorrida.

Este problema ha atraído mucha atención a nivel de literatura académica básicamente por dos razones.

- El problema aparece en una gran cantidad de casos prácticos.
- El problema es teóricamente interesante, y no siempre es fácil su solución.

El Problema de Ruteo de Vehículos en la práctica involucra muchas restricciones acerca de la ruta que los vehículos de entrega pueden seguir, a continuación se consideran algunas de las más comunes. Se pueden clasificar estas restricciones según su relación con los vehículos o con los clientes. Nótese que en cualquier caso particular no todas las restricciones se pueden aplicar, de todas maneras pensando con una concepción global acerca del problema es útil listar todas las restricciones que potencialmente se pueden aplicar.

## Vehículos

- Cada vehículo tiene un límite (capacidad, generalmente en peso o volumen) en cuanto a la mercadería que transporta, por ejemplo un camión cisterna que traslada gasolina a una estación de servicio tiene límite de volumen, los autobuses tienen límite en cuanto a la cantidad de personas que pueden legalmente llevar en su interior, etc.
- Cada vehículo tiene un tiempo total de trabajo desde el momento que parte hasta el que arriba al depósito, este tiempo por lo general corresponde al determinado por las horas de trabajo del conductor del vehículo.
- Cada vehículo tiene un período de tiempo en el cual debe abandonar el depósito, esto se debe para asegurarse de que haya espacio disponible para la llegada de otros vehículos que vengan a reabastecerse al depósito.
- Cada vehículo tiene una cantidad determinada de períodos de tiempo para no hacer nada (tiempo en el cual el conductor descansa).

- Cada vehículo tiene un costo asociado a su uso.

### Clientes

- Cada cliente tiene una cierta cantidad (demanda de mercadería) la cual le debe ser entregada, comúnmente se habla de operaciones de entrega absoluta.
- Cada cliente tiene una cantidad definida de períodos de tiempo en los cuales acepta la llegada de mercadería. Esos períodos de tiempo son llamados ventanas de tiempo. Por ejemplo un cliente podría solo recibir la mercadería entre las horas 10:30 y 11:30 o entre la 14:00 y las 16:15. Estos dos períodos de tiempo son las ventanas de tiempo para ese cliente. Las ventanas de tiempo son de ayuda para los clientes, dado que les sirven para poder organizar otras tareas para que se ajusten a ese patrón de trabajo definido. Así como decimos que las ventanas de tiempo son convenientes para los clientes, no lo son para las compañías que deben entregar las mercaderías, debido a que esto les quita flexibilidad implicando un aumento en los costos.
- Cada cliente tiene asociado un tiempo en el cual se procede a “descargar” la mercadería.
- No todos los vehículos pueden ser utilizados para entregar mercadería a todos los clientes. Cada cliente tiene un conjunto de vehículos que pueden ser utilizados para recibir sus entregas (restricciones de acceso).
- Los clientes tienen prioridad para sus entregas (esto se utiliza en el caso de que los vehículos no puedan cumplir con las entregas de todos los clientes asignados). Esto ocurre debido a indisponibilidad tanto del conductor como del vehículo que debía de hacer la entrega, o por condiciones meteorológicas que reduzcan dramáticamente la velocidad de los vehículos.
- Los clientes pueden recibir visitas compartidas (una entrega que requiera más de un vehículo para su realización).

### Otros factores

- Viajes múltiples realizados por el mismo vehículo en un mismo día, en los cuales los vehículos deben retornar al depósito para luego partir nuevamente. (por ejemplo las camionetas de entrega de cartas del correo)
- Viajes efectuados por un vehículo que requieren más de un día (por ejemplo con paradas nocturnas).
- Vehículos compartimentados con diferentes tipos de productos a ser entregados. Los vehículos de transporte de gasolina suelen estar compartimentados para poder llevar gasolina con y sin plomo, así como los vehículos de transporte de comida (comida congelada y no congelada).
- Más de un depósito, donde los vehículos pueden comenzar, visitar o terminar su trayecto.

En el diseño de rutas para vehículos que cumplan las restricciones mencionadas existe un gran número de objetivos que pueden ser adoptados. Entre ellos se pueden distinguir tres de los más básicos:

- Minimizar el número de vehículos a utilizar (los vehículos y sus respectivos conductores son costos fijos).
- Minimizar la distancia total (o tiempo) a recorrer. Esto corresponde al uso de variables de costo.
- Minimizar ciertas condiciones de las dos anteriores, por ejemplo minimizar el costo total de las variables de costo y los costos fijos.

El costo de los vehículos de la flota es a menudo considerado como un costo fijo, por lo tanto el primer objetivo corresponde a minimizar los costos fijos; el segundo objetivo corresponderá a minimizar las variables de costo; y el tercer objetivo será minimizar el costo total.

El problema de ruteo de vehículos consiste en determinar un conjunto de rutas para realizar pedidos a varios nodos, las cuales se originan en un depósito central, buscando minimizar la distancia total recorrida por la totalidad de la flota. Los vehículos constan de cierta capacidad de carga y pudieran tener restricciones de tiempo, entre otras. Todo vehículo comienza y culmina su recorrido en el depósito central. Nos referiremos a la siguiente formulación del problema como el Problema de Ruteo de Vehículos Genérico:

Minimizar  $\sum_{i=1}^n \sum_{j=1}^n \sum_{v=1}^{NV} c_{ij} x_{ij}^v$  siendo NV la cantidad de vehículos.

Se puede ver como minimizar la suma de las rutas de cada uno de los vehículos, por lo tanto consideramos el mismo x y c de la anterior definición (TSP). El superíndice v de x significa que es referente al camino por donde transitó el vehículo V.

Sujeto a:

$$\left\{ \begin{array}{l} \sum_{i=1}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (j = 2, \dots, n) \quad \{ \text{a cada nodo solo llegue un arco} \} \\ \sum_{j=1}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (i = 2, \dots, n) \quad \{ \text{de cada nodo solo salga un arco} \} \\ \sum_{i=1}^n x_{ip}^v - \sum_{j=1}^n x_{pj}^v = 0 \quad (v = 1, \dots, NV; p = 1, \dots, n) \\ \{ \text{continuidad de las rutas, si un vehículo entra a un nodo, debe salir de él} \} \\ \sum_{i=1}^n d_i \left( \sum_{j=1}^n x_{ij}^v \right) \leq K_v \quad (v = 1, \dots, NV) \\ \{ \text{suma de los requerimientos de los nodos visitados por un vehículo no sobrepase la capacidad del vehículo} \} \\ \sum_{i=1}^n t_i^v \sum_{j=1}^n x_{ij}^v + \sum_{i=1}^n \sum_{j=1}^n t_{ij}^v x_{ij}^v \leq T_v \quad (v = 1, \dots, NV) \\ \{ \text{implica que se cumplan las restricciones de tiempo impuestas por los clientes} \} \\ \sum_{j=2}^n x_{1j}^v \leq 1 \quad (v = 1, \dots, NV) \\ \{ \text{garantiza que cada vehículo salga 0 o 1 vez del depósito} \} \\ \sum_{i=2}^n x_{i1}^v \leq 1 \quad (v = 1, \dots, NV) \\ \{ \text{garantiza que cada vehículo llegue 0 o 1 vez del depósito} \} \\ X = (x_{ij}) \in S \\ x_{ij} = 0 \text{ o } 1 \quad (i, j = 1, \dots, n) \end{array} \right.$$

En donde  $n$  es el número de nodos, NV es la cantidad de vehículos,  $K_v$  es la capacidad del vehículo  $v$ ,  $T_v$  es el tiempo máximo que tiene el vehículo  $v$  permitido para andar en la ruta,  $d_i$  es la demanda del cliente que se encuentra en el nodo  $i$  ( $d_1 = 0$ ),  $t_i^v$  es el

tiempo que le lleva al vehículo  $v$  entregar o recoger su pedido en el nodo  $i$  ( $t_i^v = 0$ ),  $t_{ij}^v$  es el tiempo que le lleva viajar al vehículo  $v$  desde el nodo  $i$  al nodo  $j$  ( $t_{ii}^v = \infty$ ),  $c_{ij}$  es el costo que lleva trasladarse del nodo  $i$  al nodo  $j$ ,  $x_{ij}^v = 1$  si el arco  $(i, j)$  es atravesado por el vehículo  $v$ , de lo contrario su valor es de cero,  $X$  es la matriz que especifica las conexiones sin importar el vehículo, y sus componentes son  $x_{ij} \equiv \sum_{v=1}^{NV} x_{ij}^v$ .

La distancia es la que se quiere minimizar, alternatively se podría minimizar costos reemplazando  $c_{ij}$  por el coeficiente de costo  $c_{ij}^v$  el cual depende del tipo del vehículo.

Se puede encontrar un desarrollo mas profundo de esta formulación matemática en el Apéndice "Estado del Arte" y en [RSV], páginas 95 a 97.

De esta definición se desprende que la demanda en cada nodo es atendida por exactamente un solo vehículo. También queda determinada la continuidad de la ruta, por ejemplo si un vehículo entra a un nodo con demanda, debe salir de ese nodo. También se indican las restricciones de capacidad de los vehículos, y del tiempo total de ruta transcurrido. Además se debe asegurar que la disponibilidad del vehículo no sea excedida. Por último no se deben permitir las formaciones de subtours. Por otro lado se asume que la demanda de cada nodo no excede la capacidad de los vehículos. En este modelo también se asume que cuando un nodo con demanda es atendido, sus requerimientos son satisfechos completamente, esto quiere decir que no existen servicios parciales.

Existe un número de extensiones al problema básico del VRP que fueron consideradas por varios investigadores. Estas apuntan a crear un modelo más real del problema imponiendo nuevas restricciones. Entre estas se puede encontrar restricciones en cuanto al tamaño de la flota, decisiones mixtas y permitir diferente formas de rutas entre otras. Una de las más conocidas es la investigación de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW), esta especifica que cada cliente debe ser atendido dentro de un período de tiempo específico llamado ventana de tiempo. VRPTW y otras extensiones del VRP son consideradas en el anexo "Estado del arte" a este documento y también se pueden encontrar en [RSV].

## 2.2 A considerar en este proyecto

### TSP

Sea un grafo  $G = [N, A, C]$ , donde  $N$  es el conjunto de nodos,  $A$  es el conjunto de aristas y  $C=[c_{ij}]$  la matriz de costos, donde  $c_{ij}$  es el costo asociado a la arista entre el nodo  $i$  y el nodo  $j$ .

En este proyecto se considerará resolver el TSP simétrico sujeto a las siguientes restricciones:

- existe un camino entre cualquier par de nodos (grafo completo).

- el costo  $c_{ij}$  entre el nodo  $i$  y el nodo  $j$  es la distancia euclidiana entre ellos medida en un eje de coordenadas cartesianas positivas.

## VRP

Sea un grafo  $G = [N, A, C, D]$ , donde  $N$  es el conjunto de nodos,  $A$  es el conjunto de aristas,  $C = [c_{ij}]$  la matriz de costos, donde  $c_{ij}$  es el costo asociado a la arista entre el nodo  $i$  y el nodo  $j$ , y  $D = [d_i]$ .

En este proyecto se considerará resolver el VRP sujeto a las siguientes restricciones:

- Existe un camino entre cualquier par de nodos (grafo completo).
- El costo  $c_{ij}$  entre el nodo  $i$  y el nodo  $j$  es la distancia euclidiana entre ellos medida en un eje de coordenadas cartesianas positivos.
- Existe un solo vehículo. Esto no hace que se pierda la generalidad del problema, ya que al obtener una solución con un solo vehículo se puede plantear que los distintos tours se realicen con más de un vehículo a la vez. La única restricción que hay que tener en cuenta es que los vehículos posean las mismas características.
- Existe un solo depósito.
- La demanda de ningún cliente es mayor que la capacidad del vehículo cuando es visitado un cliente, se debe satisfacer toda su demanda (entrega absoluta).
- Existe un solo "tipo" de mercadería.
- Los clientes aceptan la entrega a cualquier hora (no se consideran ventanas de tiempo).
- No hay prioridad entre los clientes.

### 3 Heurísticas y metaheurísticas conocidas

Las heurísticas son algoritmos no exactos que dan como resultado una solución, pero no nos aseguran que ésta sea la óptima. Lo que se espera es que en un menor tiempo de ejecución que un algoritmo exacto brinde una solución "cercana" a la óptima. Dependerá de la bondad de la heurística cuan cerca o lejos de la solución óptima se encuentre el resultado obtenido. Esto se puede evaluar al comparar los resultados alcanzados de ambas formas. En nuestro caso evaluaremos la bondad de las heurísticas propuestas al compararlas con las publicadas en el TSPLIB95 [TSPLIB]. En caso de utilizar una entrada no especificada en la TSPLIB95 se especifica el origen de la misma.

En la resolución del TSP y del VRP, el espacio de potenciales soluciones crece exponencialmente con un aumento lineal de la entrada (cantidad de nodos) y por lo tanto es un caso característico en el cual se utilizan heurísticas en vez de algoritmos exactos para su resolución.

Las metaheurísticas son heurísticas que toman ideas, emulan o siguen reglas de comportamiento de otros fenómenos de la naturaleza.

Para introducirnos en la resolución de los problemas de TSP y VRP estudiamos distintas heurísticas conocidas que se pueden clasificar en tres tipos.

- Heurísticas constructivas
- Heurísticas de mejoramiento iterativo
- Heurísticas híbridas

A continuación se presenta un resumen de algunas de estas heurísticas estudiadas. Un desarrollo más detallado y otras heurísticas se encuentran en el Apéndice 1.

#### 3.1 Heurísticas Constructivas

Una heurística constructiva es un algoritmo que determina un tour en función de alguna regla de construcción, con la característica de que una vez determinado el mismo no busca la manera de mejorarlo. Los tours que ya fueron creados permanecen sin cambios a lo largo de la ejecución del algoritmo.

##### **Vecino más cercano**

El vendedor comienza en una ciudad que va a ser su origen y destino. Desde ahí decide visitar la ciudad más cercana a ella. Una vez en ésta continúa su recorrido visitando la ciudad más cerca a donde se encuentra en ese momento y que aún no fue visitada. Este último paso se repite hasta que no queden ciudades sin visitar. Una vez llegado a esa instancia retorna a la ciudad origen.

Dentro de las heurísticas constructivas se encuentra una clasificación de algunas particulares:



## Heurísticas constructivas de Inserción

Estas heurísticas comienzan con la construcción de un subtour inicial, y luego lo van extendiendo al insertar los nodos restantes uno tras otro hasta que todos hallan sido insertados. Estos procedimientos de inserción, toman un subtour de  $k$  nodos en la iteración  $k$  y tratan de determinar cual de los nodos que no pertenecen al recorrido debe unirse a este (paso selectivo), y en donde (paso de inserción).

### Inserción más cercana

Comenzamos con un subtour que solo contiene al nodo  $i$ . Hallamos el nodo  $k$  tal que  $c_{ik}$  es el mínimo para todos los demás nodos, y formamos el subtour  $i-k-i$ . El paso selectivo considera un subtour y se busca el nodo  $k$  que no se encuentra en el mismo, tal que la distancia a cualquier miembro del subtour sea mínima.

El paso de inserción corresponde a hallar el arco  $(i, j)$  en el subtour el cual minimice la ecuación  $c_{ik} + c_{kj} - c_{ij}$ , e inserta  $k$  entre  $i$  y  $j$ . Esto se realiza hasta encontrar un ciclo Hamiltoniano.

### Inserción aleatoria

En esta heurística se procede igual que en la inserción mas cercana, con la diferencia que en el paso selectivo se selecciona a un nodo aleatoriamente.

### Inserción de la cáscara convexa

La cáscara convexa se le llama al menor conjunto convexo que contiene a todos los puntos. Esta probado [RSV] que si los costos  $c_{ij}$  representan distancias Euclidianas, y que si  $H$  es la cáscara convexa del conjunto de nodos, en un espacio bidimensional, el orden en el cual aparecen los nodos pertenecientes a los bordes del tour óptimo, es el mismo en el que aparecen en  $H$ . Esta apreciación sirve como impulso para el desarrollo de la *Heurística de la Cáscara Convexa*.

El procedimiento a seguir es el siguiente: primero se construye la cáscara convexa del conjunto de nodos con el cual voy a trabajar. Este ciclo se utiliza como subtour inicial. En el siguiente paso se itera sobre todos los nodos que aun no se encuentran en el recorrido, por lo tanto, para cada nodo  $k$  se decide entre cual pareja de nodos  $i$  y  $j$  pertenecientes al subtour debo insertarlo. Esto se realiza siguiendo el criterio de inserción que minimiza la ecuación  $c_{ik} + c_{kj} - c_{ij}$  para los nodos  $i, j, y k$ . En la próxima iteración se determina la terna  $(i^*, k^*, j^*)$  tal que es mínimo entre todas las ternas  $(i, k, j)$  encontradas en el paso anterior. Luego se inserta el nodo  $k^*$  entre los nodos del subtour  $i^*$  y  $j^*$ . Se repiten estos pasos (a excepción del primero) hasta encontrar un ciclo Hamiltoniano.

## 3.2 Heurísticas de mejoramiento iterativo

Las heurísticas de mejoramiento iterativo siguen los siguientes pasos:

- Encontrar un tour inicial. Este puede ser seleccionado de manera aleatoria, definido por una heurística de tipo constructiva o utilizando otro algoritmo.

- Mejorar el tour seleccionado utilizando una heurística de intercambio.
- Realizar el paso 2 hasta que no sea posible realizar más mejoras sobre el tour.

Como ejemplo de una heurística de estas características, la heurística de mejoramiento iterativo Intercambio 2-opt se presenta en el anexo 1.

### 3.3 Heurísticas híbridas

Las heurísticas híbridas o compuestas combinan una heurística constructiva para hallar un tour inicial y luego este se lo toma como entrada para una heurística de mejoramiento iterativo.

La triangulación de Delaunay y OCIDT son ejemplo de heurísticas híbridas, también desarrollados en el anexo 1.

## 4 Propuestas para la resolución del TSP

En este capítulo se presenta un conjunto de heurísticas para la resolución del TSP. Estas heurísticas se basan en el concepto de centro de masas tomado del tema "Impulso y cantidad de movimiento". [FIS1] Con respecto a la aplicación en los estudios de la física, las entidades participantes y el comportamiento descrito en este tema son útiles para la descripción de la rotación y simplifican considerablemente el análisis de los choques. Por otra parte, este concepto puede relacionarse con el problema que nos ocupa, el TSP. Tomando ideas basadas en el centro de masas surgen las propuestas de resolución planteadas. Al relacionar las entidades participantes (ciudades o nodos), se puede calcular el centro de masas del sistema y tomarlo como referencia para determinar el orden en que deben ser visitadas.

Las heurísticas propuestas construyen la solución incrementalmente, es decir que van "armando" una solución a medida que se ejecuta el algoritmo, y no consideran una solución inicial sobre la cual se va iterando para mejorarla. Son heurísticas constructivas de inserción, tomando la clasificación expuesta en el capítulo anterior. Se referirá a la solución final brindada por la heurística como tour y al camino que se va obteniendo durante la construcción del mismo como subtour.

Dentro de la "construcción" de la solución (tour) consideramos tres criterios:

- Criterio de selección del primer nodo
- Criterio de selección de nodo (sin ser el primero)
- Criterio de inserción de nodo

El criterio de selección del primer nodo indica cual nodo será el primero a ser considerado. El por mi mismo será el primer subtour. El tour final depende de la selección de este nodo, por eso se considera como un criterio separado del segundo. El criterio de selección de nodo indica cual será el siguiente nodo a ser seleccionado para el subtour que se está construyendo. Y el tercer criterio indica (una vez seleccionado un nodo por el criterio de selección de nodo) en que lugar del subtour se insertará, por lo tanto determina el orden en que serán recorridos los nodos del tour.

Se presenta a continuación la definición de centro de masas y los criterios de selección e inserción que conforman las diferentes heurísticas propuestas.

### 4.1 Centro de masas

Un criterio particularmente útil para analizar el movimiento de un sistema de muchas partículas, o un cuerpo finito, es el de *centro de masas* (abreviado CM de aquí en adelante). La posición del CM de un sistema de N partículas de masas  $m_1, m_2, \dots, m_N$ , cuya ubicación en el espacio está dada por los vectores  $r_1, r_2, \dots, r_N$  está dada por  $m_1 r_1 + m_2 r_2 + \dots + m_N r_N$ , siendo "+" la suma vectorial y la multiplicación  $m_i r_i$  el producto escalar. Por propiedades de estas operaciones, si consideramos  $M = m_1 + m_2 + \dots + m_N$

la masa de todo el sistema y  $r_{CM} = r_1 + r_2 + \dots + r_N$  la suma vectorial de  $r_1 + r_2 + \dots + r_N$  podemos definir el CM del sistema planteado como  $Mr_{CM}$  [FIS1].

La aplicación del concepto de CM al problema que nos ocupa la planteamos al hacer una analogía del mapa (ciudades o nodos y su respectiva ubicación) con un sistema de partículas. Consideraremos este sistema sin la influencia de fuerzas externas, lo que nos permite calcular el CM del sistema con solamente la información referente a las partículas. De cada partícula debemos saber su masa y su ubicación (vector de posición) para poder calcular el CM del sistema. En la analogía planteada supondremos la masa de cada nodo constante e igual a uno. Esto es porque cada nodo es indistinto. Si consideráramos masas diferentes, el CM del sistema estaría "desequilibrado" hacia los nodos a los que les haya sido asignado mayor masa. La ubicación de los nodos en el plano es un dato que ya está dado como entrada al TSP. La ubicación del centro de coordenadas del sistema de nodos ubicados en un plano euclidiano que se brinda como entrada para el TSP no influye en los cálculos del CM. Todos los vectores de posición  $r_i$  de los nodos del sistema estarán referidos al mismo sistema de coordenadas, por lo tanto el vector  $r_{CM}$  del CM que se calcula como su suma vectorial también lo estará. Teniendo en cuenta estos dos aspectos queda planteada la analogía que consideramos para la resolución del TSP con un sistema de partículas al considerar el CM del mismo.

## 4.2 Heurísticas del centro de masas

Como hemos mencionado al principio de este capítulo las heurísticas propuestas están conformadas por tres criterios: selección del nodo inicial, selección de nodo e inserción. Las diferentes heurísticas diferirán entre sí de acuerdo a los distintos criterios aplicados. El orden de aplicación y la forma general propuesta es la siguiente:

- Aplicar del criterio de selección del nodo inicial
- Marcar como visitado el nodo seleccionado  
(En este momento el subtour está conformado solamente por un nodo)
- Mientras haya nodos no marcados:
  - Aplicar el criterio de selección de nodo
  - Aplicar el criterio de inserción de nodo al nodo seleccionado
  - Marcar como visitado el nodo seleccionado e insertado y quitar la selección
- Fin Mientras

A continuación se presentan los criterios propuestos y su aplicación.

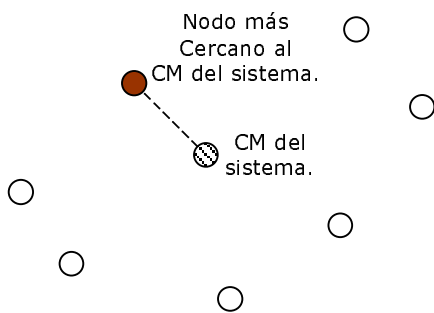
### 4.2.1 Criterios de selección del nodo inicial

Utilizando la analogía del mapa con un sistema de partículas se puede calcular el CM del mismo considerando la masa de cada nodo igual a uno y sus ubicaciones en el plano. Este cálculo brinda la posición del CM en el sistema de coordenadas que estaban dadas las ciudades.

Utilizando el CM calculado de esta manera proponemos dos criterios de selección del nodo inicial que serán presentados a continuación.

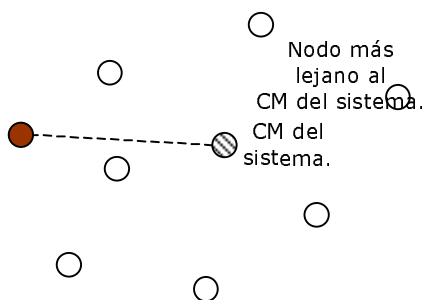
### El más cercano al centro de masas

El primer criterio propuesto selecciona como nodo inicial el más cercano al CM del sistema. Se toma la distancia de cada nodo al CM medida como la distancia euclidiana entre ellos y se selecciona el nodo que está a menor distancia. Si las coordenadas del nodo  $i$  son  $(x_i, y_i)$  y las coordenadas del CM son  $(x_{CM}, y_{CM})$ , la distancia euclidiana entre ellos es  $\sqrt{(x_i - x_{CM})^2 + (y_i - y_{CM})^2}$ . Este criterio considera el primer subtour como el nodo más cercano al CM del sistema. Se espera que en el "entorno" de este nodo seleccionado haya "bastantes" nodos, ya que están "cerca" del CM y que por lo tanto puedan ser visitados logrando que las distancias entre ellos sea "corta".



### El más lejano al centro de masas

El segundo criterio propuesto selecciona como nodo inicial el más lejano al CM del sistema. También se mide la distancia euclidiana de los nodos al CM, como en la propuesta anterior, pero se selecciona el de mayor distancia. Este criterio considera el primer subtour como el nodo más lejano al CM buscando empezar la construcción del tour final visitando a los nodos que están en una zona de "baja densidad". La idea es que (dependiendo también del criterio de selección de nodo utilizado) estos nodos no influyan en los cálculos posteriores (mas avanzados en la construcción) del CM que utilizará el criterio de selección.



De todas formas, como criterio de selección del primer nodo, se podría seleccionar uno arbitrariamente. En particular, en la herramienta desarrollada para la ejecución de las propuestas se tiene la opción de que el usuario seleccione un nodo como inicial si no quiere utilizar ninguna de las dos propuestas anteriores.

## Criterios de inserción de nodos

En cada iteración de la heurística se selecciona un nodo utilizando un criterio de selección. Según un criterio de inserción se inserta el mismo al subtour resultado de la iteración anterior y se obtiene el subtour de la iteración actual.

En esta sección se presentan tres criterios de inserción. Las entradas a los mismos son un subtour y un nodo que no pertenece al subtour (que fue seleccionado por un criterio de selección). Provee como salida un subtour que contiene a los nodos dados en la entrada.

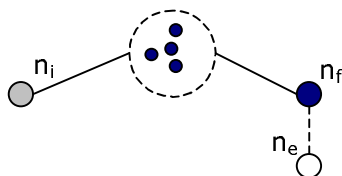
A continuación se presentarán los criterios de inserción propuestos, y se utilizará la siguiente nomenclatura:

- $\text{subtour}_e$  para referirnos al subtour de entrada
- $n_1-...n_i-...n_f$  para referirnos a los nodos del subtour de entrada
- $n_e$  para referirnos al nodo de entrada (criterio de selección)
- $\text{subtour}_s$  para referirnos al subtour de salida

### A continuación del último

Este criterio de inserción es el más sencillo y el más rápido de ejecutar, simplemente considera agregar el nodo seleccionado a continuación del último ingresado en el subtour. Es decir que la entrada en la aplicación de este criterio será  $\text{subtour}_e = n_1-...n_i-...n_f$  y  $n_e$  y la salida será  $\text{subtour}_s = n_1-...n_i-...n_f - n_e$ .

Podemos verlo gráficamente representado en la siguiente figura:



Este criterio prioriza la eficiencia ya que no debe hacer cálculos, por lo que tiene  $O(1)$ . Su eficacia depende mucho de la ubicación del nodo seleccionado con respecto al anterior seleccionado.

### En el extremo más cercano

Se miden las distancias  $n_i - n_e$  y  $n_f - n_e$ . Y se inserta a continuación del nodo tal que la distancia con él sea mínima. Por lo tanto el subtours estará conformado por  $n_1-...n_i -...n_f - n_e$  o por  $n_e - n_1-...n_i -...n_f$  según cual sea la distancia mínima.

En la siguiente figura se muestran los dos casos posibles:

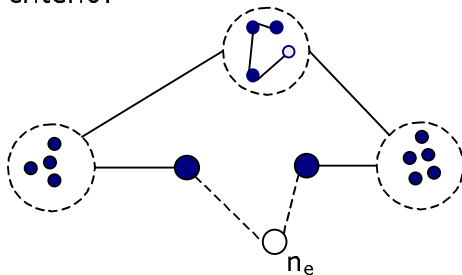


Este criterio, al igual que el anterior, prioriza la eficiencia ya que debe hacer solamente dos cálculos de distancias y una comparación, lo que lo hace de  $O(1)$ . Su eficacia depende mucho de la ubicación del nodo seleccionado con respecto a los extremos del subtour de entrada.

### En la mejor ubicación

Este criterio inserta al  $n_e$  en el  $\text{subtour}_e$  tal que esa inserción resulte en el mínimo  $\text{subtour}_s$ . Elige entre todas las posibles ubicaciones de  $n_e$  entre nodos adyacentes de  $\text{subtour}_e$  y se queda con la que minimice la longitud total de  $\text{subtour}_s$ .

En la siguiente figura se muestran una representación gráfica de la inserción con este criterio:



Este criterio prioriza la eficacia del resultado incrementando el orden de ejecución en busca de una mejor solución. Calcula la longitud de todos los potenciales subtours "probando" con el  $n_e$  entre todo par de nodos adyacentes en el  $\text{subtour}_e$ . El  $n_i$  se considera adyacente al  $n_f$  para este cálculo. Esto hace que el algoritmo tenga  $O(n)$  pues calcula las distancias de todos los posibles subtours para quedarse con el de menor longitud.

## 4.3 Criterios de selección de nodo y construcción de las heurísticas

En la ejecución de las heurísticas constructivas propuestas, luego de haber seleccionado el nodo inicial, éste se marca como visitado y se obtiene el primer subtour conformado por solamente un nodo. Luego se itera sobre los nodos que aún no han sido procesados (no han sido marcados como visitados) seleccionándolos de acuerdo a un criterio de selección e insertándolos al subtour actual para obtener el siguiente subtour. En esta sección se presentan doce criterios de selección de nodos basados en el concepto de CM y la construcción y ejecución de heurísticas utilizando los criterios antes mencionados.

Para aplicar el cálculo del CM se debe considerar un sistema conformado por un conjunto de nodos. En el caso de los criterios de selección del nodo inicial, el conjunto considerado es el sistema en su totalidad, todas las ciudades del mapa. A medida que el algoritmo va avanzando en su ejecución se puede considerar dos conjuntos de nodos más: los procesados (marcados como visitados) y los aún no procesados (no marcados como visitados). Los diferentes criterios de selección utilizan el CM calculado sobre los tres grupos de nodos antes mencionados. Utilizaremos la siguiente nomenclatura para referirnos a ellos:

- $CM_S$  = Centro de masas del sistema conformado por todos los nodos del mapa
- $CM_V$  = Centro de masas del sistema conformado por todos los nodos visitados
- $CM_{NV}$  = Centro de masas del sistema conformado por todos los nodos no visitados

El  $CM_S$  es invariante durante la ejecución de la heurística, ya que se calcula a partir de los datos de entrada del TSP. Tanto el  $CM_V$  como el  $CM_{NV}$  varían en cada iteración de selección e inserción, ya que en cada paso un nodo no visitado "pasa" del conjunto de los nodos no visitados al conjunto de los nodos visitados.

Proponemos dos conjuntos de criterios de selección de nodos. El primero (conformado por los seis primeros criterios) utiliza la noción de nodos más cercanos al CM considerado y el segundo (conformado por los seis últimos criterios) la noción de nodos más lejanos al CM considerado.

Combinando los diferentes criterios presentados a continuación con los expuestos anteriormente se define un conjunto de heurísticas para la resolución del TSP al combinar uno de selección del nodo inicial, uno de selección de nodo y uno de inserción. Al tener dos criterios de selección del nodo inicial, doce de selección de nodo y tres de inserción existen setenta y dos (dos por doce por tres) combinaciones posibles de criterios que resultan en diferentes (aunque similares) heurísticas para la resolución del problema que nos ocupa. De todas maneras los distintos criterios no están pensados para ser utilizados en todas sus combinaciones. La idea es utilizar el criterio de selección del nodo inicial más cercano al  $CM_S$  con los seis primeros criterios de selección de nodo (que consideran la cercanía de los nodos seleccionados al CM) y utilizar el criterio de selección del nodo inicial más lejano al  $CM_S$  con los últimos seis criterios de selección de nodo (que consideran la lejanía de los nodos seleccionados al CM).

A continuación se presentan los doce criterios de selección propuestos y los resultados de la aplicación de doce combinaciones (una para cada criterio de selección) y su comparación con el mejor resultado publicado en la TSPLIB95.

Las combinaciones de criterios que conforman las 12 heurísticas presentadas se resumen en la siguiente tabla:

Criterio de selección de nodo inicial	Criterio de selección de nodo	Criterio de inserción
El más cercano al centro de masas	Criterio 1	En la mejor ubicación
	Criterio 2	
	Criterio 3	
	Criterio 4	
	Criterio 5	
	Criterio 6	
El más lejano al Centro de masas	Criterio 7	
	Criterio 8	
	Criterio 9	
	Criterio 10	
	Criterio 11	
	Criterio 12	

Se decidió utilizar en todos los casos el criterio de inserción de la mejor ubicación ya que, si bien el tiempo de ejecución es mayor que los criterios de inserción "a



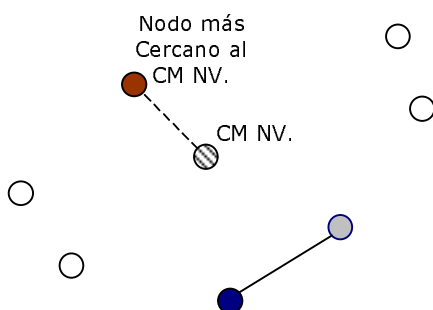
continuación del último” y del “extremo más cercano”, no demora tanto como para utilizar el otro criterio de mayor eficiencia pero peor eficacia.

También se presenta una comparación de los resultados con los obtenidos al aplicar el algoritmo de cáscara convexa y del vecino más cercano. En la aplicación de ambos algoritmos se utiliza en las seis primeras ejecuciones (comparaciones con las seis primeras heurísticas) como primer nodo el más cercano al CMS. Esto es para comparar los dos métodos de resolución partiendo desde el mismo nodo. Y en los últimos seis se utilizó como nodo inicial el más lejano al CMS.

Obs.: Para la especificación de la plataforma en la cual se ejecutaron todos los algoritmos que se presentan tablas en las siguientes secciones referirse al capítulo 6.

### 4.3.1 Selección uno: el más cercano al $CM_{NV}$

En el proceso de selección, se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y se selecciona el nodo más cercano a su  $CM$ , es decir a  $CM_{NV}$ . La idea de aplicar este criterio de selección es que luego de haber sido seleccionado el nodo será insertado en el siguiente subtour, y como está “cerca” del  $CM_{NV}$  es de esperar que en el “entorno” haya una cantidad importante de nodos. Por lo tanto en los pasos sucesivos se seguirán eligiendo en ese entorno y los siguientes nodos se insertarán al subtour con distancias “cortas” al mismo, ya que nodos cercanos a ellos han sido insertados recientemente y el subtour “pasa cerca”. En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Obs.: En esta explicación de la idea del criterio de selección, así como en las siguientes, se utilizan términos informales como ser “cerca”, “entorno”, “cantidad importante” “pasa cerca” y otros. Si bien no están formalmente definidos expresan la idea del criterio. Esta “informalidad” en los términos no degrada la especificación del criterio ya que está inmerso en una heurística y como tal no tiene bases exactas, sino ideas que se espera que conduzcan a una solución aceptable.

En este criterio, cada paso de selección es  $O(nv)$ , siendo  $nv$  la cantidad de nodos no visitados aún. En cada iteración se selecciona un nodo, entonces en la ejecución completa de la metaheurística se itera  $n$  veces, siendo  $n$  la cantidad de nodos del mapa. Si sumamos el orden de ejecución en cada paso obtenemos

$$n + (n-1) + (n-2) + \dots + 1 = \frac{n(n+1)}{2} = \frac{n^2 + n}{2}. \text{ Por lo tanto, el orden de ejecución que}$$

aporta el criterio de selección 1 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodos del sistema.

## Construcción de una heurística utilizando el criterio de selección 1

En este caso utilizaremos el criterio de selección de nodo inicial más cercano al CM<sub>s</sub> y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	559	31	1.5
Berlin52	52	7542	9669	28	1.6
st70	70	675	855	25	3.5
Eil76	76	538	655	21	4.5
Pr76	76	108159	134002	23	4.6
Rat99	99	1211	1501	23	10.1
KroA100	100	21282	26648	25	11
Eil101	101	629	795	26	10.6
Pr107	107	44303	46672	5	13.7
Pr124	124	59030	66190	12	20
Bier127	127	118282	148728	25	22.7
Ch130	130	6110	7757	26	23.5
Pr136	136	96772	111827	15	31.3
Pr144	144	58537	66503	13	36.1
Pr152	152	73682	86952	18	41.5

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM – Selección 1".

Promedio del % de exceso sobre el óptimo conocido: **21%**.

Promedio de tiempo de ejecución: **15.7 seg.**

Comparaciones								
Nombre Instancia	#Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	559	444	25	3.2	525	6	< 1
Berlin52	52	9669	7624	26	2.4	9014	7	< 1
st70	70	855	704	21	7.4	783	9	< 1
Eil76	76	655	576	13	22.6	619	5	< 1
Pr76	76	134002	112479	19	78.2	151142	-11	< 1
Rat99	99	1501	1258	19	289.3	1493	-0.5	< 1
KroA100	100	26648	22055	20	17.7	25781	3	< 1
Eil101	101	795	679	17	40.4	808	-1	< 1
Pr107	107	46672	49695	-6	290.8	52145	-10	< 1
Pr124	124	66190	61627	7	216.2	68321	-3	< 1
Bier127	127	148728	126650	17	313.7	148512	-0.1	< 1
Ch130	130	7757	6731	15	81.5	7216	7	< 1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM – Heurística de la cáscara convexa" y "TSP - Heurísticas del CM – Heurística del vecino más cercano" respectivamente.

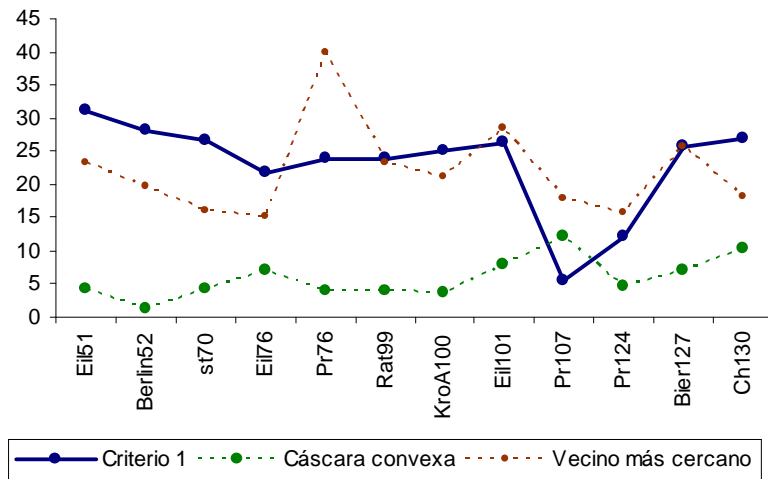
Promedio del % de exceso sobre la Cáscara Convexa: **16%**.

Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**

Promedio del % de exceso sobre el Vecino más cercano: **1%**.

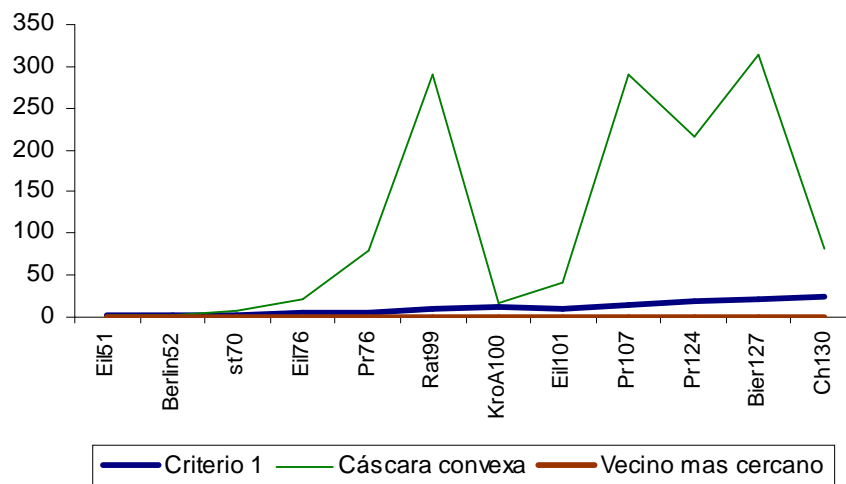
Promedio de tiempo de ejecución Vecino más cercano: **< 1 seg.**

Porcentajes sobre mejor solución conocida (TSPLIB)



Se aprecia un comportamiento estable en la comparación de los resultados con la mejor solución conocida en las instancias cuya topología presenta una distribución uniforme. En el caso Pr107, donde se puede ver que el resultado da mejor se tiene una topología en la cual los nodos se encuentran distribuidos en la periferia agrupados en bandas. No hay una gran diferencia entre la distancia al CM de los nodos más cercanos y más lejanos. A partir de esto se puede inferir que este criterio se está comportando (en estos casos) de forma análoga a una selección por nodos más lejanos al CM.

Comparación de tiempos de ejecución

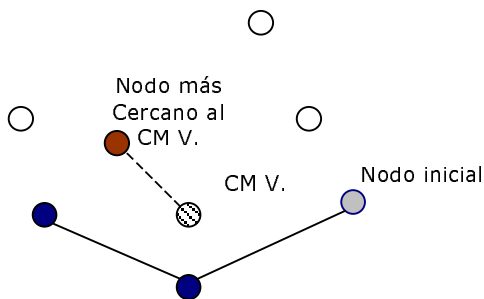


Con respecto al tiempo de ejecución, se puede ver que es estable y creciente con respecto a la cantidad de nodos considerados, a diferencia del algoritmo de cáscara convexa el cual es claramente dependiente de la topología del problema.

### 4.3.2 Selección dos: el más cercano al $CM_V$

En el proceso de selección, se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se selecciona el nodo más cercano a su centro de masas, es decir a  $CM_V$ . La idea de aplicar este criterio de selección es que como el nodo seleccionado se encuentra "cerca" del  $CM_V$  y este no varía mucho con la inserción de un solo nodo en las etapas avanzadas de la heurística (muchas iteraciones), se encontrará "cerca" del anterior. Esto aportara de esta forma una distancia "corta" en el siguiente subtour. De todas formas depende del criterio de inserción que tan "en cuenta" tenga la inserción del nodo anterior con el siguiente en cuanto a la distancia que hay entre ellos.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Realizando un cálculo análogo al expuesto en el anterior criterio se puede ver que el orden de ejecución que aporta el criterio de selección 2 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

### Construcción de una heurística utilizando el criterio de selección 2

En este caso utilizaremos el criterio de selección de nodo inicial más cercano al  $CM_S$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	487	14	1.5
Berlin52	52	7542	9606	27	1.6
st70	70	675	836	23	3.8
Eil76	76	538	667	23	4.9
Pr76	76	108159	141377	30	4.9
Rat99	99	1211	1498	23	12.2
KroA100	100	21282	27484	29	11.2
Eil101	101	629	770	22	11.6
Pr107	107	44303	47266	6	14.4
Pr124	124	59030	77645	31	23.4
Bier127	127	118282	136579	15	23.7
Ch130	130	6110	7764	27	25.2
Pr136	136	96772	111453	15	29.3
Pr144	144	58537	71203	21	37.3
Pr152	152	73682	83859	13	41.6

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM – Selección 2".

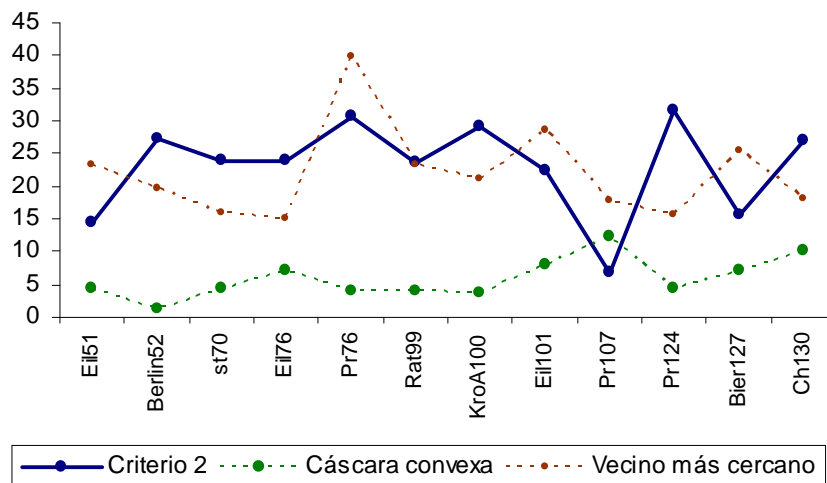
Promedio del % de exceso sobre el óptimo conocido: **21%**.  
 Promedio de tiempo de ejecución: **16.1 seg.**

Comparaciones								
Nombre Instancia	#Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	487	444	9	3.2	525	-7	< 1
Berlin52	52	9606	7624	25	2.4	9014	6	< 1
st70	70	836	704	18	7.4	783	6	< 1
Eil76	76	667	576	15	22.6	619	7	< 1
Pr76	76	141377	112479	25	78.2	151142	-6	< 1
Rat99	99	1498	1258	19	289.3	1493	-0.3	< 1
KroA100	100	27484	22055	24	17.7	25781	6	< 1
Eil101	101	770	679	13	40.4	808	-4	< 1
Pr107	107	47266	49695	-5.0	290.8	52145	-9	< 1
Pr124	124	77645	61627	25	216.2	68321	13	< 1
Bier127	127	136579	126650	7.0	313.7	148512	-8	< 1
Ch130	130	7764	6731	15	81.5	7216	7	< 1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

Promedio del % de exceso sobre la Cáscara Convexa: **15.8%**  
 Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**  
 Promedio del % de exceso sobre el Vecino más cercano: **0.9%**  
 Promedio de tiempo de ejecución Vecino más cercano: **< 1**

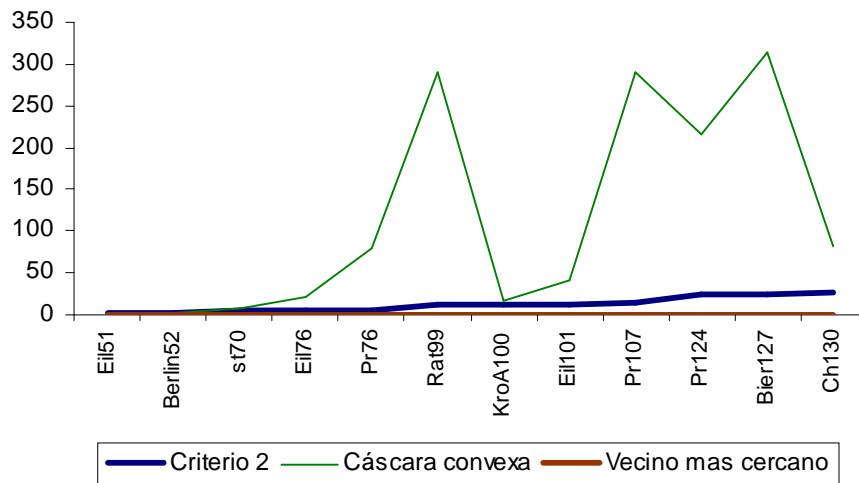
Porcentajes sobre mejor solución conocida (TSPLIB)



En este caso, al igual que en el anterior se puede ver un comportamiento estable en la comparación de los resultados con la mejor solución conocida en las instancias cuya topología presenta una distribución uniforme. Tanto en el caso de Prt107 como en el caso de Prt124 se ve un comportamiento diferente de la media observada en los demás para dar un resultado mejor y otro peor respectivamente. Con respecto al comportamiento en el Prt107 la explicación está dada por la misma razón que en el

caso anterior, ya que el  $CM_V$  y  $CM_{NV}$  en esa topología no varían mucho entre sí. En el caso del Prt124 se puede ver que empeoró considerablemente respecto del criterio 1 y esto se atribuye a que si bien la mayoría de los nodos están en la periferia y no es una topología uniforme existen nodos cercanos al CM que fueron incluidos en el tour de una manera distinta (más “temprana”) que al considerar el  $CM_{NV}$ .

Comparación de tiempos de ejecución

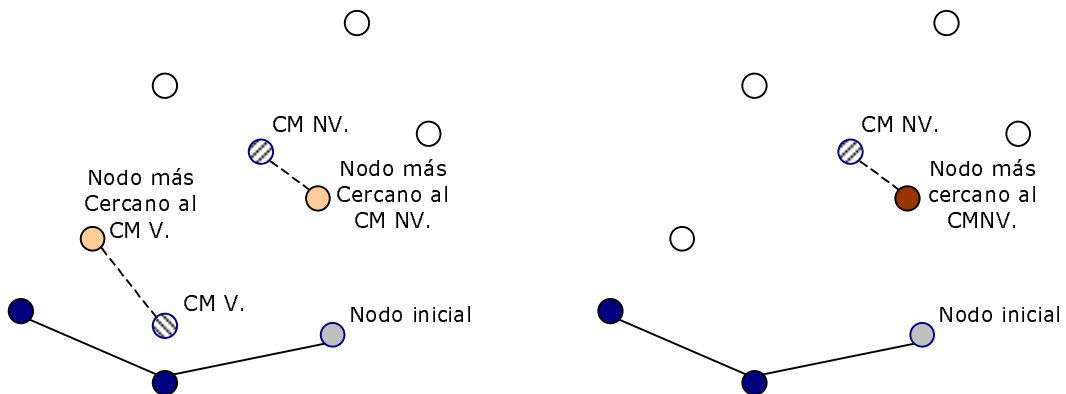


En este caso se aprecia un comportamiento análogo al caso anterior con respecto a los tiempos de ejecución.

### 4.3.3 Selección 3: el más cercano del $CM_V$ o $CM_{NV}$

Este criterio de selección toma las ideas de los dos anteriores intentando adaptarse a la mejor selección dependiendo del caso. Se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y se calcula el nodo más cercano a su centro de masas, es decir a  $CM_{NV}$ . Luego se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se calcula el nodo más cercano a su centro de masas, es decir a  $CM_V$ . De ambos nodos calculados se selecciona el que a menor distancia se encuentre del CM correspondiente.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



En este criterio, cada paso de selección es  $O(nv+v)$ , siendo  $nv$  la cantidad de nodos no visitados aún y  $v$  la cantidad de nodos ya visitados. En cada iteración se selecciona un nodo, entonces en la ejecución completa de la metaheurística se itera  $n$  veces, siendo  $n$  la cantidad de nodos del sistema. Además  $nv + v = n$  en todas las iteraciones. Si sumamos el orden de ejecución en cada paso obtenemos  $n+n+\dots+n=n^2$ . Por lo tanto, el orden de ejecución que aporta el criterio de selección 1 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

### Construcción de una heurística utilizando el criterio de selección 3

En este caso utilizaremos el criterio de selección de nodo inicial más cercano al  $CM_s$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	559	31	1.6
Berlin52	52	7542	9210	22	1.7
st70	70	675	862	27	4.1
Eil76	76	538	660	22	5.1
Pr76	76	108159	133996	23	5.2
Rat99	99	1211	1597	31	11.7
KroA100	100	21282	26242	23	11.7
Eil101	101	629	776	23	13.1
Pr107	107	44303	52279	18	14.7
Pr124	124	59030	77593	31	24.2
Bier127	127	118282	146767	24	25.4
Ch130	130	6110	7742	26	26.3
Pr136	136	96772	116518	20	26.8
Pr144	144	58537	67452	15	31.3
Pr152	152	73682	87301	18	40.5

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM - Selección 3"

Promedio del % de exceso sobre el óptimo conocido: **23%**  
 Promedio de tiempo de ejecución: **16.2 seg.**

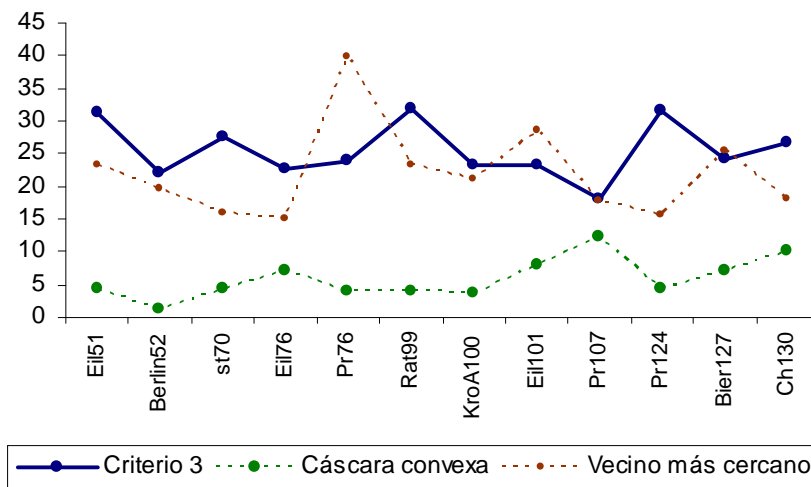
Comparaciones								
Nombre Instancia	#Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más Cercano	% exceso	Tiempo (seg.)
Eil51	51	559	444	25	3.2	525	6	< 1
Berlin52	52	9210	7624	20	2.4	9014	2	< 1
st70	70	862	704	22	7.4	783	10	< 1
Eil76	76	660	576	14	22.6	619	6	< 1
Pr76	76	133996	112479	19	78.2	151142	-11	< 1
Rat99	99	1597	1258	26	289.3	1493	6	< 1
KroA100	100	26242	22055	18	17.7	25781	1	< 1
Eil101	101	776	679	14	40.4	808	-3	< 1
Pr107	107	52279	49695	5	290.8	52145	-0.2	< 1
Pr124	124	77593	61627	25	216.2	68321	13	< 1
Bier127	127	146767	126650	15	313.7	148512	-1	< 1
Ch130	130	7742	6731	15	81.5	7216	7	< 1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la

cáscara convexa” y “TSP - Heurísticas del CM – Heurística del vecino más cercano” respectivamente.

Promedio del % de exceso sobre la Cáscara Convexa: **18.1%**  
 Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**  
 Promedio del % de exceso sobre el Vecino más cercano: **3.0%**  
 Promedio de tiempo de ejecución Vecino más cercano: **< 1**

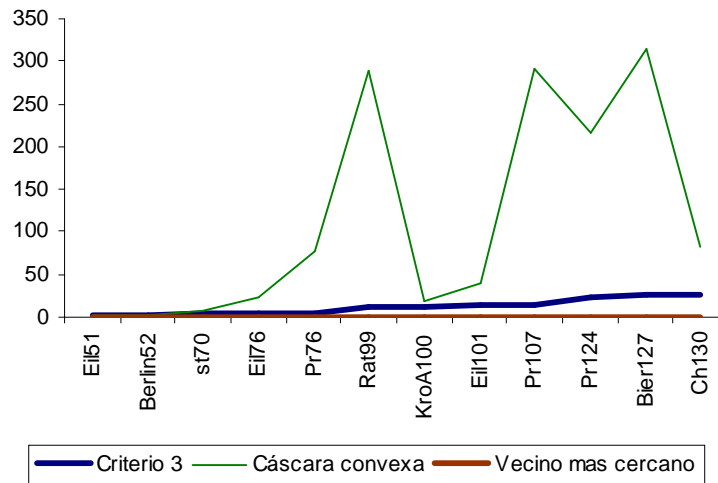
Porcentajes sobre mejor solución conocida (TSPLIB)



En este caso se puede ver una estabilidad, aunque de valores altos, en el comportamiento de las distintas soluciones encontradas al compararlas con la mejor conocida. La estabilidad se explica por la “compensación” que existe entre  $CM_V$  y el  $CM_{NV}$ , ya que cada vez que se visita un nodo, este pasa de ser considerado en el  $CM_{NV}$  a ser considerado en el  $CM_V$ . Por lo tanto, como el criterio utilizado selecciona el mas cercano de ambos se da un “intercambio” de selección con respecto a uno y a otro que no le deja alcanzar la eficacia obtenida en los casos anteriores al seleccionar siempre con respecto a la cercanía de un solo CM.



Comparación de tiempos de ejecución

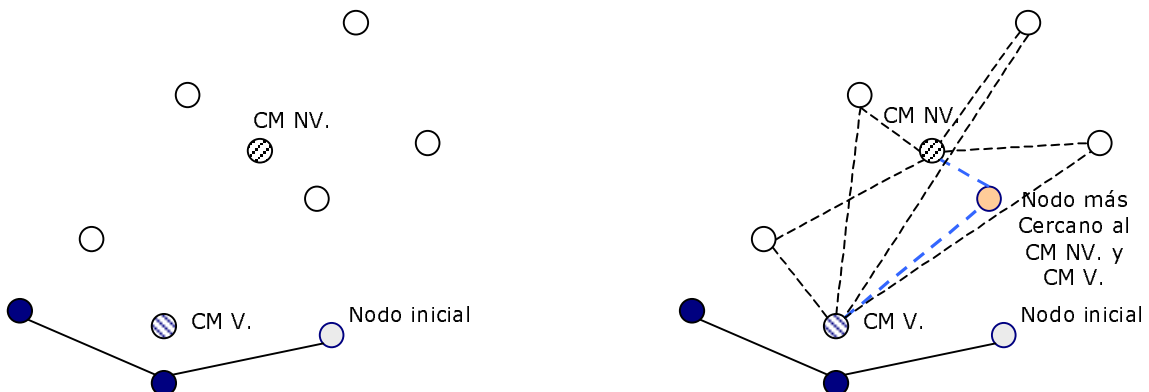


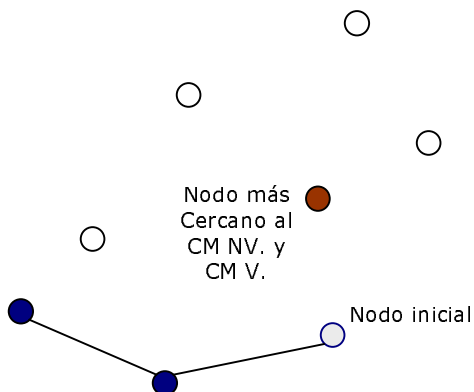
En este caso se aprecia un comportamiento análogo al caso anterior con respecto a los tiempos de ejecución.

#### 4.3.4 Selección 4: el más cercano al $CM_V$ y $CM_{NV}$

Éste y el siguiente criterio son variantes del anterior, el más cercano del  $CM_V$  o  $CM_{NV}$ . Estos criterios intentan "estabilizar" el área del mapa en donde se seleccionan los nodos. En el criterio anteriormente expuesto los nodos "compiten" para ver quién es el que está más cerca de algún CM de los considerados. Dependiendo de quien "gane" el siguiente nodo seleccionado estará más cerca de uno u otro. En estos criterios ambos CM "pesan" en la selección del siguiente nodo y por lo tanto se espera que el área en la cual se encuentre el nodo seleccionado no varíe "mucho" de una iteración a la otra. Se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y se calcula su centro de masas, es decir  $CM_{NV}$ . Luego se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se calcula su centro de masas, es decir  $CM_V$ . Se selecciona el nodo tal que la suma de las distancias a los centros de masas calculados sea mínima.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:





En este criterio, al igual que en el criterio anterior, el cálculo de los dos CM aportan un  $O(n)$  en cada iteración. Además el calculo del nodo cuya distancia a ambos CM sea mínima aporta un  $O(nv)$ . En cada iteración se selecciona un nodo, entonces en la ejecución completa de la metaheurística se itera  $n$  veces, siendo  $n$  la cantidad de nodos del sistema. Además  $nv < n$  en todas las iteraciones. Por lo tanto, el orden de ejecución que aporta el criterio de selección 4 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

#### Construcción de una heurística aplicando el criterio de selección 4

En este caso utilizaremos el criterio de selección de nodo inicial más cercano al CMs y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	527	23	1.6
Berlin52	52	7542	9684	28	1.5
st70	70	675	861	27	3.7
Eil76	76	538	692	28	4.8
Pr76	76	108159	136704	26	5
Rat99	99	1211	1611	33	11.2
KroA100	100	21282	26487	24	10.7
Eil101	101	629	774	23	10.8
Pr107	107	44303	48301	9	12.9
Pr124	124	59030	70653	19	20.1
Bier127	127	118282	150599	27	22
Ch130	130	6110	7763	27	23.6
Pr136	136	96772	113858	17	27
Pr144	144	58537	66930	14	32.6
Pr152	152	73682	87301	18	38.2

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM – Selección 4"

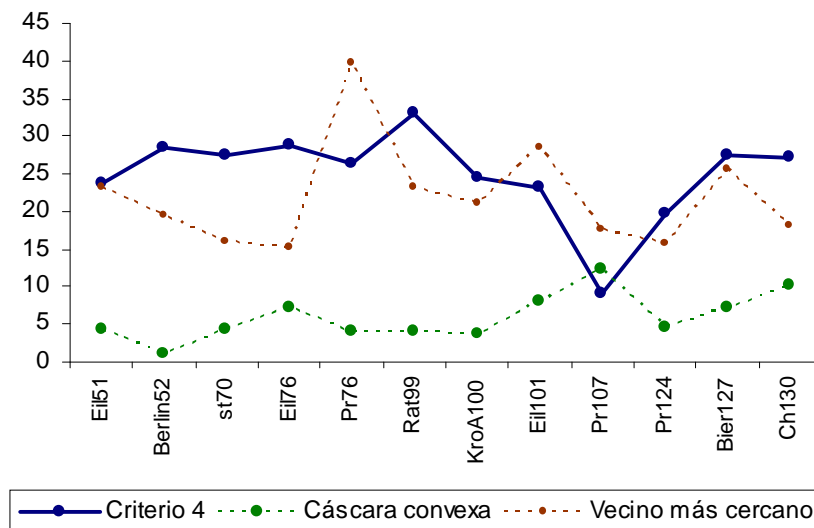
Promedio del % de exceso sobre el óptimo conocido: **22%**  
 Promedio de tiempo de ejecución: **15 seg.**

Comparaciones								
Nombre Instancia	# Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	527	444	18	3.2	525	0.3	< 1
Berlin52	52	9684	7624	27	2.4	9014	7	< 1
st70	70	861	704	22	7.4	783	9	< 1
Eil76	76	692	576	20	22.6	619	11	< 1
Pr76	76	136704	112479	21	78.2	151142	-9	< 1
Rat99	99	1611	1258	28	289.3	1493	7	< 1
KroA100	100	26487	22055	22	17.7	25781	2	< 1
Eil101	101	774	679	13	40.4	808	-4	< 1
Pr107	107	48301	49695	1	290.8	52145	-7	< 1
Pr124	124	70653	61627	14	216.2	68321	3	< 1
Bier127	127	150599	126650	18	313.7	148512	1	< 1
Ch130	130	7763	6731	15	81.5	7216	7	< 1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

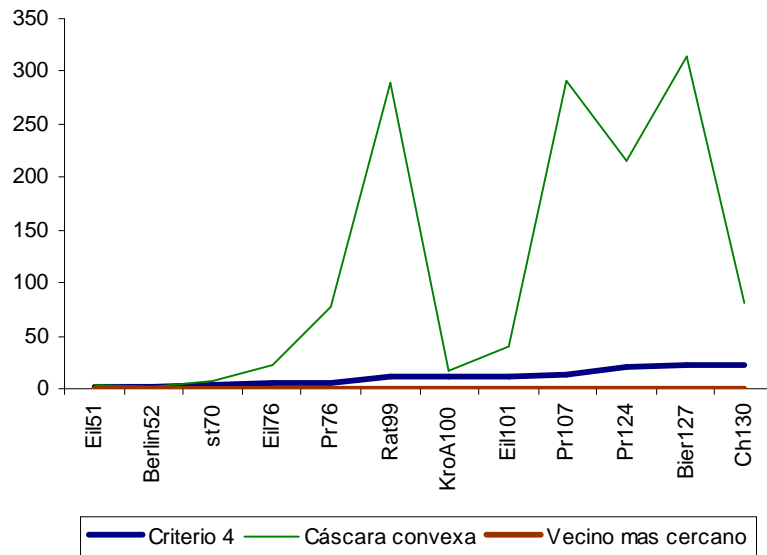
Promedio del % de exceso sobre la Cáscara Convexa: **18.2%**  
 Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**  
 Promedio del % de exceso sobre el Vecino más cercano: **2.2%**  
 Promedio de tiempo de ejecución Vecino más cercano: **< 1**

Porcentajes sobre mejor solución conocida (TSPLIB)



En este caso, nuevamente se puede ver un comportamiento estable en la comparación de los resultados con la mejor solución conocida en las instancias cuya topología presenta una distribución uniforme. También se puede observar un comportamiento mejor en Prt107 como fue analizado en casos anteriores.

Comparación de tiempos de ejecución

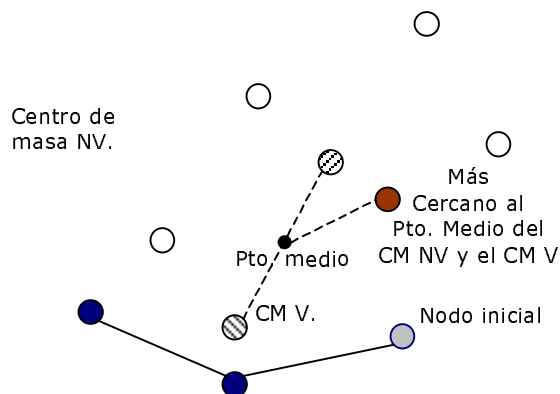


En este caso se aprecia un comportamiento análogo a los casos anteriores con respecto a los tiempos de ejecución.

#### 4.3.5 Selección 5: el más cercano al punto medio entre $CM_V$ y $CM_{VN}$

Se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y se calcula el nodo más cercano a su centro de masas, es decir a  $CM_{NV}$ . Luego se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se calcula el nodo más cercano a su centro de masas, es decir a  $CM_V$ . Se selecciona el nodo más cercano al punto medio entre los dos CM calculados.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Realizando un cálculo análogo al expuesto en el anterior criterio se puede ver que el orden de ejecución que aporta el criterio de selección 5 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

## Construcción de una heurística utilizando el criterio de selección 5

En este caso utilizaremos el criterio de selección de nodo inicial más cercano al CM<sub>s</sub> y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	527	23	1.4
Berlin52	52	7542	9335	23	1.5
st70	70	675	861	27	3.7
Eil76	76	538	692	28	5
Pr76	76	108159	133884	23	5
Rat99	99	1211	1611	33	10.9
KroA100	100	21282	26487	24	10.8
Eil101	101	629	774	23	11.5
Pr107	107	44303	47871	8	17
Pr124	124	59030	73193	23	21.9
Bier127	127	118282	150599	27	24.3
Ch130	130	6110	7090	16	25
Pr136	136	96772	113957	17	28.6
Pr144	144	58537	66930	14	32.1
Pr152	152	73682	87301	18	44.2

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM – Selección 5"

Promedio del % de exceso sobre el óptimo conocido: **21%**

Promedio de tiempo de ejecución: **16.2 seg.**

Comparaciones								
Nombre Instancia	#Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	527	444	18	3.2	525	-0.3	< 1
Berlin52	52	9335	7624	22	2.4	9014	3	< 1
st70	70	861	704	22	7.4	783	9	< 1
Eil76	76	692	576	20	22.6	619	11	< 1
Pr76	76	133884	112479	19	78.2	151142	-11	< 1
Rat99	99	1611	1258	28	289.3	1493	7	< 1
KroA100	100	26487	22055	20	17.7	25781	2	< 1
Eil101	101	774	679	13	40.4	808	-4	< 1
Pr107	107	47871	49695	-3	290.8	52145	-8	< 1
Pr124	124	73193	61627	18	216.2	68321	7	< 1
Bier127	127	150599	126650	18	313.7	148512	1	< 1
Ch130	130	7090	6731	5	81.5	7216	-1	< 1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM – Heurística de la cáscara convexa" y "TSP - Heurísticas del CM – Heurística del vecino más cercano" respectivamente.

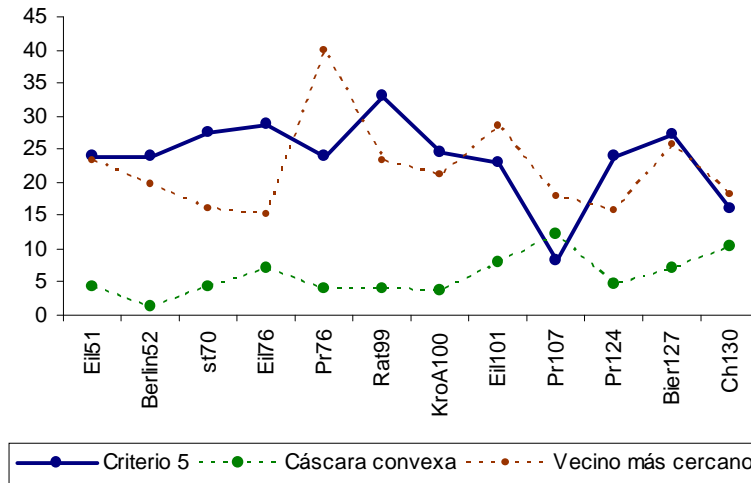
Promedio del % de exceso sobre la Cáscara Convexa: **16.6%**

Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**

Promedio del % de exceso sobre el Vecino más cercano: **1.3%**

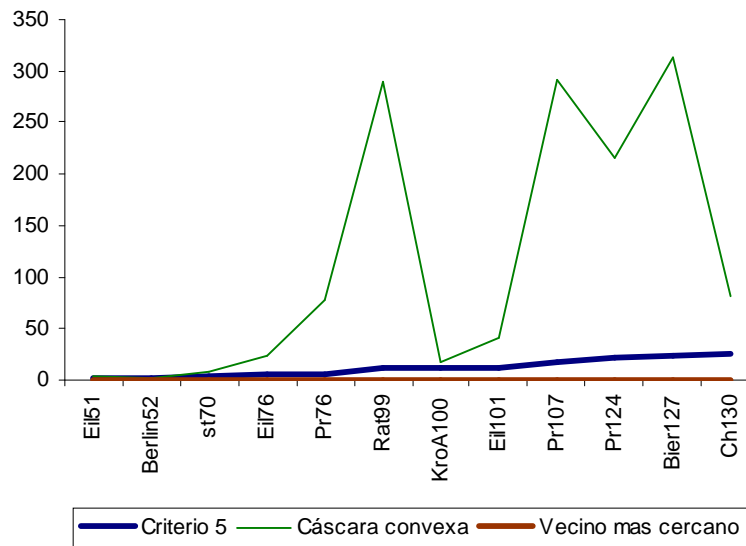
Promedio de tiempo de ejecución Vecino más cercano: **< 1**

Porcentajes sobre mejor solución conocida (TSPLIB)



Este criterio de selección es una variante del anterior y se aprecia un comportamiento análogo al mismo. La variación observada es casi despreciable y del orden del 1% con respecto a la propuesta 4.

Comparación de tiempos de ejecución

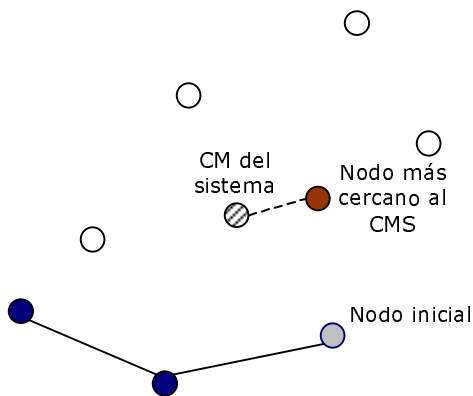


En este caso se aprecia un comportamiento análogo a los casos anteriores con respecto a los tiempos de ejecución.

### 4.3.6 Selección 6: el más cercano al $CM_S$

Para este criterio de selección, el CM calculado es el CM del sistema,  $CM_S$ , y es invariante a lo largo de toda la ejecución de la heurística, por lo tanto basta con calcularlo una vez al principio y se podrá utilizar durante toda la ejecución. En cada paso se selecciona el nodo más cercano al  $CM_S$ , de los que no han sido ya visitados. La idea de aplicar este criterio de selección es que durante toda la ejecución de la heurística se tenga como referencia el mismo punto, invariante, y de esta manera realizar una selección en "amplitud" con respecto al  $CM_S$ , que es donde se espera haya una mayor "concentración" de nodos.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



En este caso, como el orden de selección puede quedar determinado desde el principio el orden que aporta el criterio de selección a toda la ejecución es el necesario para ordenar a todos los nodos en orden creciente de distancia al  $CM_S$ . Existen algoritmos de ordenación de  $O(n \log n)$  [EDA] pág. 253, por lo tanto en este caso, al aplicar un algoritmo de ordenación con estas características se puede lograr un orden menor de ejecución que los criterios anteriores.

### Construcción de una heurística utilizando el criterio de selección 6

En este caso utilizaremos el criterio de selección de nodo inicial más cercano al  $CM_S$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	559	31	1.4
Berlin52	52	7542	9188	21	2.3
st70	70	675	855	26	3.8
Eil76	76	538	692	28	5
Pr76	76	108159	133884	23	4.6
Rat99	99	1211	1489	22	10.3
KroA100	100	21282	26277	23	11.2
Eil101	101	629	804	27	11.3
Pr107	107	44303	47229	6	13.1
Pr124	124	59030	68139	15	20.4
Bier127	127	118282	144851	22	21.6
Ch130	130	6110	7763	27	23.2
Pr136	136	96772	11827	12	28.8
Pr144	144	58537	66672	13	32.7
Pr152	152	73682	87286	18	43.6

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM - Selección 6"

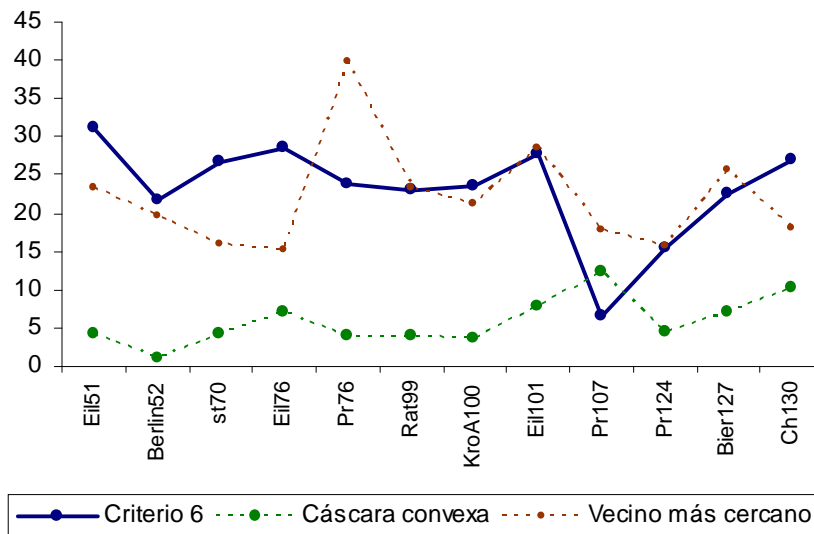
Promedio del % de exceso sobre el óptimo conocido: **20%**  
 Promedio de tiempo de ejecución: **15.5 seg.**

Comparaciones								
Nombre Instancia	# Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	559	444	25	3.2	525	6	< 1
Berlin52	52	9188	7624	20	2.4	9014	1	< 1
st70	70	855	704	21	7.4	783	9	< 1
Eil76	76	692	576	20	22.6	619	11	< 1
Pr76	76	133884	112479	19	78.2	151142	-11	< 1
Rat99	99	1489	1258	18	289.3	1493	0.2	< 1
KroA100	100	26277	22055	19	17.7	25781	1	< 1
Eil101	101	804	679	18	40.4	808	0	< 1
Pr107	107	47229	49695	-5	290.8	52145	-9	< 1
Pr124	124	68139	61627	10	216.2	68321	0	< 1
Bier127	127	144851	126650	14	313.7	148512	-2	< 1
Ch130	130	7763	6731	15	81.5	7216	7	< 1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

Promedio del % de exceso sobre la Cáscara Convexa: **16.1%**  
 Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**  
 Promedio del % de exceso sobre el Vecino más cercano: **1.0%**  
 Promedio de tiempo de ejecución Vecino más cercano: **< 1**

Porcentajes sobre mejor solución conocida (TSPLIB)



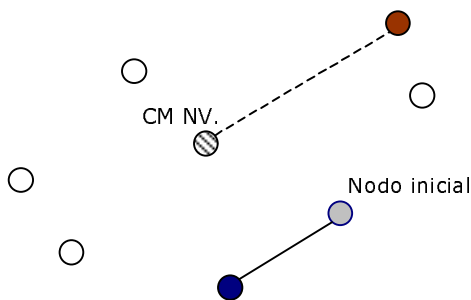
También en este caso se aprecia un comportamiento análogo a los casos anteriores con respecto a los tiempos de ejecución.



### 4.3.7 Selección 7: el más lejano al $CM_{NV}$

En el proceso de selección, se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y selecciono el nodo más lejano a su centro de masas, es decir a  $CM_{NV}$ . La idea de aplicar este criterio de selección es ir "estabilizando" el  $CM_{NV}$  al visitar a los mas alejados haciendo "converger" a la selección al  $CM_{NV}$  a medida que se suceden las iteraciones.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Realizando un cálculo análogo al del criterio 1 se concluye que el orden de ejecución que aporta el criterio de selección 7 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

### Construcción de una heurística utilizando el criterio de selección 7

En este caso utilizaremos el criterio de selección de nodo inicial más lejano al  $CM_{NV}$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	472	10	4.1
Berlin52	52	7542	8421	11	4.3
st70	70	675	792	17	51.5
Eil76	76	538	605	12	13.3
Pr76	76	108159	115984	7	13.2
Rat99	99	1211	1360	12	30.9
KroA100	100	21282	27333	28	30.8
Eil101	101	629	718	14	31.4
Pr107	107	44303	50936	14	47.1
Pr124	124	59030	74327	25	66.4
Bier127	127	118282	138585	17	75.3
Ch130	130	6110	7005	14	83.6
Pr136	136	96772	106795	10	82.2
Pr144	144	58537	65672	12	96.2
Pr152	152	73682	78508	6	123

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM – Selección 7"

Promedio del % de exceso sobre el óptimo conocido: **13.9%**

Promedio de tiempo de ejecución: **50,2 seg.**

Comparaciones								
Nombre Instancia	# Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	472	444	6	3.2	525	-20	< 1
Berlin52	52	8421	7624	10	2.4	9014	-18	< 1
st70	70	792	704	12	7.4	783	-4	< 1
Eil76	76	605	576	5	22.6	619	-7	< 1
Pr76	76	115984	112479	3	78.2	151142	-29	< 1
Rat99	99	1360	1258	9	289.3	1493	-15	< 1
KroA100	100	27333	22055	23	17.7	25781	-3	< 1
Eil101	101	718	679	5	40.4	808	-10	< 1
Pr107	107	50936	49695	2	290.8	52145	6	< 1
Pr124	124	74327	61627	20	216.2	68321	-3	< 1
Bier127	127	138585	126650	9	313.7	148512	-0.4	< 1
Ch130	130	7005	6731	4	81.5	7216	-20	< 1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

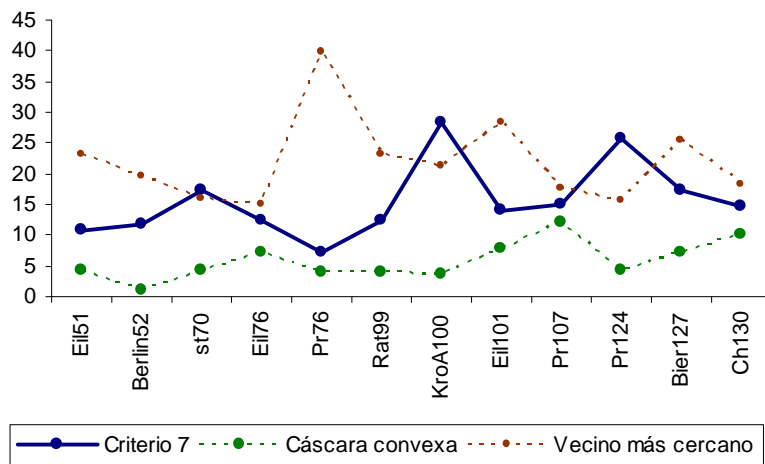
Promedio del % de exceso sobre la Cáscara Convexa: **9.0%**

Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**

Promedio del % de exceso sobre el Vecino más cercano: **-11.1%**

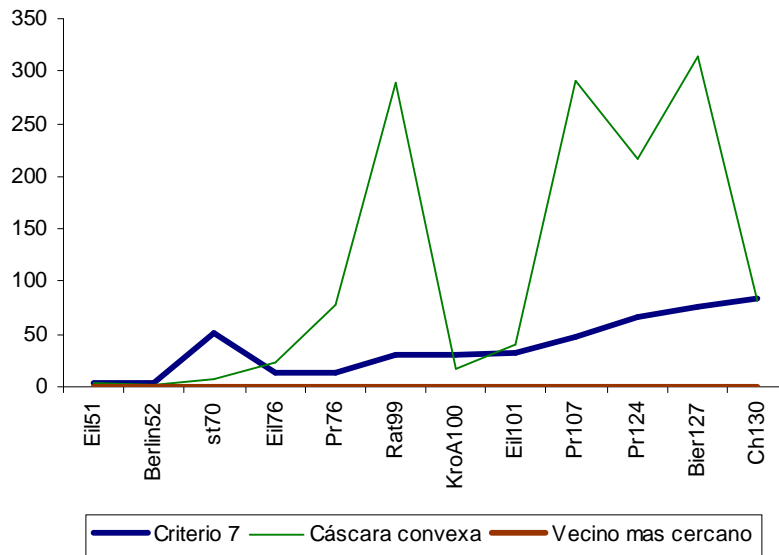
Promedio de tiempo de ejecución Vecino más cercano: **< 1**

Porcentajes sobre mejor solución conocida (TSPLIB)



En este caso podemos apreciar una estabilidad en los resultados obtenidos al compararlos con las mejores soluciones conocidas. De todas formas podemos ver que la media (13.9%) de exceso es menor que en la aplicación de los criterios que consideraban la cercanía con el CM.

### Comparación de tiempos de ejecución

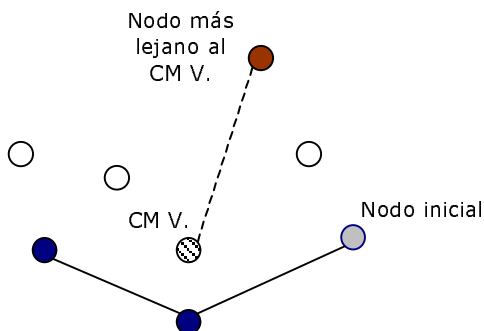


Al observar los valores de tiempo se puede ver que en general es proporcional a la cantidad de nodos, al igual que en los casos anteriores. De todas maneras se nota un incremento en los tiempos de ejecución superando en algunos casos el tiempo requerido para la ejecución al de la cáscara convexa, caso que no se había dado hasta ahora en los primeros seis criterios ejecutados.

#### 4.3.8 Selección 8: el más lejano al $CM_V$

En el proceso de selección, se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se selecciona el nodo más cercano a su centro de masas, es decir a  $CM_V$ . La idea de aplicar este criterio de selección es ir "estabilizando" el  $CM_V$  al visitar a los mas alejados haciendo "converger" a la selección al  $CM_V$  a medida que se suceden la iteraciones.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Realizando un cálculo análogo al expuesto en criterio 1 se puede ver que el orden de ejecución que aporta el criterio de selección 8 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

### Construcción de una heurística utilizando el criterio de selección 8

En este caso utilizaremos el criterio de selección de nodo inicial más lejano al  $CM_s$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	494	15	5
Berlin52	52	7542	8521	12	4.6
st70	70	675	770	14	11.6
Eil76	76	538	603	12	16.6
Pr76	76	108159	115906	7	16.5
Rat99	99	1211	1380	13	36.4
KroA100	100	21282	24511	15	37.3
Eil101	101	629	712	13	36.6
Pr107	107	44303	46174	4	44.7
Pr124	124	59030	63883	8	66.6
Bier127	127	118282	135518	14	72
Ch130	130	6110	7035	15	77.3
Pr136	136	96772	105282	8	85.6
Pr144	144	58537	61744	5	104.6
Pr152	152	73682	78739	6	116.5

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM - Selección 8"

Promedio del % de exceso sobre el óptimo conocido: **10.7%**

Promedio de tiempo de ejecución: **48.8 seg.**

Nombre Instancia	#Ciudades	Comparaciones						
		Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	494	444	11	3.2	525	-13	< 1
Berlin52	52	8521	7624	11	2.4	9014	-14	< 1
st70	70	770	704	9	7.4	783	-7	< 1
Eil76	76	603	576	4	22.6	619	-7	< 1
Pr76	76	115906	112479	3	78.2	151142	-22	< 1
Rat99	99	1380	1258	9	289.3	1493	-11	< 1
KroA100	100	24511	22055	11	17.7	25781	-13	< 1
Eil101	101	712	679	4	40.4	808	-10	< 1
Pr107	107	46174	49695	-7	290.8	52145	-3	< 1
Pr124	124	63883	61627	3	216.2	68321	-17	< 1
Bier127	127	135518	126650	7	313.7	148512	-2	< 1
Ch130	130	7035	6731	4	81.5	7216	-16	< 1

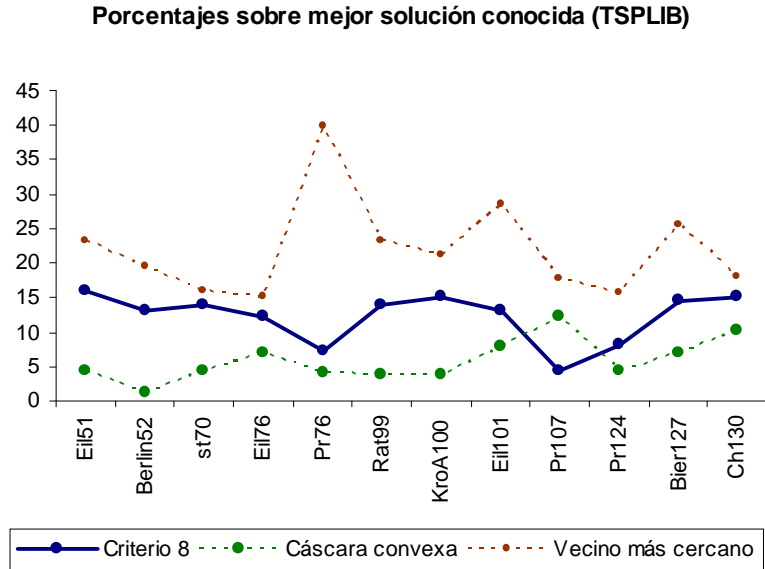
*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

Promedio del % de exceso sobre la Cáscara Convexa: **5.75%**

Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**

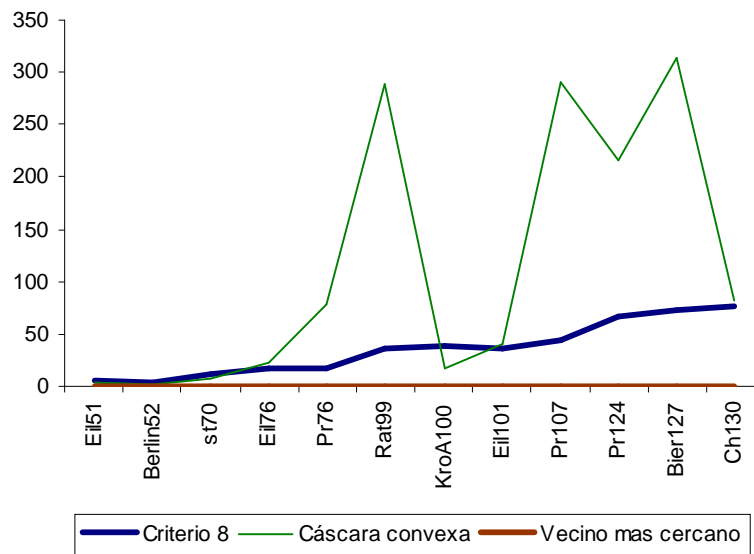
Promedio del % de exceso sobre el Vecino más cercano: **-11.2%**

Promedio de tiempo de ejecución Vecino más cercano: < 1



En este caso podemos apreciar una mayor estabilidad de los resultados con respecto de la mejor solución conocida que en el caso anterior. Es de notar que en todos los casos de prueba se superó en eficacia de la solución al algoritmo del vecino más cercano, y en el mismo caso (Pr107) que en la mayoría de las ejecuciones con la aplicación de los criterios anteriores se superó al algoritmo de cáscara convexa.

**Comparación de tiempos de ejecución**



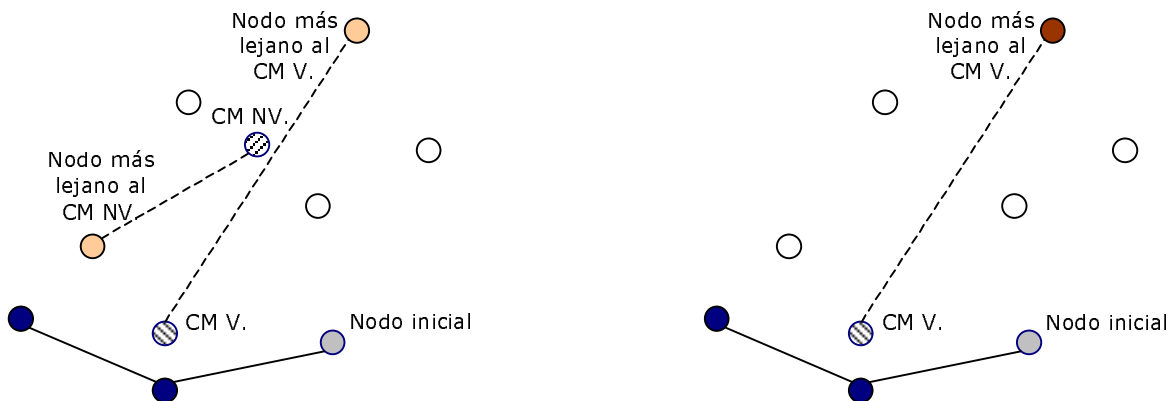
Nuevamente se puede apreciar una proporcionalidad entre la cantidad de nodos y el tiempo requerido para la ejecución. Es de notar también que en la aplicación de este

criterio de selección se logró mejorar la solución con respecto a los primeros criterios expuestos, pero a un costo de mayor tiempo de procesamiento.

#### 4.3.9 Selección 9: el más lejano al $CM_V$ o $CM_{NV}$

Este criterio de selección toma las ideas del criterio 3 intentando adaptarse a la mejor selección, pero esta vez de los criterios de selección 7 y 8 en vez de los criterios de selección 1 y 2. Se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y se calcula el nodo más cercano a su centro de masas, es decir a  $CM_{NV}$ . Luego se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se calcula el nodo más cercano a su centro de masas, es decir a  $CM_V$ . De ambos nodos calculados se selecciona el que a mayor distancia se encuentre del CM correspondiente.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Realizando un cálculo análogo al expuesto en el criterio 3 se puede ver que el orden de ejecución que aporta el criterio de selección 9 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

#### Construcción de una heurística utilizando el criterio de selección 9

En este caso utilizaremos el criterio de selección de nodo inicial más lejano al  $CM_S$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	494	15	4.5
Berlin52	52	7542	8458	12	4.9
st70	70	675	748	10	12.2
Eil76	76	538	603	12	15.2
Pr76	76	108159	115906	7	16.9
Rat99	99	1211	1375	13	36.5
KroA100	100	21282	24650	15	35.6
Eil101	101	629	711	13	34.9
Pr107	107	44303	46174	4	42.5
Pr124	124	59030	63261	7	73.9
Bier127	127	118282	137866	16	71.5
Ch130	130	6110	6988	14	80.6
Pr136	136	96772	105282	8	81.4
Pr144	144	58537	61744	5	117.3
Pr152	152	73682	78379	6	130

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM - Selección 9"

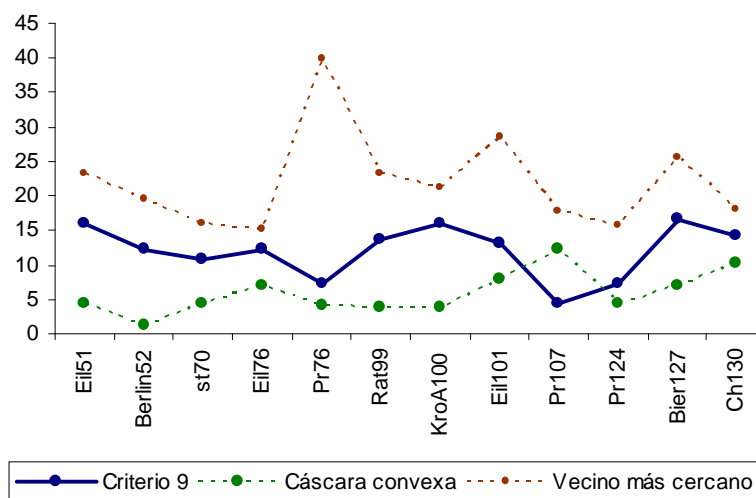
Promedio del % de exceso sobre el óptimo conocido: **10.4%**  
 Promedio de tiempo de ejecución: **50.5 seg.**

Comparaciones								
Nombre Instancia	#Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	494	444	11	3.2	525	-13	< 1
Berlin52	52	8458	7624	10	2.4	9014	-15	< 1
st70	70	748	704	6	7.4	783	-9	< 1
Eil76	76	603	576	4	22.6	619	-7	< 1
Pr76	76	115906	112479	3	78.2	151142	-22	< 1
Rat99	99	1375	1258	9	289.3	1493	-12	< 1
KroA100	100	24650	22055	11	17.7	25781	-12	< 1
Eil101	101	711	679	4	40.4	808	-10	< 1
Pr107	107	46174	49695	-7	290.8	52145	-3	< 1
Pr124	124	63261	61627	2	216.2	68321	-18	< 1
Bier127	127	137866	126650	8	313.7	148512	0	< 1
Ch130	130	6988	6731	3	81.5	7216	-17	< 1

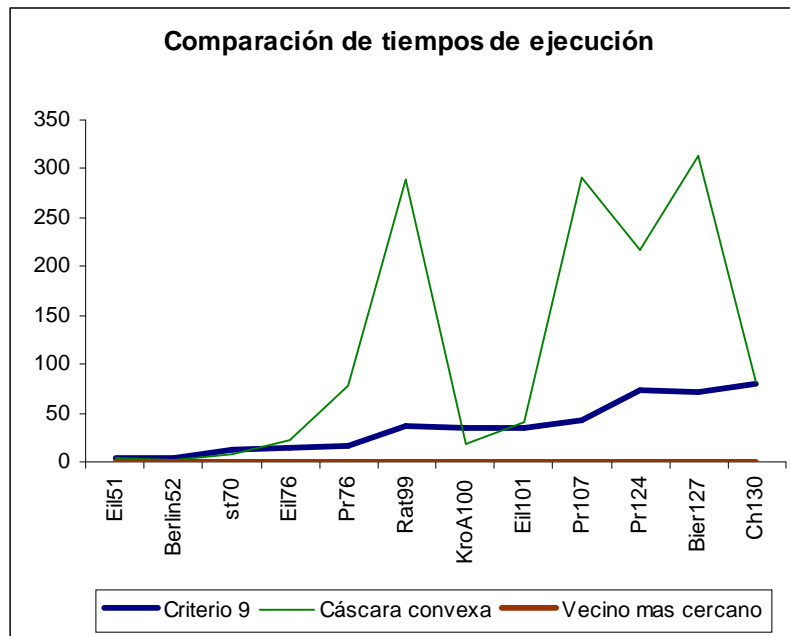
Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

Promedio del % de exceso sobre la Cáscara Convexa: **5.3%**  
 Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**  
 Promedio del % de exceso sobre el Vecino más cercano: **-11.5%**  
 Promedio de tiempo de ejecución Vecino más cercano: **< 1**

Porcentajes sobre mejor solución conocida (TSPLIB)



La aplicación de este criterio muestra un comportamiento similar a los anteriores destacándose nuevamente la estabilidad de las soluciones y las soluciones encontradas.



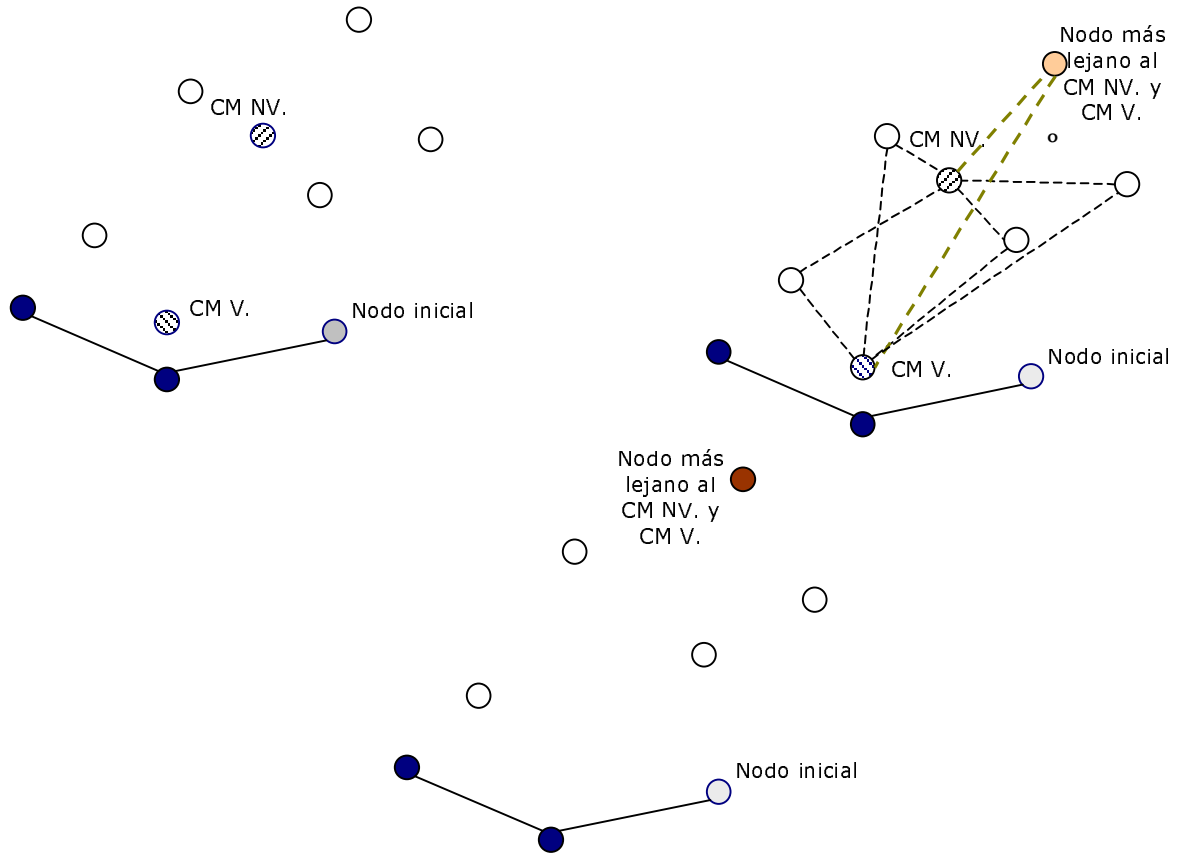
En este caso se aprecia un comportamiento análogo a los casos anteriores con respecto a los tiempos de ejecución.

#### 4.3.10 Selección 10: el más lejano al $CM_V$ y $CM_{NV}$

Éste y el siguiente criterio son variantes del anterior, el más lejano del  $CM_V$  o  $CM_{NV}$ . Estos criterios intentan "estabilizar" el área del mapa en donde se seleccionan los nodos. En el criterio anteriormente expuesto los nodos "compiten" para ver quién es el que está más lejos de algún CM de los considerados. Dependiendo de quien "gane" el siguiente nodo seleccionado estará más lejos de uno u otro. En estos criterios ambos CM "pesan" en la selección del siguiente nodo y por lo tanto se espera que el área en la cual se encuentre el nodo seleccionado no varíe "mucho" de una iteración a la otra. Se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y se calcula el nodo más lejano a su centro de masas, es decir a  $CM_{NV}$ . Luego se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se calcula el nodo más lejano a su centro de masas, es decir a  $CM_V$ . Se selecciona el nodo tal que la suma de las distancias a los centros de masas calculados sea máxima.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:





Realizando un cálculo análogo al expuesto en criterio 4 se puede ver que el orden de ejecución que aporta el criterio de selección 10 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

### Construcción de una heurística utilizando el criterio de selección 10

En este caso utilizaremos el criterio de selección de nodo inicial más lejano al CM<sub>s</sub> y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	494	15	4.1
Berlin52	52	7542	8513	12	4.4
st70	70	675	770	14	10.2
Eil76	76	538	606	12	13.7
Pr76	76	108159	115661	6	13.6
Rat99	99	1211	1381	14	31
KroA100	100	21282	24489	15	32.7
Eil101	101	629	712	13	33.2
Pr107	107	44303	46515	4	40.5
Pr124	124	59030	64195	8	65.5
Bier127	127	118282	138131	16	70.3
Ch130	130	6110	7035	15	73.7
Pr136	136	96772	107544	11	102.3
Pr144	144	58537	61744	5	112
Pr152	152	73682	77198	4	131.6

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM - Selección 10"

Promedio del % de exceso sobre el óptimo conocido: **10.9%**

Promedio de tiempo de ejecución: **49.25 seg.**

Comparaciones								
Nombre Instancia	#Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	494	444	11	3.2	525	-15	< 1
Berlin52	52	8513	7624	11	2.4	9014	-17	< 1
st70	70	770	704	9	7.4	783	-18	< 1
Eil76	76	606	576	5	22.6	619	-7	< 1
Pr76	76	115661	112479	9	78.2	151142	-30	< 1
Rat99	99	1381	1258	9	289.3	1493	-1	< 1
KroA100	100	24489	22055	11	17.7	25781	-15	< 1
Eil101	101	712	679	4	40.4	808	-11	< 1
Pr107	107	46515	49695	-6	290.8	52145	-2	< 1
Pr124	124	64195	61627	4	216.2	68321	-25	< 1
Bier127	127	138131	126650	9	313.7	148512	-9	< 1
Ch130	130	7035	6731	4	81.5	7216	-25	< 1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

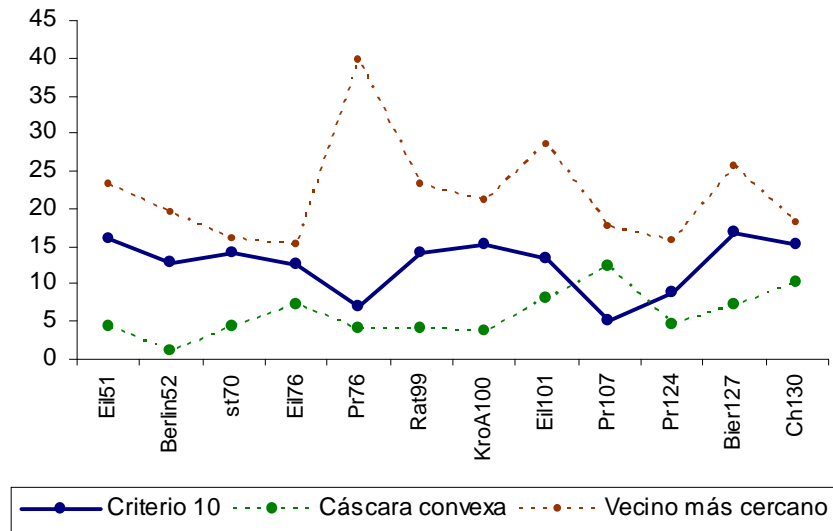
Promedio del % de exceso sobre la Cáscara Convexa: **6.6%**

Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**

Promedio del % de exceso sobre el Vecino más cercano: **14.5%**

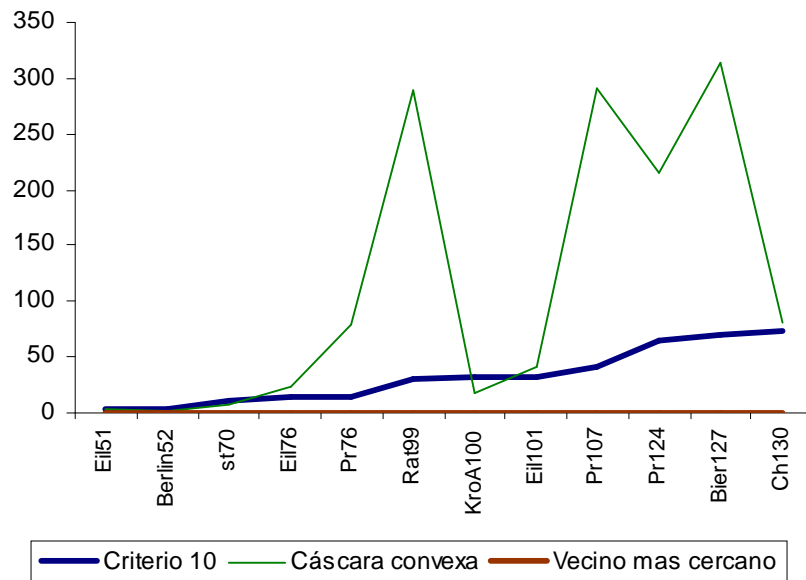
Promedio de tiempo de ejecución Vecino más cercano: **< 1**

**Porcentajes sobre mejor solución conocida (TSPLIB)**



La aplicación de este criterio muestra un comportamiento similar a los anteriores destacándose nuevamente la estabilidad de las soluciones y las soluciones encontradas.

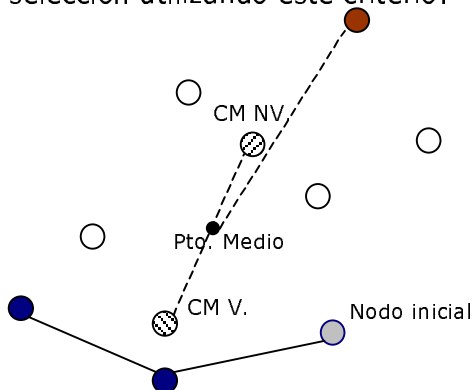
**Comparación de tiempos de ejecución**



Para este caso se aprecia un comportamiento análogo a los casos anteriores con respecto a los tiempos de ejecución.

#### 4.3.11 Selección 11: el más lejano al punto medio entre $CM_V$ y $CM_{NV}$

Se considera el conjunto de nodos que aún no han sido seleccionados (no están marcados como visitados) y se calcula el nodo más lejano a su centro de masas, es decir a  $CM_{NV}$ . Luego se considera el conjunto de nodos que ya han sido seleccionados (están marcados como visitados) y se calcula el nodo más lejano a su centro de masas, es decir a  $CM_V$ . Se selecciona el nodo más lejano al punto medio entre los dos  $CM$  calculados. En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Realizando un cálculo análogo al expuesto en criterio 5 se puede ver que el orden de ejecución que aporta el criterio de selección 11 en la aplicación de esta heurística es  $O(n^2)$ , siendo  $n$  la cantidad del nodo del sistema.

#### Construcción de una heurística utilizando el criterio de selección 11

En este caso utilizaremos el criterio de selección de nodo inicial más lejano al  $CM_S$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	494	15	4.3
Berlin52	52	7542	8513	12	5.1
st70	70	675	770	14	11.5
Eil76	76	538	606	12	16.5
Pr76	76	108159	115661	7	15.5
Rat99	99	1211	1349	11	33.3
KroA100	100	21282	24489	15	34.1
Eil101	101	629	712	13	33.7
Pr107	107	44303	46174	4	40.7
Pr124	124	59030	64195	8	64.4
Bier127	127	118282	138131	16	65.4
Ch130	130	6110	7035	15	70.9
Pr136	136	96772	107544	11	93.8
Pr144	144	58537	61744	5	101.9
Pr152	152	73682	78539	6	119

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM - Selección 11"

Promedio del % de exceso sobre el óptimo conocido: **10.9%**  
 Promedio de tiempo de ejecución: **47.34 seg.**

Comparaciones								
Nombre Instancia	# Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	494	444	11	3.2	525	-15	< 1
Berlin52	52	8513	7624	11	2.4	9014	-17	< 1
st70	70	770	704	9	7.4	783	-7	< 1
Eil76	76	606	576	5	22.6	619	-7	< 1
Pr76	76	115661	112479	2	78.2	151142	-30	< 1
Rat99	99	1349	1258	7	289.3	1493	-15	< 1
KroA100	100	24489	22055	11	17.7	25781	-15	< 1
Eil101	101	712	679	4	40.4	808	-11	< 1
Pr107	107	46174	49695	-7	290.8	52145	-3	< 1
Pr124	124	64195	61627	4	216.2	68321	-20	< 1
Bier127	127	138131	126650	9	313.7	148512	-9	< 1
Ch130	130	7035	6731	4	81.5	7216	-25	< 1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

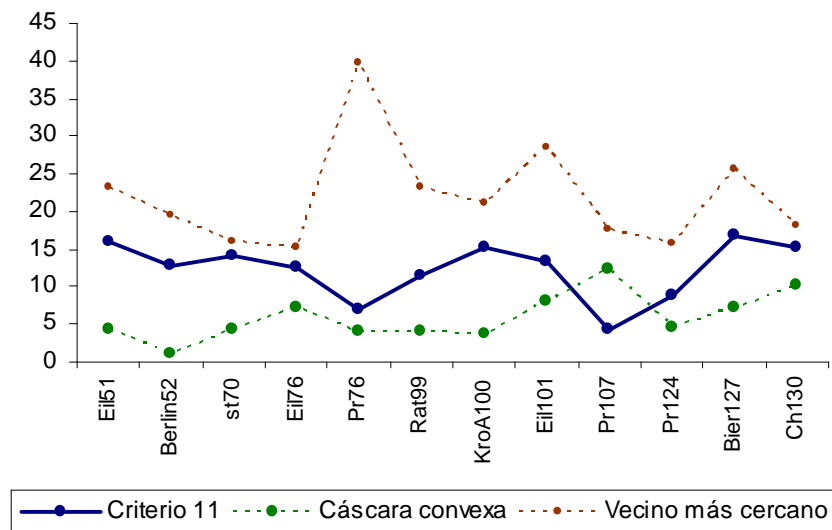
Promedio del % de exceso sobre la Cáscara Convexa: **5.8%**

Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**

Promedio del % de exceso sobre el Vecino más cercano: **-14.5%**

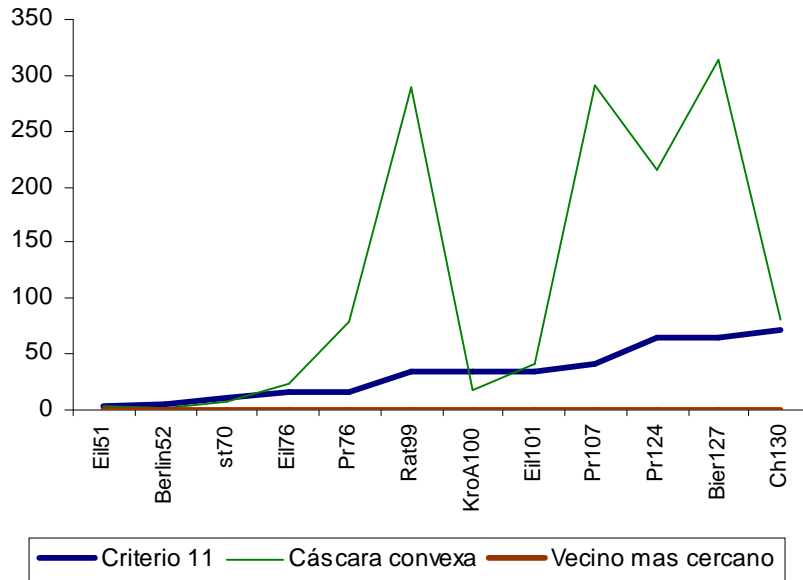
Promedio de tiempo de ejecución Vecino más cercano: **< 1**

Porcentajes sobre mejor solución conocida (TSPLIB)



El criterio de selección aplicado en esta ejecución es una variante del anterior y se aprecia un comportamiento análogo al mismo. La variación observada es casi despreciable y del orden del 1% con respecto a la propuesta 10.

### Comparación de tiempos de ejecución



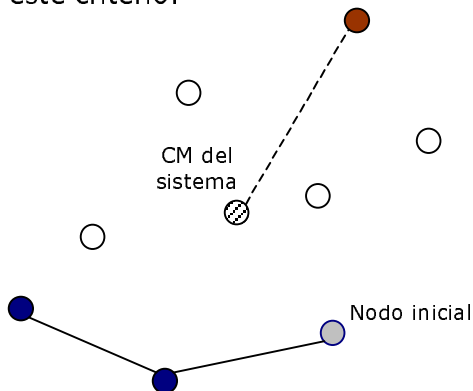
En este caso se aprecia un comportamiento análogo a los casos anteriores con respecto a los tiempos de ejecución.

#### 4.3.12 Selección 12: el más lejano al $CM_S$

El CM calculado es el CM del sistema,  $CM_S$ , y es invariante a lo largo de toda la ejecución de la metaheurística, por lo tanto basta con calcularlo una vez al principio y se podrá utilizar durante toda la ejecución. En cada paso se selecciona el nodo más lejano al  $CM_S$ , de los que no han sido ya visitados.

La idea de aplicar este criterio de selección es que durante toda la ejecución de la metaheurística se tenga como referencia el mismo punto, invariante, y de esta manera realizar una selección en "amplitud" pero de manera de "afuera hacia adentro" con respecto al  $CM_S$ , y de "adentro hacia fuera" como el criterio 6.

En la siguiente figura se muestra una representación gráfica de la selección utilizando este criterio:



Realizando un cálculo análogo al expuesto en criterio 6 se puede ver que el orden de ejecución que aporta el criterio de selección 12 en la aplicación de esta heurística es  $O(n \log n)$ , siendo  $n$  la cantidad del nodo del sistema.

### Construcción de una heurística utilizando el criterio de selección 12

En este caso utilizaremos el criterio de selección de nodo inicial más lejano al  $CM_s$  y el criterio de inserción "en la mejor ubicación".

Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)	Costo obtenido	% exceso	Tiempo (seg.)
Eil51	51	426	494	15	4
Berlin52	52	7542	8521	12	4.4
st70	70	675	770	14	12.2
Eil76	76	538	603	12	13.3
Pr76	76	108159	115906	7	14.5
Rat99	99	1211	1380	13	33.2
KroA100	100	21282	24511	15	34.8
Eil101	101	629	712	13	35.5
Pr107	107	44303	46174	4	42.2
Pr124	124	59030	63883	8	63.9
Bier127	127	118282	135518	14	70.5
Ch130	130	6110	7035	15	78.5
Pr136	136	96772	105282	8	83.9
Pr144	144	58537	61744	5	98.1
Pr152	152	73682	78379	6	118.3

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "TSP - Heurísticas del CM - Selección 12"

Promedio del % de exceso sobre el óptimo conocido: **10.7%**

Promedio de tiempo de ejecución: **47.15 seg.**

Comparaciones								
Nombre Instancia	#Ciudades	Nuestra Solución	Solución Cáscara Convexa	% exceso	Tiempo (seg.)	Solución Vecino más cercano	% exceso	Tiempo (seg.)
Eil51	51	494	444	11	3.2	525		< 1
Berlin52	52	8521	7624	11	2.4	9014	-17	< 1
st70	70	770	704	9	7.4	783	-7	< 1
Eil76	76	603	576	4	22.6	619	-7	< 1
Pr76	76	115906	112479	3	78.2	151142	-29	< 1
Rat99	99	1380	1258	9	289.3	1493	-13	< 1
KroA100	100	24511	22055	11	17.7	25781	-15	< 1
Eil101	101	712	679	4	40.4	808	-11	< 1
Pr107	107	46174	49695	-7	290.8	52145	-3	< 1
Pr124	124	63883	61627	3	216.2	68321	-20	< 1
Bier127	127	135518	126650	7	313.7	148512	-2	< 1
Ch130	130	7035	6731	4	81.5	7216	-20	< 1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "TSP - Heurísticas del CM - Heurística de la cáscara convexa" y "TSP - Heurísticas del CM - Heurística del vecino más cercano" respectivamente.

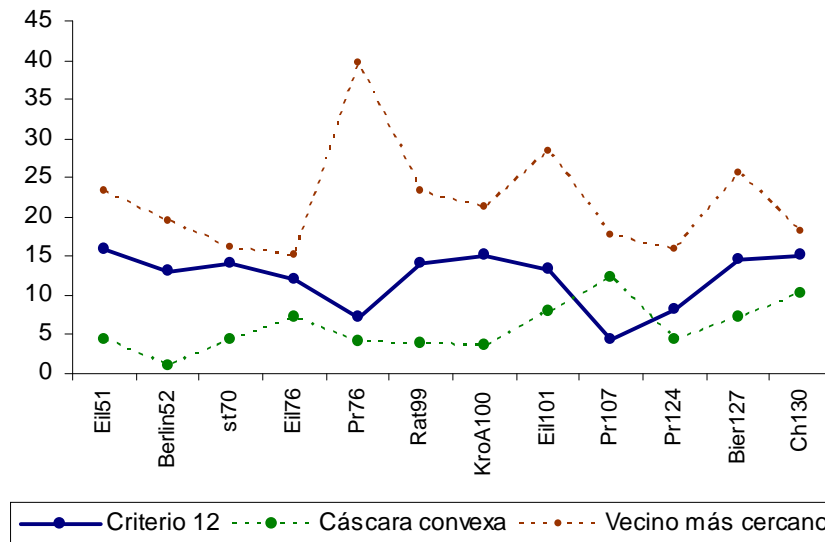
Promedio del % de exceso sobre la Cáscara Convexa: **5.7%**

Promedio de tiempo de ejecución Cáscara Convexa: **113.6 seg.**

Promedio del % de exceso sobre el Vecino más cercano: **-13.2%**

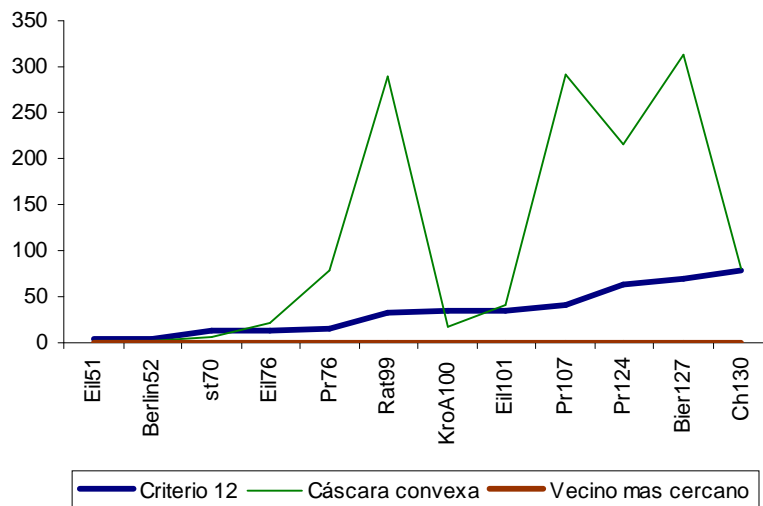
Promedio de tiempo de ejecución Vecino más cercano: < 1

Porcentajes sobre mejor solución conocida (TSPLIB)



En esta última aplicación de un criterio de selección utilizando la lejanía a un CM no se aprecian cambios destacados al comparar su comportamiento con las cinco anteriores.

Comparación de tiempos de ejecución



También en este caso se aprecia un comportamiento análogo a los casos anteriores con respecto a los tiempos de ejecución.



## 4.4 Análisis de resultados y conclusiones

En la siguiente tabla se resumen todos los valores promedio de los porcentajes de exceso obtenidos en la aplicación de las distintas heurísticas:

Criterio de selección de nodo inicial	Criterio de selección de nodo	Criterio de inserción	% de exceso promedio sobre la mejor solución conocida (TSPLIB95)	% de exceso promedio sobre la cáscara Convexa	% de exceso promedio sobre el vecino más cercano	
El más cercano al centro de masas	Criterio 1	En la mejor ubicación	21	16	1	
	Criterio 2		21	15.8	0.9	
	Criterio 3		23	18.1	3.0	
	Criterio 4		22	18.2	2.2	
	Criterio 5		21	16.6	1.3	
	Criterio 6		20	16.1	1.0	
El más lejano al centro de masas	Criterio 7			13.9	9.0	-11.1
	Criterio 8			10.7	5.75	-11.2
	Criterio 9			10.4	5.3	-11.2
	Criterio 10			10.9	6.6	-14.5
	Criterio 11			10.9	5.8	-14.5
	Criterio 12			10.7	5.7	-13.2

En función de los resultados obtenidos observamos que no hay gran variación en los porcentajes promedio sobre la mejor solución conocida al aplicar las seis primeras propuestas que utilizan la idea de la cercanía al centro de masa. Se ve un comportamiento estable en los resultados, existiendo casos particulares en los cuales, dependiendo de la topología, se obtienen resultados alejados de la media.

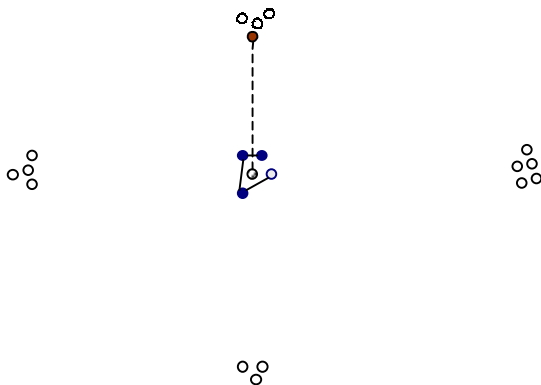
Con respecto al tiempo de ejecución, se puede ver que en todos los casos tuvo un comportamiento creciente proporcional a la cantidad de nodos considerados. Comparando los tiempos de ejecución con los observados para la cáscara convexa se puede ver que se obtienen resultados más eficientes en promedio, e independientes de la topología.

En cuanto a los criterios de selección que utilizan la idea de lejanía del centro de masas y que fueron motivados por mejorar la estabilidad de los mismos, se observó una mejoría en relación a las seis primeras propuestas. Al realizar el análisis de variación de los centros de masa, en la ejecución de las heurísticas que aplican estos criterios, se puede ver en los casos estudiados que es despreciable con respecto a los otros seis.

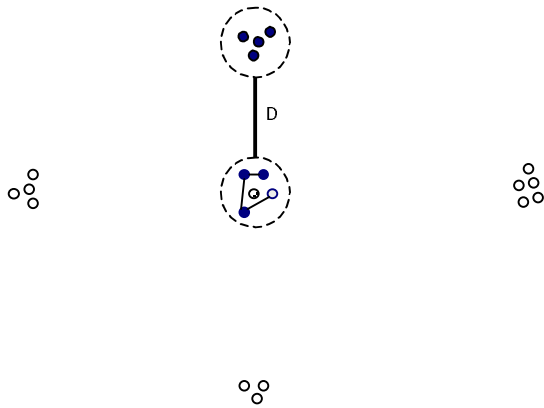
Al observar los resultados obtenidos al aplicar los cinco primeros criterios de selección y no ser tan buenos como lo esperado, atribuimos la causa a la variación que

experimentaban los CM de los nodos visitados y no visitados durante la ejecución de la heurística. Luego al aplicar el criterio 6 que dejaba invariante (fijo) el CM considerado y no obtener mejoras sobre los cinco anteriores atribuimos la deficiencia de ese criterio a que el punto de referencia no se adecuaba al avance de la ejecución como era lo esperado en la utilización de los criterios por cercanía al CM. Pensando en esto es que se propusieron los seis criterios de selección con la idea de lejanía del CM. Al haber observado los resultados obtenidos en la utilización de los criterios 7 al 11 pensamos que habíamos logrado estabilizar adecuadamente los CM durante la ejecución de las heurísticas ya que eran buenos en comparación con los obtenidos anteriormente. Al aplicar el criterio 12 se esperaba obtener resultados similares a los observados con el criterio 6 pero no fue así. Se vieron resultados que no variaban mucho a los obtenidos con los cinco criterios anteriores. Esto motivó un análisis de variación de los  $CM_V$  y  $CM_{NV}$  en la ejecución de los criterios 7 al 11 y se pudo ver que no variaban lo suficiente en comparación con el 12 como para hacer una diferencia en los resultados obtenidos. ¿Por qué entonces con los criterios que utilizaban la idea de lejanía del CM se obtienen mejores resultados que al aplicar los criterios que utilizan la idea de cercanía al CM? Un análisis en la topología de los casos estudiados aclara esta situación.

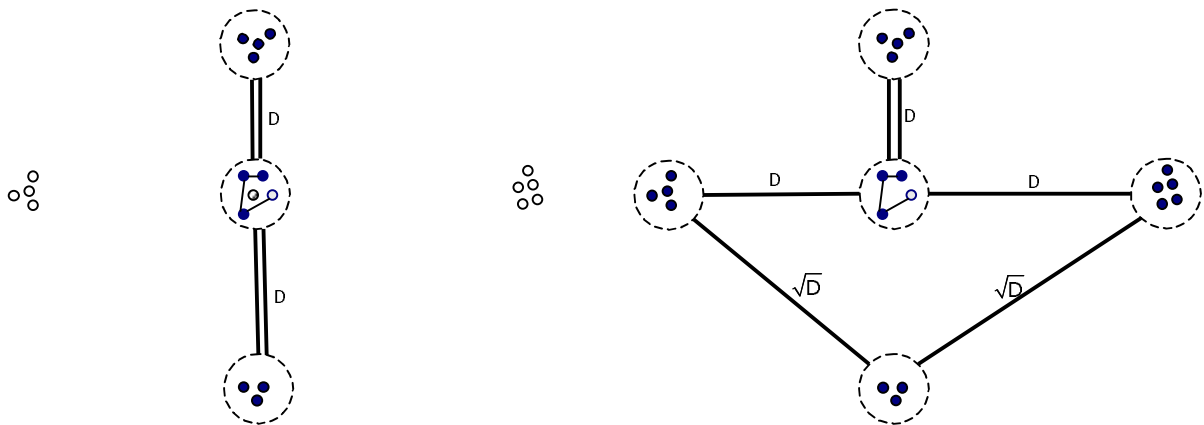
Para este análisis se utilizó un determinado caso de estudio con una topología particular. En él se puede ver que, en el caso de la selección del nodo en función de la cercanía al centro de masa se generan caminos no deseados en el momento de realizar el ruteo. Como se puede ver en la siguiente figura, el algoritmo comienza su ejecución en centro de masa del sistema de nodos, seleccionando y ruteando los nodos que se encuentran en el radio más próximo a este. Una vez que dentro de este radio no se encuentren más nodos sin seleccionar, el criterio procederá a buscar el más cercano dentro de los nodos que se encuentran más lejos con respecto al centro de masa.



El centro de masa se inclinará hacia el sector en donde se encuentra este nodo, lo que implicará que en las próximas selecciones se consideren todos los nodos que habitan en ese radio, generándose el subtour como se muestra la siguiente figura. En este análisis se considerará que las distancias que existen entre la concentración de nodos en el centro del esquema, y los cuatro sectores aledaños es semejante, y será referenciada como D. A su vez vamos a asumir que las diferencias en cuanto a los costos de ruteo son despreciables en cada sección.



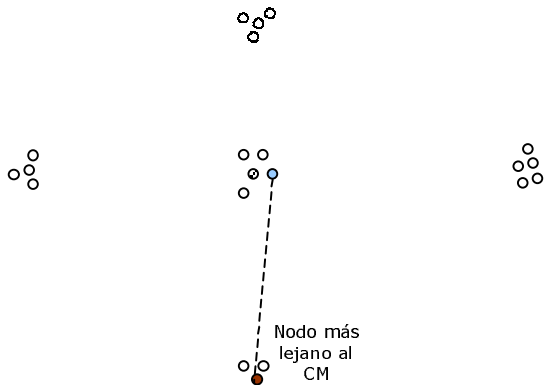
Los pasos sucesivos del algoritmo hacen que el próximo sector a ser considerado para la selección sea el ubicado en el extremo opuesto al sector previamente ruteado, lo que genera el subtour que se presenta en la figura de la izquierda. A partir de esta composición, los próximos sectores se seleccionan y rutean conformando el tour de la figura de la derecha.



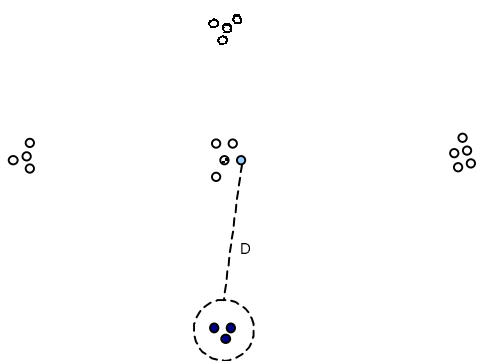
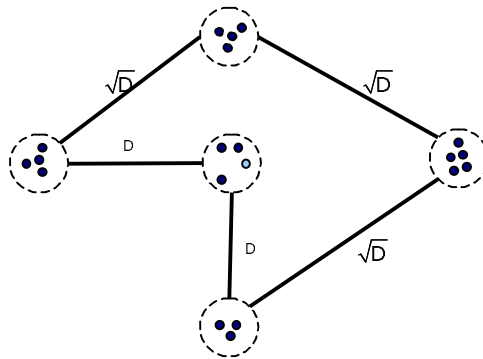
En este caso la distancia comprendida entre los sectores va a estar dada por la suma conformada por los caminos hallados:

$$4D + 2\sqrt{D}$$

Si ahora consideramos el caso de la selección de aquellos nodos más lejanos al centro de masa, primero se comenzará por seleccionar y rutear aquellos sectores que se encuentren alejados del centro de masa y por último se seleccionaran los nodos que se encuentren cerca de este, como se presenta la siguiente figura.



La siguientes figuras muestran, por un lado el comportamiento en los primeros pasos de ejecución (figura de la izquierda), y la culminación de este conformando el tour final desplegado en la figura de la derecha.



Para este caso, las distancias comprendidas entre los sectores van a estar dadas por la suma conformada por los caminos hallados:

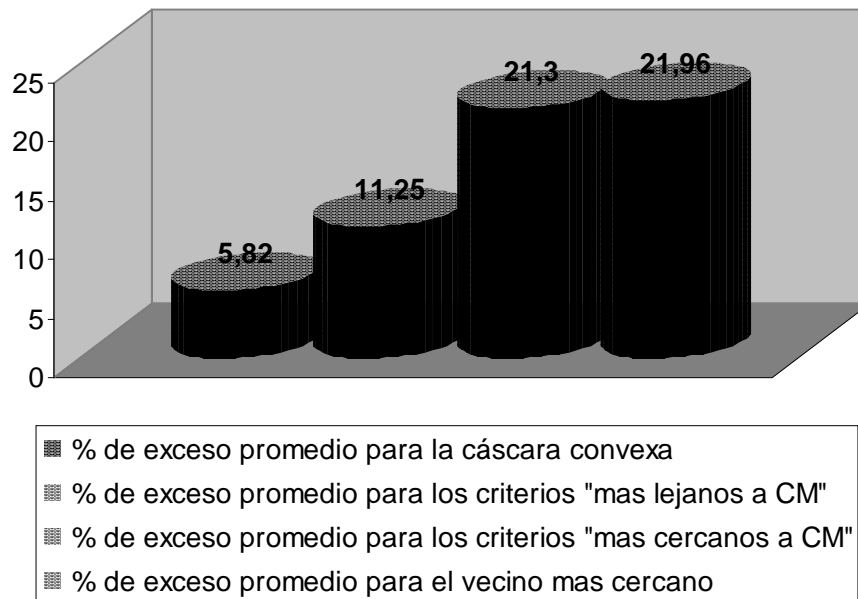
$$2D + 3 \sqrt{D}$$

Realizando la comparación de distancias contra el total obtenido para el caso del nodo más cercano al centro de masa, llegamos a la siguiente desigualdad que se presenta a continuación:

$$2D > \sqrt{D}$$

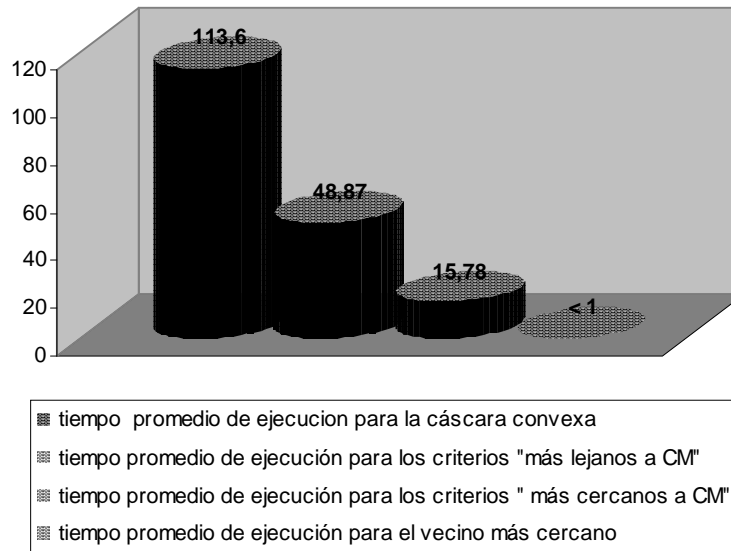
Esto nos está indicando que cuando exista este tipo de topología inmersa en el sistema de nodos, el costo de los subtours generados por el método del vecino más lejano podrá dar mejores resultados sobre los obtenidos por el método del centro de masa más lejano. Separando y promediando los resultados en dos grupos, uno para cada conjunto de criterios de selección (más cercanos y más lejanos a los CM considerados respectivamente) podemos ver la siguiente comparación:

**Comparación de % de exceso sobre la mejor solución conocida (TSPLIB)**



En la siguiente tabla se resumen todos los valores promedio de tiempos de ejecución obtenidos en la aplicación de las distintas heurísticas:

### Comparación de tiempos de ejecución



Como se puede apreciar en la gráfica, el tiempo de ejecución para los criterios que utilizan la idea de lejanía al CM considerado es mayor que los que utilizan la idea de cercanía al CM. Se puede ver una relación entre la eficacia lograda por las heurísticas que utilizan los criterios de lejanía al CM y el tiempo que estos utilizan para su ejecución en relación con el tiempo que utilizan las heurísticas que utilizan la idea de cercanía al CM. Las heurísticas que logran mejores resultados (son más eficaces) necesitan más tiempo para su ejecución (son menos eficientes). Al considerar el algoritmo de cáscara convexa se puede ver que también esta relación se mantiene al ser la que necesita más tiempo para su ejecución siendo la que logra los mejores resultados, como se puede apreciar en la anterior gráfica de comparación del % de exceso sobre la mejor solución conocida. Se destaca que la relación entre el tiempo que necesita la cáscara convexa para lograr la mejor solución entre las comparadas es menor, en promedio, al que necesitan las heurísticas que utilizan los criterios de selección del más lejano al CM. Logrando estas heurísticas encontrar soluciones cercanas a las encontradas por la cáscara convexa se desprende que la relación eficacia – eficiencia es mayor en un caso que en el otro:

Relación eficacia/eficiencia promedio (% de exceso promedio sobre mejor solución conocida / segundos promedio para su ejecución)	
Heurísticas que utilizan los criterios de selección por "lejanía al CM" (criterios 7 al 12)	23.02
Cáscara convexa	5.12

Se observa un tiempo de ejecución casi despreciable para el algoritmo del vecino más cercano, pero esto a costa de resultados muy pobres como se pudo apreciar en la anterior gráfica de comparación del % de exceso sobre la mejor solución conocida. En este caso se puede decir que se obtiene una alta eficiencia a costa de una baja eficacia.



## 5 Propuestas para la resolución del VRP

En este capítulo se presentan dos conjuntos de metaheurísticas para la resolución del VRP. El primero, basado en el tema de la microfísica cargas eléctricas, del cual se utilizó el concepto de la fuerza de atracción entre cargas eléctricas. El segundo basado en el tema campo eléctrico del cual se utilizó el concepto vector campo eléctrico. [FIS1]

Cada propuesta de resolución del VRP especifica una manera de encontrar un conjunto de tours que lo resuelven. Las metaheurísticas planteadas son incrementales, es decir que se va armando la solución y una vez que definimos un tour no iteramos sobre él para optimizarlo. Dentro de la "construcción" de cada tour consideramos dos criterios:

- Criterio de selección de nodo.
- Criterio de ruteo de nodos en un tour.

El criterio de selección de nodo indica cual será el siguiente nodo a ser seleccionado para el tour que se está construyendo. Una vez seleccionados todos los nodos pertenecientes a un tour, el segundo criterio indica el orden en que serán recorridos, esto es, correr un algoritmo de resolución del TSP sobre ese conjunto de nodos.

### 5.1 Metaheurísticas de fuerza de atracción

En el conjunto de metaheurísticas presentados en esta sección se aplica el concepto de fuerza de atracción entre cargas eléctricas. Si las cargas son opuestas (una positiva y otra negativa) experimentan una fuerza de atracción entre sí que es proporcional al valor de cada una e inversamente proporcional al cuadrado de la distancia entre ellas. Para calcular esta fuerza de atracción entre dos cargas se utiliza la ley de Coulomb que dice lo siguiente:

$$F = \frac{k \cdot q_1 \cdot q_2}{d^2} \quad (k = 8.9874 \times 10^9 \text{ N.m}^2/\text{C}^2) \quad [\text{FIS1}]$$

Siendo  $F$  la fuerza que experimentan entre sí las cargas,  $q_1$  y  $q_2$  el valor de cada una y  $d$  la distancia que hay entre ellas. Para un desarrollo mas detallado de la teoría de interacción entre cargas ver anexo "Conceptos físicos".

En la metaheurística propuesta utilizamos esta ley de la física al plantearnos una analogía entre el vehículo o camión con una carga positiva y de las ciudades o nodos con cargas negativas. El valor de la "carga vehículo" se considera constante e igual a 1 y el valor de las "cargas nodo" se consideran igual a su demanda. El valor de la "carga vehículo" se considera constante porque nos interesa que lo que influya en el cálculo de la fuerza sea la demanda de los nodos e igual a 1 simplemente para facilitar los cálculos. Las distancias entre nodos se calculan como la distancia en coordenadas cartesianas al igual que lo consideramos en el capítulo anterior.



Como a nosotros nos interesa comparar (y no cuantificar) la fuerza de atracción que hay entre el vehículo y los nodos no es necesario multiplicar por la constante  $K$ , ya que solo aporta un factor de escalamiento a las fuerzas ejercidas por los nodos al vehículo. Por lo tanto, teniendo en cuenta la analogía planteada, el cálculo de la fuerza entre el vehículo y un nodo en particular se realiza de la siguiente manera:

$$F = q/d^2$$

Siendo  $F$  la fuerza que ejerce el nodo sobre el vehículo,  $q$  la demanda del nodo y  $d$  la distancia entre el vehículo y el nodo.

Como fue mencionado al principio de este capítulo las metaheurísticas propuestas están conformadas por dos criterios: selección de nodo y ordenamiento de nodos en un tour. Las diferentes metaheurísticas diferirán entre sí de acuerdo a los criterios aplicados.

A continuación se presenta un conjunto de metaheurísticas que utilizan el cálculo de la fuerza entre el vehículo y los nodos como fue planteado anteriormente. Las diferentes propuestas son constructivas, no iterativas. Por lo tanto construiremos la solución insertando a la misma diferentes tours (disjuntos entre sí) hasta haber incluido todos los nodos considerados. Es por esto que la lógica de la propuesta se centra en la construcción de un tour. En cada paso de iteración se construirá un tour y para ello se definirán los criterios de selección y de ruteo. Lo que hará la diferencia entre cada propuesta será el criterio de selección de nodo. Como se mencionó anteriormente, el criterio de ruteo consta del TSP para los nodos seleccionados. Se utilizará para ello la propuesta 8 del capítulo anterior: el más lejano al  $CM_V$  que consta del criterio de selección del nodo inicial más lejano al  $CM$ , del criterio de selección 8 y del criterio de inserción en la mejor ubicación. Se eligió esta propuesta de TSP por tres motivos:

- es una propuesta que ya fue utilizada observándose resultados buenos y estables
- está basada en un concepto de la física (Centro de masa) por lo que al aplicarla quedará "construida" una metaheurística para la resolución del VRP basada en su totalidad en conceptos de física (Fuerza de atracción entre cargas y Centro de masa)
- fue propuesta por nosotros, por lo que al aplicarla quedará "construida" una metaheurística para la resolución del VRP en su totalidad hecha por nosotros

Ya que será utilizado frecuentemente, a partir de este momento, se hará referencia a la propuesta 8 del capítulo anterior para la resolución del TSP como TSPCM. En las distintas metaheurísticas presentadas a continuación, además del criterio de ruteo TSPCM, se utiliza el algoritmo de cáscara convexa para comparar los resultados obtenidos con uno y otro criterio.

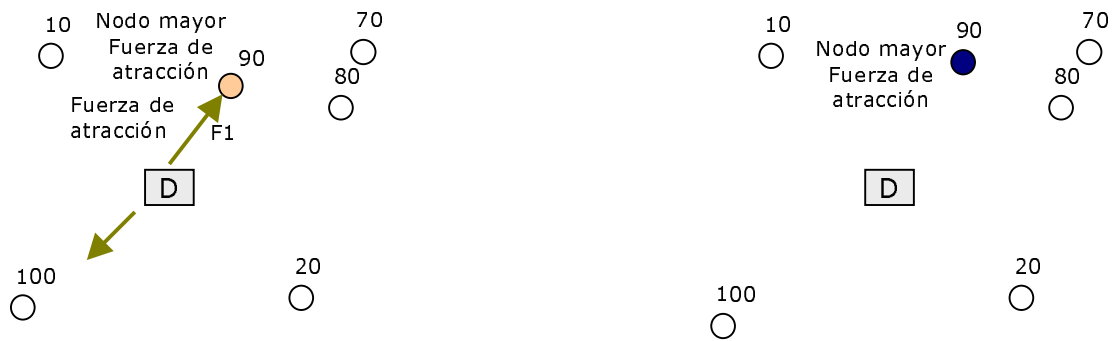
*Obs.:* Para la especificación de la plataforma en la cual se ejecutaron todos los algoritmos que se presentan tablas en las siguientes secciones referirse al capítulo 6.

### 5.1.1 Propuesta 1: Fuerza de atracción desde el depósito

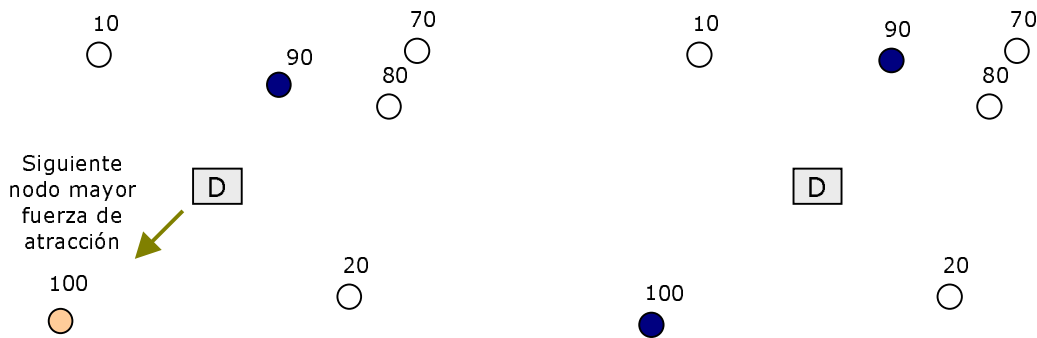
La metaheurística propuesta selecciona el siguiente nodo a visitar en un tour como el que mayor fuerza ejerza sobre el vehículo. De esta manera, se tiende a que el vehículo visite primero a los nodos de mayor demanda, pero ponderado por la distancia a que se encuentran los nodos logrando que los más alejados (mayor costo llegar hasta ellos)

tengan menor probabilidad de ser seleccionados. La intención de visitar primero a los de mayor demanda busca que cuando al vehículo le quede menor capacidad haya nodos con menor demanda que aún pueda satisfacer y no tenga que volver al depósito teniendo aún "bastante" capacidad. En esta propuesta se calcula la fuerza de atracción que ejerce cada nodo sobre el vehículo estando ubicado este en el depósito. Cada tour queda conformado por la mayor cantidad de nodos que la capacidad del camión pueda satisfacer en orden decreciente de fuerza ejercida sobre él por los nodos no visitados aún. El criterio de selección entonces es el de mayor fuerza ejercida por un nodo desde el depósito y el criterio de ruteo es TSPCM. A continuación se muestra una secuencia de figuras que ilustra este comportamiento.

Las primeras dos figuras muestran como se selecciona al nodo con demanda 90 sobre el de demanda 100, debido a que este se encuentra a menor distancia y esto hace que ejerza mayor fuerza sobre el depósito.



Luego, como se muestra en las siguientes figuras, en una próxima instancia se procederá a seleccionar al próximo nodo que ejerce la mayor fuerza sobre el depósito, solo considerando aquellos que aun no han sido seleccionados.



### Pseudocódigo

Marco el depósito como visitado

Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el depósito

Mientras halla nodos sin visitar:

- Seleccione el depósito
- Loop //cálculo de un tour
  - Seleccione el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer
  - Si no me da la capacidad para satisfacer a ninguno salgo del loop
- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 1: Fuerza de atracción desde el depósito							
Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 1			
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	595	4	58	<1
Eil30	29	545	3	721	4	32	<1
Eil33	32	910	4	988	4	8	<1
Eil51	50	521	5	817	5	56	<1
Eila76	75	847	10	1540	10	81	<1
Eilb76	75	1058	15	1735	14	63	<1
Eilc76	75	745	8	1312	8	76	<1
Eild76	75	692	7	1253	7	81	<1
Eila101	100	825	8	1527	8	85	<1
Eilb101	100	1114	14	2110	14	89	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP – Metaheurísticas Fuerza de atracción – Propuesta 1"

Promedio del % de exceso sobre el óptimo conocido: **62.9%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 1 con cáscara convexa			
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	592	4	57	<1

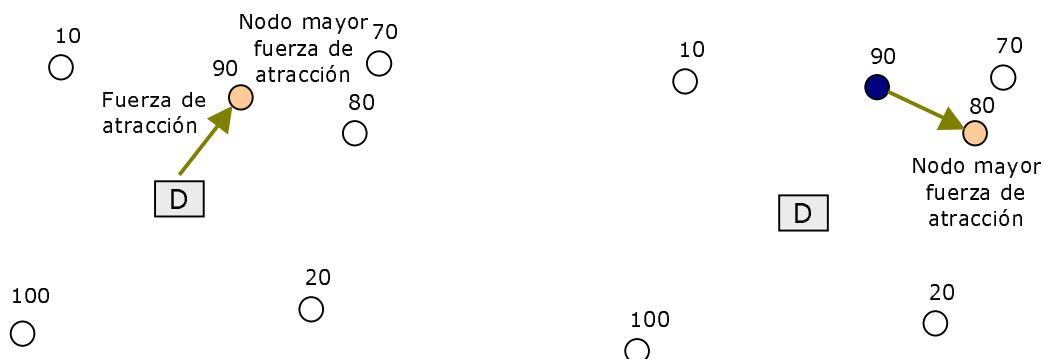
Eil30	29	545	3	721	3	32	<1
Eil33	32	910	4	988	4	8	<1
Eil51	50	521	5	815	5	56	<1
Eila76	75	847	10	1538	10	81	<1
Eilb76	75	1058	15	1735	14	63	<1
Eilc76	75	745	8	1307	8	75	<1
Eild76	75	692	7	1251	7	80	<1
Eila101	100	825	8	1525	8	84	<1
Eilb101	100	1114	14	2106	14	89	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa – Propuesta 1".

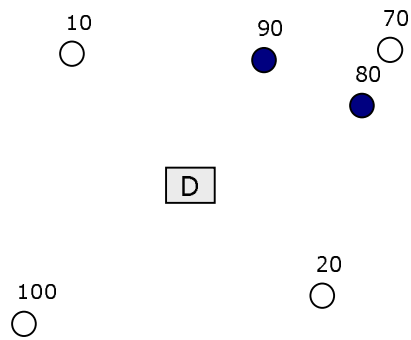
Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **62.5%**  
 Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

### 5.1.2 Propuesta 2: Fuerza de atracción desde cada nodo

En esta propuesta calculamos la fuerza de atracción que ejercen los nodos sobre el vehículo pero estando éste ubicado en el último nodo visitado. Como se parte del depósito, la selección del primer nodo no diferirá de la anterior propuesta, pero la diferencia está en la selección de los siguientes nodos del tour. Se seleccionarán calculando la fuerza que ejercen los nodos con respecto al último seleccionado. Esto implica que, cada vez que se seleccione un nodo para pertenecer al tour se recalculará la fuerza que ejerce cada nodo al vehículo ubicado en el último nodo seleccionado. Si bien hay que recalcular las fuerzas de atracción en cada paso de selección se fortalece la idea principal de la metaheurística que intenta seleccionar primero a los que tengan mayor demanda, ponderando la distancia a la que se encuentran. De todas maneras, al ir avanzando en la ejecución la cantidad de cálculos disminuirá al haber cada vez más nodos seleccionados, no considerados en el cálculo. El criterio de selección entonces es el de mayor fuerza ejercida por un nodo desde el último seleccionado y el criterio de ruteo el TSPCM. El proceso de ejecución es presentado en las siguientes figuras.



El resultado parcial de la selección se muestra en la siguiente figura.



### Pseudocódigo

Marco el depósito como visitado

Selecciono el depósito

Mientras halla nodos sin visitar:

- Loop //cálculo de un tour
  - Calculo la fuerza que ejerce cada nodo no visitado ni seleccionado al vehículo ubicado en el último nodo seleccionado
  - Selecciono el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer
  - Si no me da la capacidad para satisfacer a ninguno salgo del loop
- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 2: Fuerza de atracción desde cada nodo							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 2			
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	439	4	17	<1
Eil30	29	545	3	558	3	2	<1
Eil33	32	910	4	996	4	9	<1
Eil51	50	521	5	620	5	19	<1
Eila76	75	847	10	1030	10	21	<1
Eilb76	75	1058	15	1342	14	26	<1
Eilc76	75	745	8	950	8	27	<1
Eild76	75	692	7	928	7	34	<1
Eila101	100	825	8	1120	8	35	<1
Eilb101	100	1114	14	1473	14	32	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP - Metaheurísticas Fuerza de atracción - Propuesta 2"

Promedio del % de exceso sobre el óptimo conocido: **22.2%**  
 Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 2 con cáscara convexa			
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	439	4	17	<1
Eil30	29	545	3	558	3	2	<1
Eil33	32	910	4	996	4	9	<1
Eil51	50	521	5	615	5	18	<1
Eila76	75	847	10	1029	10	21	<1
Eilb76	75	1058	15	1338	14	26	<1
Eilc76	75	745	8	949	8	27	<1
Eild76	75	692	7	926	7	33	<1
Eila101	100	825	8	1125	8	36	<1
Eilb101	100	1114	14	1470	14	31	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa - Propuesta 2".

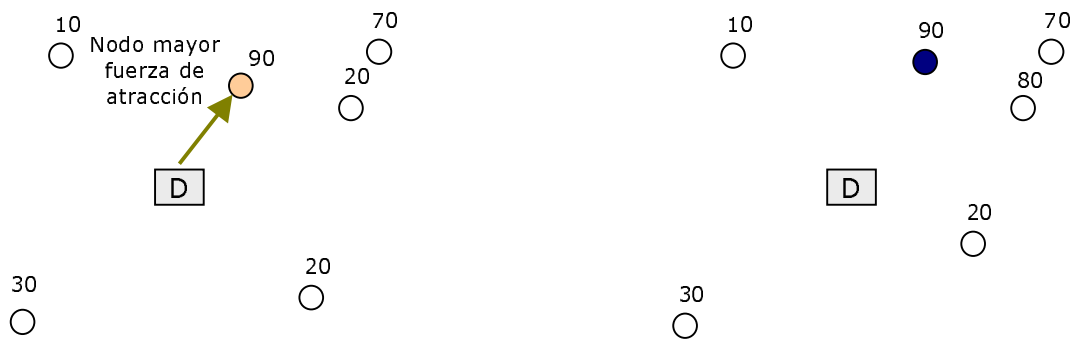
Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **22%**  
 Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

### 5.1.3 Propuesta 3: Fuerza de atracción desde el depósito "con vuelta"

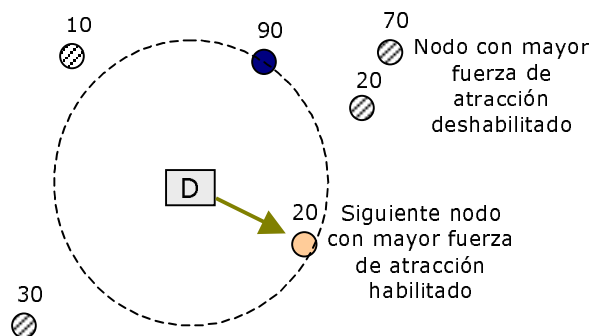
Esta propuesta es una variante de la propuesta 1. Se calcula la fuerza que ejerce cada nodo al vehículo desde el depósito y no se vuelve a calcular la fuerza entre el vehículo y otro nodo durante el resto del proceso. La diferencia está en que, a partir del momento en que la suma de las solicitudes de los nodos seleccionados sobrepasen cierto porcentaje de la capacidad del vehículo, no se seleccionarán nodos tales que su distancia al depósito sea mayor que la distancia al depósito del último seleccionado. Esto es para evitar que cuando el vehículo se "vacíe" no esté muy "lejos" del depósito, y por lo tanto el recorrido de vuelta hasta él sea muy extenso. En otras palabras: obligamos a que el vehículo, cuando le queda menos de cierto porcentaje de su carga, no visite nodos que lo alejen del depósito. Si hay algún nodo posible para ser visitado, pero eso implica alejarse del depósito no le será permitido visitarlo en este tour. Cuando no alcanza la capacidad para satisfacer a ningún nodo no visitado que esté más cerca del depósito que el vehículo, entonces se obtuvo el conjunto de nodos

seleccionados para un tour. La selección del porcentaje de la capacidad del camión para dar la "vuelta" y solo considerar los nodos tales que al ser visitados no se aleje mas del depósito es arbitraria y paramétrica. Elegimos como ejemplo el 50% de la capacidad del vehículo porque se espera que en cada tour utilice aproximadamente la mitad de su capacidad en "la ida" (alejamiento del depósito) y la otra mitad en la vuelta (acercamiento "obligado" al depósito). El criterio de selección entonces es el de mayor fuerza ejercida por un nodo desde el depósito sin considerar los nodos que se alejen de él a partir de que la capacidad del vehículo haya disminuido cierto porcentaje y el criterio de inserción es TSPCM.

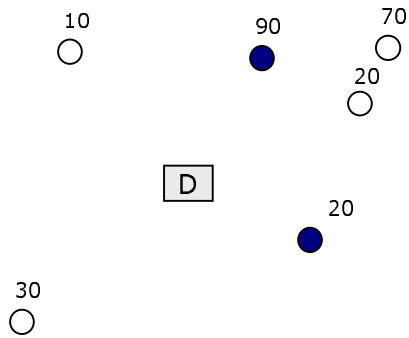
En las siguientes figuras se representan un caso en el cual el primero nodo que se selecciona hace que se llegue al porcentaje de capacidad del vehículo definido.



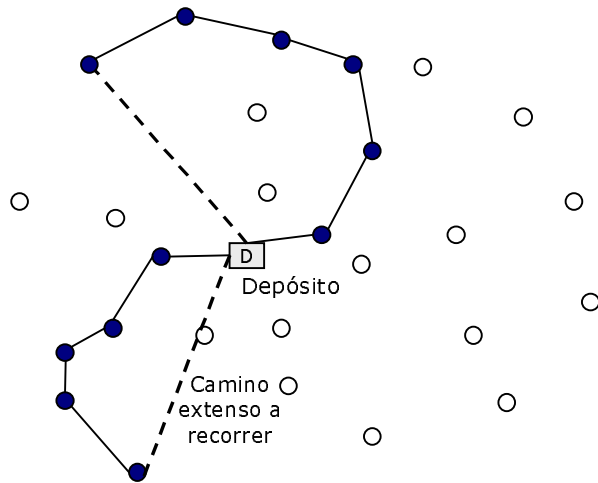
Los nodos que solo se considerarán a partir de esta instancia serán solo aquellos que pertenezcan al círculo definido que se representa en la figura. Esto se hará a pesar de que existan otros nodos de mayor fuerza con demanda capaz de ser atendida, pero que habitan fuera del círculo definido.



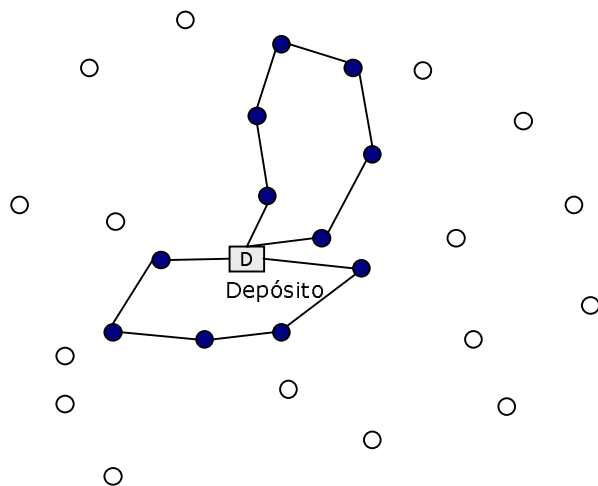
La selección quedará determinada de la siguiente manera.



A continuación se ilustra gráficamente la situación que esta propuesta intenta mejorar con respecto a la propuesta 1.

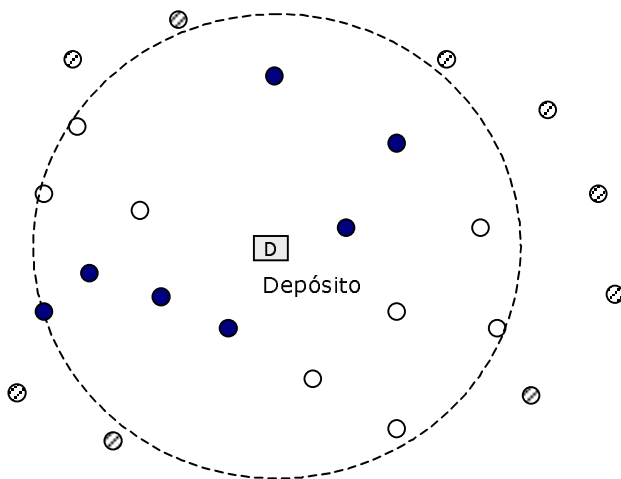


La figura de arriba muestra el comportamiento de un caso no deseado, en el que se genera un camino larga distancia al retornar el vehículo al depósito. La figura de abajo indica el comportamiento del vehículo al no permitírsele seguir visitando nodos una vez alcanzado el porcentaje de capacidad preestablecido.





Una vez que el vehículo llega a cierto porcentaje de su capacidad de carga no le permitimos alejarse más del depósito. Se puede ver como definir una circunferencia de centro el depósito y de radio la distancia actual del vehículo al depósito y no permitirle "salirse" de ella. Esta circunferencia se recalcula cada vez que seleccionamos un nodo. Gráficamente, en un tour del ejemplo anterior se vería de la siguiente manera.



### Pseudocódigo

Marco el depósito como visitado

Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el depósito

Mientras halla nodos sin visitar:

- Loop //cálculo de la primera parte de un tour (ida)
  - Seleccione el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer.
  - Si ningún nodo fue seleccionado o la capacidad del camión es menor que el 50% salgo del loop
- Fin loop
- Loop //cálculo de la segunda parte de un tour (vuelta)
  - Seleccione el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer dentro de los nodos que están a menor distancia del depósito que el último seleccionado
  - Si ningún nodo fue seleccionado salgo del loop
- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 3: Fuerza de atracción desde el depósito con "vuelta"							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 3			
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	622	5	65	<1
Eil30	29	545	3	839	4	53	<1
Eil33	32	910	4	1169	5	28	<1
Eil51	50	521	5	967	6	85	<1
Eila76	75	847	10	1514	11	78	<1
Eilb76	75	1058	15	1780	15	68	<1
Eilc76	75	745	8	1298	9	74	<1
Eild76	75	692	7	1112	7	60	<1
Eila101	100	825	8	1501	8	81	<1
Eilb101	100	1114	14	2080	14	86	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP - Metaheurísticas Fuerza de atracción - Propuesta 3"

Promedio del % de exceso sobre el óptimo conocido: **67.8%**  
 Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 3 con cáscara convexa			
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	622	5	65	<1
Eil30	29	545	3	838	4	53	<1
Eil33	32	910	4	1163	5	27	<1
Eil51	50	521	5	967	6	85	<1
Eila76	75	847	10	1513	11	78	<1
Eilb76	75	1058	15	1780	15	68	<1
Eilc76	75	745	8	1298	9	74	<1
Eild76	75	692	7	1111	7	60	<1
Eila101	100	825	8	1501	8	81	<1
Eilb101	100	1114	14	2080	14	86	<1

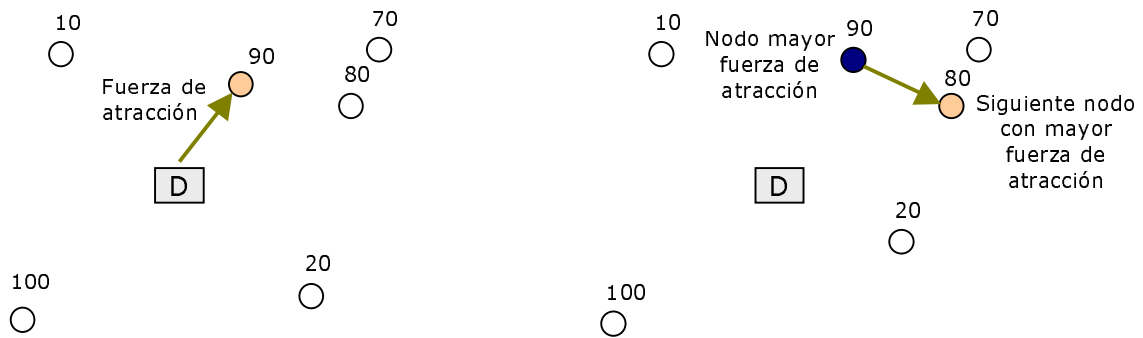
Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa - Propuesta 3".

Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **67.7%**  
 Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

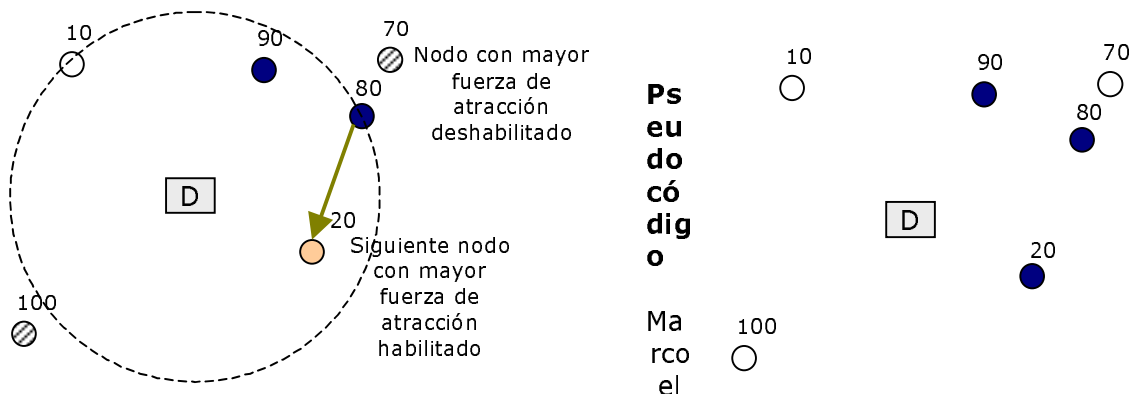
### 5.1.4 Propuesta 4: Fuerza de atracción desde cada nodo con “vuelta”

Esta propuesta es una variante de la propuesta anterior agregándole el recalcular de las fuerzas de atracción de los nodos no seleccionados en cada paso de la iteración. Al igual que en la propuesta 2, se seleccionará el próximo nodo perteneciente al tour calculando la fuerza que ejerce cada nodo sobre el vehículo, estando este ubicado en el último nodo agregado al tour. El criterio de selección entonces es el de mayor fuerza ejercida por un nodo desde el último seleccionado sin considerar los nodos que se alejen de él a partir de que la capacidad del camión haya disminuido cierto porcentaje y el criterio de inserción el TSPCM.

En las figuras se representa la selección de los dos primeros nodos para un caso particular:



Una vez seleccionado el segundo nodo se llega al porcentaje de capacidad del vehículo definida, por lo que se procederá a realizar la restricción en cuanto a la distancia de este al depósito.



depósito como visitado

Mientras halla nodos sin visitar:

- Loop //cálculo de la primera parte de un tour (ida)
  - Calculo la fuerza que ejerce cada nodo no visitado ni seleccionado al camión ubicado en el último nodo seleccionado
  - Seleccione el nodo no visitado de mayor fuerza que me da la capacidad de satisfacer
  - Si ningún nodo fue seleccionado o la capacidad del camión es menor que el 50% salgo del loop

- Fin loop
- Loop //cálculo de la segunda parte de un tour (vuelta)
  - Calculo la fuerza que ejerce cada nodo no visitado ni seleccionado al camión ubicado en el último nodo seleccionado
  - Selecciono el nodo no visitado de mayor fuerza que me da la capacidad de satisfacer dentro de los nodos que están a menor distancia del depósito que el último seleccionado
  - Si ningún nodo fue seleccionado salgo del loop
- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 4: Fuerza de atracción desde cada nodo con "vuelta"							
Nombre Instancia	#Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 4			
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	484	5	29	<1
Eil30	29	545	3	558	3	2	<1
Eil33	32	910	4	1008	4	10	<1
Eil51	50	521	5	682	6	30	<1
Eila76	75	847	10	1012	11	19	<1
Eilb76	75	1058	15	1263	15	19	<1
Eilc76	75	745	8	865	8	16	<1
Eild76	75	692	7	896	8	29	<1
Eila101	100	825	8	1082	8	31	<1
Eilb101	100	1114	14	1440	14	29	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP – Metaheurísticas Fuerza de atracción – Propuesta 4"

Promedio del % de exceso sobre el óptimo conocido: **21.4%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación								
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 4 con cáscara convexa				
		costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)	
Eil22	21	525	4	484	5	29	<1	
Eil30	29	545	3	558	3	2	<1	
Eil33	32	910	4	1007	4	10	<1	
Eil51	50	521	5	679	6	30	<1	
Eila76	75	847	10	1011	11	19	<1	
Eilb76	75	1058	15	1257	15	18	<1	
Eilc76	75	745	8	864	8	15	<1	
Eild76	75	692	7	895	8	29	<1	
Eila101	100	825	8	1087	8	31	<1	
Eilb101	100	1114	14	1439	14	29	<1	

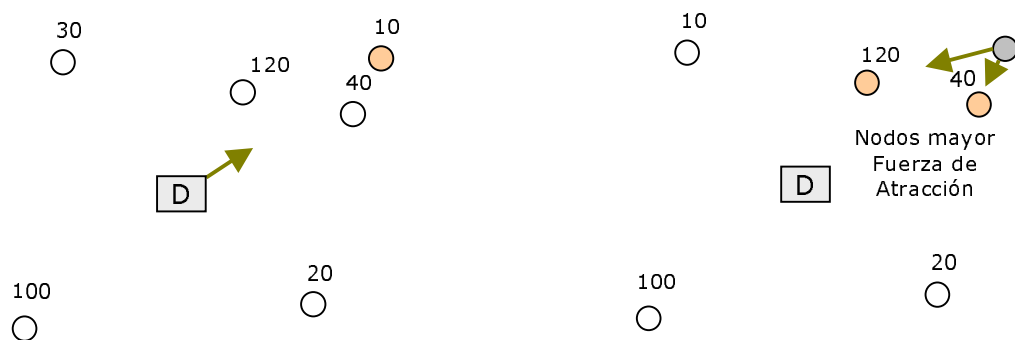
Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa - Propuesta 4".

Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **21.2%**

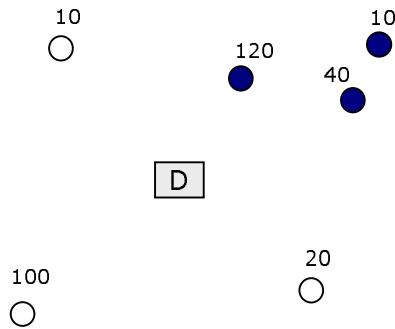
Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

### 5.1.5 Propuesta 5: Fuerza de atracción desde el primer nodo, variante "primero el de menor fuerza"

Esta propuesta es el resultado de una variación a la propuesta 1. La variación consiste en la elección del primer nodo visitado cuando se parte del depósito. Se calcula la fuerza que ejercen todos los nodos sobre el vehículo ubicado en el depósito y se selecciona primero el de menor fuerza, como se muestra en la figura de la izquierda.



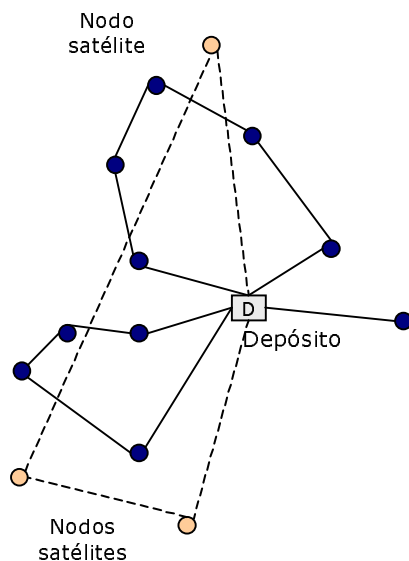
Luego el criterio de selección seguirá comportándose como en la propuesta 1 con la variante de que no calculará la fuerza que ejercen los nodos sobre el vehículo posicionado en el depósito, si no que posicionado en el primer nodo seleccionado de la manera anteriormente descrita, como se representa en la figura de arriba a la derecha. Cada tour queda entonces conformado por la mayor cantidad de nodos que la capacidad del camión pueda satisfacer en orden decreciente de fuerza ejercida sobre él por los nodos no visitados aún. La siguiente figura indica los nodos seleccionados para el caso representado.



El criterio de selección entonces es el de mayor fuerza ejercida por un nodo desde el primero seleccionado y el criterio de ruteo es TSPCM.

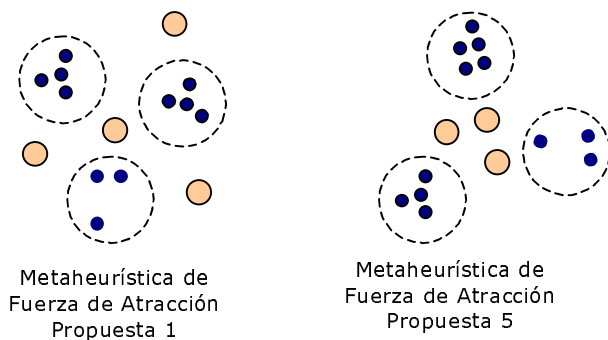
En la propuesta 1, para seleccionar los nodos pertenecientes a cada tour se utiliza la fuerza que ejerce cada nodo sobre el vehículo ubicado éste en el depósito. En particular, se selecciona el primer nodo del tour como el de mayor fuerza de estos. Este algoritmo tiende a recorrer primero (incluir en los tours) a los nodos más cercanos al depósito. Esto es porque los más lejanos ejercen menor fuerza que los más cercanos al ser esta inversamente proporcional al cuadrado de la distancia. En etapas avanzadas del algoritmo se presentan casos de nodos o ciudades "satélites" que están muy alejados del depósito. Esto genera tours "finales" de largo recorrido y que visitan pocos nodos. Por lo tanto la eficiencia lograda en los primeros tours se descompensa con la necesidad de tours finales muy costosos. Una idea gráfica del problema se presenta en las siguientes dos figuras:

Los tres primeros tours dejan tres nodos "satélites", que provocan la generación de un tour "costoso" (en comparación con los tres anteriores) para atender a los tres nodos "satélites".

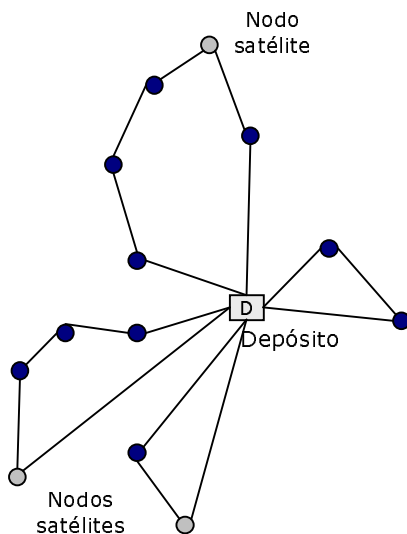


La nueva propuesta consta de seleccionar como primer nodo de cada tour el que menor fuerza ejerza sobre el vehículo ubicado en el depósito, y no el de mayor fuerza, como en la propuesta 1. La intención es que los tours que alcancen estos nodos

lejanos al depósito no sean casi "exclusivos" para su atención y se puedan aprovechar en la atención de otros nodos. Lo que se pretende es que los nodos "finales" (últimos en pertenecer a algún tour) queden cercanos al depósito y no lejanos. Luego, con uno o más de un tour se visitarán estos nodos, pero se espera que estén cercanos al depósito, y por lo tanto cercanos entre sí. Gráficamente podemos apreciar en la siguiente figura la idea que intenta cambiar la variante de este algoritmo:



El caso de los nodos satélites descrito anteriormente se resolverá como se muestra en la siguiente figura, generando tours más eficaces.



### Pseudocódigo

Marco el depósito como visitado

Mientras halla nodos sin visitar:

- Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el depósito
- Selecciono como primer nodo del tour al de menor fuerza de los calculados
- Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el último nodo seleccionado
- Loop //cálculo de un tour
  - Selecciono el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer
  - Si no me da la capacidad para satisfacer a ninguno salgo del loop
- Fin loop

- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 5: Fuerza de atracción desde el primer nodo, variante "primero el de menor fuerza"							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 5			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	484	4	29	<1
Eil30	29	545	3	584	3	7	<1
Eil33	32	910	4	1030	4	13	<1
Eil51	50	521	5	644	5	23	<1
Eila76	75	847	10	954	10	12	<1
Eilb76	75	1058	15	1151	14	8	<1
Eilc76	75	745	8	906	8	21	<1
Eild76	75	692	7	857	7	23	<1
Eila101	100	825	8	1097	8	32	<1
Eilb101	100	1114	14	1364	14	22	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP – Metaheurísticas Fuerza de atracción – Propuesta 5"

Promedio del % de exceso sobre el óptimo conocido: **19%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 5 con cáscara convexa			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	484	4	29	<1
Eil30	29	545	3	581	3	6	<1
Eil33	32	910	4	1029	4	13	<1
Eil51	50	521	5	640	5	22	<1
Eila76	75	847	10	952	10	12	<1
Eilb76	75	1058	15	1152	14	8	<1
Eilc76	75	745	8	908	8	21	<1
Eild76	75	692	7	851	7	22	<1
Eila101	100	825	8	1087	8	31	<1
Eilb101	100	1114	14	1361	14	22	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa – Propuesta 5".

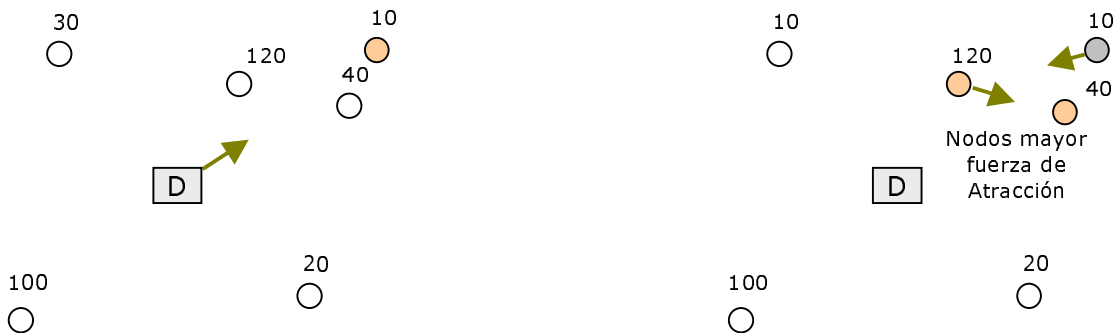
Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **18.6%**



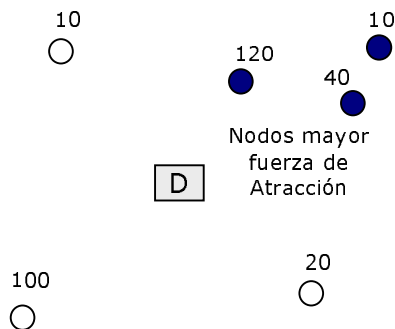
Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

### 5.1.6 Propuesta 6: Fuerza de atracción desde cada nodo, variante “primero el de menor fuerza”

Esta propuesta utiliza el mismo criterio de selección del nodo inicial que la anterior. Es decir que se calcula la fuerza de atracción ejercida sobre el vehículo en el depósito y se elige al nodo que ejerza la menor de estas.



La diferencia radica en la selección de los siguientes nodos de cada tour. No se seleccionarán calculando la fuerza que ejercen los nodos con respecto al primer nodo seleccionado, si no que se seleccionarán calculando la fuerza que ejercen los nodos con respecto al último seleccionado.



Esto implica que, cada vez que se seleccione un nodo para pertenecer al tour se recalculará la fuerza que ejerce cada nodo al camión ubicado en el último nodo seleccionado. Esto lleva a tener que recalcular las fuerzas de atracción en cada paso de selección pero fortalece la idea principal de la metaheurística que intenta seleccionar primero a los que tengan mayor demanda, ponderando la distancia a la que se encuentran. De todas maneras, al ir avanzando en la ejecución la cantidad de cálculos disminuirá al haber cada vez más nodos seleccionados, no considerados en el cálculo. El criterio de inserción de los nodos seleccionados para el tour está dado por el orden de la selección.

#### Pseudocódigo

Marco el depósito como visitado  
Mientras halla nodos sin visitar:

- Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el depósito
- Selecciono como primer nodo del tour al de menor fuerza de los calculados
- Loop //cálculo de un tour
  - Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el último nodo seleccionado
  - Selecciono el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer
  - Si no me da la capacidad para satisfacer a ninguno salgo del loop
- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 6: Fuerza de atracción desde cada nodo, variante "primero el de menor fuerza"							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 6			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	409	4	9	<1
Eil30	29	545	3	665	3	22	<1
Eil33	32	910	4	947	4	4	<1
Eil51	50	521	5	640	5	22	<1
Eila76	75	847	10	1065	11	25	<1
Eilb76	75	1058	15	1282	15	21	<1
Eilc76	75	745	8	911	8	22	<1
Eild76	75	692	7	1057	7	52	<1
Eila101	100	825	8	1116	8	35	<1
Eilb101	100	1114	14	1393	14	25	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP – Metaheurísticas Fuerza de atracción – Propuesta 6"

Promedio del % de exceso sobre el óptimo conocido: **23.7%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

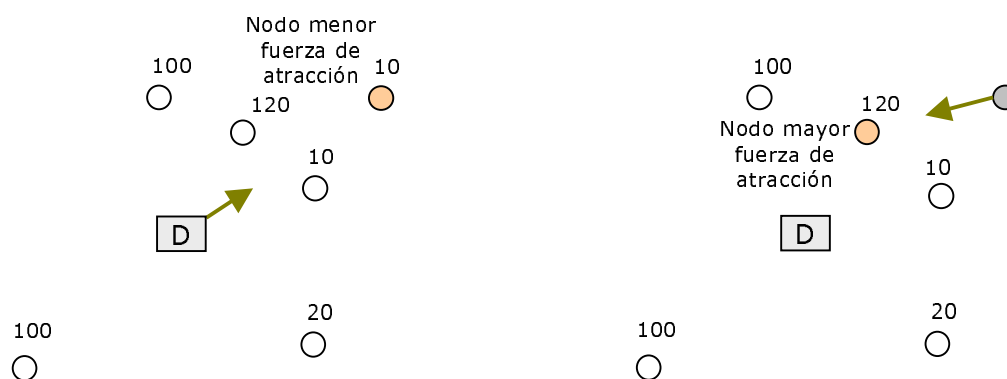
Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 6 con cáscara convexa			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	409	4	9	<1
Eil30	29	545	3	665	3	22	<1
Eil33	32	910	4	946	4	3	<1
Eil51	50	521	5	640	5	22	<1
Eila76	75	847	10	1065	11	25	<1
Eilb76	75	1058	15	1282	15	21	<1
Eilc76	75	745	8	910	8	22	<1
Eild76	75	692	7	1050	7	51	<1
Eila101	100	825	8	1104	8	33	<1
Eilb101	100	1114	14	1393	14	25	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa - Propuesta 6".

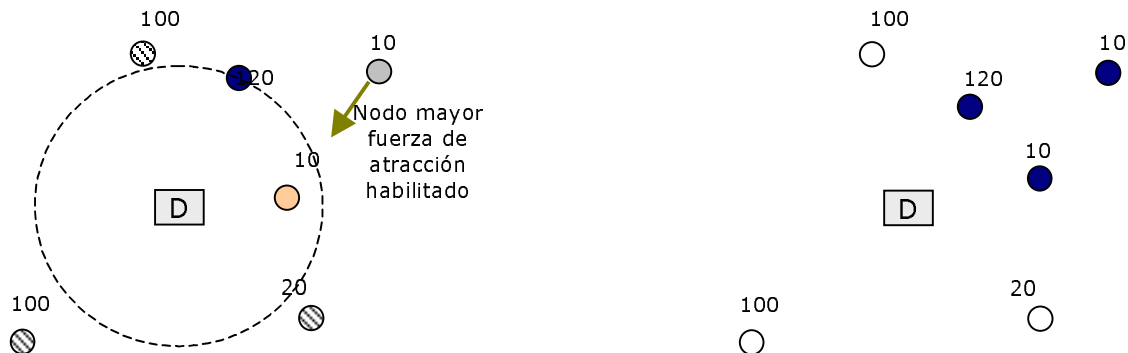
Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **23.3%**  
 Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

### 5.1.7 Propuesta 7: Fuerza de atracción desde el primer nodo con "vuelta", variante "primero el de menor fuerza"

Con respecto a la selección del primer nodo se comporta igual que las dos propuestas anteriores. Es decir que se calcula la fuerza de atracción ejercida sobre el vehículo en el depósito y se elige al nodo que ejerza la menor.



La diferencia está en el criterio de selección del siguiente nodo perteneciente al tour. No se recalculará la fuerza ejercida por cada nodo al vehículo ubicado en el último nodo seleccionado como en la propuesta anterior. Solamente se calculará la fuerza ejercida al vehículo por todos los nodos no seleccionados estando ubicado éste (el vehículo) en el primer nodo seleccionado. Se elegirá de acuerdo al de mayor fuerza que se pueda satisfacer, al igual que en la propuesta 5.



La diferencia está en que, a partir del momento en que la suma de las solicitudes de los nodos seleccionados sobrepasen cierto porcentaje de la capacidad del camión, no se seleccionarán nodos tales que su distancia al depósito sea mayor que la distancia al depósito del último seleccionado, al igual que en la propuesta 3. Esto es para evitar que cuando el camión se "vacíe" no esté muy "lejos" del depósito, y por lo tanto el recorrido de vuelta hasta él sea muy extenso. En otras palabras: obligamos a que el vehículo, cuando le queda menos de cierto porcentaje de su carga, no visite nodos que lo alejen del depósito. Si hay algún nodo posible para ser visitado, pero eso implica alejarse del depósito no le será permitido visitarlo en este tour. Cuando no alcanza la capacidad para satisfacer a ningún nodo no visitado que esté más cerca del depósito que el vehículo, entonces se obtuvo el conjunto de nodos seleccionados para un tour. La selección del porcentaje de la capacidad del camión para dar la "vuelta" y solo considerar los nodos tales que al ser visitados no se aleje más del depósito es arbitraria y paramétrica. Elegimos como ejemplo el 50% de la capacidad del vehículo porque se espera que en cada tour utilice aproximadamente la mitad de su capacidad en "la ida" (alejamiento del depósito) y la otra mitad en la vuelta (acercamiento "obligado" al depósito). El criterio de selección entonces es el de mayor fuerza ejercida por un nodo desde el primero seleccionado sin considerar los nodos que se alejen del depósito a partir de que la capacidad del vehículo haya disminuido cierto porcentaje y el criterio de inserción es TSPCM.

### Pseudocódigo

Marco el depósito como visitado

Mientras halla nodos sin visitar:

- Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el depósito
- Selecciono como primer nodo del tour al de menor fuerza de los calculados
- Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el último nodo seleccionado
- Loop //cálculo de la primera parte de un tour (ida)
  - Selecciono el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer.
  - Si ningún nodo fue seleccionado o la capacidad del camión es menor que el 50% salgo del loop
- Fin loop
- Loop //cálculo de la segunda parte de un tour (vuelta)
  - Selecciono el nodo no visitado ni seleccionado de mayor fuerza que me

da la capacidad de satisfacer dentro de los nodos que están a menor distancia del depósito que el último seleccionado

- Si ningún nodo fue seleccionado salgo del loop
- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 7: Fuerza de atracción desde el primer nodo con "vuelta", variante "primero el de menor fuerza"							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 7			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	513	6	36	<1
Eil30	29	545	3	791	5	45	<1
Eil33	32	910	4	1111	5	22	<1
Eil51	50	521	5	653	7	25	<1
Eila76	75	847	10	1111	15	31	<1
Eilb76	75	1058	15	1291	19	22	<1
Eilc76	75	745	8	1008	12	35	<1
Eild76	75	692	7	849	9	22	<1
Eila101	100	825	8	1178	11	42	<1
Eilb101	100	1114	14	1445	19	29	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP – Metaheurísticas Fuerza de atracción – Propuesta 7"

Promedio del % de exceso sobre el óptimo conocido: **30.9%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

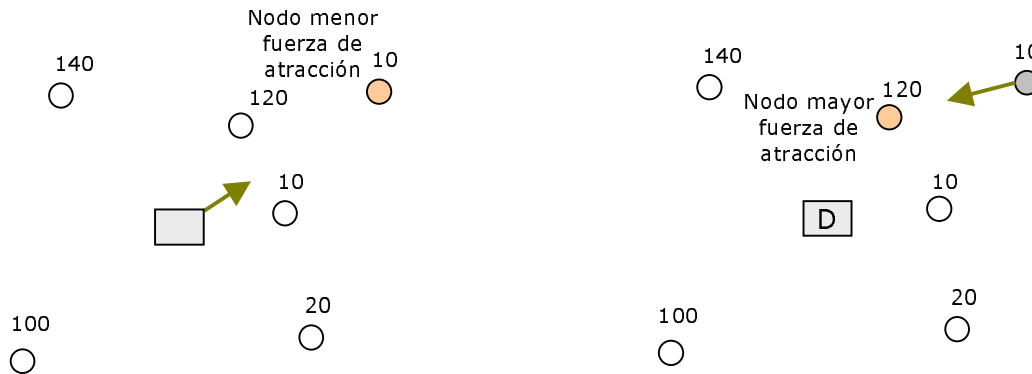
Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 7 con cáscara convexa			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	513	6	36	<1
Eil30	29	545	3	788	5	44	<1
Eil33	32	910	4	1111	5	22	<1
Eil51	50	521	5	653	7	25	<1
Eila76	75	847	10	1111	15	31	<1
Eilb76	75	1058	15	1291	19	22	<1
Eilc76	75	745	8	1007	12	35	<1
Eild76	75	692	7	849	9	22	<1
Eila101	100	825	8	1169	11	41	<1
Eilb101	100	1114	14	1445	19	29	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa - Propuesta 7".

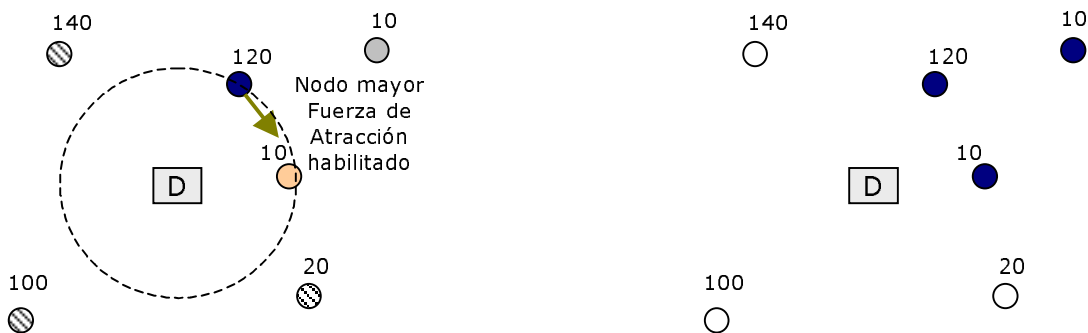
Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **30.7%**  
 Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

### 5.1.8 Propuesta 8: Fuerza de atracción desde cada nodo con "vuelta", variante "primero el de menor fuerza"

Esta variante considera el mismo criterio de selección del primer nodo que las tres anteriores. La propuesta 6 aplica a la propuesta 5 el recalcular de las fuerzas de atracción que ejercen los nodos no visitados aún sobre el vehículo en el último nodo seleccionado.



Esto está motivado por el hecho de tener "actualizado" cual es el nodo con mayor solicitud (ponderado a la distancia) con el costo adicional del recalcular en cada paso. La propuesta 7 aplica a la propuesta 5 el concepto de "acercarse" al depósito, en la selección de nodos pertenecientes al tour, una vez que se haya llegado a la mitad de la capacidad.



Esta propuesta aplica a la propuesta 5 las dos ideas expuestas en las propuestas 6 y 7. Por lo tanto se recalculan las fuerzas ejercidas por los nodos aún no seleccionados en cada paso y una vez que se llegó a la mitad de la capacidad del camión no se

consideran nodos mas lejanos al depósito que el último seleccionado. El criterio de ruteo es el TSPCM.

### Pseudocódigo

Marco el depósito como visitado

Mientras halla nodos sin visitar:

- Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el depósito
- Seleciono como primer nodo del tour al de menor fuerza de los calculados
- Loop //cálculo de la primera parte de un tour (ida)
  - Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el último nodo seleccionado
  - Seleciono el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer.
  - Si ningún nodo fue seleccionado o la capacidad del camión es menor que el 50% salgo del loop
- Fin loop
- Loop //cálculo de la segunda parte de un tour (vuelta)
  - Calculo la fuerza que ejerce cada nodo no visitado al vehículo ubicado en el último nodo seleccionado
  - Seleciono el nodo no visitado ni seleccionado de mayor fuerza que me da la capacidad de satisfacer dentro de los nodos que están a menor distancia del depósito que el último seleccionado
  - Si ningún nodo fue seleccionado salgo del loop
- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

Al ejecutar esta metaheurística y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Propuesta 8: Fuerza de atracción desde cada nodo con "vuelta", variante "primero el de menor fuerza"							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 8			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	455	5	21	<1
Eil30	29	545	3	604	4	10	<1
Eil33	32	910	4	1003	5	10	<1
Eil51	50	521	5	694	6	33	<1
Eila76	75	847	10	992	10	17	<1
Eilb76	75	1058	15	1176	15	11	<1
Eilc76	75	745	8	911	8	22	<1
Eild76	75	692	7	951	7	37	<1
Eila101	100	825	8	1183	8	43	<1
Eilb101	100	1114	14	1444	14	29	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP – Metaheurísticas Fuerza de atracción – Propuesta 8"

Promedio del % de exceso sobre el óptimo conocido: **23.3%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Propuesta 8 con cáscara convexa			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	525	4	455	5	21	<1
Eil30	29	545	3	606	4	11	<1
Eil33	32	910	4	1001	5	10	<1
Eil51	50	521	5	694	6	33	<1
Eila76	75	847	10	992	10	17	<1
Eilb76	75	1058	15	1175	15	11	<1
Eilc76	75	745	8	910	8	22	<1
Eild76	75	692	7	947	7	36	<1
Eila101	100	825	8	1175	8	42	<1
Eilb101	100	1114	14	1444	14	29	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Fuerza de atracción con cáscara convexa - Propuesta 8".

Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **23.2%**

Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**



## 5.2 Campo Eléctrico

En esta metaheurística se utiliza el concepto de campo eléctrico que se define como la suma vectorial de las fuerzas de atracción a las que está sometida una carga de valor 1 en presencia de otras [FIS1]. Aquí también la analogía planteada es del vehículo con una carga positiva y de las ciudades con cargas negativas. En las metaheurísticas de fuerza de atracción considerábamos el valor del módulo del vector fuerza entre las cargas. En la presentada aquí se tiene en cuenta, además del módulo, la dirección y sentido de la fuerza que cada carga ejerce sobre la "carga vehículo". En las metaheurísticas anteriores se consideró ir "derecho" hacia el de mayor fuerza. Ahora se considera ir en la dirección del vector campo eléctrico. Esta dirección, conceptualmente será para "el lado" en que están los nodos de mayor fuerza. La intención es lograr los tours más cortos posibles para que el vehículo se vacíe rápido.

Para calcular el campo magnético se utiliza la misma fórmula utilizada anteriormente dada por la ley de Coulomb:

$$F = \frac{k \cdot q_1 \cdot q_2}{d^2} \quad (k = 8.9874 \times 10^9 \text{ N.m}^2/\text{C}^2) \text{ [FIS1]}$$

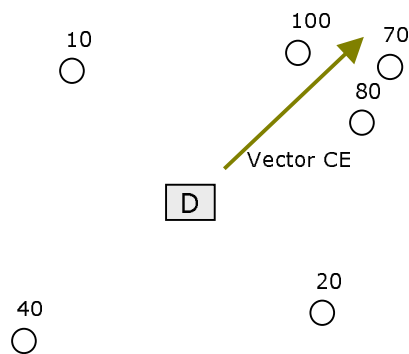
Siendo  $F$  la fuerza que experimentan entre sí las cargas,  $q_1$  y  $q_2$  el valor de cada una y  $d$  la distancia que hay entre ellas. Para un desarrollo mas detallado de la teoría de interacción entre cargas ver anexo "Conceptos físicos".

El valor de la "carga vehículo" se considera constante e igual a 1 y el valor de las "cargas nodo" se consideran igual a su demanda. El valor de la "carga vehículo" se considera constante porque nos interesa que lo que influya en el cálculo de la fuerza sea la demanda de los nodos e igual a 1 simplemente para facilitar los cálculos. Las distancias entre nodos se calculan como la distancia en coordenadas cartesianas al igual que lo consideramos en el capítulo anterior. Nuevamente apreciamos que la constante  $K$  en la fórmula no aporta más que un factor de escalamiento en la fuerza de atracción que hay entre el vehículo y los nodos. Por lo tanto, teniendo en cuenta la analogía planteada, el cálculo del módulo de la fuerza entre el vehículo y un nodo en particular se realiza de la siguiente manera:

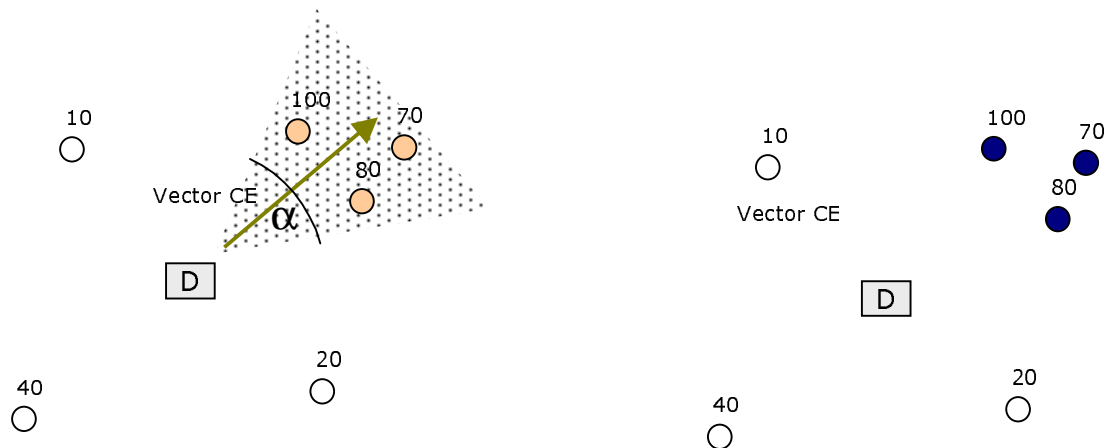
$$F = q/d^2$$

Siendo  $k$  la fuerza que ejerce el nodo sobre el vehículo,  $q$  la demanda del nodo y  $d$  la distancia entre el vehículo y el nodo. La dirección del vector fuerza ejercida por el nodo sobre el vehículo será la recta definida por los puntos en los que se encuentran el nodo y el vehículo y la dirección será desde el vehículo hacia el nodo, ya que se consideran cargas opuestas (fuerza de atracción).

Una vez determinado el módulo la dirección y el sentido del vector fuerza de atracción para cada nodo se calcula el vector campo eléctrico realizando la suma vectorial de todos los vectores fuerza de atracción.



Esta metaheurística propone simular el comportamiento de una "carga vehículo" en el campo magnético definido por las "cargas nodo" haciendo recorrer los nodos que estén "en la dirección" del vector campo eléctrico.



En la ejecución de la metaheurística se calculará el vector campo eléctrico en cada iteración, siendo utilizado (cada vez) para seleccionar los nodos que formarán parte de un tour. En cada paso se seleccionarán los nodos que puedan ser abastecidos por la capacidad del vehículo tales que el ángulo que forma su vector fuerza de atracción con el vector campo eléctrico no exceda de un cierto ángulo alfa, logrando de esta manera seleccionar para el tour correspondiente a la iteración actual nodos en "la dirección" del vector campo eléctrico. De esta manera queda definido el criterio de selección de nodo. Una vez obtenido el conjunto de nodos se rutean utilizando TSPCM como en los casos anteriores.

### Pseudocódigo

Marco el depósito como visitado

Mientras halla nodos sin visitar

- Calculo el vector campo magnético ubicado en el depósito sobre los nodos no visitados
- Loop //cálculo de un tour
  - Seleccione el nodo no visitado ni seleccionado tal que forme el menor ángulo posible (que no exceda alfa) con la dirección y sentido del vector campo eléctrico que me alcance la capacidad del camión
  - Si no me da la capacidad para satisfacer a ninguno "dentro" del ángulo alfa o no quedan nodos sin visita ni ser seleccionados en el rango de

ángulo considerado salgo del loop.

- Fin loop
- Ruteo los nodos seleccionados utilizando TSPCM y agrego el tour calculado al conjunto de tours que conforman la solución final
- Marco los nodos seleccionados como visitados y les quito la selección

Fin Mientras

A continuación se presentan los resultados de la ejecución de esta metaheurística para distintos casos de prueba y con diferentes ángulos alfa de entrada para analizar el impacto de este parámetro en los resultados obtenidos. Al igual que en las metaheurísticas de fuerza de atracción, además del criterio de ruteo TSPCM, se utiliza el algoritmo de cáscara convexa para comparar los resultados obtenidos con uno y otro criterio

*Obs.:* Para la especificación de la plataforma en la cual se ejecutaron las corridas que se presentan en la siguiente tabla referirse al capítulo 6 .

Al implementar y ejecutar esta metaheurística con *alfa* = 180 se observaron los siguientes resultados:

Metaheurística Campo eléctrico							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Campo eléctrico con <i>alfa</i> = 180			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	545	4	45	<1
Eil30	29	545	3	593	3	8	<1
Eil33	32	910	4	1009	4	10	<1
Eil51	50	521	5	686	5	31	<1
Eila76	75	847	10	1116	10	31	<1
Eilb76	75	1058	15	1403	15	32	<1
Eilc76	75	745	8	1018	8	36	<1
Eild76	75	692	7	872	7	26	<1
Eila101	100	825	8	1135	8	37	<1
Eilb101	100	1114	14	1405	14	26	<1

*Obs.:* Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP – Metaheurísticas Campo eléctrico – alfa180"

Promedio del % de exceso sobre el óptimo conocido: **28.2%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística con *alfa* = 180, pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Campo eléctrico con $\alpha = 180$ y cáscara convexa			
		Costo	Cant. vehículos	Costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	545	4	45	<1
Eil30	29	545	3	593	3	8	<1
Eil33	32	910	4	1001	4	10	<1
Eil51	50	521	5	684	5	31	<1
Eila76	75	847	10	1114	10	31	<1
Eilb76	75	1058	15	1402	15	32	<1
Eilc76	75	745	8	1013	8	35	<1
Eild76	75	692	7	870	7	25	<1
Eila101	100	825	8	1126	8	36	<1
Eilb101	100	1114	14	1405	14	26	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Campo eléctrico con cáscara convexa- alfa180".

Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **27.9%**

Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

Al implementar y ejecutar esta metaheurística con  $\alpha = 90$  se observaron los siguientes resultados:

Metaheurística Campo eléctrico							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Campo eléctrico con $\alpha = 90$			
		Costo	Cant. vehículos	Costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	561	5	49	<1
Eil30	29	545	3	613	4	12	<1
Eil33	32	910	4	1009	4	10	<1
Eil51	50	521	5	605	5	16	<1
Eila76	75	847	10	1112	12	31	<1
Eilb76	75	1058	15	1325	16	25	<1
Eilc76	75	745	8	987	8	32	<1
Eild76	75	692	7	794	7	14	<1
Eila101	100	825	8	1017	8	23	<1
Eilb101	100	1114	14	1315	14	18	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP - Metaheurísticas Campo eléctrico - alfa90"

Promedio del % de exceso sobre el óptimo conocido: **23.0%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística con  $\alpha = 90$ , pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Campo eléctrico con $\alpha = 90$ y cáscara convexa			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	561	5	49	<1
Eil30	29	545	3	613	4	12	<1
Eil33	32	910	4	1001	4	10	<1
Eil51	50	521	5	605	5	16	<1
Eila76	75	847	10	1111	12	31	<1
Eilb76	75	1058	15	1325	16	25	<1
Eilc76	75	745	8	982	8	31	<1
Eild76	75	692	7	793	7	14	<1
Eila101	100	825	8	1012	8	22	<1
Eilb101	100	1114	14	1313	14	17	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Campo eléctrico con cáscara convexa -  $\alpha 90$ ".

Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **22.7%**

Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

Al implementar y ejecutar esta metaheurística con  $\alpha = 65$  se observaron los siguientes resultados:

Metaheurística Campo eléctrico							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Campo eléctrico con $\alpha = 65$			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	467	4	24	<1
Eil30	29	545	3	613	4	12	<1
Eil33	32	910	4	1009	4	10	<1
Eil51	50	521	5	709	8	36	<1
Eila76	75	847	10	1038	12	22	<1
Eilb76	75	1058	15	1297	16	22	<1
Eilc76	75	745	8	947	9	27	<1
Eild76	75	692	7	845	8	22	<1
Eila101	100	825	8	1023	9	24	<1
Eilb101	100	1114	14	1297	14	16	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" sección "VRP - Metaheurísticas Campo eléctrico -  $\alpha 65$ "

Promedio del % de exceso sobre el óptimo conocido: **21.5%**

Promedio de tiempo de ejecución: **< 1 seg.**

Al ejecutar esta metaheurística con  $\alpha = 65$ , pero utilizando la cáscara convexa como criterio de ruteo y compararla con la mejor solución conocida tomando como referencia la TSPLIB se obtuvieron los siguientes datos:

Comparación							
Nombre Instancia	# Ciudades	Mejor solución conocida (TSPLIB95)		Campo eléctrico con $\alpha = 65$ y cáscara convexa			
		Costo	Cant. vehículos	costo	Cant. vehículos	% exceso	Tiempo (seg.)
Eil22	21	375	4	467	4	22	<1
Eil30	29	545	3	613	4	12	<1
Eil33	32	910	4	1001	4	10	<1
Eil51	50	521	5	709	8	36	<1
Eila76	75	847	10	1037	12	22	<1
Eilb76	75	1058	15	1296	16	22	<1
Eilc76	75	745	8	946	9	26	<1
Eild76	75	692	7	852	8	23	<1
Eila101	100	825	8	1018	9	23	<1
Eilb101	100	1114	14	1295	14	16	<1

Obs.: Los reportes correspondientes a estas ejecuciones se encuentran en el apéndice "Reportes de ejecuciones" secciones "VRP - Metaheurísticas Campo eléctrico con cáscara convexa -  $\alpha 65$ ".

Promedio del % de exceso sobre solución utilizando Cáscara Convexa: **21.2%**  
 Promedio de tiempo de ejecución utilizando Cáscara Convexa: **<1 seg.**

## Análisis de resultados y conclusiones

En la siguiente tabla se resumen todos los valores promedio de los porcentajes de exceso obtenidos en la aplicación de las distintas heurísticas:

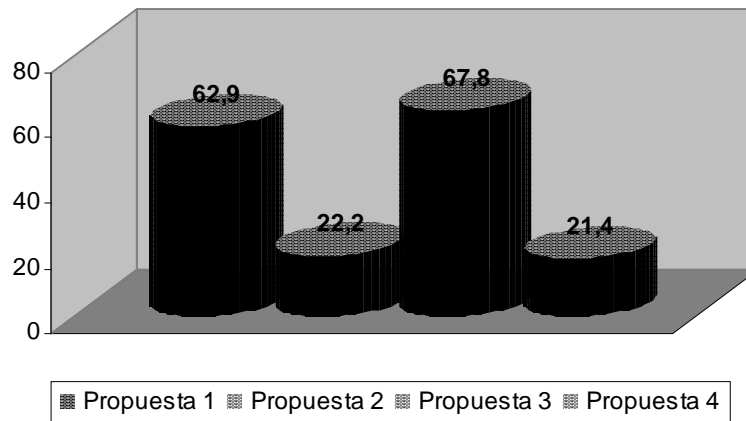
Metaheurísticas		% de exceso promedio sobre la mejor solución conocida (TSPLIB95)
Fuerza de atracción	Propuesta 1	62.9
	Propuesta 2	22.2
	Propuesta 3	67.8
	Propuesta 4	21.4
Fuerza de atracción (primero el de menor fuerza)	Propuesta 5	19.0
	Propuesta 6	23.7
	Propuesta 7	30.9
	Propuesta 8	23.3
Campo eléctrico	$\alpha = 180^\circ$	28.2
	$\alpha = 90^\circ$	23.0
	$\alpha = 65^\circ$	21.5

En el siguiente análisis de resultados se trabaja sobre los datos obtenidos en las ejecuciones anteriormente expuestas. Los promedios utilizados como referencia en ellas fueron obtenidos de la TSPLIB95, no habiendo encontrado más juegos de datos con mejor solución conocida como para poder tener un muestreo más representativo que nos permita ver un comportamiento global de las metaheurísticas.

En función de los resultados obtenidos observamos que hay una gran variación en los porcentajes promedio sobre la mejor solución conocida en todas las metaheurísticas. No pudiendo inferir un comportamiento estable, ni hacer un análisis que explique las

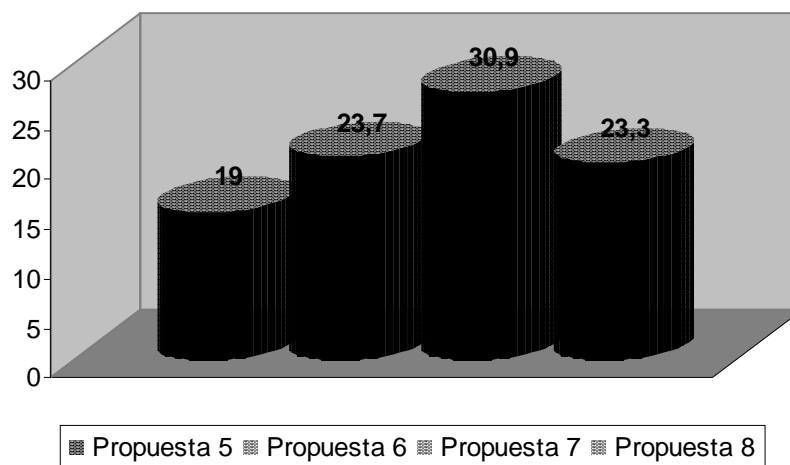
soluciones encontradas con un mejor grado de precisión, nos remitimos a analizar la diferencia entre ellas con respecto al promedio de las soluciones.

**Comparación de % de exceso de las metaheurísticas de fuerza de atracción sobre la mejor solución conocida (TSPLIB)**



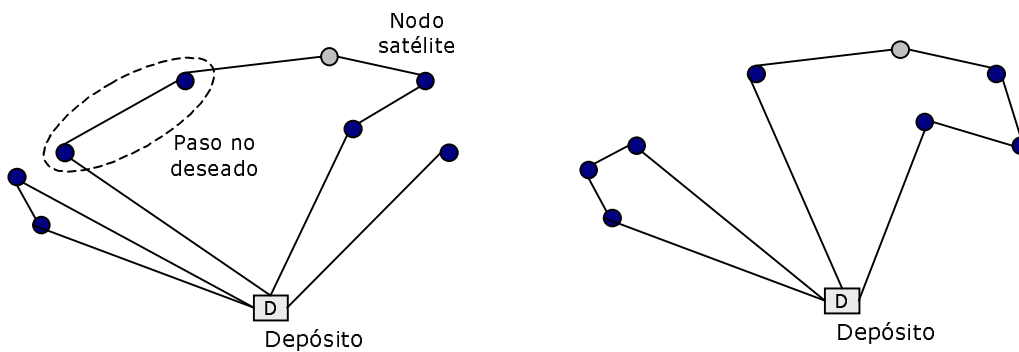
Para las primeras cuatro metaheurísticas se puede apreciar que al aplicar la idea del recálculo de la fuerza en cada nodo para la selección se logra una mejora al respecto a la selección de nodos desde el depósito. Esto se explica porque al seleccionar los nodos que son atraídos con más fuerza desde el depósito quedan, en iteraciones avanzadas de la metaheurística, nodos alejados a ser considerados en las últimas iteraciones. Estos nodos van a incurrir en un alto costo de inserción aumentando el costo total de la suma de los tours.

**Comparación de % de exceso de las metaheurísticas de fuerza de atracción "primero el de menor fuerza" sobre la mejor solución conocida (TSPLIB)**



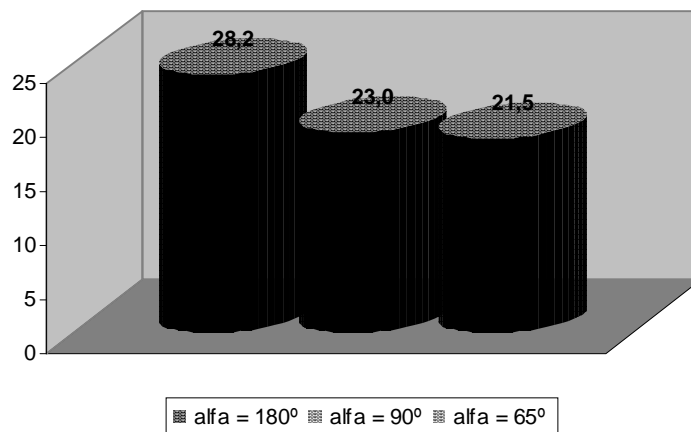
En las propuestas con la variante "primero el de menor fuerza" se pueden apreciar mejores resultados que en las cuatro primeras propuestas. Esto se debe a que la selección se realiza a partir de aquellos nodos satélites que en los casos anteriores

provocaban altos costos de ruteo. En estas propuestas se puede apreciar un comportamiento opuesto en cuanto a la aplicación de la idea del recálculo de las fuerzas de atracción en cada nodo. Esto se explica a que estamos realizando el cálculo desde un nodo satélite y por lo tanto si calculamos la fuerza de atracción desde el mismo se tiende a seleccionar nodos en su entorno. De la otra manera se corre el riesgo de alejarse de ese entorno y seleccionar nodos más lejanos provocando aperturas en cuanto a la ruta formada. En las siguientes figuras se muestra gráficamente el concepto antes explicado:



En cada caso se utilizó tanto TSPCM como cáscara convexa para el ruteo de los nodos seleccionados de cada tour. Al observar la diferencia de resultados se puede ver una mínima variación (menor al 0.4%) en la utilización de cada uno. Analizando esto vemos que los nodos a rutear en cada caso son pocos y que los dos criterios utilizados se comportan de manera similar. Por lo tanto el criterio que "pesa" más en estas metaheurísticas es el de selección de nodos y no el de ruteo.

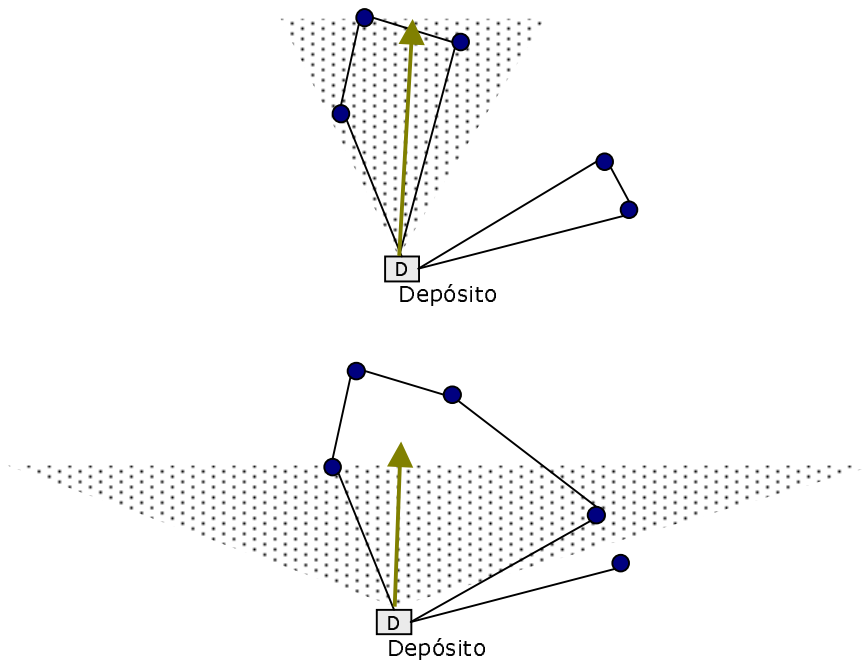
Comparación del % de exceso de las metaheurísticas de campo eléctrico sobre la mejor solución conocida (TSPLIB)



Utilizando las metaheurísticas de campo eléctrico también se observa una varianza importante, pero menor que en las anteriores. Al utilizar el parámetro de apertura con distintos valores apreciamos que los resultados obtenidos mejoran al ser menor el



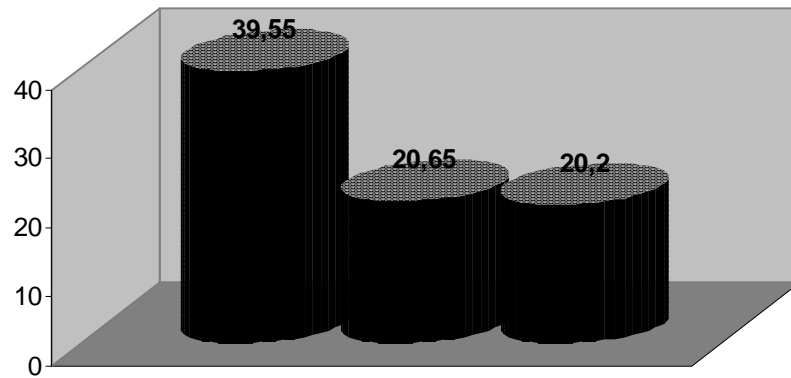
ángulo considerado. A mayor ángulo de apertura se observa menor cantidad de rutas y viceversa. En el caso de menor cantidad de rutas se abastece a más nodos en cada una, pero debe alejarse de la dirección del vector campo eléctrico para abastecer a los últimos de cada una. En el caso de mayor cantidad de rutas, si bien se abastecen a menor cantidad de nodos en cada tour no se recorren grandes distancias para abastecer a los últimos nodos. En las siguientes figuras se muestra gráficamente el concepto antes explicado:



También para estas metaheurísticas se utilizó tanto TSPCM como cáscara convexa para el ruteo de los nodos seleccionados de cada tour. Al observar la diferencia de resultados se puede ver una mínima variación (menor al 0.3%) en la utilización de cada uno. Atribuimos esta variación casi despreciable a lo mismo concluido para las metaheurísticas de fuerza de atracción, que en cada paso, son pocos los nodos a rutear los dos criterios utilizados se comportan de manera similar. Por lo tanto el criterio que “pesa” más en estas metaheurísticas, también es el de selección de nodos y no el de ruteo.

Con respecto a los tiempos de ejecución de todas las metaheurísticas, tanto para un criterio de ruteo como para otro se pueden ver tiempos prácticamente despreciables. Esto se debe a que las metaheurísticas son constructivas, es decir que no se itera luego de calculado un tour para mejorarlo y que en cada uno la cantidad de nodos considerados es “baja” con respecto al criterio de ruteo a utilizar. La siguiente gráfica muestra la comparación de los promedios obtenidos para los tres conjuntos de metaheurísticas propuestos con la mejor solución conocida.

**Comparación del % de exceso sobre la mejor solución conocida (TSPLIB)**



- Metaheurísticas Fuerza de atracción
- Metaheurísticas Fuerza de atracción, variante "primero el más lejano"
- Metaheurísticas Campo Eléctrico

Concluimos que no se han logrado buenos resultados al utilizar esta metaheurísticas y que tampoco tenemos los suficientes datos como para realizar un análisis más profundo al respecto.

## 6 Verificación de los resultados

Los algoritmos que se propusieron e implementaron fueron verificados utilizando un módulo del sistema creado específicamente para poder analizar el comportamiento de los mismos durante el proceso de su ejecución.

Este módulo, llamado *Paso a paso*, permite ser utilizado por los componentes que implementan la lógica de los algoritmos. Utilizando métodos accedidos a través de la interfaz que provee, es posible especificar dentro del código del algoritmo eventos que acceden a la interfaz gráfica de la herramienta **Visual Tour** permitiendo visualizar la ejecución del algoritmo de manera parcial.

Este componente funciona integrado a la interfaz de usuario, y es de suma ayuda para la comprensión del comportamiento para un algoritmo. Permite seguir paso a paso la ejecución del mismo, permitiendo detectar anomalías en el comportamiento del método el cual se este ejecutando. En el siguiente capítulo se dará una explicación más detallada para mejorar la comprensión de la lógica e interfaz de este componente.

### 6.1 Instancias usadas y condiciones de los experimentos

Fue necesario para poder realizar la verificación de resultados de los algoritmos que se desarrollaron, disponer de un conjunto de instancias tanto para el TSP como para el VRP que permitan realizar comparaciones de resultados. TSPLIB 95 [TSPLIB95] mantiene una base de datos de dominio público, que reúne un conjunto de instancias de ambos problemas, presentando las mejores soluciones encontradas o inclusive soluciones óptimas sobre estas. De nuestra consideración fueron utilizadas aquellas instancias de tipo euclidiano en dos dimensiones, que son aquellas con las que trabaja la herramienta **Visual Tour**.

Esta librería es accesible a través de Internet, e incluye instancias de diversos tipos de problema relacionados con el ruteo de vehículos en sus diversas variaciones.

Cada archivo de la TSPLIB 95 que fue utilizado para nuestros procesos de verificación, representa una instancia de los problemas antes mencionados. En su totalidad consiste en un encabezado que especifica el formato del archivo y características generales de la instancia tales como cantidad de ciudades y tipo de distancia utilizada, y un cuerpo que contiene los datos de las ciudades (coordenadas de cada una, y una estructura que refleja las distancias).

Esta base de datos es utilizada en el ambiente académico, y es usual que sus instancias se utilicen como ejemplos en la mayoría de los trabajos de investigación sobre los diversos problemas relacionados con el ruteo de vehículos.

Además de las instancias provistas por esta librería, se definieron y ejecutaron instancias propias creadas por nosotros, las cuales considerábamos de interés para las pruebas. Aquellas instancias utilizadas para el problema del TSP fueron comparadas contra ejecuciones de heurísticas conocidas, como son la del vecino más cercano o la cascar convexa, siendo ambas implementadas en la herramienta de ruteo **Visual Tour**. Para el caso del problema de VRP, al no contar con la implementación de alguna

heurística conocida para la ejecución del mismo, no se pueden comparar los resultados obtenidos contra resultados conocidos. De todas formas, al contar con una representación gráfica de la salida de los algoritmos, esto sirvió para determinar ciertos comportamientos que de manera intuitiva pudieron determinar comportamientos adecuados o no según el caso.

Los nuevos juegos de datos creados se realizaron en algunos casos explícitamente, indicando en su totalidad los datos de la instancia para determinar compartimientos en topologías definidas, y en otros casos de manera aleatoria utilizando una funcionalidad provista por la herramienta.

En el *cd-rom* adjunto a esta documentación se encuentran de forma detallada las distintas ejecuciones de los algoritmos planteados en un formato de reporte. Estos reportes son generados utilizando la herramienta Visual Tour, y constan con información referente a la instancia ejecutada, las rutas generadas y una imagen que corresponde al mapa que se forma para ese sistema de nodos bajo la ejecución seleccionada.

La herramienta visual Tour fue ejecutada sobre una plataforma Microsoft Windows XP Professional versión 2002 service pack 1 bajo una arquitectura AMD con procesador Athlon (TM) 1800+ 1.53GHZ con 512 MB de memoria RAM.

## 7 Proceso de desarrollo

El proceso de desarrollo seleccionado para la implementación del sistema fue de tipo iterativo incremental, lo cual indica que posibles inserciones funcionales, entre otras cosas, fueron permitidas durante toda la etapa del desarrollo.

### 7.1 Decisiones de herramientas

Desde un principio se seleccionaron las herramientas que se utilizaron para el desarrollo de la aplicación, como ser lenguaje de programación (MS Visual Basic 6.0) y el lenguaje de diseño (UML) utilizando el manual de proceso provisto por [UMLYP]. El hecho de utilizar un lenguaje orientado a eventos fue porque este provee un fácil manejo en cuanto al desarrollo de componentes visuales. La idea era desarrollar una herramienta que permitiera definir y editar distintas configuraciones de los problemas a resolver de forma gráfica mediante el uso de una interfaz amigable. Con el uso de este lenguaje pudimos balancear la carga de código requerida para la lógica y el requerido para la visualización e interfaz de controles gráficos. Además MS Visual Basic 6.0 desarrolla componentes COM ActiveX compilados como DLL [MSREF1], lo que nos permite desarrollar una herramienta escalable e interoperable.

El proceso que se presenta a continuación es el de la herramienta que fue utilizada para implementar los algoritmos propuestos, testarlos y generar reportes sobre las ejecuciones de los problemas que esta resuelven, entre otros aspectos funcionales que describiremos a lo largo de esta sección.

### 7.2 Conocimiento de los Requerimientos

En esta sección se tratarán los siguientes objetivos:

- Crear los artefactos de la fase de requerimientos, como son las especificaciones de funciones.
- Identificar y clasificar las funciones del sistema.
- Identificar y clasificar los atributos del sistema y relacionarlos con las funciones

#### Categorías de las funciones

En esta sección describiremos las funciones del sistema. Para ello vamos a necesitar categorizar las mismas en función de su importancia. Vamos a clasificar las funciones a fin de establecer prioridades entre ellas.

Categoría de la función	Significado
Evidente	Debe realizarse.
Ocultas	Debe realizarse, aunque no sea visible para el usuario.
Superflua	Opcional, su inclusión no repercute significativamente en el costo ni en otras funciones.

### 7.3 Funciones Básicas

La siguiente tabla muestra las funciones más representativas del sistema que comprende la herramienta **Visual Tour**.

Ref #	Función	Categoría
<b>R 1.1</b>	Resuelve la el problema del TSP con heurísticas conocidas (Cáscara convexa y vecino más cercano)	Evidente
<b>R 1.2</b>	Resuelve la el problema del TSP con las heurística propuestas por nosotros.	Evidente
<b>R 1.3</b>	Resuelve la el problema de CVRP con las heurística propuestas por nosotros.	Evidente
<b>R 1.4</b>	Carga instancias de la TSPLIB para ser ejecutados por los algoritmos programados.	Evidente
<b>R 1.5</b>	Permitir que el usuario pueda definir en tiempo de ejecución su propia instancia del sistema comprendido por las ciudades. Esta instancia podrá ser utilizada para que sea ejecutada por cualquiera de los algoritmos implementados en el sistema.	Evidente
<b>R 1.6</b>	Despliega en una pantalla la disposición de las ciudades en un eje de coordenadas euclidiano.	Evidente
<b>R 1.7</b>	Ejecutar cada algoritmo paso a paso, pudiéndose ver cada etapa de la ejecución del mismo, como ser la decisión del próximo nodo ingresado al tour.	Evidente
<b>R 1.8</b>	Permitir que las instancias definidas por el usuario sean almacenadas de forma persistente.	Superfluo
<b>R 1.9</b>	Cálculo y visualización de datos estadísticos acerca de las ejecuciones de las instancias en cualquiera de los métodos implementados.	Superfluo

## 7.4 Casos de Uso: Descripción de procesos

En esta sección se tratarán los siguientes objetivos:

- Identificar y escribir casos de uso.
- Diseñar diagramas de casos de uso.
- Contrastar los casos de uso de alto nivel con los expandidos.
- Contrastar los casos de uso esenciales con los reales

### Identificación de actores

En nuestra aplicación solo existirá un actor (agente externo), y este va a ser el usuario del sistema. El usuario va a interactuar con el sistema a través de una interfaz gráfica definida. La frontera del sistema será el sistema de hardware/software comprendido por la computadora donde se instale el mismo.

Actor	Procesos relevantes
Usuario	Selecciona la instancia del problema ya existente, tanto para el TSP como para CVRP. Define nuevas instancias de problemas, tanto para el TSP como para CVRP. Resuelve instancias de TSP y CVRP, con distintos métodos provistos por el sistema. Solicita estadísticas de la resolución.

### Casos de Uso en alto nivel

Ahora se describirán las secuencias de eventos de los actores que utilizan el sistema. Los siguientes casos de uso de alto nivel describen clara y concisamente los distintos procesos que ofrece el sistema.

**Caso de uso:            Seleccionar Instancia de TSP y CVRP.**

**Actores:**            Usuario.

**Tipo:**                Primario y esencial.

**Descripción:**    El usuario selecciona una instancia de datos a partir de un archivo de texto con extensión definida, que será cargado por el sistema para su uso.

**Caso de uso:            Resolver Instancia de TSP y CVRP.**

**Actores:**            Usuario.

**Tipo:**                Primario y esencial.

**Descripción:**    El usuario selecciona un método para resolver el problema de TSP o CVRP, y posteriormente el sistema resuelve esa instancia proporcionando los datos asociados a la misma.

**Caso de uso:            Diseño de instancias.**

**Actores:** Usuario.  
**Tipo:** Secundario.  
**Descripción:** El usuario mediante el uso de una herramienta de diseño, proporcionada por el sistema, crea su propia instancia de TSP o CVRP. El usuario podrá editar los datos de los nodos de la instancia que se encuentre cargada o de la definida por el usuario.

**Caso de uso:** **Resolución paso a paso.**

**Actores:** Usuario.  
**Tipo:** Secundario.  
**Resumen:** El usuario, una vez seleccionada la instancia de TSP o CVRP, le indica al sistema que ejecute un algoritmo, pero que lo vaya haciendo detenidamente en función de algún parámetro propio de cada algoritmo. De esta manera se puede ir visualizando el proceso de selección propio de cada método.

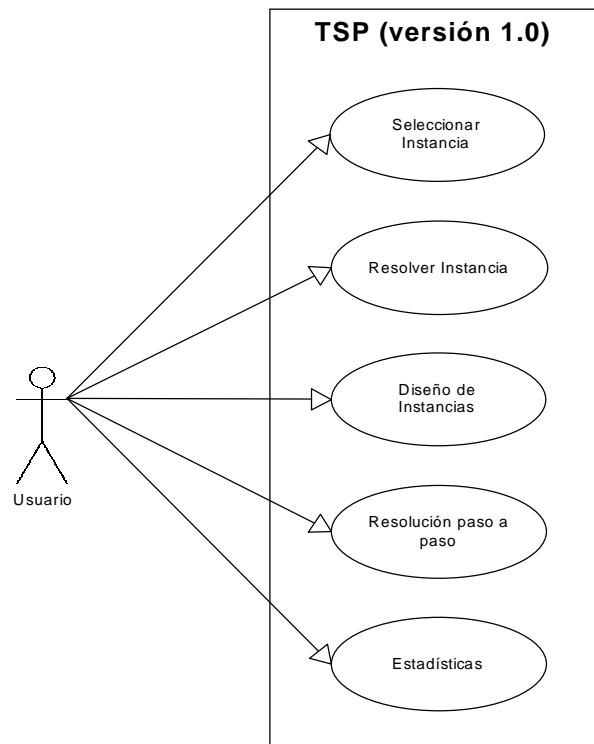
**Caso de uso:** **Presentación de estadísticas.**

**Actores:** Usuario.  
**Tipo:** Secundario.  
**Descripción:** Posterior a la resolución de la instancia, el sistema le brindará al usuario la posibilidad de consultar distintas estadísticas referentes a la ejecución del algoritmo.



## 7.5 Diagramas de los casos de uso

En esta sección se describirán los procesos utilizando diagramas de casos de uso.



## 7.6 Casos de Uso expandidos

### **Caso de uso: Seleccionar Instancia del TSP.** **Sección principal**

**Caso de uso:** Seleccionar Instancia del TSP.

**Actores:** Usuario.

**Propósito:** Seleccionar una instancia ya existente del TSP.

**Resumen:** El usuario selecciona una instancia de datos a partir de un archivo de texto con extensión definida, que será cargado por el sistema para su uso.

**Tipo:** Primario y esencial.

**Referencias**

**cruzadas:** *Funciones:* R 1.4

Curso normal de los eventos	
Acción del actor	Respuesta del sistema
Este caso de uso comienza cuando el usuario decide resolver una instancia de datos con alguno de los métodos de resolución tanto de TSP como de CVRP.	
El usuario selecciona, mediante el uso de un <i>common dialog</i> , una instancia ya existente.	<p>El sistema valida los datos de esa instancia verificando la estructura y la consistencia de los mismos.</p> <p>Se cargan los parámetros correspondientes a esa instancia en estructuras para su visualización. Se habilita la posibilidad de resolver la misma con alguno de los métodos implementados.</p>

*Cursos alternos:*

- Línea 3: De no ser válidos los datos seleccionados por el usuario, el sistema aborta la carga e indica que hubo un error.

**Caso de uso: Resolver Instancia del TSP o CVRP.**

**Sección: principal**

**Caso de uso:** Resolver Instancia de TSP o CVRP.

**Actores:** Usuario.

**Propósito:** Procesar la información de la instancia de datos para resolver el problema del TSP o CVPR en función del método seleccionado por el usuario.

**Resumen:** El usuario selecciona un método para resolver el problema, y posteriormente el sistema resuelve esa instancia proporcionando los datos asociados a la misma una vez culminada la ejecución.

**Tipo:** Primario y esencial.

**Referencias**

**cruzadas:** *Funciones:* R 1.1, R 1.2, R 1.3

*Casos de Uso:* el Usuario debe haber terminado el caso de uso: Seleccionar Instancia.

Curso normal de los eventos	
Acción del actor	Respuesta del sistema
El usuario selecciona uno de los diferentes métodos para resolver la instancia seleccionada y le solicita al sistema que lo ejecute.	Se procede a resolver el problema con el método seleccionado por el usuario.
	Se muestra el tiempo transcurrido desde que se inició el proceso.
	Una vez finalizado el proceso de cálculo, se le muestra al usuario los datos correspondientes a la solución (distancia total, distancias parciales, definición de los caminos resultantes, tiempo de ejecución).

*Cursos alternos:*

- Línea 2: De haber un problema durante el proceso de ejecución del algoritmo, el sistema aborta la ejecución e indica que hubo un error.

**Caso de uso: Diseño de Instancias.**

**Sección: principal**

**Caso de uso:** Diseño de instancias.

**Actores:** Usuario.

**Propósito:** Diseñar nuevas instancias del TSP o CVRP o modificar las ya existentes para ser resueltas bajo los distintos algoritmos.

**Resumen:** El usuario crea su propia instancia del problema, o modifica la que este en curso. Esto comprende la inserción y eliminación de nodos de forma manual, y la edición de los atributos de cada uno de ellos.

**Tipo:** Secundario.

**Referencias**

**cruzadas:** *Funciones:* R 1.5, R 1.6, R 1.8

**Curso normal de los eventos**

**Acción del actor**

1. El usuario posicionando el cursor sobre la ventana de visualización del mapa decide:
  - a. Insertar un nodo, véase la sección *Insertar un nodo*.
  - b. Editar datos de un nodo existente, véase la sección *Editar datos de nodos*.

**Respuesta del sistema**

2. El sistema registra las posiciones e inserta en la colección de nodos actual al nuevo nodo .

## Sección: Insertar un nodo

### Curso normal de los eventos

#### Acción del actor

1. El usuario presiona el botón izquierdo del ratón sobre un espacio vacío de la ventana de visualización del mapa.

#### Respuesta del sistema

2. El sistema registra las posiciones e inserta en la colección de nodos actual al nuevo nodo. Los datos de este nodo serán los definidos como datos por defecto.

## Sección: Editar datos de nodos

### Curso normal de los eventos

#### Acción del actor

1. El usuario presiona el botón derecho del ratón sobre un nodo ya existente en la ventana de visualización del mapa.
3. El usuario selecciona la opción a configurar:
  - a. Definir al nodo como depósito.
  - b. Modificar datos del nodo.
  - c. Eliminar a ese nodo.

#### Respuesta del sistema

2. El sistema registra las posiciones y despliega un *popup* menú que contiene información y las posibles modificaciones que se pueden efectuar sobre el nodo seleccionado.
4. Eventos en función de la selección:
  - a. Marca al nodo seleccionado como depósito, y deselecta al nodo que estaba marcado como depósito.
  - b. Se despliega un cuadro de diálogo en el cual el usuario puede modificar datos como las coordenadas del nodo, su demanda y su estado como depósito del sistema.
  - c. Elimina al nodo seleccionado de la colección de nodos.

## Caso de uso: Resolución paso a paso.

### Sección: principal

#### Caso de uso: Resolución paso a paso.

**Actores:** Usuario.

**Propósito:** Resolver una instancia de tal manera que el usuario pueda visualizar el proceso de formación del ciclo propuesto por el método seleccionado.

**Resumen:** El usuario, una vez seleccionada la instancia de TSP o CVRP, le indica al sistema que ejecute un algoritmo, pero que lo vaya haciendo detenidamente en función de algún parámetro propio de cada algoritmo. De esta manera se puede ir visualizando el proceso de selección propio de cada método.

**Tipo:** Secundario.

**Referencias**

**cruzadas:** *Funciones:* R 1.7

*Casos de Uso:* debe haber al menos un nodo en la colección de nodos del sistema. Para eso el usuario debe haber terminado el caso de uso: *Seleccionar Instancia*, o haber efectuado al menos una inserción provista por el caso de uso *Diseño de instancias*.

Curso normal de los eventos	Función
Acción del actor	Respuesta del sistema
1. El usuario selecciona la función de resolución paso a paso.	2. El sistema valida la solicitud, y habilita la opción para seleccionar el algoritmo a ser ejecutado.
3. Selecciona un problema a resolver, TSP como CVRP, y un algoritmo para que lo resuelva.	4. Se habilita el comienzo de ejecución para selección
5. Mientras que el algoritmo no culmine, el usuario presionará un botón que corresponderá a la ejecución parcial del algoritmo seleccionado.	6. El sistema ejecuta parcialmente el algoritmo seleccionado hasta el próximo punto de quiebra, visualizando en pantalla el proceder del mismo. Se desplegará un cuadro de texto que irá informando las diferentes decisiones y acciones del proceso.
	Al terminar el proceso el sistema le indica al usuario que finalizó con éxito.

**Caso de uso: Presentación de estadísticas.**

**Sección: principal**

**Caso de uso: Presentación de estadísticas.**

**Actores:** Usuario.

**Propósito:** Presentarle al usuario determinada información perteneciente al calculo del algoritmo seleccionado.

**Resumen:** Posterior a la resolución de la instancia, el sistema le brindará al usuario la posibilidad de consultar distintas estadísticas referentes a la ejecución del algoritmo.

**Tipo:** Secundario.

**Referencias**

**cruzadas:** *Funciones:* R 1.9

*Casos de Uso:* el Usuario debe haber terminado el caso de uso:  
*Resolución paso a paso o Resolución de Instancia.*

Curso normal de los eventos	Función
Acción del actor	Respuesta del sistema
1. El usuario ingresa a la sección de estadísticas del sistema.	2. El sistema le presenta información correspondiente a la ejecución del algoritmo.

**Nota:** Esta funcionalidad aún no ha sido implementada, se definirá en una próxima iteración en el desarrollo del sistema, de no ser posible quedará como trabajo futuro del proceso de desarrollo.

## 7.7 Clasificación de los casos de uso

Vamos a clasificar los casos de uso de tal manera que los casos de alto rango sean tratados al inicio de los ciclos de desarrollo. La estrategia consistirá en escoger primero aquellos casos que influyan profundamente en la arquitectura básica de nuestro sistema. Vamos a utilizar una lista de cualidades que aumentan la clasificación de un caso.

- a. Tener fuerte repercusión en el diseño arquitectónico.
- b. Con relativamente poco esfuerzo obtener información e ideas importantes sobre el diseño.
- c. Incluir funciones riesgosas, urgentes o complejas.
- d. Requerir una investigación a fondo o tecnología nueva y riesgosa.
- e. Representar procesos primarios de línea de negocios.

Caso de Uso	a	b	c	d	e	Suma
Seleccionar Instancia del TSP.	6	2	6	2	10	<b>26</b>
Resolver Instancia del TSP.	3	2	8	3	10	<b>26</b>
Diseño de Instancias.	5	2	8	8	4	<b>27</b>
Resolución paso a paso.	5	2	6	6	4	<b>23</b>
Presentación de estadísticas	1	1	5	8	6	<b>21</b>

Con base a los criterios anteriores de clasificación, a continuación se incluye una clasificación informal y poco rigurosa de los casos de uso de nuestra aplicación.

Clasificación	Caso de uso	Justificación
<b>Alto</b>	Seleccionar Instancia del TSP. Resolver Instancia del TSP.	Corresponde a los criterios de clasificación más altos.
<b>Medio</b>	Diseño de Instancias. Resolución paso a paso.	Agregado funcional. Agregado funcional.
<b>Bajo</b>	Presentación de estadísticas	Agregado funcional.

## 7.8 Construcción del modelo conceptual

En esta sección se tratarán los siguientes objetivos:

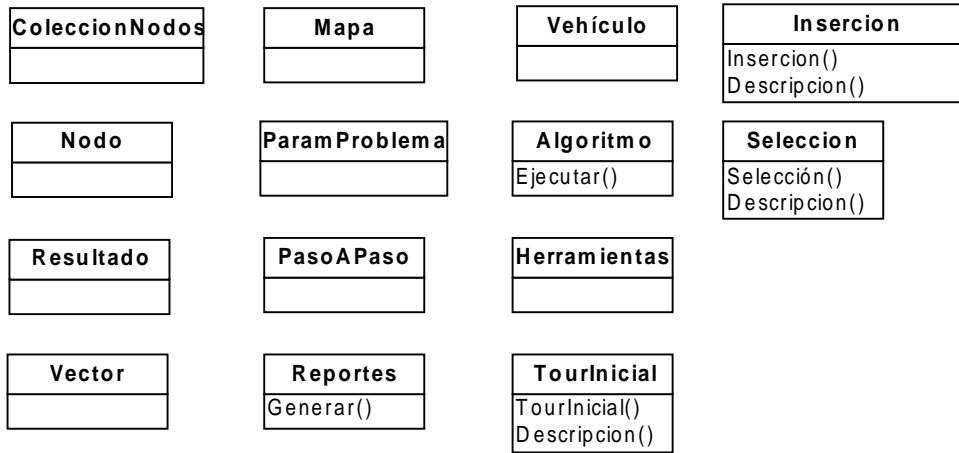
- Identificar los conceptos y asociaciones relacionados con los requerimientos del ciclo actual.
- Crear un modelo conceptual.

### Conceptos

A partir de los casos de uso identificados en la etapa anterior generamos una lista de conceptos adecuados para incluirlos en nuestra aplicación. Esta lista esta sujeta a la restricción de los requerimientos y simplificaciones o agregados que se modelaran posteriormente en la etapa de diseño en donde se ondeará en cuestiones más específica.

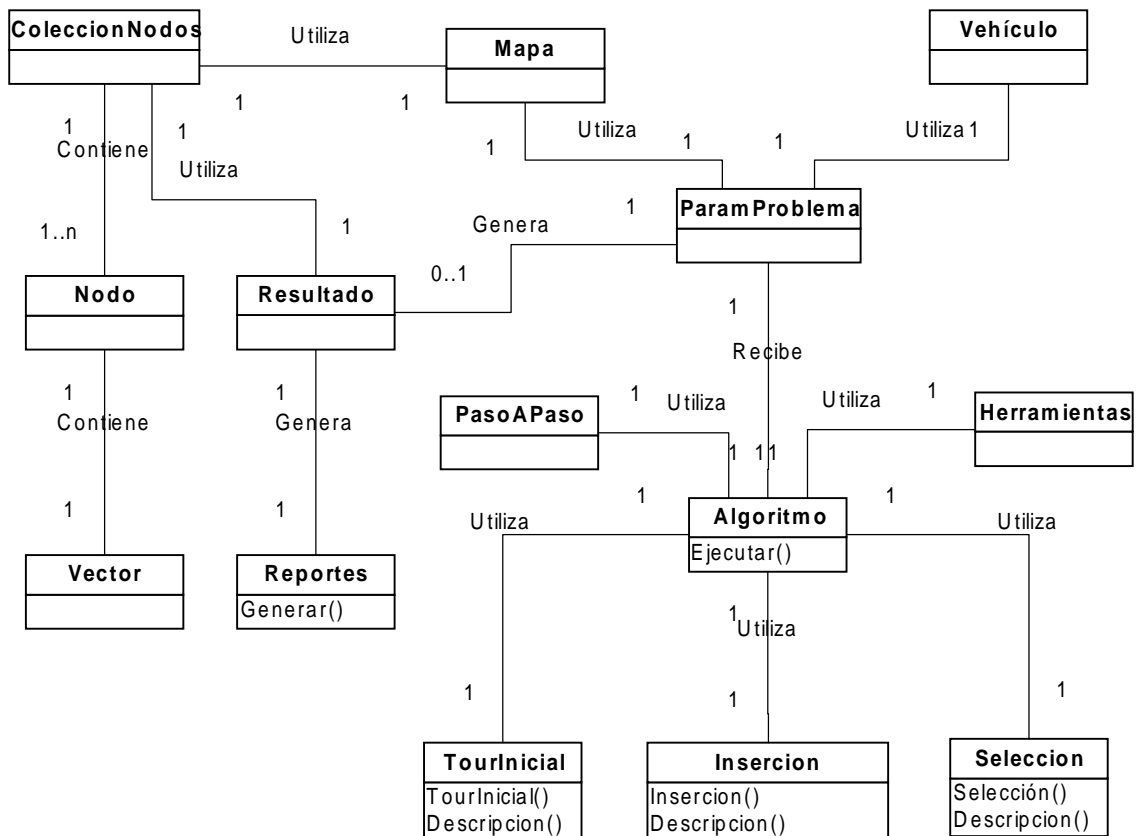
<i>ColecciónNodos</i>	<i>Algoritmo</i>
<i>Mapa</i>	<i>Selección</i>
<i>Vehículo</i>	<i>Resultado</i>
<i>Nodo</i>	<i>PasoAPaso</i>
<i>ParamProblema</i>	<i>Herramientas</i>
<i>Inserción</i>	<i>Vector</i>
<i>Reportes</i>	<i>TourInicial</i>

La lista anterior de nombres de conceptos puede representarse gráficamente en el siguiente diagrama de estructura estática de UML, a fin de mostrar la génesis del modelo conceptual. En la misma ya se agregaron algunos de los atributos necesarios para cumplir con las necesidades de información.



## Modelo Conceptual

Es necesario identificar las asociaciones de los conceptos que se requieren para satisfacer los requerimientos de información de los casos de uso en cuestión. Al combinar los conceptos, asociaciones y atributos obtenemos el modelo de la figura que se muestra debajo. Hemos creado un modelo conceptual relativamente útil del dominio de la aplicación.



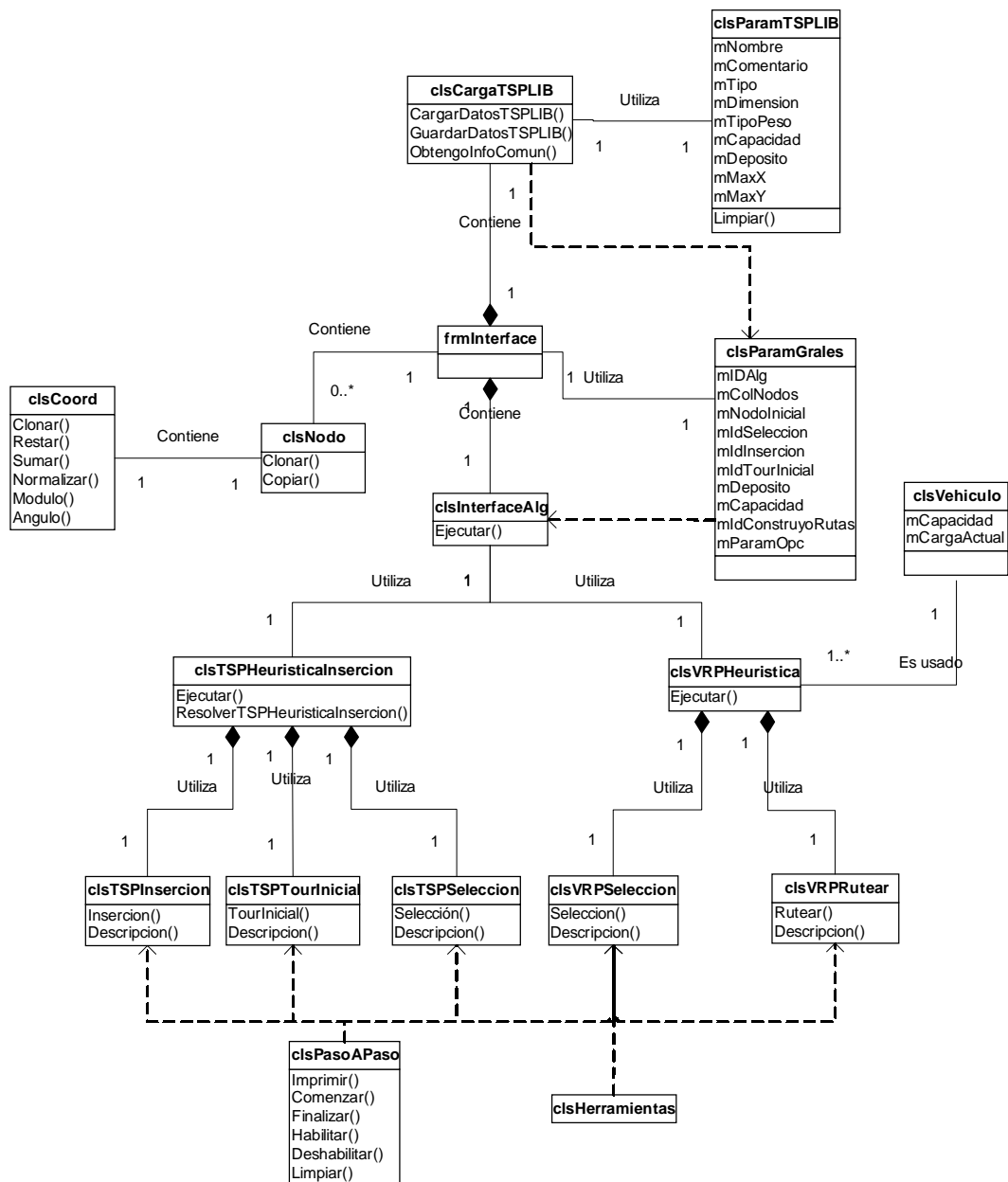


## 7.9 Diagrama de Clases del Diseño

En esta sección se tratarán los siguientes objetivos:

- Crear el diagrama de clases del diseño.
- Identificar las clases, los métodos y las asociaciones que deben incluirse en un diagrama de clases de este tipo.

### Diagrama



En este diagrama se modelan los conceptos que se identificaron durante el proceso de análisis de requerimientos. Los más destacados son,

- frmInterface  
Formulario que contiene los controles utilizados para el desarrollo de la interfaz de usuario. Interactúa con la clase clsInterfaceAlg para resolver los problemas de ruteo.
- clsInterfaceAlg  
Capa intermedia entre la lógica de los algoritmos y la interfaz de usuario.
- clsParamGrales  
Contiene los parámetros necesarios para la ejecución de los algoritmos.
- clsTSPHeurísticaInsercion  
Esqueleto de la heurística de Inserción para la resolución del problema del TSP. Tiene una dependencia con las clases clsTSPInsercion, clsTSPSeleccion, y clsTourInicial que son las encargadas de proveer la implementación de las instancias del algoritmo.
- clsVRPHeurística  
Esqueleto de la heurística propuesta para la resolución del problema del CVRP. Tiene una dependencia con las clases clsTSPSeleccion, y clsRuteo que son las encargadas de proveer la implementación de las instancias del algoritmo.
- clsTSPInsercion  
Es la clase que contiene los distintos métodos desarrollados para la selección de nodos dentro del proceso iterativo de la heurística de inserción de la clase clsTSPHeurísticaInserción.
- clsTSPSeleccion  
Es la clase que contiene los distintos métodos desarrollados para la inserción de nodos dentro del proceso iterativo de la heurística de inserción de la clase clsTSPHeurísticaInserción.
- clsTourInicial  
Es la clase que contiene los distintos criterios desarrollados para seleccionar el tour inicial en la heurística de inserción de la clase clsTSPHeurísticaInserción.
- clsVRPSeleccion  
Es la clase que contiene los distintos métodos desarrollados para la selección de nodos dentro del proceso iterativo de la heurística del CVRP de la clase clsVRPHeurística.
- clsVRPRuteo  
Es la clase que contiene los distintos métodos desarrollados para el ruteo de nodos dentro del proceso iterativo de la heurística del CVRP de la clase clsVRPHeurística.
- clsPasoAPaso  
Esta clase provee una interfaz para el desarrollador que le permite agregar a su implementación la posibilidad de realizar detenciones con visualizaciones gráficas en los secciones del código que considere necesarias.

## 7.10 Descripción e Implementación

El componente que contiene la lógica para la resolución de los distintos algoritmos propuestos es clsInterfaceAlg. Esta clase se utiliza como capa intermedia entre el formulario de interfaz de usuario y los componentes que contienen la lógica de los algoritmos propuestos. La idea tras esta arquitectura es proveer independencia entre ambas capas, permitiendo la creación e inserción de nuevas propuestas sin la necesidad de regenerar toda la aplicación. Los parámetros para la ejecución de los

algoritmos se pasan a través de un componente definido por la clase `clsParamGenerales` el cual fue definido con ese propósito.

Los parámetros que controla este componente son los que se muestran a continuación.

- `Private mIDAlg As String`  
Identificador del algoritmo.
- `Private mColNodos As Collection`  
Nombre y camino del archivo donde se encuentran los datos.
- `Private mNodoInicial As Long`  
Identificador del nodo Inicial.
- `Private mIdSeleccion As Long`  
Identifica el tipo de selección de nodo al tour para ciertos algoritmos.
- `Private mIdInsercion As Long`  
Identifica el tipo de inserción del nodo al tour para ciertos algoritmos.
- `Private mIdTourInicial As Long`  
Identifica un tipo de construcción para el Tour Inicial.
- `Private mDeposito As String`  
Identifica al nodo depósito.
- `Private mCapacidad As Long`  
Capacidad del Vehículo.
- `Private mIdConstruyoRutas As String`  
Identifica el algoritmo de asignación de rutas entre los nodos seleccionados por el método de selección del VRP.
- `Private mParamOpc As Integer`  
Parámetro opcional para los algoritmos.

La capa lógica de la herramienta será un componente COM MS Windows compilado como DLL. Este componente proveerá la interfaz que permitirá a la herramienta realizar un despliegue gráfico de tanto los resultados como de el proceso de desarrollo de los algoritmos, de ser esto provisto por el mismo.

La clase `clsInterfaceAlg` utiliza dos componentes autónomos que fueron desarrollaron para implementar las heurísticas definidas. Estos dos componentes implementan los algoritmos para la resolución de los problemas de TSP y CVPR, y están modelados por las clases `clsTSPHeurísticaInsercion` y `clsVRPHeurística` respectivamente.

### **clsTSPHeurísticaInsercion**

Los algoritmos propuestos para la resolución del problema del vendedor viajero se basan en el esquema de la heurística de inserción. Esto nos permite identificar tres instancias separadas dentro del proceso de ejecución del mismo. La primera de ellas será la que comprende la selección de un tour inicial, la segunda determinará un criterio de selección y la tercera un criterio de inserción, estas dos últimas pertenecientes a la etapa iterativa del algoritmo.

La interfaz pública de esta clase es la función `Ejecutar`, que retornará el tour inicial en función de la configuración provista por el parámetro de entrada. La siguiente figura nos muestra el cuerpo de la función `Ejecutar`.

```
Public Function Ejecutar(pParamGrales As clsParamGrales) As Variant

    *** Ejecuto el algoritmo
1   Set Ejecutar = ResolverTSPHeuristicaInsercion(pParamGrales)

End Function
```

La función ResolverTSPHeuristicaInsercion implementa el esqueleto de la heurística de inserción, a continuación se presenta el código de la misma.

```
Private Function ResolverTSPHeuristicaInsercion(
    pParamG As clsParamGrales) As Collection

Dim pTour As Collection
Dim pColNoVisitados As Collection
Dim pSeleccion As Object
Dim pTSPTourInicial As New clsTSPTourInicial
Dim pTSPSeleccion As New clsTSPSeleccion
Dim pTSPInsercion As New clsTSPInsercion
Dim pTools As New clsHerramientas

    *** Creo la colección donde se almacenará el tour y la
    *** la colección de nodos no visitados
1   Set pTour = New Collection
2   Set pColNoVisitados = New Collection

    *** Copio Colección de nodos no visitados
3   pTools.CopiarColeccion pParamG.ColNodos, _
        pColNoVisitados

    *** Construyo un tour inicial en función a algún criterio
4   Set pTour = pTSPTourInicial.TourInicial(pParamG, _
        pTour, _
        pColNoVisitados)
5   If frmInterface.Tag = "0" Then Exit Function

    *** Mientras halla nodos sin visitar
6   Do While pColNoVisitados.Count <> 0

        *** Selecciono el nodo no visitado según un criterio
7       Set pSeleccion = pTSPSeleccion.Selección(pParamG.IdSeleccion,
            pColNoVisitados, _
            pTour, _
            pParamG.ColNodos)
8       If frmInterface.Tag = "0" Then Exit Function

        *** Inserto el nodo seleccionado según un criterio de
        *** inserción
9       pTSPInsercion.Insercion pParamG.IdInsercion, _
            pSeleccion, _
            pTour, _
            pColNoVisitados
10      If frmInterface.Tag = "0" Then Exit Function
11  Loop

    *** Agrego el último nodo con el primero
```

```

12 pTour.Add pTour.Item(1).Clonar
13 Set ResolverTSPHeuristicaInsercion = pTour
End Function
    
```

Estas tres instancias las modelamos con las siguientes clases, con sus respectivas interfaces. En líneas generales cada clase provee dos métodos, uno de ellos es el que permite ejecutar la lógica implementada, y el otro brinda una descripción del método seleccionado. Dicha descripción debe ser provista por el programador, en tiempo de desarrollo.

#### **clsTSPTourInicial**

- Public Function TourInicial(pParamGrales As clsParamGrales, \_  
pTour As Collection, \_  
pColNoVisitados As Collection) As Collection
- Public Function Descripcion(pIdTourInicial As Long) As String

#### **clsTSPSeleccion**

- Public Function Selección(pCriterio As Integer, \_  
pColNoVisitados As Collection, \_  
pTour As Collection, \_  
pSistemaNodos As Collection) As Object
- Public Function Descripcion(pSeleccion As Long) As String

#### **clsTSPInserción**

- Public Sub Insercion(pCriterio As Integer, \_  
pSeleccion As Object, \_  
pTour As Collection, \_  
pColNoVisitados As Collection)
- Public Function Descripcion(pInsercion As Long) As String

#### **clsVRPHuerística**

El siguiente componente es el que implementa los algoritmos para resolver el problema de ruteo de vehículos. El diseño es análogo al componente de la heurística de inserción. Se considera el cuerpo de un algoritmo en el cual se identifican dos procesos que juntos conformarán una solución a un problema. La idea general de este algoritmo es definir un depósito y entrar en un proceso iterativo en el cual por cada paso del mismo se seleccionan nodos para luego ser ruteados. La selección y el ruteo se realizan en función de criterios que deben ser provistos por los componentes que implementan los mismos. La lógica de las nuevas propuestas esta implementada en dos clases, clsVRPSeleccion y clsVRPRuteo correspondientes a la selección y al ruteo respectivamente.

A continuación se muestra el escueto del algoritmo planteado.

```

Private Function ResolverVRPFuerzaDeAtraccion(pParamG As sParamGrales)
    As Collection

Dim pDeposito As clsNodo
Dim pNodeSel As clsNodo
Dim pVehiculo As New clsVehiculo
Dim pColNoVisitados As New Collection
Dim pTour As New Collection
    
```

```

Dim pTourAux As New Collection
Dim pSeleccion As New Collection
Dim pVRPSeleccion As New clsVRPSeleccion
Dim pVRPRutear As New clsVRPRutear
Dim pTools As New clsHerramientas

    '** Copio Colección de nodos no visitados
1   pTools.CopiarColeccion pParanG.ColNodos, _
        pColNoVisitados

    '** Seleccionar el depósito
2   Set pDeposito = pTools.BuscarNodo(pParanG.ColNodos, _
        pParanG.Deposito)

    '** Inicializo el vehículo con carga total
3   With pVehiculo
4       .Capacidad = pParanG.Capacidad
5       .CargaActual = pParanG.Capacidad
6   End With

    '** Inicializo al nodo con el cual voy a comenzar la ejecución del
    '** algoritmo como el depósito
7   Set pNodeSel = pDeposito

    '** Marco a este nodo como visitado
8   pTools.EliminarNodo pColNoVisitados, _
        pNodeSel.NodoId

    '** Mientras halla nodos sin visitar
9   Do While pColNoVisitados.Count <> 0

        '** Selecciona los nodos según el criterio elegido
10      Set pSeleccion = pVRPSeleccion.Seleccion(pParanG.IdSeleccion,
            pNodeSel, _
            pColNoVisitados, _
            pDeposito, _
            pVehiculo, _
            pParanG.ParamOpc)

11      If pSeleccion.Count = 0 Then Exit Function

        '** Rutea los nodos seleccionados según el criterio elegido
12      Set pTourAux = pVRPRutear.Rutear(pParanG.IdInsercion, _
            pSeleccion, _
            pDeposito)

        '** Agrego el Tour auxiliar al tour Final
13      Set pTour = AgregoAlTour(pTour, pTourAux)

        '** Inicializo los datos
14      pVehiculo.CargaActual = pVehiculo.Capacidad
15      Set pNodeSel = pDeposito
16      Set pSeleccion = Nothing
17      Set pSeleccion = New Collection

18  Loop

```

```

    *** Agrego el último nodo con el primero
19     pTour.Add pTour.Item(1).Clonar

20     Set ResolverVRPFuerzaDeAtraccion = pTour

End Function

```

Al igual que las clases `clsTSPSeleccion` y `clsTSPInsercion` descritas anteriormente, cada clase provee dos métodos, uno de ellos es el que permite ejecutar la lógica implementada, y el otro brinda una descripción del método seleccionado. Dicha descripción debe ser provista por el programador, en tiempo de desarrollo.

### **clsVRPSeleccion**

- Public Function `Seleccion(pCriterio As Integer, _  
pNodoActual As clsNodo, _  
pColNoVisitados As Collection, _  
pDeposito As clsNodo, _  
pVehiculo As clsVehiculo, _  
Optional pParamOpt As Integer) As Collection`
- Public Function `Descripcion(pIdSeleccion As Long) As String`

### **clsVRPRuteo**

- Public Function `Rutear(pCriterio As Integer, _  
pColNodos As Collection, _  
pDeposito As clsNodo) As Collection`
- Public Function `Descripcion(pRuteo As Long) As String`

### **clsPasoAPaso**

La siguiente clase provee una interfaz que permite al programador proveer a su algoritmo con la modalidad de paso a paso. Esto es interpretado, por el formulario contenido en la interfaz de usuario, como una funcionalidad de ese algoritmo que permite la ejecución del mismo realizando detenciones en puntos definidos por el implementador en el momento del diseño del algoritmo. El componente está integrado directamente con el formulario `clsInterface` puesto que utiliza primitivas de componentes de este para su procesamiento. El implementador contará con las siguientes primitivas.

- Public Sub `Imprimir(pNodos1 As Variant, _  
pLeyenda1 As String, _  
pColor As ColorConstants, _  
Optional pNodos2 As Variant, _  
Optional pLeyenda2 As String)`
- Public Sub `AvanzarPaso(pColNodos As Collection, _  
pLeyenda As String, _  
Optional pPersistente As Boolean = False)`
- Public Sub `Comenzar()`
- Public Sub `Finalizar()`
- Public Sub `Habilitar()`
- Public Sub `Deshabilitar()`
- Public Sub `Limpiar()`

- Private Sub Inicializar()
- Public Sub ImprimirRutas(pColNodos As Variant)



## 8 Conclusiones y trabajo futuro

El objetivo planteado en este proyecto fue el de desarrollar nuevas metaheurísticas para la resolución del TSP y VRP basándonos en conceptos de la física y la microfísica. En esta incursión las propuestas definidas se centran en tres campos de estudios. Para el TSP nos basamos en el concepto de centro de masas de un sistema de partículas, y para el caso del VRP nos basamos en los conceptos de fuerza eléctrica de las cargas y campo eléctrico.

En el proceso de búsqueda de soluciones, se propuso en una primera instancia un conjunto de metaheurísticas para la resolución del TSP que se basaron en la idea de la selección de nodos en función de la cercanía al centro de masas. Este supuesto no brindó los resultados esperados, por lo que luego de un análisis se procedió a formular diversas alternativas considerando los nodos más lejanos. La nueva propuesta fue claramente más eficaz que la anterior, y fue posteriormente utilizada para las instancias de ruteo en las metaheurísticas de VRP propuestas.

Por otro lado, la utilización de los conceptos de fuerza eléctrica y campo eléctrico para la resolución del VRP no resultó ser del todo exitosa. Las metaheurísticas propuestas no lograron un comportamiento estable en el conjunto de instancias utilizadas. Los porcentajes sobre las mejores soluciones conocidas no fueron los deseados, y se obtuvo una gran variación de resultados a causa de pequeñas variaciones en los criterios definidos. Esto deja abiertas las puertas para continuar estudiando diversas alternativas a estas propuestas, de tal manera que se logre generar mejores soluciones.

El campo temático de la física es muy amplio y las analogías posibles con los problemas planteados muy variadas, por lo que queda mucho campo por explorar. Otras posibles áreas de estudio son las ondas, rayos y la refracción de la luz los cuales fueron considerados pero no profundizados para la búsqueda de soluciones (Ver anexos "Conceptos Físicos").

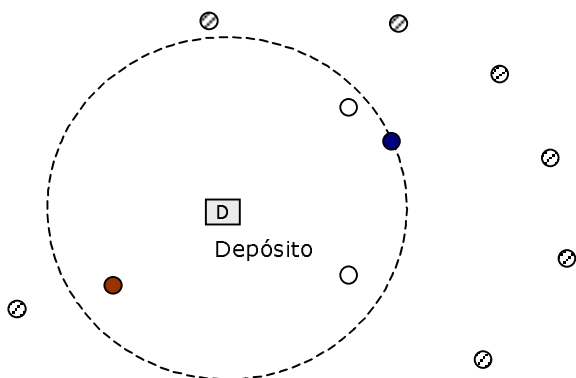
Una vez planteada, implementada y ejecutada una metaheurística el análisis de resultados no resultó trivial. Las instancias de pruebas de la TSPLIB utilizadas no son suficientes para determinar con certeza la eficacia de los algoritmos. Los casos que representan grandes dimensiones o diversidad en las topologías no fueron considerados durante el proceso de verificación y análisis debido a que la librería de referencia utilizada no los ofrecía.

A comienzos del proyecto se tuvo como objetivo desarrollar un Algoritmo Genético utilizando ideas de la física para los operadores de cruzamiento y mutación. Para esto se estudió e investigó sobre el tema, e inclusive posteriormente, se cursó la asignatura electiva Algoritmos Genéticos dictada en la facultad, en parte con el objetivo de que fuera de ayuda para la búsqueda de soluciones. Lamentablemente las ideas que surgieron no parecían ser eficaces, y requerían de operadores de reparación haciendo que se perdiera el concepto de la física en la solución determinada. (Ver anexos "Algoritmos Genéticos")

Planteamos como trabajo futuro seguir refinando los criterios utilizados en las metaheurísticas propuestas con el objetivo de disminuir la varianza observada en los

resultados obtenidos al contrastarlos con las soluciones conocidas. Es de considerar la necesidad de utilizar para la verificación de resultados una base de datos mayor para poder trabajar sobre un muestreo más representativo que permitir realizar un mejor análisis de resultados.

En particular en dos de las heurísticas y metaheurísticas planteadas, se puede continuar con el análisis de varianza de resultados al utilizar diferentes valores para los parámetros de entrada. Uno de los parámetros de entrada considerado en unas propuestas es el porcentaje de la capacidad del vehículo que se considera para "dar la vuelta" al depósito y no visitar nodos que se encuentren a mayor distancia de él. En los casos de estudio propuestos se utilizó el 50% como parámetro pudiéndose comparar los resultados obtenidos con otros valores. En las metaheurísticas que utilizan esta idea de "vuelta" el valor del parámetro es planteado como fijo durante toda la ejecución. Se puede considerar un valor del parámetro variable en función del avance de la ejecución de la metaheurística o inclusive ser dinámico en función de la capacidad con que llegue de cada tour el vehículo. A las metaheurísticas que utilizan esta idea de "vuelta" cuando han llegado a cierto porcentaje de su capacidad se le ha notado una característica que degrada sus soluciones. Una vez decidida "la vuelta" de acuerdo al parámetro de entrada, el vehículo considera como factibles a pertenecer al tour que se está formando a los nodos tales que su distancia al depósito sea menor que la del vehículo en el momento del cálculo. Esto define una circunferencia de centro el depósito y radio la distancia del vehículo en la cual puede haber nodos mas alejados del vehículo que el propio depósito. Esta situación se ilustra en la siguiente figura:

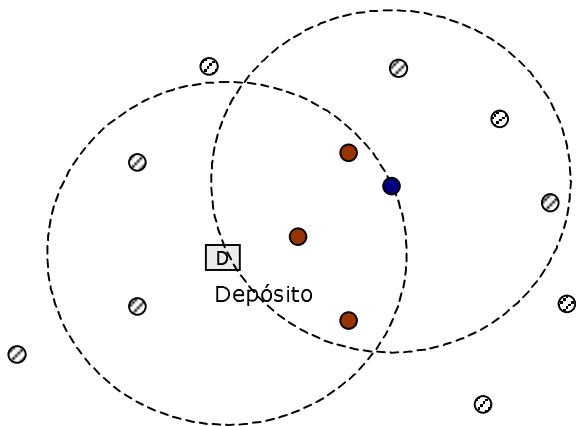


Como se puede apreciar, visitar un nodo con estas características tiene un costo alto, ya que si bien está más cerca del depósito que el vehículo la distancia a recorrer puede llegar a ser hasta el doble. Tres nuevos criterios se pueden plantear para atacar esta solución:

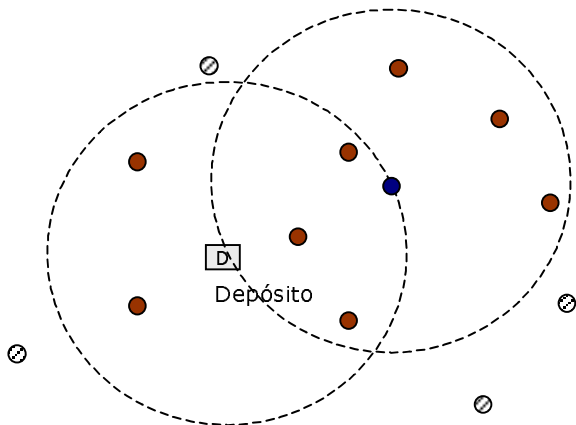
*Obs.:* En las siguientes explicaciones de criterios de selección se utilizan términos informales, los cuales se marcan entre "" para su lectura como tal. La especificación de los criterios se da utilizando términos de estas características porque se intenta transmitir la idea de la manera mas intuitiva posible.

1. Una vez que está "decidido" dar "la vuelta", no considerar aquellos nodos tales que su distancia al depósito sea mayor que la del vehículo al mismo y que tampoco visitarlo tengo un costo mayor que ir al depósito. Esto define otra

circunferencia de centro el vehículo y radio la distancia del mismo al depósito. Los nodos que se consideran (si le alcanza la capacidad al vehículo) para visitar son entonces los pertenecientes a la intersección de ambas circunferencias. Gráficamente se puede ver de la siguiente manera:



2. Tanto en las soluciones propuestas e implementadas, como en la anteriormente descrita, los "radios de acción" o distancias a considerar para la selección de nuevos nodos se recalculan en cada paso. Esto lleva a que rápidamente los nodos cercanos al depósito sean visitados, es decir que en las primeras iteraciones que resultan en los primeros tours quedan la mayoría de los nodos cercanos al depósito visitados. A partir de que estos nodos quedan visitados, el vehículo llegará al depósito con suficiente carga como para abastecer a otros nodos de los aún no visitados, pero como queda "acotado" a los cercanos al depósito se "desperdiciará" esa carga en el tour. Una posible alternativa a las propuestas anteriores para mitigar esto es considerar la o las circunferencias fijas a partir del momento en que se decide dar la vuelta y no recalcularla en cada paso.
3. En la propuesta 1 se puede llegar a "converger" muy rápido al depósito sin "darle tiempo" al vehículo a visitar otros nodos para los cuales le permitiera la capacidad sin alejarse mucho del depósito. En la propuesta 2 se considera no ir recalculando la o las circunferencias en cada paso, lo que podría terminar en que se le terminara la capacidad del vehículo (o le quedara tan poco que no pudiera abastecer a ningún nodo) bastante "lejos" del depósito. Esto podría contrarrestar el beneficio buscado al "obligar la vuelta" del vehículo al depósito. Es por esto que se puede plantear una alternativa "intermedia" entre ambas propuestas. Se considera el recálculo de las circunferencias propuestas en la propuesta 1, pero en vez de considerar los nodos pertenecientes a la intersección de ambas como potenciales a visitar se consideran los nodos de la unión. Esto hace que se "converja" al depósito, pero de una manera mas lenta permitiendo visitar nodos cada vez mas cerca al mismo pero no acotando tanto el "radio de acción" para el vehículo. Gráficamente se puede ver de la siguiente manera:



La analogía planteada, tanto por los criterios propuestos e implementados como en estos se trata de que el depósito empieza a ejercer una fuerza cada vez mayor sobre el vehículo a medida que este se va vaciando. Esperando se encuentre "cerca" de él cuando ya no pueda abastecer a otro nodo y de esa manera logra que el costo de volver al depósito sea bajo.

La performance de los algoritmos en parte está relacionada con las estructuras de datos que se utilizaron para modelar los diferentes conceptos del sistema. En la programación de la herramienta de ruteo se utilizaron estructuras implementadas por nosotros, como también las predefinidas por el lenguaje de programación MS Visual Basic 6.0. En algunas ocasiones el manejo de estas estructuras puede no ser el más adecuado debido a que los órdenes operacionales de sus métodos no son los óptimos. Una posibilidad para mejorar estos tiempos es la de utilizar otras estructuras de datos, por ejemplo las planteadas por *Data structures of travelling Salesman [DSTSP]*. En este documento se realiza un análisis para diferentes estructuras y se proponen diferentes alternativas para mejorar las búsquedas y las operaciones más comunes en la implementación de estos tipos de problemas.

La herramienta implementada, **Visual Tour**, fue diseñada con la capacidad de interoperar con otros sistemas a través de la tecnología XML (*Extensible Markup Language*). A partir de la ejecución de una instancia de datos, utilizando Visual Tour, es posible generar la salida resultante del algoritmo ejecutado en formato XML. Para poder lograr independencia en cuanto a la estructura del XML generado, el módulo encargado de la formación del mismo, fue definido en un componente externo con una interfaz conocida, el cual es utilizado por la Herramienta de ruteo para la generación de este tipo de salidas. Esto permite que pueda tener una estructura variable en función del uso que se le quiera dar a esta salida. De todas maneras, en la estructura por defecto, se trató de incluir de forma clara y bien definida la mayor cantidad de información pertinente tanto a los datos en juego como a los parámetros correspondientes a la ejecución del algoritmo, para facilitar la integración del mismo con una aplicación externa. Una mejor descripción en cuanto a la estructura de datos del archivo XML generado por Visual Tour es especificada en el apéndice del Manual de Usuario.

Un Sistema de Información Geográfica (SIG) puede requerir el uso de este tipo de salidas. Debido a la capacidad de interoperabilidad sería posible conectar los sistemas de captura y almacenamiento de datos de un SIG, ofreciendo la información de datos espaciales a través de coordenadas euclidianas con el fin de realizar operaciones que se pueden realizar sobre ellos. Actualmente la herramienta Visual Tour toma como

entrada archivos de texto que contienen la información de la instancia, con una estructura bien definida propuesta por la TSPLIB [TSPLIB95]. Los SIG almacenan archivos espaciales en forma numérica, para un trabajo futuro, se podría analizar la posibilidad de utilizar esa información como entrada a nuestro sistema.

## 9 Apéndice: Estado del arte del TSP y VRP

El problema básico de ruteo consiste en un conjunto de nodos que pueden o no estar vinculados por aristas, los cuales requieren de un servicio que es brindado por una flota de vehículos. En este caso base no hay restricciones en cuanto al orden en el cual estas entidades deben ser atendidas ni tampoco el momento en el tiempo que requieren el servicio. [RSVSA]

El problema se centra en construir un conjunto de rutas factibles (llamado tour) de bajo costo para cada vehículo, en donde una ruta se entiende como una secuencia de ciudades que un vehículo debe visitar.

El ruteo de vehículos es en primer instancia un problema espacial, por lo tanto se asume que no hay restricciones temporales o de cualquier otro tipo que influyan en la decisión de las rutas a tomar. Solo se considera la restricción de minimizar el largo total de los recorridos.

A continuación vamos a definir los problemas de ruteo de vehículos más conocidos en el ámbito de la investigación operativa y el análisis combinatorio.

El primero de ellos es el Problema del Vendedor Viajero, el cual busca determinar el costo mínimo del ciclo que pasa por todos los nodos de un grafo una sola vez (conocido como ciclo Hamiltoniano). Si los costos de las aristas son simétricos, o sea que el costo de viajar entre dos nodos no depende de la dirección del viaje, entonces tenemos un Problema del Vendedor Viajero Simétrico; de otra forma nos encontramos frente al Problema del Vendedor Viajero Asimétrico o Dirigido. El acrónimo utilizado para definir este problema es TSP por Travelling Salesman Problem.

El Problema del Vendedor Viajero Múltiple es una generalización del Problema del Vendedor Viajero en donde se necesita tener más de un vendedor (o vehículo). La cantidad  $M$  de vehículos de la flota deben de iniciar y culminar su recorrido en un único depósito común. No hay restricciones en cuanto a la cantidad de nodos que cada vehículo debe visitar, solo se requiere que cada vehículo visite al menos un nodo.

El Ruteo de Múltiples Vehículos con un único Depósito (clásico Problema de Ruteo de Vehículos VRP) busca determinar un conjunto de rutas de reparto para los vehículos que se encuentran alojados en un depósito central, de tal forma que abastezca a todos los nodos y que minimice la distancia total recorrida. La demanda de cada nodo se asume que es determinista y que cada vehículo tiene una capacidad de carga conocida. El Problema del Vendedor Viajero Múltiple es un problema VRP con una flota de  $M$  vehículos donde cada uno tiene capacidad infinita.

El Ruteo de Vehículos con Múltiples Depósitos es una generalización del problema previo en el cual la flota de vehículos ahora puede hacer uso de  $D$  depósitos en lugar de solo uno. Todas las restricciones del problema clásico de VRP se mantienen para este caso. Como agregado se solicita que cada vehículo debe regresar al mismo depósito en el cual comenzó su recorrido.

Por último vemos el Ruteo de Múltiples Vehículos con un único Depósito y Demandas Estocásticas el cual es similar al clásico VRP excepto que las demandas de los nodos no se conocen con certeza. Estas surgen a partir de una función de distribución específica.

## Un poco de Historia

### Antesala

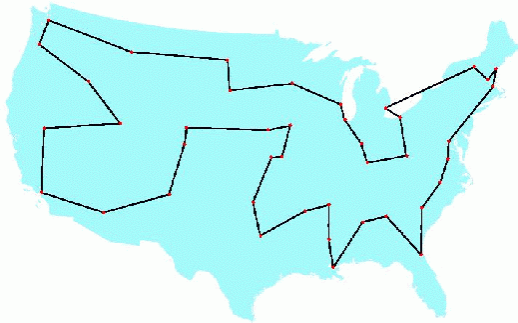
Los problemas matemáticos relacionados con el problema del vendedor que viajaba fueron tratados en el 1800s por el sir irlandés Guillermo Rowan Hamilton y por el matemático británico Thomas Penyngton Kirkman. El cuadro abajo es una fotografía del juego de Icosian de Hamilton que requiere a jugadores terminar viajes a través de los 20 puntos usando solamente las conexiones especificadas. [HIS]



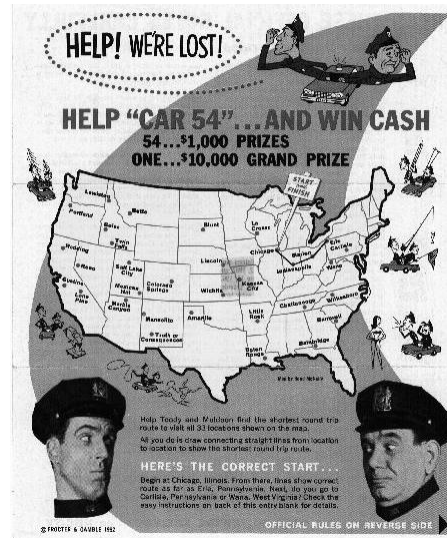
### 1900

En los años 20, el matemático y el economista Karl Menger publicó entre sus colegas en Viena este problema. La forma general del TSP aparece en los años 30 en los círculos matemáticos de Princeton. En los años 40, fue estudiado por los estadísticos (Mahalanobis (1940), Jessen (1942), Gosh (1948), Marks (1948)) en la conexión de uso agrícola y la inundación de Merill. El problema se popularizó entre los matemáticos en el RAND Corporation. A lo largo del tiempo, el TSP ganó notoriedad como el prototipo de un problema difícil en la optimización combinatoria, debido a que examinar los viajes uno por uno es inadmisibles debido a su número grande. Por ello es que se estudian heurísticas y métodos de aproximación que, sin incurrir en un costo elevado den una solución aproximada apropiada al problema. [HIS]

En 1954 George Dantzig, Ray Fulkerson, y Selmer Johnson publicaron una descripción de un método para solucionar el TSP que mostró el poder de este método solucionando un caso con 49 ciudades, un tamaño impresionante para aquella época. Para esa demostración escogieron una ciudad de cada uno de los 48 estados en los E.E.U.U.. (Alaska y Hawaii se convirtió en un solo estado 1959) y agregando Washington; los costos del recorrido entre estas ciudades fueron definidos por las distancias de los caminos. En realidad solucionaron el problema 42-city obtenido de quitar Baltimore, Wilmington, Philadelphia, Newark, York nueva, Hartford, y Providence.

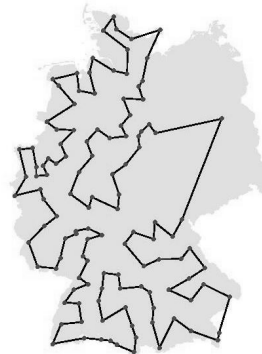
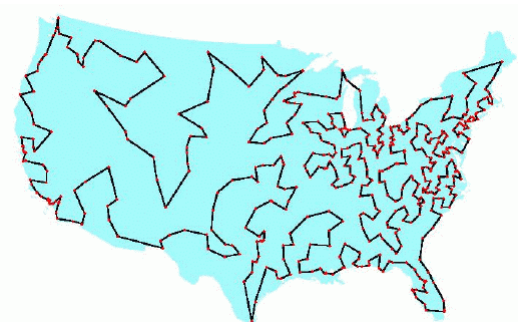


Proctor y Gamble establecieron una competencia en el año 1962. La competencia requirió solucionar un TSP en ciudades especificadas. Uno de los primeros científicos investigadores del TSP, profesor Gerald Thompson de la universidad de Carnegie Mellon, fue uno de los ganadores.

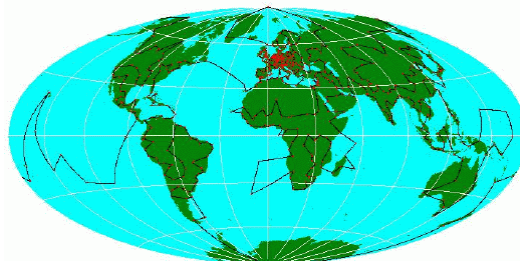


las 33

Groetschel (1977) encontró el viaje óptimo de 120 ciudades de la cuál era entonces la República Federal de Alemania. Padberg y Rinaldi (1987) encontraron el camino óptimo de 532 interruptores de AT&T en los E.E.U.U.

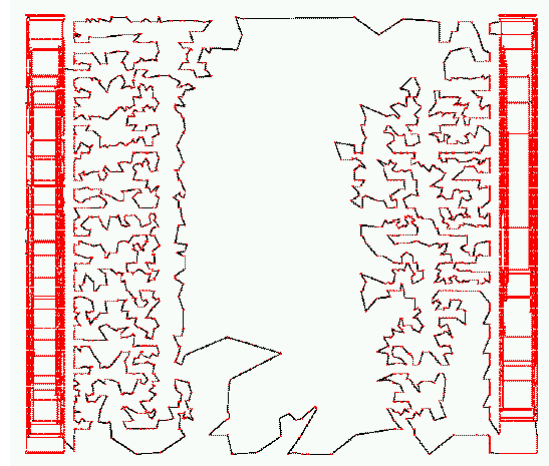
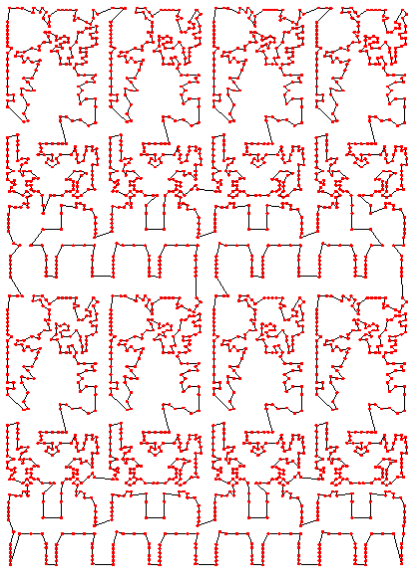


Groetschel y Holanda (1987) encontraron el viaje óptimo de 666 lugares interesantes en el mundo.

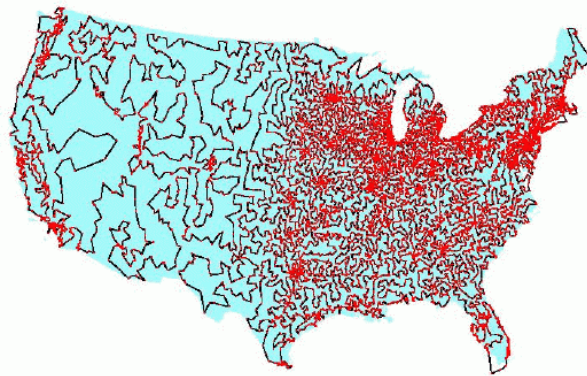


Padberg y Rinaldi (1987) encontraron el camino óptimo de una disposición de 2.392 items que fue obtenida de Tektronics inc.





Applegate, Bixby, Chvátal, y Cook (1994) encontraron el camino óptimo para un 7,397-city TSP que se presentó en una aplicación en los laboratorios de AT&T Bell. Applegate, Bixby, Chvátal, y Cook (1998) encontraron el viaje óptimo de las 13.509 ciudades en los E.E.U.U. con las poblaciones mayor de 500.



Applegate, Bixby, Chvátal, y Cook (2001) encontraron el viaje óptimo de 15.112 ciudades en Alemania.

## TSP

El TSP consiste en visitar un conjunto de  $n$  nodos, comenzando en un nodo específico (nodo origen), debiendo regresar a él, luego de haber visitado todas los demás y sin haber visitado dos veces a alguno. El objetivo es minimizar la longitud de la secuencia de visita de los nodos, bajo el criterio de minimización de la función de costo.

Entonces, el TSP busca determinar el costo mínimo del ciclo que pasa por todos los nodos de un grafo, una sola vez. Si los costos de las aristas son simétricos, o sea que el costo de viajar entre dos nodos no depende de la dirección del viaje, entonces tenemos un TSP Simétrico; de la otra forma tenemos un TSP Asimétrico o Dirigido.

La importancia del TSP dentro del mundo de problemas NP radica en que es utilizado como conjunto de pruebas para los nuevos algoritmos asociados a la resolución de esta clase de problemas. Su naturaleza NP Completo hace posible que se puedan resolver una gran familia de problemas equivalentes mediante transformaciones particulares para cada caso. Además, como el TSP presenta una gran cantidad de aplicaciones en el mundo real, se ha convertido en uno de los problemas más estudiados por la comunidad científica mundial.

Las definiciones y conceptos desarrollados en este capítulo fueron extraídos de la bibliografía y webgrafía de referencia [CH95], [RSVSA], [JEBOR], [DHMCE], [SMAA]

### Definición

Sea un grafo  $G=[N,A,C]$  donde  $N$  es el conjunto de nodos,  $A$  el conjunto de aristas, y  $C=[cij]$  la matriz de costos, donde  $cij$  es el costo de desplazarse o la distancia que existe entre el nodo  $i$  y  $j$ . El problema del vendedor viajero busca determinar el Ciclo Hamiltoniano en  $G$  de menor costo. Se ha demostrado que este problema es de tipo NP-completo. Esto implica que no es factible que exista un algoritmo exacto para el TSP que determine un límite polinomial para este.

El problema puede ser representado a través de una matriz  $n \times n$  (simétrica ó asimétrica) de distancias  $d$  donde  $dij$  es la distancia entre la ciudad  $i$  y la ciudad  $j$ . El valor de  $dii$  es siempre 0. Si tenemos  $n$  ciudades, existen a lo más  $(n-1)!$  circuitos diferentes (y  $(n-1)! / 2$  en el caso del TSP simétrico).

### Complejidad computacional

El problema es del tipo NP-Completo, es decir, es un problema representativo de la clase NP cuyo tiempo de resolución crece en forma exponencial ante incrementos lineales del número de elementos que intervienen en el problema. Por tanto no se ha encontrado un algoritmo que lo resuelva en tiempo polinomial. [DHMCE]

Optimizar significa hallar el máximo o mínimo de una cierta función, definida en algún dominio. Las teorías clásicas de optimización (por ejemplo, el cálculo diferencial) tratan con el caso en que este dominio es infinito. Desde este ángulo, los problemas de optimización combinatoria, donde el dominio es típicamente finito, pueden parecer triviales, pues existe siempre un procedimiento elemental para determinar la solución óptima buscada, por ejemplo realizar una exploración exhaustiva del conjunto de soluciones. Es decir, generar todas las soluciones factibles, calcular para cada una el costo asociado, y elegir finalmente la que haya dado lugar al mejor de ellos. Sin embargo, aunque este método teóricamente llega siempre a la solución buscada, no es eficiente, pues el tiempo de cálculo necesario crece exponencialmente con el número de items del problema. Las posibilidades a revisar pueden incluir todos los árboles de  $N$  nodos, o todos los circuitos hamiltonianos de un grafo completo, y por lo tanto listarlas todas para encontrar la mejor entre ellas es prácticamente imposible aún para instancias de tamaño moderado. Existen problemas combinatorios para los cuales no se conocen algoritmos de resolución, más que aquellos en los cuales se produce una explosión exponencial del tiempo de cálculo al aumentar el tamaño del problema. Son problemas computacionalmente difíciles de tratar. Por el contrario, para otros

problemas combinatorios existen algoritmos que sólo crecen en tiempo polinomial con el tamaño del problema. Matemáticamente se trató de caracterizar a uno y otro tipo de problemas. De aquellos para los cuales se conocen algoritmos que necesitan un tiempo polinomial para ofrecer la solución óptima se dice que pertenecen a la clase P, y se considera que son resolubles eficientemente. Sin embargo, se comprobó que la mayoría de los principales problemas de optimización pertenecen a un superconjunto de P, la clase denominada NP, en la cual se incluyen aquellos problemas para los que no se conoce un algoritmo polinomial de resolución, tales como el TSP.

Estos problemas "difíciles" en NP, para los cuales todos los algoritmos conocidos requieren tiempo exponencial, tienen la peculiaridad de que todos los problemas en NP pueden ser "reducidos" polinomialmente a ellos, es decir, si se puede dar una solución en tiempo polinomial para uno de ellos, se podría dar para todos los de NP. Esta propiedad ha sido usada para definir una subclase separada en NP, la de los problemas NP-hard. Se dice que un problema es NP-hard si cualquier problema en NP es polinomialmente transformable en él, aunque el problema en sí no pertenezca a NP. Si el problema además pertenece a NP, entonces se lo denomina NP-completo. Lamentablemente, muchos de los problemas importantes que aparecen en Investigación Operativa son NP-completos.

## Enfoque Óptimo

### Formulación

Los métodos o estrategias óptimas para el TSP se basan en formulaciones de programación matemática. Por simplicidad vamos a asumir que los costos son simétricos (o sea  $c_{ij} = c_{ji}$ ) y vamos a definir a  $c_{ii} = 0$  para  $i=1,2,\dots,n$ . El problema consiste en determinar un tour que pase por todos los nodos comenzando y finalizando en el nodo origen, el cual vamos a definir como el nodo 1, de tal manera que se recorra la mínima distancia o se realice el menor costo. Vamos a definir entonces,

$$x_{ij} = \begin{cases} 1 & \text{si } \overline{arc\ ij} \text{ esta en el tour} \\ 0 & \text{si no esta en el tour} \end{cases}$$

Para formular el hecho de que solo debe haber un camino que entre y uno que salga por cada nodo, vamos a definir un requerimiento llamado formulación basada en la asignación, para ello vamos a tomar la matriz de decisión  $X = (x_{ij})$  de tal manera que solo un arco  $(i,j)$  salga del nodo  $i$ , y que exactamente un arco entre al nodo  $j$ . Esto implica una asignación para cada nodo con su respectivo nodo sucesor en el tour. Este requerimiento de asignación por si solo no determina que la matriz X corresponda a un tour, puesto que las asignaciones pueden resultar en subtours. Para eliminar la posibilidad de que se formen subtours, vamos a imponerle algunas restricciones más para la selección de arcos de la matriz X. Por lo tanto el problema planteado se

formula de la siguiente manera como: Minimizar  $\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$

$$\text{Sujeto a } \left\{ \begin{array}{l} \sum_{i=1}^n x_{ij} = b_j = 1 \quad (j = 1, \dots, n) \quad \{\text{Implica que a cada nodo solo llegue un arco}\} \\ \sum_{j=1}^n x_{ij} = a_i = 1 \quad (i = 1, \dots, n) \quad \{\text{Implica que de cada nodo solo salga un arco}\} \end{array} \right.$$

$$X = (x_{ij}) \in S$$
$$x_{ij} = 0 \text{ o } 1 \quad (i, j = 1, \dots, n)$$

El conjunto  $S$  puede contener cualquier restricción que prohíba que soluciones con subtours satisfagan las restricciones previamente definidas. Estas restricciones son llamadas restricciones destructoras de subtours.

Por ejemplo:

$$S = \left\{ (x_{ij}) : \sum_{i \in Q} \sum_{j \notin Q} x_{ij} \geq 1 \text{ para cada subconjunto } Q \text{ de } N \right\}$$

Esta restricción destructora de subtours dice que todo subconjunto  $Q$  de nodos debe estar conectado a los otros nodos del grafo en la solución  $X$ .

Como se dijo al comienzo de esta sección, no vamos a dar un método de resolución exacto del problema TSP. En función a la definición del problema descrita anteriormente y mediante la utilización de métodos y formulas matemáticas es que se llega a una solución óptima del problema. Puesto que los métodos exactos para resolver el TSP son, muy ineficientes en cuanto a tiempo de cómputo para problemas de gran porte, una variedad de heurísticas y metaheurísticas han aparecido para poder aproximar soluciones en este campo.

## Enfoque Heurístico

### Introducción a las Heurísticas

Las heurísticas son herramientas desarrolladas para ayudar en la resolución de muchos tipos de problemas. Aunque la capacidad científica de la humanidad parece haber alcanzado un nivel que permite dar respuesta prácticamente a cualquier reto que se le plantee, lamentablemente no es ésta la realidad. En efecto, si pensamos en los problemas reales que suelen ser de interés, para los cuales el objetivo es encontrar la solución que optimiza algún tipo de criterio (por ejemplo, menor costo bajo determinadas condiciones), la situación es justamente la opuesta: tan sólo una pequeña parte de ellos pueden ser resueltos fácilmente. En estas circunstancias, las heurísticas adquieren cada vez una mayor importancia. Las heurísticas que nosotros vamos a considerar caen en estas tres clases:

- Heurísticas Constructivas
- Heurísticas de Mejoramiento Iterativas
- Heurística Híbridas

A continuación vamos a dar una breve descripción de cada una de ellas.

### Heurísticas Constructivas

Una heurística constructiva es un algoritmo que determina un tour en función de alguna regla de construcción, con la característica de que una vez determinado el mismo no busca la manera de mejorarlo. Los tours que ya fueron creados permanecen sin cambios a lo largo de la ejecución del algoritmo. Estas heurísticas construyen una aproximación del tour óptimo haciendo uso de la matriz de distancias.

Vamos a presentar,

- Heurística del Vecino más Cercano
- Ahorro de Clark y Wright
- Heurística de Inserción
  - Inserción más Barata
  - Inserción más Cercana
  - Inserción Arbitraria
  - Inserción más Lejana
  - Inserción más Rápida
  - Inserción de la Cáscara Convexa

### **Heurística del Vecino más Cercano**

El vendedor comienza en una ciudad que va a ser su origen y destino. Desde ahí decide visitar la ciudad más cercana a ella. Una vez en esta continúa su recorrido visitando la ciudad más cerca a donde se encuentra en ese momento la cual aún no fue visitada. Este último paso se repite hasta que no queden ciudades sin visitar. Una vez llegado a esa instancia retorna a la ciudad origen.

Esta es una heurística intuitiva que suele ocurrírsele a cualquier persona. Una de las deficiencias de este método es que en el armado del tour muchas ciudades no son consideradas durante el curso del algoritmo, teniéndose que ser insertadas al final del mismo a costos muy altos.

Aunque este algoritmo no sea del todo bueno, el tour definido por la Heurística del Vecino más Cercano sirve como un buen tour de comienzo para el uso de los Métodos de Mejoramiento Iterativos. El método del vecino más cercano contiene unos pocos grandes errores, pero al mismo tiempo contiene gran cantidad de segmentos que conectan nodos con los caminos más cortos.

El algoritmo de la Heurística del Vecino más Cercano es de orden  $n^2$ , pero para obtener mejores resultados se repite el procedimiento  $n$  veces, en cada ocasión seleccionando un nuevo nodo como inicial. Esto conlleva a que esta nueva estrategia corra en un tiempo proporcional a  $n^3$ .

### **Ahorro de Clark y Wright**

Este método consiste en seleccionar aleatoriamente un nodo, el cual va ser considerado como el nodo depósito y se le llamara nodo 1. Luego se realiza el siguiente calculo para cada pareja de nodos:

$$s_{ij} = c_{1i} + c_{1j} - c_{ij} \text{ para } i, j=2,3,\dots,n$$

y los almaceno de mayor a menor en una estructura de datos llamada lista de ahorros. Comenzando en el tope de la lista de ahorros y recorriendo hacia abajo, voy formando subtours de máxima distancia mediante la unión de nodos  $i$  y  $j$ . Repito este proceso hasta que se forme un tour.

El calculo de la matriz  $S=[s_{ij}]$  requiere  $cn^2$  operaciones para alguna constante  $c$ . En el momento de realizar el ordenamiento de la lista de ahorros, suponiendo el uso del algoritmo de "heapsort" se computara en orden  $cn^2 \lg(n)$ . El último paso requiere a lo sumo  $n^2$  operaciones puesto que son la cantidad máxima de elementos en la lista de

ahorros. En conclusión el método heurístico de Ahorros de Clark y Wright requiere un orden de  $n^2 \lg(n)$  computaciones.

Igual que en el método anterior, en el caso de que se quisiera buscar una mejor solución se podría considerar ejecutar  $n$  veces el procedimiento descrito, y por cada vez seleccionar un nodo diferente como depósito. De esta forma el orden de ejecución queda de  $n^3 \lg(n)$  computaciones.

### Heurísticas de Inserción

Un método intuitivo para resolver el problema del TSP es comenzar con la construcción de un subtour, por ejemplo un tour en un conjunto reducido de nodos, y luego extenderlo insertando los nodos restantes uno tras otro hasta que todos los nodos hallan sido insertados. Estos procedimientos de inserción, toman un subtour de  $k$  nodos en la iteración  $k$  y tratan de determinar cual de los próximos nodos que no pertenece al recorrido debe unirse a este (paso selectivo), y en donde (paso de inserción). Hay una gran variedad de posibilidades para implementar este diseño de inserción. En general se pueden clasificar en base a estas tres características:

- Como construir el tour inicial
- Que criterio utilizar para seleccionar el próximo nodo a ser insertado
- Donde insertar ese nodo

El tour por el cual se comienza comprende por lo general a aquellos nodos que forman el triángulo más grande. Para los problemas Euclidianos, un buen tour para comenzar es aquel que determina la Cáscara Convexa de todos los nodos. Esta es una elección razonable, puesto que la secuencia que siguen los nodos en la cáscara convexa es la misma que la de un tour óptimo.

Un nuevo nodo es insertado en el tour de tal forma que cause el mínimo incremento en el largo total del mismo. La mayor diferencia entre los distintos diseños de inserción se da en el orden en el cual los nodos son insertados. A continuación vamos a pasar a ver alguno de ellos.

### Inserción más Cercana

Comenzamos con un subtour que solo contiene al nodo  $i$ . Hallamos el nodo  $k$  tal que  $c_{ik}$  es el mínimo para todos los demás nodos, y formamos el subtour  $i-k-i$ . Ahora vamos a ver lo que sería el paso selectivo; dado un subtour buscamos el nodo  $k$  que no se encuentra en el mismo, tal que la distancia a cualquier miembro del subtour sea mínima.

El paso de inserción correspondería a hallar el arco  $(i, j)$  en el subtour el cual minimice la ecuación  $c_{ik} + c_{kj} - c_{ij}$ , e inserto  $k$  entre  $i$  y  $j$ . Esto lo realizo hasta encontrar un ciclo Hamiltoniano.

Este algoritmo requiere un tiempo de ejecución del orden de  $n^2$  computaciones.  
Inserción más Barata

El procedimiento para este algoritmo es similar al de la inserción más cercana excepto por los pasos de selección y de inserción. El paso selectivo consiste en hallar el par de nodos  $(i, j)$  dentro del subtour y un nodo  $k$  que no pertenece al mismo de tal forma que la ecuación  $c_{ik} + c_{kj} - c_{ij}$ , sea mínima. Una vez hallado el trío de nodos insertamos al nodo  $k$  entre los nodos  $i$  y  $j$ . La ecuación nos dice que hallemos entre todos los nodos que aún no han sido insertados, aquel cuya inserción cause el mínimo incremento en el largo total del tour.

El número de computaciones de este método es del orden de  $n^2 \log(n)$ .

### **Inserción Arbitraria**

Este método es bastante simple, se procede igual que en la inserción más cercana, con la diferencia que en el paso selectivo se selecciona a un nodo aleatoriamente. La inserción arbitraria tiene un orden de  $n^2$  computaciones.

### **Inserción más Lejana**

Este método es prácticamente igual al de Inserción más Cercana. En el paso selectivo en lugar de seleccionar al más cercano, se elige a aquel nodo cuya distancia a un nodo del subtour sea máxima. La selección del nodo que formara parte del primer subtour se efectúa seleccionando a aquel que se encuentre más lejos del origen.

### **Inserción más Rápida**

El procedimiento para esta heurística consiste en primero seleccionar cualquier nodo al azar y formar el subtour  $T$ , formado por un nodo y cero arcos. El siguiente paso dice que dado el circuito  $T_k$ , encontrar el nodo  $z_k$  que no está en  $T_k$ , que se encuentra más cerca del nodo al cual llamaremos  $y_k$  de  $T_k$ . Sea  $T_{k+1}$ , el circuito obtenido luego de haber insertado el nodo  $z_k$  inmediatamente después de  $y_k$  en  $T_k$ . El último paso consiste en repetir el segundo y tercer paso hasta encontrar un ciclo Hamiltoniano. La cantidad de computaciones es igual al método de la inserción más cercana, o sea  $n^2$  computaciones.

### **Inserción De la Cáscara Convexa**

La cáscara convexa se le llama al menor conjunto convexo que contiene a todos los puntos. Está probado [RSV] que si los costos  $c_{ij}$  representan distancias Euclidianas, y que si  $H$  es la cáscara convexa del conjunto de nodos, en un espacio bidimensional, el orden en el cual aparecen los nodos pertenecientes a los bordes del tour óptimo, es el mismo en el que aparecen en  $H$ . Esta apreciación sirve como impulso para el desarrollo de la Heurística de la Cáscara Convexa.

El procedimiento a seguir es el siguiente, primero construyo la cáscara convexa del conjunto de nodos con el cual voy a trabajar. Este ciclo lo vamos a utilizar como subtour inicial. En el siguiente paso vamos a iterar sobre todos los nodos que aun no se encuentran en el recorrido, por lo tanto, para cada nodo  $k$  decido entre cual pareja de nodos  $i$  y  $j$  pertenecientes al subtour debo insertarlo. Esto se realiza siguiendo el siguiente criterio de inserción, el que minimiza la ecuación  $c_{ik} + c_{kj} - c_{ij}$ , para los nodos  $i$ ,  $j$ , y  $k$ . En un próximo paso determino la terna  $(i^*, k^*, j^*)$  tal que es mínimo entre todas las ternas  $(i, k, j)$  encontradas en el paso anterior. Luego inserto el nodo  $k^*$  entre los nodos del subtour  $i^*$  y  $j^*$ . Voy a repetir todos estos pasos (a excepción del primero) hasta encontrar un ciclo Hamiltoniano.

### **Heurísticas de Mejoramiento Iterativo**

Los tours que determinan varias de las heurísticas constructivas mencionadas en la sección anterior tienen la característica general de ser de una calidad moderada. En esta sección vamos a tratar el hecho de cómo mejorar esos tours. Comúnmente, las heurísticas de mejoramiento iterativo se caracterizan por realizar cierto tipo básico de movimiento que altera el tour formado hasta el momento.

Estas heurísticas trabajan de la siguiente forma,

- Encontrar un tour inicial. Este puede ser seleccionado de manera aleatoria, o definido por una heurística de tipo constructiva.
- Mejorar el tour seleccionado utilizando una heurística de intercambio.
- Realizar el paso 2 hasta que no sea posible realizar más mejoras sobre el tour.

### **Intercambio 2-opt**

Para el TSP se considerará como heurística básica la siguiente: dado un tour  $T$ , se tratará de mejorarlo eliminando  $k$  arcos y reemplazándolos por un nuevo conjunto de  $k$  arcos que reconectan el tour. Estos borrados y reemplazos serán llamados  $k$ -Changes. Se dice que un tour es  $k$ -óptimo cuando no puede ser mejorado por un  $k$ -Change; es decir, es un mínimo local bajo la movida del  $k$ -Change. Para esta operación de intercambio, dos configuraciones son vecinas si existe un  $k$ -Change que transforma a una de ellas en la otra. Es fácil ver que existe una relación entre el tamaño del vecindario y el valor de  $k$ . En general, un paso del  $k$ -Opt mejora un tour borrando  $k$  arcos existentes, e insertando  $k$  nuevos arcos. El algoritmo  $k$ -Opt obvio requiere un tiempo de  $O(Nk)$  para chequear por un único intercambio. Cuando se usa un mayor  $k$ , el mínimo local es mejor, pero el costo computacional es mayor. Las heurísticas más usadas de este tipo son las que producen tours 2-óptimos y 3-óptimos.

Una estrategia o movimiento del tipo 2-opt consiste en eliminar dos de los bordes que determinan el tour para luego reconectarlos de diferente forma, por ejemplo combinándolos entre ellos, de tal manera de obtener un nuevo tour. Solo hay una manera de reconectar los caminos que lleva a la formación de un tour diferente. Entre todo par de aristas cuyo intercambio 2-opt hace que se decremente el largo total del tour elegimos el par que nos da como resultado el mínimo del mismo.

Este procedimiento se efectúa de forma iterativa hasta que no queden más pares de aristas por intercambiar. El tour resultante es llamado 2-optimal.

En el 3-Opt, el intercambio reemplaza hasta 3 arcos del tour actual. Aplicado a tours de distribución uniforme, el algoritmo 3-Opt produce tours sustancialmente más cortos que el 2-Opt, al costo de un significativo aumento en el tiempo de ejecución.

### **Heurísticas Compuestas o Híbridas**

Muchas heurísticas para el TSP son generalmente compuestas por dos fases, una heurística constructiva seguida de una iteración mejorativa hasta que un mínimo local es alcanzado. A este tipo de heurísticas es que las llamamos híbridas. La razón es que luego de que una ciudad ha sido introducida, un período de mejoramiento local es llevado a cabo para llegar a un mínimo local antes de que se vuelva a insertar una nueva ciudad.

El procedimiento básico a realizarse en las heurísticas compuestas se puede definir como sigue, primero obtener un tour inicial utilizando un método constructivo. Luego aplico un procedimiento 2-opt al tour determinado en el paso anterior, y por último aplico un procedimiento pero esta vez de tipo 3-opt al tour que formó el paso anterior. Estos métodos son relativamente rápidos de computar y dan excelentes resultados. La idea detrás de los procedimientos compuestos es hallar una buena solución inicial rápido y confiar en que los métodos 2-opt y 3-opt determinen una solución casi óptima.



A continuación presentamos otro método compuesto desarrollado por Natalio Krasnor, Pablo Moscazo y Michael G. Norman, llamado "Una nueva heurística híbrida para el TSP de gran porte basado en la Triangulación de Delaunay" [TSPDT].

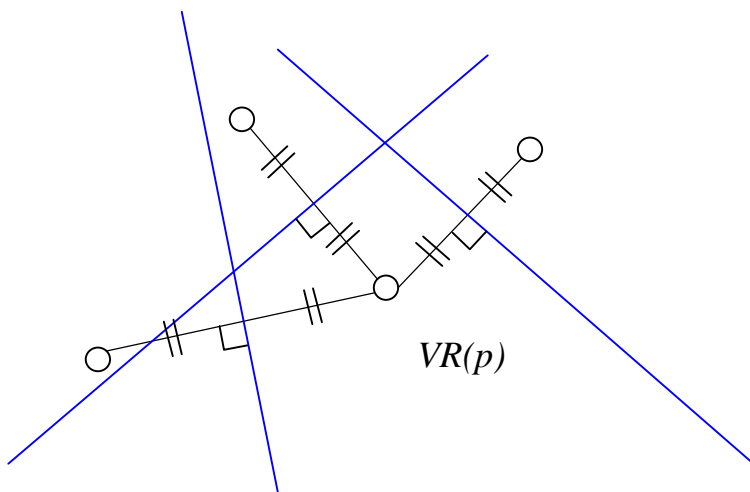
### TSP Basado en la Triangulación Delaunay

Vamos a ver una nueva heurística híbrida para los problemas geométricos del Vendedor Viajero de gran porte. El método se basa en el uso de la Triangulación de Delaunay para un conjunto de ciudades.

### Diagrama Voronoi y la Triangulación Delaunay

El diagrama Voronoi resuelve el siguiente problema, "dado un conjunto  $S$  de  $N$  puntos en el plano, para cada punto  $p$  en  $S$ , ¿Cuál es el lugar geométrico de los puntos  $(x, y)$  en el plano tal que están más cerca de  $p$  que de cualquier otro punto en  $S$ ?".

Dado dos puntos  $p$  y  $q$ , el conjunto de puntos que están más cerca de  $p$  que de  $q$  es el semiplano que contiene a  $p$  definido por la mediatriz a  $\overline{pq}$ . Vamos a denominar a este semiplano como  $H(p,q)$ . El polígono Voronoi asociado a un punto  $p$  (denotado como  $VR(p)$ ) es la región convexa del polígono que no tiene más de  $N-1$  lados, puesto que es la intersección de  $N-1$  semiplanos mediatrices definidos por el punto  $p$  y los otros  $N-1$  puntos en el conjunto  $S$ .



Por lo tanto,

$$V_R(p) = \bigcap_{p,q \in S, p \neq q} H(p,q)$$

Dado el diagrama de Voronoi para el conjunto  $S$  existe un grafo que es dual al mismo. Este se llama Triangulación de Delaunay ( $D_T$ ), y se denota  $(S, D_T)$ . Es un grafo no dirigido que tiene a  $S$  como el conjunto de vértices y a  $D_T$  como conjunto de aristas, tal que  $D_T = \{\{p, q\} / V_R(p) \cap V_R(q) \neq \emptyset\}$ . Otra manera de definir la Triangulación Delaunay sobre un conjunto de puntos del plano es decir que, consiste en todos los segmentos tal que existe un círculo que pasa por  $p$  y  $q$  que no contiene ningún otro

punto del conjunto  $S$ . De manera equivalente vamos a decir que  $\overline{pq}$  es una arista *Delaunay* si y solo si las regiones *Voronoi* para  $p$  y  $q$  son adyacentes. La región *Voronoi* de un vértice  $p$  es el polígono con la propiedad de que todos sus puntos interiores están más cerca de  $p$  que de cualquier otro punto. En este sentido, las aristas *Delaunay* comprenden la noción de "vecindario". Hemos asumido de forma implícita que no existen dos vértices iguales, que no hay tres vértices colineales, y que no existen cuatro vértices que puedan pertenecer a un mismo círculo. Si esas condiciones no se satisfacen, el procedimiento estándar que se aplicara perturbará ligeramente las posiciones de esos puntos de tal forma que las condiciones se cumplan. El grafo *Delaunay*, el cual denotamos como  $(S, D_G)$ , se crea a partir de  $(S, D_T)$  excluyendo las aristas  $\{p, q\} \in D_G$  para los cuales  $|V_R(p) \cap V_R(q)| = 1$  (esto son los polígonos *Voronoi* de los dos vértices que se interseccionan en un solo punto). En contraste a  $(S, D_T)$ ,  $(S, D_G)$  no es más una triangulación pero se garantiza que  $|D_G| = O(N)$ .

### Inserción más Lejana a partir de la Mínima Cáscara Convexa

FAMCH es una técnica que comienza por el tour (posiblemente incompleto) que comprende la mínima cáscara convexa (esta es la cáscara convexa de la cual se la han extraído todos los vértices tales que sus aristas adyacentes forman un ángulo de 180 grados). Iterativamente elegimos una ciudad que no pertenece al tour de tal manera que sea la que se encuentra a mayor distancia de su vecino más cercano perteneciente al tour, rompiendo enlaces de manera aleatoria, e insertándolo entre dos ciudades consecutivas en el subtour de tal manera que dicha inserción provoca el mínimo incremento en el largo del tour total.

La pregunta que se hace acerca de cual es el orden apropiado de inserción es la que nos conduce a comenzar a considerar diferentes variantes del procedimiento de selección descrito arriba: "Iterativamente elegimos una ciudad que no pertenece al tour de tal manera que sea la que se encuentra a mayor distancia de su vecino más cercano perteneciente al tour, rompiendo enlaces de manera aleatoria". El DT era un candidato adecuado para una determinada definición de vecindario de un punto que pertenece un conjunto de puntos. Como ya habíamos visto, resultaba mejor insertar primero aquellas ciudades que tenían menor ángulo en DT. Pensamos que esto era razonable y pronto establecimos este procedimiento como el primer criterio de selección para la inserción.

Hay un punto interesante a notar en esta estrategia. Se sabe que podemos asociar una superficie poliedra a toda triangulación, en particular la  $D_T$ , solicitando que todas las uniones sean de largo unitario. Es claro que esa superficie, compuesta por triángulos equiláteros, no puede ser "flat" puesto que el único caso en el cual es "flat" es el caso en que una ciudad tiene 6 vecinos en el  $D_T$ . Cinco vecinos tendrán una curvatura positiva, mientras que siete formaran una "silla de montar" con curvatura negativa. Dicho esto, tenemos el ángulo de una ciudad en el  $D_T$  como primer criterio de selección para clasificar las ciudades ha ser insertadas y puede ser reinterpretado o mejor aun "refinado", en futuros estudios, con el concepto de curvatura así como otros.

Vamos a necesitar otro criterio para sacar la aleatoriedad a la selección, puesto que hay muchas ciudades con el mismo número de vecinos en  $D_T$ . Por el momento vamos a llamar a este criterio como  $H\#$  y vamos a hacer explícita la estructura básica de OCIDT.

### Basic OCIDT

1. Crear la Triangulación de Delaunay para el conjunto de puntos  $S$ .
2. Repito hasta que todos los puntos hallan sido insertados
  - a. Encontrar la ciudad que aun no esta en el tour que tiene el ángulo mas bajo en  $DT$ . Rompo los lazos relacionados a  $H\#$ .
  - b. Insertar la ciudad seleccionada entre dos ciudades consecutivas del subtour para las cuales dicha inserción causa el mínimo incremento en el largo total del tour, rompiendo lazos aleatoriamente.
  - c. Comenzar una iteración mejorativa por la que en cada paso se inserte una ciudad.

Observamos el hecho de que el paso iterativo mejorativo es realizado cada vez que una ciudad ha sido insertada, de esta manera se reduce drásticamente el chequeo en la inserción. Es interesante ver como la heurística evoluciona, a veces una "cascada de eventos" cambia dramáticamente la forma total de un tour parcial. La heurística tiene un bajo costo de complejidad puesto que  $D_T$  puede ser calculado en  $O(N \log N)$ .

Hemos elegido utilizar la inserción de una ciudad por vez (osea, chequear por la inserción de una ciudad entre otras dos) debido a su simplicidad y su bajo grado de complejidad. Otro tipo de movimientos, como son el 2-Opt, podrían también ser aplicados pero no son parte de este estudio.

### Enfoque Meta Heurístico

Las heurísticas pueden ser simples o complejas. Los algoritmos simples tienden a tener reglas de terminación bien definidas, y se detienen en un óptimo local. Los algoritmos más complejos pueden no tener reglas de terminación estándar, y típicamente buscan soluciones mejores hasta alcanzar un punto de parada arbitrario. Las metaheurísticas son ejemplos de algoritmos más complejos. Estos son alguno de ellos.

- Búsqueda Tabú
- Simulated Annealing
- Algoritmos Genéticos
- Redes Neuronales
- Sistema de la Colonia de Hormigas

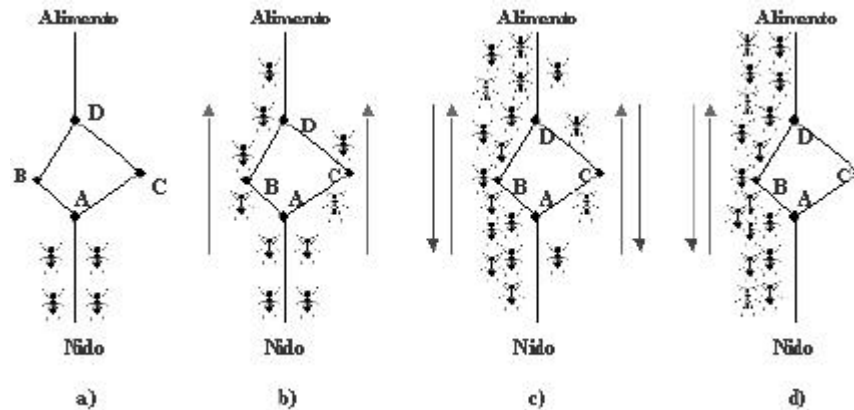
En esta documentación vamos a ver solo el caso del Sistema de la Colonia de Hormigas. Esta información fue tomada del documento "Algoritmo ACO aplicado al TSP: Resumen de una experiencia práctica" [TSPACO].

A continuación, se presentará la analogía de la Colonia de Hormigas con el problema del TSP, y se culminara con la presentación de las Hormigas Artificiales.

### La analogía de la colonia de hormigas

La Metaheurística se basa en el comportamiento de una colonia de Hormigas. Las hormigas reales son capaces de encontrar el camino más corto entre una fuente de comida y su nido, por ejemplo sin usar mecanismos visuales, sino sólo explotando el rastro de Feromona.

Una forma en que las hormigas explotan la feromona para encontrar el camino más corto entre dos puntos se muestra en la figura.



En a) las hormigas llegan a un punto en el cual deben decidirse por uno de los dos caminos a seguir; en b) se realiza una elección de camino, la que puede ser aleatoria al haber bajos niveles de feromona o guiada al haber una diferencia notable entre la cantidad de feromona de cada camino; en c), dado que la velocidad de una hormiga se considera aproximadamente constante, es claro decir que las hormigas que eligieron el camino más corto se demoran menos que las otras en llegar hasta el otro extremo, lo cual conduce a una mayor acumulación de feromona en el camino. En d), la feromona acumulada en mayor cantidad en el camino más corto y más circulado guía a las hormigas a la fuente de alimento de la forma más rápida, donde el camino menos transitado pierde la feromona depositada en él a causa de la evaporación. Informalmente, esta metaheurística trabaja de la siguiente manera:

“Cada hormiga genera un tour completo escogiendo ciudades de acuerdo a una regla de transición de estados probabilística. Las hormigas prefieren moverse a ciudades que están conectadas a caminos de bajo costo con una alta cantidad de feromona. Una vez que todas las hormigas han completado su tour, se aplica una regla de actualización de la feromona global. A continuación se evapora una fracción de la feromona, mientras que por otra parte algunas hormigas depositan una cantidad proporcional de ella sobre los caminos que formaron parte de su tour (de acuerdo al costo total de cada uno), lo cual provoca un incremento a través del tiempo. Luego, el proceso comienza a iterar. A través de este mecanismo, la colonia genera una convergencia común, dirigida hacia el camino más corto.” La actualización de feromona es la coordinación entre los distintos tours realizados en forma concurrente por cada una de las hormigas.

### Las hormigas artificiales

De acuerdo a este comportamiento es posible definir el concepto de Hormiga Artificial. Un ejemplo de pseudo código de la implementación de la metaheurística ACO se presenta a continuación:

```

procedimiento ACO()
  establecer_feromona_inicial();
  mientras (el criterio de término no esté satisfecho)
    crear_las_hormigas();
    para (cada hormiga)
      {mover_hormiga();
      } hasta (completar tour);
  fin para
  actualizar_feromona();

```

```
    destruir_hormigas();  
fin mientras  
fin procedimiento  
  
procedimiento mover_hormiga()  
para (todo el vecindario factible)  
    calcular_probabilidades_de_movimiento();  
    fin para  
    ciudad_seleccionada:=seleccionar_el_movimiento();  
    llevar_hormiga_a_ciudad(ciudad_seleccionada);  
fin procedimiento
```

Para este estudio se utilizó una implementación del algoritmo Ant Colony System (ACS). En él cada una de las  $n$  hormigas construye una solución (tour) del TSP. Inicialmente, cada una de las hormigas es colocada en alguna ciudad (en forma aleatoria o ciclomática). Una vez hecho esto, a cada hormiga se le aplica una regla de elección probabilística de los caminos a seguir. En esta regla se utilizan parámetros del estilo la influencia relativa del camino de feromona y la distancia respectiva, y el vecindario factible para la hormiga  $k$ .

Este algoritmo corresponde a un algoritmo clásico voraz estocástico (con múltiples puntos de inicio desde que las hormigas están distribuidas en las ciudades). Existe el caso en el cual sólo trabajará la amplificación de feromona, este método podría dirigirnos a una rápida situación de estancamiento con la correspondiente generación de tours los cuales, en general, son fuertemente subóptimos. La búsqueda de estancamiento se define como la situación donde todas las hormigas siguen la misma ruta y construyen un óptimo local. Por lo tanto, debe existir una mezcla entre la influencia de la información heurística y los caminos de feromona. Después que todas las hormigas han construido su tour, los caminos de feromona son actualizados. Esto se hace primero disminuyendo el nivel de feromona en todos los arcos por un factor constante, para luego permitir a cada hormiga agregar feromona a los arcos que fueron visitados.

En este caso entra en juego el factor de evaporación de la feromona, el cual es utilizado para evitar acumulación ilimitada de feromona en los caminos y le permite al algoritmo "olvidar" las malas decisiones previamente tomadas. Si un arco no es elegido por las hormigas, su feromona asociada decrece exponencialmente

El mejor tour de una hormiga es el que más feromona recibió por arcos a través del tour. En general, los arcos que son usados por muchas hormigas y los cuales están contenidos en tours cortos podrían recibir más feromona y por consiguiente son más probables a ser seleccionados en futuras iteraciones del algoritmo.

## El Problema del Vendedor Viajero Múltiple

### Definición

El MTSP es una generalización del TSP que se aproxima más a modelar los problemas de ruteo que ocurren en la vida real, donde hay necesidad de hacer uso de más de un vendedor (o vehículo).

En este problema,  $M$  vendedores van a visitar los  $n$  nodos de un grafo previamente definido. Lo van a hacer de una forma tal que la distancia recorrida por los  $M$  vendedores sea mínima. Cada vendedor va a viajar a través de un subtour de nodos, el cual incluye un depósito común, y cada nodo excepto el depósito debe de ser visitado exactamente una sola una vez y por un solo vendedor.

### Formulación

La formulación matemática para la programación del MTSP es una extensión que deriva de la formulación basada en asignación para el problema del vendedor viajero. [RSV]

$$\text{Minimizar } \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

Sujeto a:  $\left\{ \begin{array}{l} \sum_{i=1}^n x_{ij} = b_j = \begin{cases} M & \text{if } j=1 \\ 1 & \text{if } j=2,3,\dots,n \end{cases} \\ \sum_{i=1}^n x_{ij} = a_i = \begin{cases} M & \text{if } i=1 \\ 1 & \text{if } i=2,3,\dots,n \end{cases} \\ X = (x_{ij}) \in S \\ x_{ij} = 0 \text{ or } 1 \text{ (} i,j=1,\dots,n \text{)} \end{array} \right.$

para cualquier elección de  $S$  que prohíba la formación de soluciones con subtours que no incluyan al nodo origen (nodo 1). En particular se puede utilizar aquella mencionada en la formulación del TSP. Vale la pena destacar que se requiere que los  $M$  vendedores sean usados.

Cualquier problema del tipo MTSP puede ser convertido a un TSP equivalente. Las formulaciones equivalentes del TSP sobre el MTSP se obtienen mediante la creación de  $M$  copias del depósito cada una conectada a los otros nodos tal como si fuera el depósito original. De todas formas no vamos a profundizar en este problema.

## Problema de Ruteo de Vehículos

### Definición

Además de su relevancia práctica, el problema de ruteo de vehículos es un área de gran importancia para la Investigación Operativa. La siguiente definición del problema de ruteo de vehículos fue tomada de las notas introductorias de J.E. Beasley tituladas "OR-Notes" [JEBOR].

Considere la situación que presenta la figura de abajo, tenemos un depósito rodeado por un número finito de clientes que deben ser abastecidos por el depósito indicado en la figura.

El administrador del depósito se encuentra con la tarea de diseñar y asignar rutas para realizar sus entregas utilizando una flota de vehículos los cuales dispone. Este problema de diseño de rutas se conoce como "Problema de ruteo de vehículos".

Consecuentemente el problema de ruteo de vehículos puede ser definido como el problema de asignar rutas a vehículos de reparto (de capacidades conocidas) los cuales operan a partir de un único depósito para proveer a un conjunto de clientes cuya localización y demanda son conocidas. Las rutas para los vehículos son diseñadas para minimizar ciertos objetivos tales como la distancia total recorrida.

Este problema ha atraído mucha atención a nivel de literatura académica básicamente por dos razones.

- El problema aparece en una gran cantidad de casos prácticos.
- El problema es teóricamente interesante, y no siempre es fácil su solución.

### **El problema práctico**

El problema de ruteo de vehículos en la práctica involucra muchas restricciones acerca de la ruta que los vehículos de entrega pueden seguir, a continuación vamos a considerar algunas de las más comunes que se pueden ver. Podemos clasificar a estas restricciones según su relación con los vehículos o con los clientes. Nótese que en cualquier caso particular no todas las restricciones se pueden aplicar, de todas maneras pensando con una concepción global acerca del problema es útil listar todas las restricciones que potencialmente se pueden aplicar.

### **Vehículos**

- Cada vehículo tiene un límite (capacidad, generalmente en peso o volumen) en cuanto a la mercadería que transporta, por ejemplo un camión cisterna que traslada gasolina a una estación de servicio tiene límite de volumen, los autobuses tienen límite en cuanto a la cantidad de personas que pueden legalmente llevar en su interior, etc.
- Cada vehículo tiene un tiempo total de trabajo desde el momento que parte hasta el que arriba al depósito, este tiempo por lo general corresponde al determinado por las horas de trabajo del conductor del vehículo.
- Cada vehículo tiene un período de tiempo en el cual debe abandonar el depósito, esto se debe para asegurarse de que haya espacio disponible para la llegada de otros vehículos que vengan a reabastecerse al depósito.
- Cada vehículo tiene una cantidad determinada de períodos de tiempo para no hacer nada (tiempo en el cual el conductor descansa).
- Cada vehículo tiene un costo asociado a su uso.

### **Clientes**

- Cada cliente tiene una cierta cantidad (demanda de mercadería) la cual se le debe ser entregada, comúnmente se habla de operaciones de entrega absoluta
- Cada cliente tiene una cantidad definida de períodos de tiempo en los cuales acepta la llegada de mercaderías. Esos períodos de tiempo son llamados ventanas de tiempo. Por ejemplo un cliente podría solo recibir la mercadería entre las horas 10:30 y 11:30 o entre la 14:00 y las 16:15. Estos dos períodos de tiempo son las ventanas de tiempo para ese cliente. Las ventanas de tiempo

son de ayuda para los clientes, dado que les sirven para poder organizar otras tareas para que se ajusten a ese patrón de trabajo definido. Así como decimos que las ventanas de tiempo son convenientes para los clientes, no así lo son para las compañías que deben entregar las mercaderías, debido a que esto les quita flexibilidad.

- Cada cliente tiene asociado un tiempo en el cual se procede a “descargar” la mercadería (drop time).
- No todos los vehículos pueden ser utilizados para entregar mercadería a los clientes. Cada cliente tiene un conjunto de vehículos que pueden ser utilizados para recibir sus entregas (restricciones de acceso).
- Los clientes tienen prioridad para sus entregas (esto se utiliza en el caso de que los vehículos no puedan cumplir con las entregas de todos los clientes asignados). Esto ocurre debido a indisponibilidad tanto del conductor como del vehículo que debía de hacer la entrega, o por condiciones meteorológicas que reduzcan dramáticamente la velocidad de los vehículos.
- Los clientes pueden recibir visitas compartidas (una entrega que requiera más de un vehículo para su realización).

### Otros Factores

- Viajes múltiples realizados por el mismo vehículo en un mismo día, en los cuales los vehículos deben retornar al depósito para luego partir nuevamente. (por ejemplo las camionetas de entrega de cartas del correo)
- Viajes efectuados por un vehículo que requieren más de un día (por ejemplo con paradas nocturnas).
- Vehículos compartimentados con diferentes tipos de productos a ser entregados. Los vehículos de transporte de gasolina suelen estar compartimentados para poder llevar gasolina con y sin plomo, así como los vehículos de transporte de comida (comida congelada y no congelada).
- Más de un depósito, donde los vehículos pueden comenzar, visitar o terminar su trayecto.

### Objetivos

En el diseño de rutas para vehículos que cumplan las restricciones mencionadas existen un gran número de objetivos que pueden ser adoptados. Entre ellos se pueden distinguir tres de los más básicos los cuales veremos a continuación.

- Minimizar el número de vehículos a utilizar (los vehículos y sus respectivos conductores son costos fijos).
- Minimizar la distancia total (o tiempo) a recorrer (esto corresponde al uso de variables de costo).
- Minimizar ciertas condiciones de (1) y (2), por ejemplo minimizar el costo total de las variables de costo y los costos fijos.  
El costo de los vehículos de la flota es a menudo considerado como un costo fijo, por lo tanto el primer objetivo corresponde a minimizar los costos fijos; el segundo objetivo corresponderá a minimizar las variables de costo; y el tercer objetivo será minimizar el costo total.

### Formulación

El problema de ruteo de vehículos consiste en determinar un conjunto de rutas para realizar pedidos a varios nodos, las cuales se originan en un depósito central, buscando minimizar la distancia total recorrida por la totalidad de la flota. Los vehículos constan de capacidad de carga y pudieran tener restricciones de tiempo, entre otras. Todo



vehículo comienza y culmina su recorrido en el depósito central. Nos vamos a referir a la siguiente formulación de este problema como el Problema de Ruteo de Vehículos Genérico (VRP). Formalmente queda formulado de la siguiente manera:

Minimizar  $\sum_{i=1}^n \sum_{j=1}^n \sum_{v=1}^{NV} c_{ij} x_{ij}^v$  siendo  $NV$  la cantidad de vehículos. Se puede ver como

minimizar la suma de las rutas de cada uno de los vehículos, por lo tanto consideramos el mismo  $x$  y  $c$  de la anterior definición del TSP. El superíndice  $v$  de  $x$  significa que es referente al camino por donde transitó el vehículo  $V$ .

Sujeto a:  $\left\{ \begin{array}{l} \sum_{i=1}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (j = 2, \dots, n) \quad \text{\{cada nodo solo llegue un arco\}} \\ \sum_{j=1}^n \sum_{v=1}^{NV} x_{ij}^v = 1 \quad (i = 2, \dots, n) \quad \text{\{cada nodo solo llegue un arco\}} \\ \sum_{i=1}^n x_{ip}^v - \sum_{j=1}^n x_{pj}^v = 0 \quad (v = 1, \dots, NV; p = 1, \dots, n) \\ \text{\{continuidad de las rutas, si un vehículo entra a un nodo, debe salir de él\}} \\ \sum_{i=1}^n d_i \left( \sum_{j=1}^n x_{ij}^v \right) \leq K_v \quad (v = 1, \dots, NV) \\ \text{\{suma de los requerimientos de los nodos visitados por un vehículo no sobrepase la capacidad del vehículo\}} \\ \sum_{i=1}^n t_i^v \sum_{j=1}^n x_{ij}^v + \sum_{i=1}^n \sum_{j=1}^n t_{ij}^v x_{ij}^v \leq T_v \quad (v = 1, \dots, NV) \\ \text{\{implica que se cumplan las restricciones de tiempo impuestas por los clientes\}} \\ \sum_{j=2}^n x_{1j}^v \leq 1 \quad (v = 1, \dots, NV) \\ \text{\{garantiza que cada vehículo salga 0 o 1 vez del depósito\}} \\ \sum_{i=2}^n x_{i1}^v \leq 1 \quad (v = 1, \dots, NV) \\ \text{\{garantiza que cada vehículo llegue 0 o 1 vez del depósito\}} \\ X = (x_{ij}) \in \mathcal{S} \\ x_{ij} = 0 \text{ o } 1 \quad (i, j = 1, \dots, n) \end{array} \right.$

Se puede encontrar un desarrollo mas profundo de esta formulación matemática en el Apéndice "Estado del Arte" y en [RSV], páginas 95 a 97.

En donde  $n$  es el número de nodos,  $NV$  es la cantidad de vehículos,  $K_v$  es la capacidad del vehículo  $v$ ,  $T_v$  es el tiempo máximo que tiene el vehículo  $v$  permitido para andar en la ruta,  $d_i$  es la demanda del cliente que se encuentra en el nodo  $i$  ( $d_1=0$ ),  $t_1^v$  es el tiempo que le lleva al vehículo  $v$  entregar o recoger su pedido en el nodo  $i$  ( $t_1^v = 0$ ),  $t_{ij}^v$  es el tiempo que le lleva viajar al vehículo  $v$  desde el nodo  $i$  al nodo  $j$  ( $t_{ii}^v = \infty$ ),  $c_{ij}$  es el costo que lleva trasladarse del nodo  $i$  al nodo  $j$ ,  $x_{ij}^v = 1$  si el arco  $(i, j)$  es atravesado por

el vehículo  $v$ , de lo contrario su valor es de cero,  $X$  es la matriz que especifica las conexiones sin importar el vehículo, y sus componentes son  $x_{ij} \equiv \sum_{v=1}^{NV} x_{ij}^v$ .

La distancia es la que se quiere minimizar, alternatively se podría minimizar costos reemplazando  $c_{ij}$  por el coeficiente de costo  $c_{ij}^v$  el cual depende del tipo del vehículo. La demanda en cada nodo es atendida por exactamente un solo vehículo. También queda determinada la continuidad de la ruta, por ejemplo si un vehículo entra a un nodo con demanda, debe salir de ese nodo. También se indican las restricciones de capacidad de los vehículos, y del tiempo total de ruta transcurrido. Además se debe asegurar que la disponibilidad del vehículo no sea excedida. Por último no se permiten las formaciones de subtours. Por otro lado asumimos que la demanda de cada nodo no excede la capacidad de los vehículos. En este modelo también se asume que cuando un nodo con demanda es atendido, sus requerimientos son satisfechos completamente., esto quiere decir que no existen servicios parciales.

### Extensiones del VRP

Existe un número de extensiones al problema básico del VRP que fueron consideradas por varios investigadores. Estos apuntan a crear un modelo más real del problema imponiendo nuevas restricciones, entre estas se puede encontrar restricciones en cuanto al tamaño de la flota, decisiones mixtas, permitir diferentes formas de rutas entre otras. Una de las más conocidas es la investigación de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW), esta especifica que cada cliente debe ser atendido dentro de un periodo de tiempo específico llamado ventana de tiempo. Aquí vamos a enunciar los siguientes problemas:

- El Ruteo de Vehículos con Ventanas de Tiempo (VRPTW)
- El Ruteo de Vehículos con Entregas Particionadas (SDVRP)
- El Ruteo de Vehículos con Múltiples Depósitos (MDVRP)

### Ruteo de Vehículos con ventanas de Tiempo

Este documento fue obtenido del informe de "Análisis Probabilístico y Algoritmos Prácticos para el problema de Ruteo de Vehículos con ventanas de Tiempo", realizado por Julien Bramel y David Simchi-Levi en Setiembre de 1992 [VRPTW].

En el problema de ruteo de vehículos con ventas de tiempo, un conjunto de clientes debe ser abastecido por una flota de vehículos con capacidad limitada, que inicialmente se encuentran en el depósito. Cada cliente tiene asociado un período de tiempo en el cual requieren el servicio, que puede consistir en trabajos de reparación tanto como carga o descarga del vehículo. El objetivo es encontrar tours para los vehículos, de tal forma que todos los clientes sean abastecidos dentro de sus ventanas de tiempo. La carga total de cada vehículo no es mayor a la capacidad de carga del mismo, y el total de distancia recorrida por cada vehículo debe de ser el menor posible.

En este informe se presenta una caracterización de la solución óptima asintótica para distribuciones generales en el tiempo de duración del servicio, ventanas de tiempo, carga y ubicación de los clientes. Esta caracterización lleva al desarrollo de un nuevo algoritmo basado en formular el problema como un problema de ubicación estilizado. Los resultados computados muestran que el algoritmo es muy eficiente en un conjunto de problemas estándares de prueba.

El problema de Ruteo de Vehículos con Ventanas de Tiempo (VRPTW) puede definirse de la siguiente forma:

Un conjunto de clientes dispersados en una región geográfica deben de ser abastecidos por una flota de vehículos inicialmente posicionados en un depósito determinado. Cada cliente tiene una carga que debe ser recogida y el cliente a su vez especifica un período de tiempo, llamado ventana de tiempo, en el cual este proceso de recogida debe ser efectuado. El cliente es atendido por vehículos de capacidad limitada, esto quiere decir que la carga total del vehículo no puede superar la capacidad del vehículo. El objetivo es encontrar un conjunto de rutas para cada vehículo, donde cada ruta comience y termine en el depósito, de servicio a un subconjunto de clientes sin sobrepasar la capacidad de carga del vehículo y dentro de la ventana de tiempo de cada cliente. Todas estas restricciones deben cumplirse mientras se minimiza la longitud total de las rutas. Dado a la amplia aplicabilidad y la importancia económica de este problema en el área de servicios industriales, variantes del mismo han sido extensamente estudiados en la literatura de ruteo de vehículos. La mayoría de los trabajos se han enfocado en el análisis empírico, donde las heurísticas son implementadas en problemas estándares de prueba y su performance es comparada con otras heurísticas.

En contraste, muy pocos documentos han estudiado el problema desde un punto de vista analítico en busca de caracterizar el comportamiento teórico de las heurísticas.

El trabajo realizado se basa en un análisis probabilístico del VRPTW que se efectúa en busca de caracterizar la estructura de la solución óptima asintótica. Esta solución óptima está directamente relacionada con la solución óptima del problema de la planificación de itinerarios de una máquina estilizadora. De todas maneras no se va a profundizar en esta solución, o al menos por el momento, en esta primera versión del estado del arte.

### **Ruteo de Vehículos con Entregas Particionadas**

El VRP no permite que un cliente sea atendido por más de un vehículo. El SDVRP permite que a un cliente lo atiendan más de un vehículo siempre y cuando esto reduzca el costo total del problema. Esto es de mucha ayuda en los casos en los que la demanda del cliente es comparable al de los vehículos.

### **Ruteo de Vehículos con Múltiples Depósitos**

Una compañía puede tener varios depósitos desde los cuales abastece a sus clientes. Si los clientes están aglomerados alrededor de los depósitos, entonces el problema de distribución debe ser tratado como un conjunto independiente de VRPs. Sin embargo, si los clientes y los depósitos están entremezclados se debe utilizar el Ruteo de Vehículos con Múltiples Depósitos para tratar el problema.

Así como el caso base de VRP el MDVRP requiere depósitos asociados a clientes. Por cada depósito existe una flota de vehículos, los cuales inician su recorrido en ese depósito, atienden a los clientes asociados al mismo para retornar luego a este.

## 10 Apéndice 2: Definiciones matemáticas

[MAT]

### Grafo

Sea  $V$  un conjunto finito no vacío, y sea  $E \subseteq V \times V$ . El par  $(V, E)$  es un grafo dirigido (sobre  $V$ ), o digrafo (sobre  $V$ ), donde  $V$  es el conjunto de vértices, o nodos y  $E$  es su conjunto de aristas. Escribimos  $G = (V, E)$  para denotar tal digrafo.

### Grafos dirigidos

Son aquellos en los que la relación  $E$  no es simétrica. Es decir que si está la relación  $(a,b)$  en  $E$ , siendo  $a$  y  $b$  vértices, no tiene porque estar la relación  $(b,a)$ .

### Grafos no dirigidos

Son aquellos en los que la relación  $E$  es simétrica. Se puede representar la relación  $E$  como duplas no ordenadas de vértices, ya que si está la relación  $(a,b)$  en  $E$ , siendo  $a$  y  $b$  vértices, necesariamente tiene que estar la relación  $(b, a)$ .

### Lazo

Decimos que hay un lazo cuando existe la relación  $(a, a)$  en  $E$ , siendo  $a$  un vértice.

### Camino

Sean  $a, b$  vértices (no necesariamente distintos) de un grafo no dirigido  $G = (V, E)$ . Un camino  $a$ - $b$  en una sucesión alternada finita (sin lazos).

$a = a_0, e_1, a_1, e_2, a_2, e_3, \dots, e_{n-1}, a_{n-1}, e_n, a_n = b$   
de vértices y aristas de  $G$ , que comienza en el vértice  $a$  y termina en el vértice  $b$  y que contiene las  $n$  aristas  $e_i = \{x_{i-1}, x_i\}$  donde  $1 \leq i \leq n$ .

### Longitud de un camino

La longitud de un camino es  $n$ , el número de aristas que hay en el camino. Si  $n = 0$ , no existen aristas,  $x = y$ , y el camino se denomina trivial.

### Camino cerrado

Cualquier camino  $x - y$  donde  $x = y$  ( $y \neq x$ ) es un camino cerrado. En caso contrario es un camino abierto.

### Recorrido

Un recorrido es un camino en el que no se repite ninguna arista.

### Circuito

Un recorrido cerrado  $x - x$  es un circuito.

### **Camino simple**

Cuando ningún vértice del camino  $x - y$  se presenta más de una vez, el camino es un camino simple.

### **Ciclo**

Es un camino simple cerrado.

### **Grafo conexo**

Sea  $G = (V, E)$  un grafo no dirigido. Decimos que  $G$  es conexo si existe un camino simple entre cualesquiera dos vértices distintos de  $G$ .

Sea  $G = (V, E)$  un grafo dirigido. Su grafo no dirigido asociado es el grafo obtenido de  $G$  si no se tienen en cuenta las direcciones de las aristas. Si se obtiene más de una arista no dirigida de un par de vértices distintos de  $G$ , entonces sólo una de las aristas se dibuja en el grafo no dirigido asociado. Cuando este grafo asociado es conexo, consideramos que  $G$  es conexo.

### **Grafo desconexo**

Un grafo que no es conexo es desconexo.

#### **Grado de un vértice**

Sea  $G$  un grafo o multigrafo no dirigido. Para cualquier vértice  $v$  de  $G$ , el grado de  $v$ , que se denota  $\text{grad}(v)$ , es el número de aristas en  $G$  que son incidentes con  $v$ . En este caso, un lazo en un vértice  $v$  se considera como dos aristas incidentes en  $v$ .

### **Circuitos y recorridos eulerianos**

Sea  $G = (V, E)$  un grafo o multigrafo no dirigido sin vértices aislados. Entonces  $G$  tiene un circuito euleriano si existe un circuito en  $G$  que recorre cada arista del grafo exactamente una vez. Si existe un recorrido abierto de  $a$  a  $b$  en  $G$  que recorre cada arista de  $G$  exactamente una vez, este recorrido se llamará recorrido euleriano.

### **Grado de entrada y de salida de un vértice**

Sea  $G = (V, E)$  un grafo o un multigrafo dirigido. Para cualquier  $v \in V$ :

- a) El grado de entrada de  $v$  es el número de aristas de  $G$  que llegan a  $v$  y se denota con  $g_e(v)$
- b) El grado de salida de  $v$  es el número de aristas de  $G$  que parten de  $v$  y se denota con  $g_s(v)$ .

Si el grafo o multigrafo dirigido tiene uno o mas lazos, cada lazo de un vértice dado  $v$  contribuye con una unidad a  $g_e(v)$  y a  $g_s(v)$ .

### **Grafo o multigrafo plano**

Un grafo o multigrafo  $G$  es plano si podemos dibujar  $G$  en el plano de modo que sus aristas se intersequen sólo en los vértices de  $G$ .

### **Ciclos y caminos hamiltonianos**

Si  $G = (V, E)$  es un grafo o multigrafo con  $|V| \geq 3$ , decimos que  $G$  tiene un ciclo hamiltoniano si existe un ciclo en  $G$  que contenga cada vértice de  $V$ . Un camino hamiltoniano es un camino simple (y no un ciclo) de  $G$  que contiene todos los vértices.

## Aplicación a TSP

Dado un grafo con un ciclo hamiltoniano, la eliminación de cualquier arista en el ciclo produce un camino hamiltoniano. Sin embargo, es posible que un grafo tenga un camino hamiltoniano sin que tenga un ciclo hamiltoniano.

Podría parecer que la existencia de un ciclo (camino) hamiltoniano y la existencia de un circuito (recorrido) euleriano para un grafo son problemas similares. El ciclo (camino) hamiltoniano tiene como objetivo pasar por cada vértice de un grafo una sola vez; el circuito (recorrido) euleriano recorre el grafo pasando por cada arista exactamente una vez. Por desgracia, no existe una relación útil entre las dos ideas y, a diferencia de los circuitos (recorridos) eulerianos, no existen condiciones necesarias y suficientes en un grafo  $G$  que garanticen la existencia de un ciclo (camino) hamiltoniano. Si un grafo tiene un ciclo hamiltoniano, entonces al menos será conexo.

## Teoremas

### Teorema1

Sea  $G = (V, E)$  un grafo sin lazos,  $|V| = n \geq 2$ . Si  $\text{grad}(x) + \text{grad}(y) \geq n - 1$  para todos  $x, y \in V, x \neq y$ , entonces  $G$  tiene un camino hamiltoniano.

### Corolario

Sea  $G = (V, E)$  un grafo sin lazos con  $n (\geq 2)$  vértices. Si  $\text{grad}(v) \geq (n - 1)/2$  para todo  $v \in V$ , entonces  $G$  tiene un camino hamiltoniano.

### Teorema2

Sea  $G = (V, E)$  un grafo no dirigido sin lazos con  $|V| = n \geq 3$ . Si  $\text{grad}(x) + \text{grad}(y) \geq n$  para todos los vértices  $x, y \in V$  no adyacentes, entonces  $G$  tiene un ciclo hamiltoniano.

### Corolario

Si  $G = (V, E)$  es un grafo no dirigido sin lazos, con  $|V| = n \geq 3$  y si  $\text{grad}(v) \geq n/2$ , para todo  $v \in V$ , entonces  $G$  tiene un ciclo hamiltoniano.

## Aplicación

El problema del vendedor viajante que sale de su casa y debe visitar varios lugares antes de regresar encontrando una forma de hacer su viaje más eficiente en términos de distancia recorrida se puede modelar mediante un grafo etiquetado (las aristas tienen distancias o costos asociados a ellas). El problema entonces es, encontrar un ciclo hamiltoniano eficiente.

## 11 Apéndice 3: Conceptos de física

### 11.1 Estructura atómica y espectros atómicos

#### Introducción

Al comienzo del siglo XX el electrón estaba firmemente establecido como partícula fundamental. Se sabía que los electrones se liberaban en la emisión termiónica (de un metal caliente) y en el efecto fotoeléctrico. Antes de 1900, Becquerel ya había encontrado que ciertos elementos emiten los llamados rayos  $\beta$ , que en realidad son electrones.

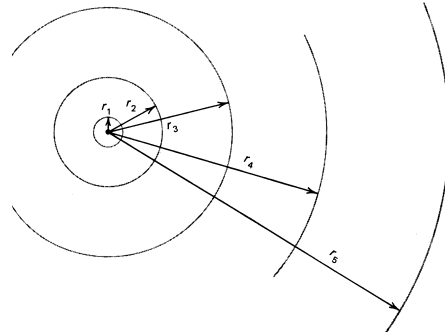
Durante los primeros años del siglo XX los físicos se ocupaban en construir modelos de átomos que fueran compatibles con los datos físicos y químicos ya conocidos. Tales modelos debían tener electrones como componentes, alguna carga positiva que neutralizara a las cargas negativas de los electrones y algún esquema para explicar los distintos pesos atómicos y las diferentes propiedades químicas de los átomos.

A uno de estos modelos, propuesto aproximadamente en 1904 por J.J. Thomson, se le apodaba el modelo del budín de pasas. Se imaginaba al átomo como una burbuja esférica que contenía  $N$  electrones embebidos en una distribución uniforme de carga positiva cuyo valor total compensaba exactamente el de los electrones. Las "pasas" o electrones quedaban fijos dentro del "budín" esférico positivo.

#### El modelo de Bohr para el átomo de hidrógeno

En 1913 Niels Bohr propuso un nuevo modelo para el átomo de hidrógeno para el cual utilizó las siguientes hipótesis:

- Los electrones sólo pueden existir en ciertas órbitas especiales alrededor del núcleo central denominadas órbitas estacionarias o estados estacionarios.
- El equilibrio dinámico del sistema en los estados estacionarios está controlado por la mecánica newtoniana.
- Las transiciones del sistema entre distintos estados estacionarios se acompañan de la emisión o absorción de radiación, cuya frecuencia  $\nu$  está dada por la fórmula de Planck  $\nu = \Delta E/h$  siendo  $\Delta E$  la diferencia de energías entre los dos estados estacionarios.



Órbitas circulares de los estados estacionarios del átomo de hidrógeno, de acuerdo con el modelo de Bohr.

### Electrostática: cargas, campos eléctricos y potenciales

Las investigaciones mas relevantes sobre los efectos de la electricidad comenzaron hacia el final del siglo XVII. Charles Francois de Cisternay du Fay efectuó un descubrimiento importante en 1734:

*"...la suerte ha puesto otro principio en mi camino...que arroja nueva luz sobre el tema de la electricidad. Este principio es que hay dos electricidades distintas, que varían mucho entre sí: a una de ellas la llamo electricidad vítrea; a la otra, electricidad resinosa. La primera es la del hule, la roca, el cristal, las piedras preciosas, el pelo de los animales, la lana y muchos otros cuerpos (frotados). La segunda es la del ámbar, el copal, la goma laca, la seda, la fibra, el papel y un gran número de substancias (frotadas).*

*La característica de esas dos electricidades es que un cuerpo de , digamos, electricidad vítrea, repele a todos los que sean de la misma electricidad; y por el contrario, atrae a todos los de electricidad resinosa".*

Actualmente se utilizan otros nombres: hablamos de cargas y no de electricidades y de positivas y negativas en vez de vítrea y resinosa. También se sabe que los efectos observados son provocados por un exceso de cargas en el vidrio y en el ámbar.

## 11.2 Origen de las cargas eléctricas

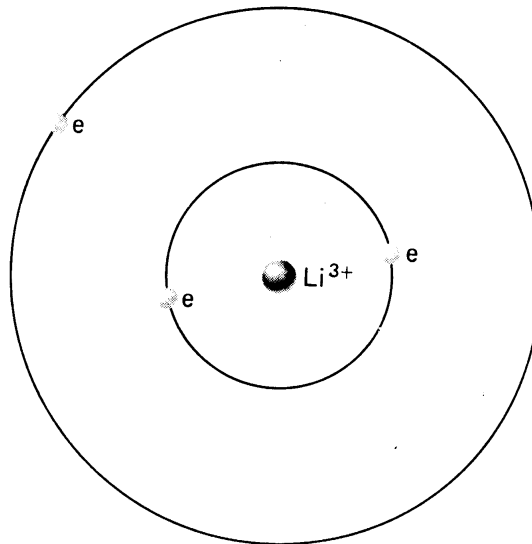
Cada átomo está formado por un núcleo muy pequeño y masivo, cargado positivamente, y de uno o varios electrones mucho más ligeros y cargados negativamente.

La carga positiva del núcleo atrae a los electrones cargados negativamente, manteniéndolos en órbitas estables. Aquellos electrones que están más cerca del núcleo se ligan fuertemente; por lo que no es fácil apartarlos. A estos, se les denomina electrones internos. A los que ocupan las órbitas que están más alejadas se les denomina electrones de valencia y se pueden separar más fácilmente.

Para mostrar un ejemplo, veremos el átomo de litio. El núcleo tiene tres protones, cada uno con un carga igual en magnitud a la de un electrón, pero de signo positivo.



Este núcleo está rodeado por tres electrones cargados negativamente, y el sistema completo es eléctricamente neutro. En este caso hay dos electrones internos (los dos más cercanos al núcleo) y un electrón de valencia.



Para algunos materiales, tales como el vidrio, los electrones que están en la superficie de los átomos se pueden llevar a la seda mediante frotamiento mecánico. Igualmente, los electrones de los átomos en la piel o en la lana tienden a emigrar a la ebonita o al ámbar. Como resultado de esta transferencia de cargas, la ebonita adquiere un exceso de cargas negativas al frotarla con la piel, mientras que una varilla de vidrio tendrá defecto de electrones y quedará cargada positivamente al frotarla contra la seda.

### Propiedades fundamentales de la carga:

La carga posee ciertos atributos fundamentales. La carga está cuantizada, la carga se conserva y la carga es invariante.

Que la carga esté cuantizada significa que puede aparecer solamente como múltiplos enteros de una unidad fundamental indivisible. Esta unidad es la carga electrónica  $e$ . Por conservación de la carga se entiende que la carga total de un sistema aislado debe permanecer constante.

El concepto de invariancia se aplica a objetos bajo condiciones relativistas, es decir, a velocidades que se aproximan a la velocidad de la luz ( $c = 3.0 \times 10^8$ ). Que la carga sea invariante significa que la carga de un electrón o de un protón o de cualquier otra partícula permanece igual, sin importar la velocidad del movimiento.

### Conductores, aisladores y semiconductores

Algunos materiales, como el cobre, el aluminio y otros metales conducen electricidad, mientras que otros grupos de materiales que incluye el vidrio y la mayoría de los plásticos se usan ampliamente como aisladores eléctricos.

Lo que distingue a estos dos tipos de materiales con respecto a sus características eléctricas es que en los conductores, los electrones externos, los llamados electrones de valencia, pueden moverse con relativa libertad en el interior del material, aunque no pueden liberarse fácilmente del él. En los aisladores hasta los electrones de valencia están firmemente ligados a sus respectivos núcleos.

Hay una tercera clase de sustancias, que incluye a los elementos germanio y silicio, así como también a numerosos compuestos tales como el antimonio de indio, y el arseniuro de galio, que ni son conductores ni aisladores. Esas sustancias, el número de electrones libres sólo es una pequeña fracción de los que hay en un metal; además, el número depende de la temperatura y de la concentración y tipo de impureza que puede estar presente. Estos semiconductores conducen electricidad, pero en ellos es mucho mayor la resistencia comparada con la de los metales, aunque a la vez mucho menor que en los aisladores.

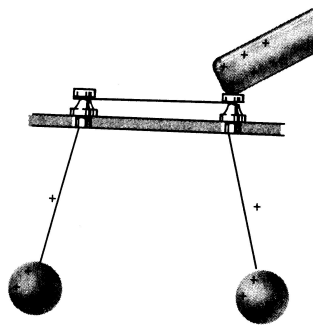
### Interacción entre cargas

Está demostrado que

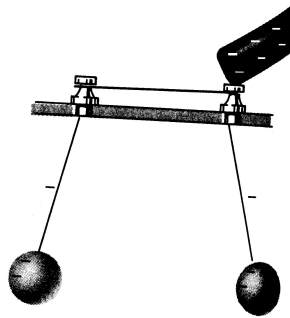
- Las cargas de igual signo se repelen entre sí.
- Las cargas de distinto signo se atraen entre sí.

Como un ejemplo de esto se puede ver que si se carga por frotamiento una barra de vidrio o de ebonita, parte de la carga se puede transferir a una pelota de médula por contacto físico:

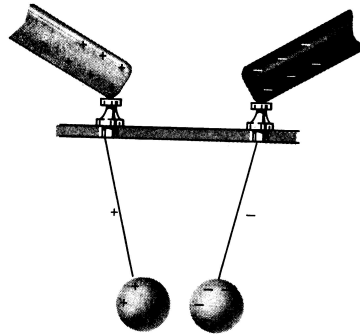
Dos pelotas de médula pintadas con pintura de aluminio están suspendidas por un alambre metálico de soporte metálicos colocados en una barra aislante horizontal.



Un alambre metálico conecta los dos soportes. Cuando una varilla de vidrio cargada positivamente toca el soporte metálico, la carga positiva se distribuye sobre las pelotas de médula. El efecto que se observa es una repulsión de las dos pelotas.

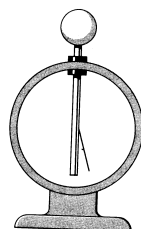


Un alambre metálico conecta los dos soportes. Cuando una varilla de ebonita cargada negativamente toca el soporte metálico, la carga negativa se distribuye sobre las pelotas de médula. El efecto que se observa es una repulsión de las dos pelotas.



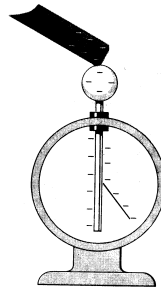
Ningún alambre metálico conecta los dos soportes. Si se carga una pelota positivamente y otra negativamente, se observa que las dos se atraen.

Para investigar las fuerzas electrostáticas y para deducir relaciones cuantitativas, se necesita un dispositivo para medir la carga eléctrica. Uno de los más antiguos instrumentos es el electroscopio de hoja de oro. Consiste en un cilindro hueco en el que se fija una hoja delgada de oro por su extremo superior a una varilla metálica vertical. La varilla se sostiene en un buje aislante y se pasa a través de la parte superior de la cámara; los laterales de la cámara se hacen de vidrio para que se pueda ver la posición de la hoja de oro.



Si, por ejemplo, una carga negativa se deposita sobre la esferita de latón que está sobre la varilla metálica, los muchos millones de electrones que sobran no permanecerán fijos allí. Podrán moverse hacia abajo y sobre la superficie del metal, y como se repelen entre sí, se redistribuirán para mantenerse lo más alejados que sea posible unos de otros. Así parte de esa carga negativa aparecerá sobre la hoja de oro y sobre la varilla metálica, y como las cargas de igual signo se repelen, la hoja flexible

de oro se separará de la varilla, y el ángulo formado dependerá de la cantidad de carga depositada sobre el electroscopio.



### Ley de Coulomb

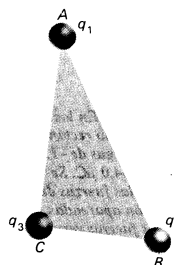
La relación matemática que controla la interacción entre las cargas de signos iguales y de signos opuestos fue enunciada por Charles Coulomb en 1785. Coulomb midió la magnitud de las fuerzas como función de la distancia entre las cargas y la magnitud de dichas cargas, usando una balanza de torsión muy sensible. Llegó a la conclusión de que la magnitud de la fuerza entre dos cargas en el aire es proporcional al producto de las dos cargas,  $q_1$ , y  $q_2$ , e inversamente proporcional al cuadrado de la distancia entre ellas:

$$F = k q_1 q_2 / r^2 \quad (k = 8.9874 \times 10^9 \text{ N.m}^2/\text{C}^2)$$

La dirección de la fuerza sobre las cargas queda a lo largo de la línea que las une. Al resolver la ecuación, una fuerza positiva significa repulsión y una fuerza negativa significa atracción. Como una fuerza es una cantidad vectorial, los adjetivos positivo y negativo no tienen significado en cualquier otro contexto (que no sea vectorial).

### Fuerza electrostática debida a distribución de cargas

Si colocamos dos cargas  $q_1$  y  $q_2$  la fuerza de cada una de ellas sobre la otra se calcula utilizando la ley de Coulomb. Pero si colocamos una tercera carga  $q_3$  de la siguiente manera:



¿cual será la fuerza sobre  $q_3$ ? La suma vectorial de la fuerza que ejerce  $q_1$  y  $q_2$ . También tenemos que tener en cuenta que la fuerza neta ejercida sobre  $q_1$ , ahora, será la suma vectorial de la fuerza ejercida por  $q_2$  y la fuerza ejercida por  $q_3$ .

La presencia de una "nueva" carga en el sistema ( $q_3$ ) no afecta a la fuerza que existe entre  $q_1$  y  $q_2$ . Esto se puede ver notando que en la ecuación de la ley de Coulomb no aparecen todas las cargas incidentes en el sistema.

De esto se desprende el principio de superposición: La presencia de otras cargas no influye con la interacción entre cualquier otro par de cargas.

La fuerza neta que actúa sobre cualquier carga es la suma vectorial de las fuerzas debidas a cada una de las cargas restantes de una distribución determinada.

## El campo eléctrico

De acuerdo con el principio de superposición, la fuerza que actúa sobre una carga  $q$  localizada en algún punto del espacio que contiene otras cargas, es la suma vectorial de las fuerzas debidas a todas esas cargas adicionales. Cada una de las fuerzas componentes es proporcional a la carga  $q$ , y por tanto la fuerza resultante también debe ser proporcional a  $q$ .

Se define el campo eléctrico  $E$ , en el punto que corresponde al vector de posición  $r$ , como la fuerza que experimentaría una carga unitaria positiva si estuviera localizada en ese punto y si su colocación no alterara la distribución de cualesquiera de las demás cargas en el espacio.

La condición de que una carga unitaria de prueba no distorsione la distribución de cualquier carga existente no es real. Pero como la fuerza sobre cualquier carga en algún punto en el espacio es proporcional a la magnitud de la misma carga, la carga de prueba se puede hacer tan pequeña como se desee. Idealmente, se podría hacer tan pequeña que fuera despreciable su efecto sobre los alrededores. Entonces, se puede redefinir el campo eléctrico como la fuerza electrostática por unidad de carga:  $E = F_{qr} / q$  Siendo  $F_{qr}$  la fuerza que actúa sobre una carga  $q$  pequeña colocada en el punto  $r$ .

De esta ecuación se desprende que las unidades del campo eléctrico son newton / coulomb.

## 11.3 La luz

Antes de 1800, el punto de vista que generalmente se aceptaba, apoyado un siglo antes por Isaac Newton, sostenía que la luz consistía de una corriente de pequeñas partículas, viajando en líneas rectas, que producían la sensación de visión al chocar con la retina del ojo.

A principios del siglo XIX, Young en Inglaterra y Fresnel en Francia demostraron fenómenos de interferencia y difracción usando luz visible. Aunque se reconoció que la luz era un fenómeno ondulatorio, la naturaleza de las ondas luminosas aún no se conocía hasta la última parte de ese siglo, cuando Maxwell predijo ondas electromagnéticas que se propagan a la velocidad de la luz.

Maxwell murió ocho años antes de que Heinrich Hertz con éxito generara y detectara ondas electromagnéticas.

Actualmente se considera que la luz – y la radiación electromagnética de todas las longitudes de onda – tienen propiedades tanto ondulatorias como de partícula; el atributo que predomine está determinado por el experimento que se efectúe.

## Velocidad de la luz

Uno de los primeros experimentos registrados para medir la velocidad de la luz fue el efectuado por Galileo; quien utilizó una lámpara, un obturador y un espejo. El espejo estaba colocado en una colina, a una distancia considerable de la lámpara y del obturador. Galileo trató de determinar la velocidad de la luz midiendo el intervalo entre la apertura del obturador para dar paso a la luz de la linterna y su percepción de la luz reflejada por el espejo. Galileo concluyó que "si no es instantánea, es extremadamente rápida".

Muchos otros intentos de medición de la velocidad de la luz continuaron en el tiempo hasta lograr calcularla. La velocidad de la luz en el vacío es una constante fundamental importante de la naturaleza, se le destina un símbolo especial, la *C* minúscula.

Por definición, su valor es  $c = 299\,792\,458 \text{ m/s} \approx 300\,000 \text{ km/s} = 3.00 \times 10^8 \text{ m/s}$

## Índice de refracción

En el vacío, la luz se propaga, entonces, a una velocidad  $c = 3.0 \times 10^8 \text{ m/s}$ . En cualquier otro medio la luz se propaga más lentamente; por ejemplo en el aire,  $v_{\text{aire}} = c/1.0003$ ; en el agua,  $v_{\text{H}_2\text{O}} = c/1.33$ ; y en el diamante,  $v_{\text{día}} = c/2.42$ . A la relación  $c/v_m$ , siendo  $v_m$  la velocidad de la luz en un medio transparente  $m$ , se le denomina índice de refracción de ese material, y se lo representa con la letra  $n$ .

$$n = c/v; \quad v = c/n$$

Así, para el agua  $n = 1.33$ , la siguiente tabla incluye los índices de refracción para la distintas sustancias (para longitud de onda 589 nm):

Material	$c/v = n$
Aire	1.0003
Agua	1.33
Etanol	1.36
Acetona	1.36
Cuarzo fundido	1.46
Benceno	1.50
Lucita o plexiglás	1.51
Vidrio crown	1.52
Cloruro de sodio	1.53
Poliestireno	1.59
Disulfuro de carbono	1.63
Vidrio flint	1.66
Yoduro de metileno	1.74
Diamante	2.42

Mientras que en el vacío las ondas electromagnéticas de todas las longitudes se propagan a la misma velocidad  $c$ , en toda la materia la velocidad de la luz varía con la longitud de onda, es decir, con la frecuencia. Esta dependencia de la frecuencia con la velocidad de propagación se conoce como dispersión, y tiene consecuencias importantes.

Como  $v = f\lambda$ , se deduce que cuando la luz entra a un medio de índice de refracción  $n$  desde el aire, deben cambiar  $f$  o  $\lambda$ , o ambos. Podemos ver que  $f$  permanece constante al considerar la situación análoga de una cuerda larga compuesta de dos secciones que tengan diferentes densidades.



De la ecuación  $v = \lambda(f/\lambda)$  se deduce que las velocidades de propagación de ondas transversales en las dos secciones será distinta. Si provocamos una onda en el extremo izquierdo y la onda se propaga hacia la derecha, el número de veces que el punto A viaja hacia arriba y hacia abajo cada segundo está determinado por la frecuencia de la onda. Pero ese movimiento del punto A es lo que da lugar a la propagación, aunque a una frecuencia menor, de la variación en la porción derecha de la cuerda. Así, la frecuencia de la onda de la izquierda y de la derecha es la misma. Pero como la velocidad es menor en el segmento del lado derecho, la distancia entre las crestas sucesivas, esto es, la longitud de onda, se reduce en la misma proporción. Por tanto, se concluye que en un medio de índice de refracción  $n$ :

$$\lambda_n = v/f = c/(nf) = \mu_0/n$$

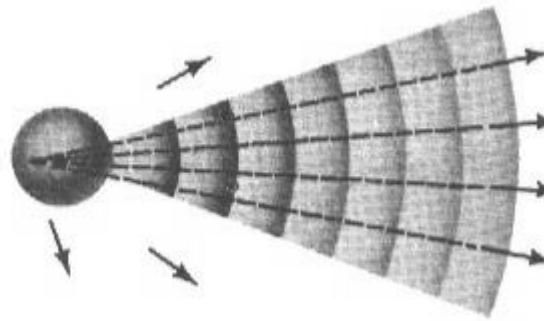
donde  $\mu_0$  es la longitud de onda en el aire (o en el vacío).

## Ondas y rayos

Si en un día soleado se abre un poco una persiana se podrá observar un haz delgado – un rayo – de luz en la habitación. Podemos reflejar este rayo con un espejo, observar su refracción (desviación) al pasar a través de una placa de vidrio o al entrar a un depósito de agua, y contemplar los colores producidos al colocar un prisma de vidrio en su camino.

Suponiendo que iniciamos un pulso en algún lugar del espacio. Así se propagará una onda hacia fuera, como una onda superficial se expande radialmente en un estanque si dejamos caer una piedra en el agua. El lugar geométrico de todos los puntos que tienen el mismo desplazamiento en la onda en el agua; o la misma magnitud del campo E en la onda electromagnética es el frente de la onda. Por ejemplo, el frente de onda en el agua podría ser la curva que conecta los puntos vecinos en cierta cresta. En tres dimensiones, el frente de onda es una superficie.

Cuando una fuente luminosa emite ondas electromagnéticas uniformemente en todas direcciones los frentes de onda viajan hacia afuera, su movimiento se puede especificar por medio de flechas:



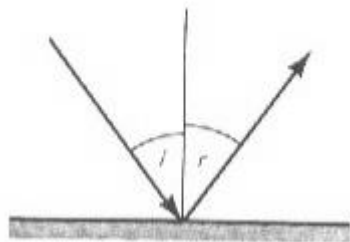
### Principio de Huygens

Huygens propuso que cada punto de un frente de onda se puede considerar como una nueva fuente de ondas. Las llamadas ondas de Huygens. La posición siguiente del frente de onda se obtiene superponiendo las ondas que emanan de estas fuentes ficticias.

### Reflexión de ondas planas

Ley de la reflexión especular (en espejo):

- El rayo reflejado queda en el plano formado por el rayo incidente y la normal a la superficie reflejante.
- El ángulo de reflexión es igual al ángulo de incidencia; se considera que dichos ángulos son los que forman los rayos incidente y reflejado con la normal a la superficie. ( $i = r$ )



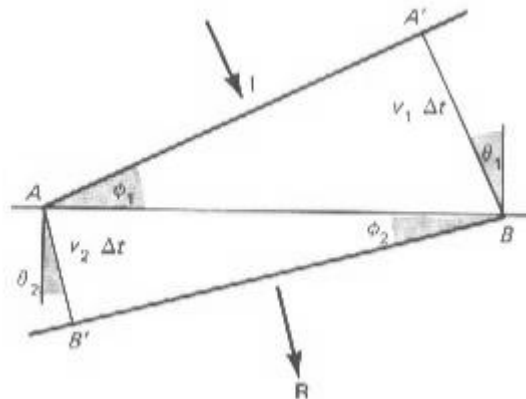
### Refracción

La refracción de la luz en la interfase entre dos medios que tienen distintos índices de refracción provoca muchos de los fenómenos comunes, como el acortamiento aparente de un objeto que está sumergido en agua y el desplazamiento de un objeto visto oblicuamente a través de un vidrio grueso.

El ángulo de refracción se define como el ángulo entre el rayo refractado y la normal a la superficie refractora. En la siguiente figura, la línea AA' representa al frente de onda de la onda incidente que se propaga en el medio 1 en la dirección indicada por el rayo I. Cuando el tiempo  $t = 0$ , el frente de onda se encuentra con la interfase entre el medio 1 y el medio 2 en A, y una onda de Huygens se propaga radialmente hacia fuera



de ese punto. En el medio 1, la onda se mueve a una velocidad  $v_1$ , y en el medio 2 a una velocidad  $v_2$ . En el tiempo  $\Delta t = A'B/v_1$ , el frente de onda incidente ha avanzado desde  $A'$  hasta  $B$ . En ese mismo intervalo de tiempo, la onda de Huygens con centro en  $A$  ha avanzado una distancia  $AB' = v_2\Delta t$  siendo  $v_2$  la velocidad de propagación en el medio 2. Así, el frente de onda de la onda plana transmitida al medio 2 es  $BB'$ , y la dirección de propagación en el medio 2 se representa por el vector  $R$ , perpendicular a  $BB'$ .



De la figura se desprende que  $AB = v_1\Delta t/\text{sen}\phi_1 = v_2\Delta t/\text{sen}\phi_2$   
 Pero como  $\theta_1 = \phi_1$  y  $\theta_2 = \phi_2$  obtenemos la ley de Snell:

$$v_1/v_2 = \text{sen } \theta_1 / \text{sen } \theta_2$$

Como  $v_1 = c/n_1$  y  $v_2 = c/n_2$  esta ley se puede escribir de la siguiente forma:

$$n_1\text{sen}\theta_1 = n_2\text{sen}\theta_2$$

Si  $n_2 > n_1$ , entonces  $\text{sen}\theta_2 < \text{sen}\theta_1$ , y  $\theta_2 < \theta_1$ . En otras palabras, cuando la luz viaja de un medio de bajo índice de refracción a uno de índice más alto, se desvía hacia la normal. Al revés, la luz se refracta alejándose de la normal al pasar de un medio de índice de refracción grande a un medio con menor índice de refracción.

## Reflexión interna total

Consideremos un rayo de luz que pasa del medio 1, con índice  $n_1$  de refracción, al medio 2 con índice de refracción  $n_2$ . De acuerdo con la ley de Snell se puede escribir

$$\text{sen}\theta_2 = (n_1/n_2)\text{sen}\theta_1$$

Cuando  $n_1 > n_2$ , habrá un rango de ángulos  $90^\circ > \theta_1 > \theta_c$  entre los ángulos para los que  $\text{sen } \theta_2 > 1$ . Bajo estas circunstancias, no se puede satisfacer la ecuación para ángulo real  $\theta_2$ , y no se propagará el rayo refractado en el seno del segundo medio; toda la luz incidente se refleja en la interfase.

Este fenómeno se conoce como reflexión interna total, ya que el haz incidente generalmente está dentro de algún material como vidrio o plástico y se refleja en la interfase vidrio-aire.

El ángulo crítico para la reflexión interna total,  $\theta_c$ , es aquel ángulo para el que  $(n_1/n_2)\text{sen}\theta_c = 1$ , o sea

$$\theta_c = \text{sen}^{-1}(n_2/n_1)$$

Si el ángulo de incidencia es mayor o igual a  $\theta_c$ , la luz se reflejará totalmente.

## Dispersión

La velocidad de propagación de la luz en todas las sustancias depende hasta cierto punto de la frecuencia de la radiación; sólo en el vacío la velocidad de propagación de las ondas electromagnéticas es independiente de la frecuencia.

Por ejemplo en índice de refracción de el silicio fundido (vidrio de cuarzo) es de 1.4636 para la luz azul con longitud de onda de 480 nm (en el aire) y 1.4561 para la luz roja con longitud de onda de 670 nm. Por lo tanto, cuando un rayo que contenga luz azul o roja llegue a un trozo de silicio fundido en ángulo de incidencia cualquiera, excepto cero grados, habrá dos rayos refractados que se propagarán en ángulos ligeramente distintos. El rayo azul será desviado hacia la normal un poco más que el rayo rojo. Un rayo de luz blanca, que abarque todas las longitudes de onda del espectro visible, será dispersado de modo semejante en todos sus varios componentes espectrales al refractarse en un vidrio. Los prismas de vidrio u otras sustancias transparentes con frecuencia se emplean en los espectroscopios que son instrumentos diseñados para identificar la composición espectral de una fuente luminosa. En la siguiente figura se muestra cómo un prisma logra esa separación. Como se puede ver, a luz se dispersa dos veces, una en la interfase aire-vidrio, y una vez más en la interfase vidrio-aire.

## Glosario

- Átomo:** Estructura que forma la unidad básica de cualquier elemento. Es la menor unidad de materia que puede intervenir en una combinación química.
- Electrón:** Partícula elemental, eléctricamente negativa, de los átomos, de masa  $9,11 \times 10^{-31}$  Kg. y carga  $1,602 \times 10^{-19}$  culombios.
- Protón:** Partícula elemental nuclear de carga positiva y masa de 1,00759 unidades másicas.
- Neutrón:** Partícula elemental del núcleo atómico, de carga eléctrica nula, cuya masa es aproximadamente igual a la del protón.
- Quarks:** M. Gell-Mann sugirió la hipótesis de que los protones y los neutrones están compuestos por partículas subnucleares que llevan cargas iguales a  $2/3$  y  $1/3$  de la carga del electrón. Siguió un período de búsqueda intensiva, aunque desafortunada, de los llamados quarks. Se han acumulado pruebas de que efectivamente existen dichos componentes con cargas fraccionarias dentro del núcleo, aunque no se les ha aislado. La existencia de los quarks no viola el concepto de cuantización de la carga; sólo significa que la unidad básica de carga podría reducirse por un factor de 3.
- Coulomb, Charles:** (1736-1806) Ingeniero y físico francés. Inició el estudio científico de los fenómenos eléctricos.
- Culombio:** Unidad de medida de la carga o masa eléctrica, simbolizada por la letra C, y definida como la cara transportada por un amperio durante un segundo. Campo eléctrico: Región del espacio en que

una masa eléctrica pasiva colocada en cualquier punto sufre la acción de una fuerza.

## 12 Apéndice 4: Manual de usuario Visual Tour

### 12.1 Antes de comenzar

#### Introducción

Este capítulo provee la información inicial acerca de las acciones posibles de ejecutar a ser realizadas en la Herramienta de ruteo Visual Tour. Incluye, además de las convenciones, contenido e instrucciones para una mejor comprensión y utilización de este manual, una lista de terminologías que permite aclarar el significado específico que tiene para la Visual Tour cada expresión aquí mencionada.

#### Acerca del manual de la Herramienta

La documentación de la herramienta de ruteo Visual Tour consiste en un manual, que entrega la información necesaria para operar los distintos módulos desarrollados para la verificación, y la comprensión en base al comportamiento de los algoritmos desarrollados.

#### Premisas

En este manual se asume que el usuario que utilice la herramienta está familiarizado tanto con el entorno MS Windows, y que sabe cómo trabajar con aplicaciones de MS Windows usando un mouse.

#### El Manual de la Herramienta de Ruteo

El Manual de la herramienta de ruteo Visual Tour está dividido en 3 capítulos específicos:

**Antes de comenzar**, entrega la información acerca de la herramienta, convenciones de uso y modo de inicio, entre otros temas.

**Descripción general**, capítulo orientado a dar una visión general del sistema. Presenta las funciones básicas y describe las peculiaridades del mismo.

**Operación**, provee instrucciones detalladas sobre cómo ejecutar las distintas funciones del módulo y entrega ejemplos completamente desarrollados e ilustrados mediante imágenes de pantalla.

#### Símbolos y convenciones

Para facilitar la búsqueda e interpretación de la información, la herramienta de ruteo Visual Tour emplea algunas convenciones visuales y cierta terminología ceñida a un significado especial dentro del contexto de la misma y de su documentación.

### Convenciones visuales de teclado

A continuación se indican las convenciones de tipografía utilizadas y su significado.

Estilo	Significado
<i>Cursiva</i>	Este tipo de letra se usa para describir las instrucciones y características de las imágenes de pantalla (cuadros de diálogo y aviso) que se mencionan en el manual.
<b>Negrita</b>	Se usa para destacar en el texto los comandos, menús, cuadros de diálogo y nombres de botones.
<b>1., 2.</b>	Secuencia de pasos numerados que componen un procedimiento.
↵	Tecla de retorno o entrada. Instrucciones para aquellos usuarios que prefieran utilizar el teclado.

La herramienta requiere el uso del teclado para ingresar información en los cuadros de texto.

### Convenciones para el mouse

La herramienta de ruteo Visual Tour usa el botón izquierdo y derecho del *mouse* para efectuar diferentes acciones en función del estado en que se encuentre el sistema.

Acción	Significado
Elegir / escoger	Utilizar el <i>mouse</i> para activar un elemento que iniciará una acción asociada.
Hacer clic	Presionar y soltar un botón del <i>mouse</i> .
Hacer doble clic	Hacer clic dos veces consecutivas en un botón del <i>mouse</i> .

### Convenciones de terminología

Los términos utilizados en el Manual de procedimientos de la herramienta de ruteo Visual Tour, en general, están conforme con los de MS Windows XP versión en español. No obstante, los términos descriptos a continuación tienen un significado especial o uso estándar dentro del contexto de Visual Tour y de su documentación.

Término	Significado
Barra de botones	Es la barra horizontal que contiene los botones de acción rápida de la herramienta de ruteo <b>Visual Tour</b> . Se encuentra ubicada sobre el área superior de la aplicación.
Barra de menú	Es la barra horizontal que contiene los nombres de los menús de la herramienta de ruteo <b>Visual Tour</b> . Se encuentra ubicada debajo de la barra de título de la ventana activa.
Botón de comando	Botón en un cuadro de diálogo que ejecuta o cancela una acción.
Comando	Item de un menú de la herramienta de ruteo <b>Visual Tour</b> que identifica

	una acción específica a realizar.
Menú	Corresponde a una lista de comandos destacados que permiten iniciar o ejecutar acciones.
Usuario	Es la persona que va a utilizar la herramienta de ruteo <b>Visual Tour</b> .

## Ingreso al ambiente de operación

### Inicio de la herramienta de ruteo Visual Tour

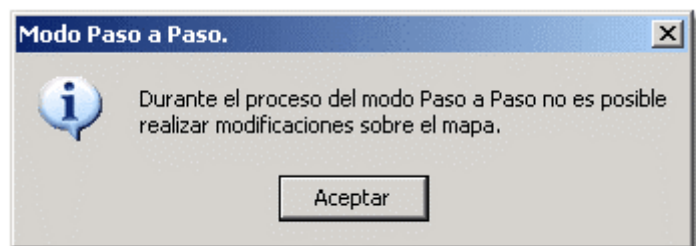
El proceso de instalación de la herramienta de ruteo Visual Tour crea un elemento de grupo de programas Windows que contiene el icono con que se puede invocar la aplicación. La aplicación la Herramienta de Ruteo se puede iniciar a través del teclado o utilizando el mouse.

Si el icono de la herramienta de ruteo Visual Tour no está seleccionado y el grupo no se encuentra activo, haga clic sobre él o presione Ctrl + Tab hasta que el grupo se active. Un grupo está activo cuando su barra de título y un icono del grupo se muestran destacados. A continuación, seleccione el icono correspondiente a la herramienta de ruteo Visual Tour, haciendo doble clic sobre él o pulsando la tecla **↵** si está previamente destacado.

### Los mensajes del sistema

Cuando la herramienta de ruteo Visual Tour no puede completar un comando, o cuando necesita advertir al usuario las consecuencias posibles de una acción en alguno de sus procedimientos, lo hace a través de cuadros de aviso, tal como se muestra en la figura siguiente.

Cuando sea necesario tomar alguna decisión antes de proceder a una acción, el cuadro incluirá los botones Aceptar, Aplicar, y Cancelar. El botón seleccionado y/o predeterminado aparecerá con un borde más oscuro que el otro.



El botón seleccionado se puede elegir pulsando la tecla **↵**, o bien, haciendo clic con el *mouse* sobre el botón deseado.

## 12.2 Descripción General

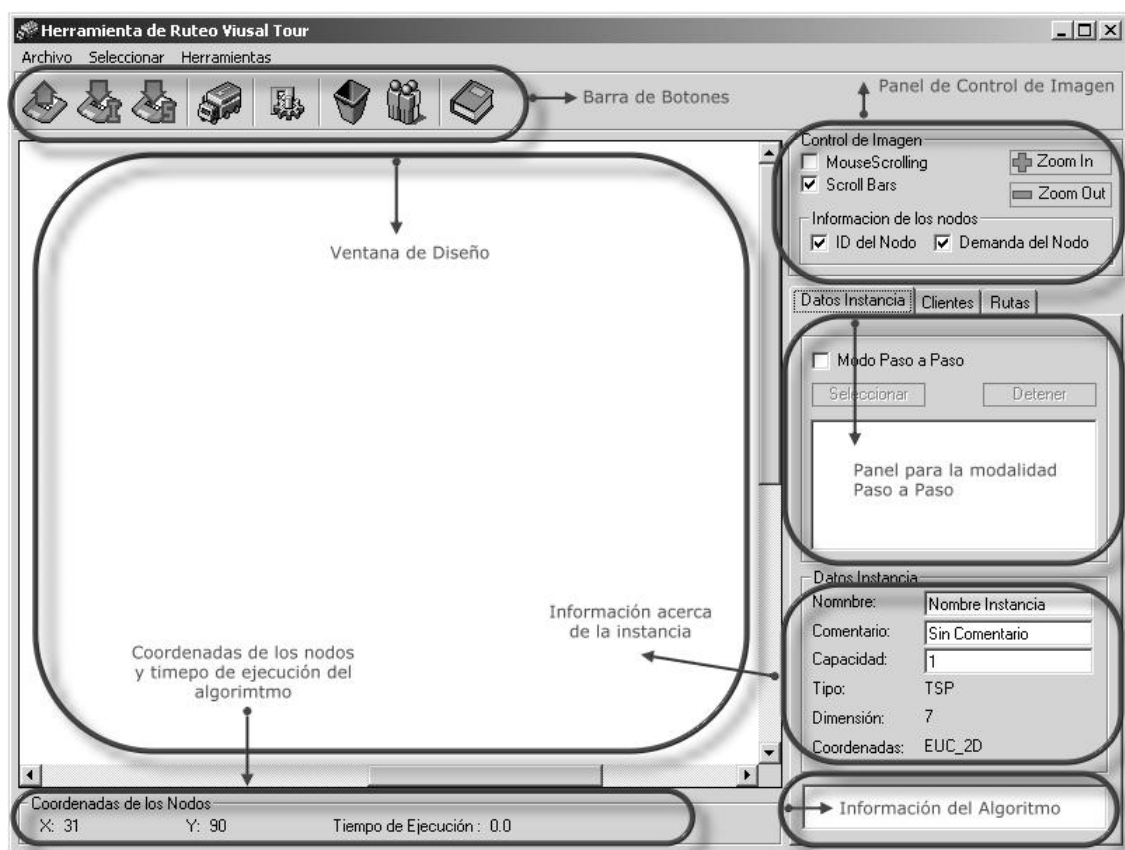
### Ingreso a la Herramienta de Ruteo

Para ingresar a la herramienta haga doble clic o pulse la tecla **↵** sobre el icono herramienta de ruteo Visual Tour en la ventana correspondiente de Windows como muestra la figura a la derecha.



Icono correspondiente al archivo ejecutable del Visual Tour

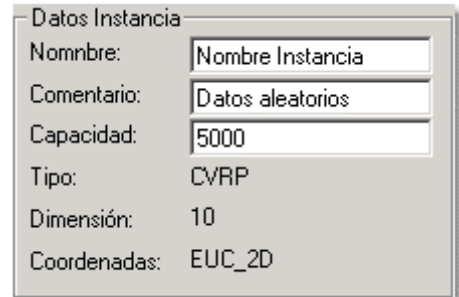
Una vez ejecutado, se cargara el ambiente de diseño de la herramienta desplegándose en pantalla una ventana con el contenido que se muestra a continuación.



### Datos de la Instancia

En la ventana de la aplicación se puede ver la información correspondiente a la instancia que se encuentra activa. Esta información que se muestran corresponde a la definida en los juegos de datos provistos por la TSPLIB 95.

- En la misma se despliegan:
- Nombre
- Comentario
- Capacidad
- Tipo
- Dimensión
- Coordenadas



Datos Instancia	
Nombre:	Nombre Instancia
Comentario:	Datos aleatorios
Capacidad:	5000
Tipo:	CVRP
Dimensión:	10
Coordenadas:	EUC_2D

En el caso de que se cargue una instancia a través de un archivo, este cuadro se formará con los datos que la misma contenga en ese archivo. Si se trata de una instancia nueva, que el usuario va a generar, este podrá definirle un Nombre, un Comentario, y de ser un problema de VRP la Capacidad del vehículo, en los cuadros de texto correspondientes. Más adelante en el manual se describirá con detalle la estructura del archivo de una instancia de datos.

### Descripción del algoritmo

En el sector inferior de la pestaña de Datos Instancia se encuentra un cuadro que contendrá información referente al último algoritmo que se selecciono para la ejecución de la instancia activa.

**Tour Inicial:**  
El tour inicial esta compuesto por el nodo inicial.

**Selección de Nodos:**  
Selecciona el nodo más cerca al CM de todos los nodos que aún no inicial.

**Inserción de Nodos:**  
Consiste en agregar el nodo seleccionado en los extremos del subtour formado hasta el momento de la ejecución.



## 12.3 Operaciones

Este capítulo está orientado a mostrar cómo se ejecutan las funciones de la herramienta, describiendo las instrucciones detalladamente para conocer el ambiente de trabajo en el cual se desenvuelve la aplicación.

Aquí se indica cómo cargar instancias ya definidas para el ambiente de prueba, creación de nuevos juegos de datos, edición, y ejecución en sus dos modalidades, entre otras funciones que provee el sistema.

Cada sección de este capítulo representa una operación e incluye:

- La descripción de la operación,
- Instrucciones precisas para ejecutarla,
- Imágenes de los cuadros de diálogo involucrados, y
- Advertencias, observaciones o excepciones cuando corresponde.

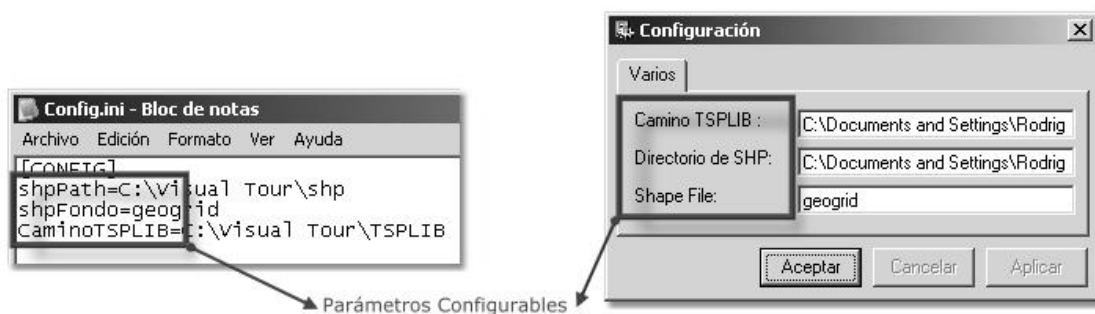
### Configuración

El sistema contiene tres parámetros que son utilizados para configurar cierta información necesaria para su correcto funcionamiento. Los mismos son ingresados al sistema a través de un archivo INI llamado Config.ini que debe estar situado siempre bajo el mismo camino que será donde se encuentra el ejecutable de la aplicación.

Dentro de la herramienta estos parámetros pueden ser modificados accediendo a esta funcionalidad presionando el comando de la Barra de Botones que aparece en la figura a continuación



La estructura de este archivo junto con la ventana de configuración es la que se presenta a continuación.



El archivo contiene tres parámetros configurables, estos son:

- *shpPath* Bajo el camino que se indique bajo este valor, es donde se deberán encontrar los archivos necesarios para la carga de los marcos necesarios utilizados por el componente **MapObject 2.1**.

Los archivos mencionados son los siguientes:

- GEOGRID.DBF
  - GEOGRID.SBN
  - GEOGRID.SBX
  - GEOGRID.SHP
  - GEOGRID.SHX
- *shpFonfo* Contiene el nombre de la estructura que permite la ubicación de figuras por su nombre dentro de la ventana de diseño, utilizada por el componente **MapObject 2.1**.
  - *CaminoTSPLIB* Este camino es el que será marcado por defecto en el momento que se quiera abrir o salvar una nueva instancia desde o hacia un archivo respectivamente dentro de la aplicación.

## Entrada y Salida de Datos

### Estructura de datos

El sistema Visual Tour permite al usuario cargar una instancia de datos a partir de un archivo, y salvar instancias que pueden ser definidas por el en tiempo de ejecución. Los archivos que el sistema reconoce son archivos de texto plano, y la información acerca de la instancia tiene una estructura definida.

### Formato del archivo

Cada archivo consta de una parte de especificación y otra parte de datos. La parte que concierne a la especificación contiene información acerca del formato de los datos y de su contenido. La parte de datos contiene explícitamente los datos.

### Parte de Especificación

Todas las entradas en esta sección tienen la forma *<clave> : <valor>*, donde *<clave>* denota un valor alfanumérico, y *<valor>* denota un valor tanto numérico como alfanumérico. Los términos *<string>*, *<integer>*, y *<real>* denotan una cadena de caracteres, enteros o un número real respectivamente. El orden de especificación de las claves es arbitrario, pero debe ser consistente.

A continuación vamos a enumerar todas las posibles claves.

- **NAME** : *<string>*  
Identifica la instancia de datos.
- **TYPE** : *<string>*  
Especifica el tipo de datos. Los dos únicos tipos de datos que interpreta **Visual Tour** son: *TSP* Traveling Salesman Problem  
*CVRP* Capacitated vehicle routing problem
- **COMMENT** : *<string>*  
Comentarios adicionales (Usualmente el nombre del creador de la instancia).
- **DIMENSION** : *<integer>*  
Para el **TSP** la dimensión es el número de sus nodos. Para el CVRP es el total de los nodos más el depósito.

- **CAPACIDAD** : *<integer>*  
Especifica la capacidad del vehículo en **CVRP**.
- **EDGE\_WEIGHT\_TYPE** : *<string>*  
Especifica como es dada la distancia (o los pesos de las conexiones entre nodos). El único valor que reconoce la herramienta es *EUC\_2*, en donde los pesos son distancias euclidianas en el plano.
- **NAME** : *<string>*  
Identifica la instancia de datos

### Parte de Datos

La información que debe ir en esta parte diferirá dependiendo en la selección del tipo de problema. De todas formas, esta información debe ir ubicada en la sección de la parte de *Datos* a continuación de la parte de *especificación*. Para indicar el final de esta sección se utiliza un identificador de fin de sección. Este identificador será EOF :

- **NODE\_COORD\_SECTION** :  
En esta sección son dadas las coordenadas de los nodos en forma de lista.  
*<integer> <real> <real>*
- **DEPOT\_SECTION**:  
Contiene el identificador correspondiente al depósito, debe tener el número -1 en la siguiente línea.
- **DEMAND\_SECTION** :  
La demanda de todos los nodos para el problema **CVRP** esta dado por la forma  
*<integer> <integer>*

El primer valor corresponde al identificador del nodo, el segundo es la demanda. El nodo depósito debe aparecer en esta lista, el valor de su demanda será de 0.

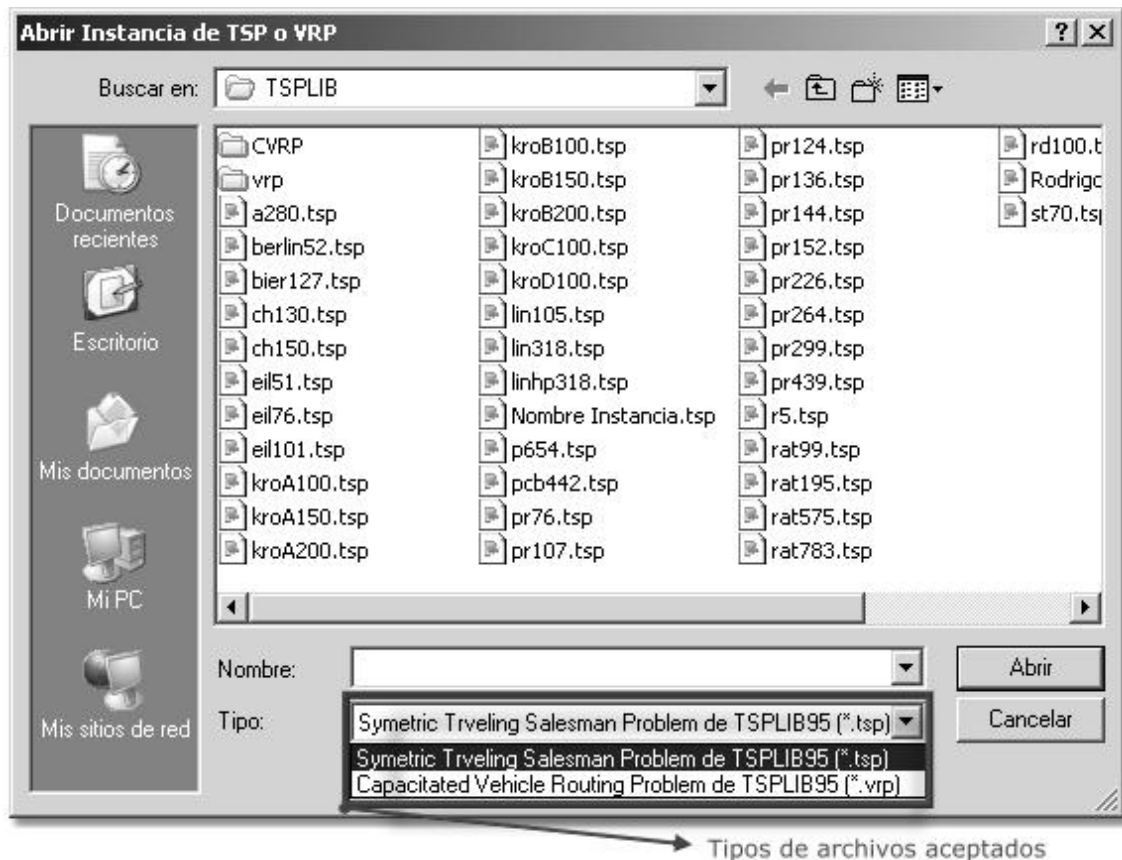
### Abrir una nueva Instancia

Para abrir una nueva instancia de datos se deberá presionar el comando de la Barra de Botones que aparece en la figura a continuación



A continuación se desplegará el cuadro de dialogo en el cual se deberá seleccionar el archivo que se desea cargar. El sistema provee la carga de dos formatos de datos posibles, estos concuerdan con el formato de datos que presentaron en la sección anterior. Indique uno de los dos tipos de datos como se muestra en la siguiente figura, y luego seleccione el archivo asociado a ese formato en el cuadro de dialogo.

Una vez que se presiona el botón Abrir, si la operación resulta satisfactoria, se desplegará en la ventana de diseño los nodos correspondientes a la instancia seleccionada. Además se cargarán otras estructuras como ser la tabla de nodos, que se describirán a más adelante en este manual.



### Guardar una Instancia

Es posible guardar juegos de datos, propuestos y diseñados en la ventana de diseño por el usuario, con el formato estructural de datos propuesto de manera detallada anteriormente.

Para guardar una instancia de datos se deberá presionar el comando de la Barra de Botones que aparece en la figura a continuación



Durante el proceso de guardar se abrirá una ventana análoga a la descrita en la sección Abrir una Nueva Instancia.

### Guardar un Resultado

Una vez ejecutado un algoritmo sobre algún juego de datos, el sistema provee la funcionalidad de guardar la información pertinente a esa instancia de ejecución bajo lo estructura de un archivo XML. Para otorgarle una mejor presentación visual a los datos contenidos en el XML, bajo la carpeta Reportes que se genera a partir de la instalación del sistema, se puede encontrar un archivo XSL que servirá para visualizar los datos contenidos en el XML con estilos de HTML.

Para poder generar el XML con los resultados de la ejecución se deberá presionar el comando de la Barra de Botones que aparece en la figura a continuación



Durante el proceso de guardar, se abrirá una ventana análoga a la descrita en la sección Abrir una Nueva Instancia, por la cual se realizara el salvado.

### Estructura del archivo XML generado

A continuación se mostrará la estructura de datos que se genera en el archivo XML para almacenar la información correspondiente a un a instancia de ejecución de un algoritmo sobre un respectivo juego de datos.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="reporte.xsl"?>
<Datos>
  <PathMapa>temp.jpg</PathMapa>
  <Nombre>Nombre Instancia</Nombre>
  <Comentario>Sin Comentario</Comentario>
  <Tipo>TSP</Tipo>
  <Dimension>3</Dimension>
  <TipoPeso>EUC_2</TipoPeso>
  <Capacidad>1</Capacidad>
  <Deposito>1</Deposito>
  <Algoritmo>
    <TourInicial>Descripción</TourInicial>
    <Seleccion>Descripción</Seleccion>
    <Insercion>Descripción</Insercion>
    <Ruteo/>
  </Algoritmo>
  <ColNodos>
    <Nodo>
      <NodoID>NodoID</NodoID>
      <CoordX>coordX</CoordX>
      <CoordY>coordY</CoordY>
      <Demanda>Demanda</Demanda>
    </Nodo>
  </ColNodos>
  <Rutas>
    <Ruta>
      <NodoID>NodoID</NodoID>
      <Distancia>Distancia</Distancia>
      <Abastecio>Cantidad Total abastecida para esta ruta</Abastecio>
    </Ruta>
  </Rutas>
</Datos>
    
```

Dentro de los tags de <ColNodos> se encuentra la lista con todos los nodos

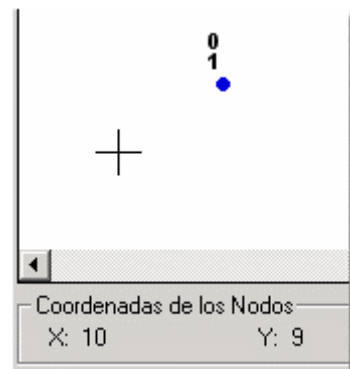
Dentro de los tags de <Rutas> se encuentra separadas por el tag <Ruta> las rutas

Aquí se colocan los nodos que generan la ruta

### Insertar un nodo

La inserción de nuevos nodos se realiza utilizando el botón izquierdo del Mouse. Para esto el puntero debe estar posicionado sobre la ventana de diseño, sobre la cual este adquirirá forma de cruz como se muestra a continuación.

En el sector inferior izquierdo de la ventana de la herramienta se muestran las coordenadas en las cuales se encuentra ubicado el puntero. En caso de realizar la inserción de un nodo, este se ubicará en las coordenadas indicadas en el momento de presionar el botón izquierdo del Mouse.



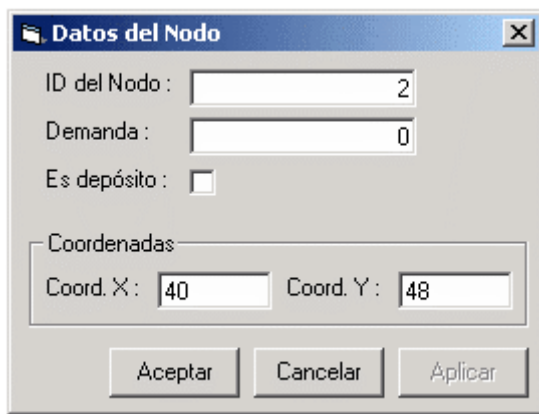


## Modificar y Eliminar Nodos

### Modificar un Nodo

Para modificar los parámetros de un nodo ya existente, se debe posicionar el puntero sobre este y presionar el botón derecho del Mouse o bien se puede hacer doble clic sobre el registro de un nodo ubicado en la tabla contenida en el marco de la pestaña de Clientes.

Una vez hecho esto se desplegará un Menú en el cual se mostrara la información correspondiente al nodo seleccionado y se permitirá editar tanto su ubicación, su demanda, y su condición de depósito. La siguiente imagen muestra la ventana correspondiente al Menú.



Posesionándose sobre la opción Datos del Nodo se accederá a otra ventana, como indica la figura de la izquierda, en la cual se podrán modificar los parámetros previamente mencionados.

Otra manera de desplegar esta ventana es haciendo doble clic con el botón de izquierdo del mouse sobre un registro, que contenga datos de un nodo, sobre la tabla de nodos de la pestaña Nodos.

### Eliminar un Nodo

Par eliminar un nodo, se debe seleccionar la opción Eliminar Nodo, en el menú de la figura. Una vez hecho esto el nodo será eliminado del sistema.



### Marcar a un nodo como depósito

Es posible seleccionar un nodo e indicar que este sea el depósito del sistema de nodos. Para poder hacerlo existen dos maneras, una de ellas es presionar el botón izquierdo del mouse sobre la opción Depósito del menú que se despliega al presionar el botón izquierdo del mouse sobre un nodo. La otra manera es a través de la ventana que se despliega al seleccionar Datos del Nodo en el mismo menú (ventana que se muestra en la figura de la sección Modificar un Nodo).

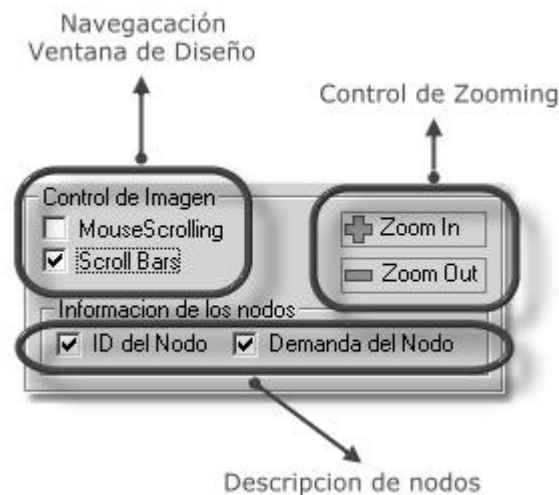


Cuando un nodo es indicado como depósito, tanto la opción Depósito de la ventana de la figura de la derecha como el check box Es depósito de la ventana Datos del Nodo serán remarcados con el símbolo de listo, como se puede ver en la figura.

En caso en que el nodo seleccionado corresponda al depósito del sistema de nodos, este se visualizará de manera diferente al resto asociándosele el icono de un triángulo rojo.

## Opciones de Visualización

Visual Tour esta compuesto de un conjunto de herramientas para la visualización, ubicados en el sector superior derecho de la interfaz bajo el marco Control de Imagen. Las funciones disponibles son las descritas a continuación.



- **Zoom In:**  
Permite realizar un acercamiento en la ventana de diseño, con una escala correspondiente a 1:2.
- **Zoom Out:**  
Análogo al Zoom In, realizando un alejamiento en la ventana de diseño, con una escala correspondiente a 2.
- **Mouse Scrolling:**  
Permite mover el mapa de la ventana de diseño manteniendo presionado el botón izquierdo del Mouse. Icono del cursor representado por una mano.
- **Información de los Nodos:**
  - **ID del Nodo:**  
Despliega en la ventana de diseño sobre cada nodo el identificador asociado a este.
  - **Demanda del Nodo:**  
Despliega en la ventana de diseño sobre cada nodo el valor de la demanda que esta tiene.

Para utilizar el Zoom, se debe primero presionar el botón correspondiente, tanto Zoom In como Zoom Out, y luego presionando el botón izquierdo del mouse cada vez sobre la Ventana de diseño se efectuará la respectiva operación escalar.



El icono que representará al cursor durante estos procesos corresponderá a una lupa.

Una vez que no desee hacer más uso de esta funcionalidad debe volver a presionar el botón del Zoom correspondiente para dar por terminado este proceso. Hasta no ser deselectiona esta modalidad no podrá efectuar otra operación sobre el sistema.



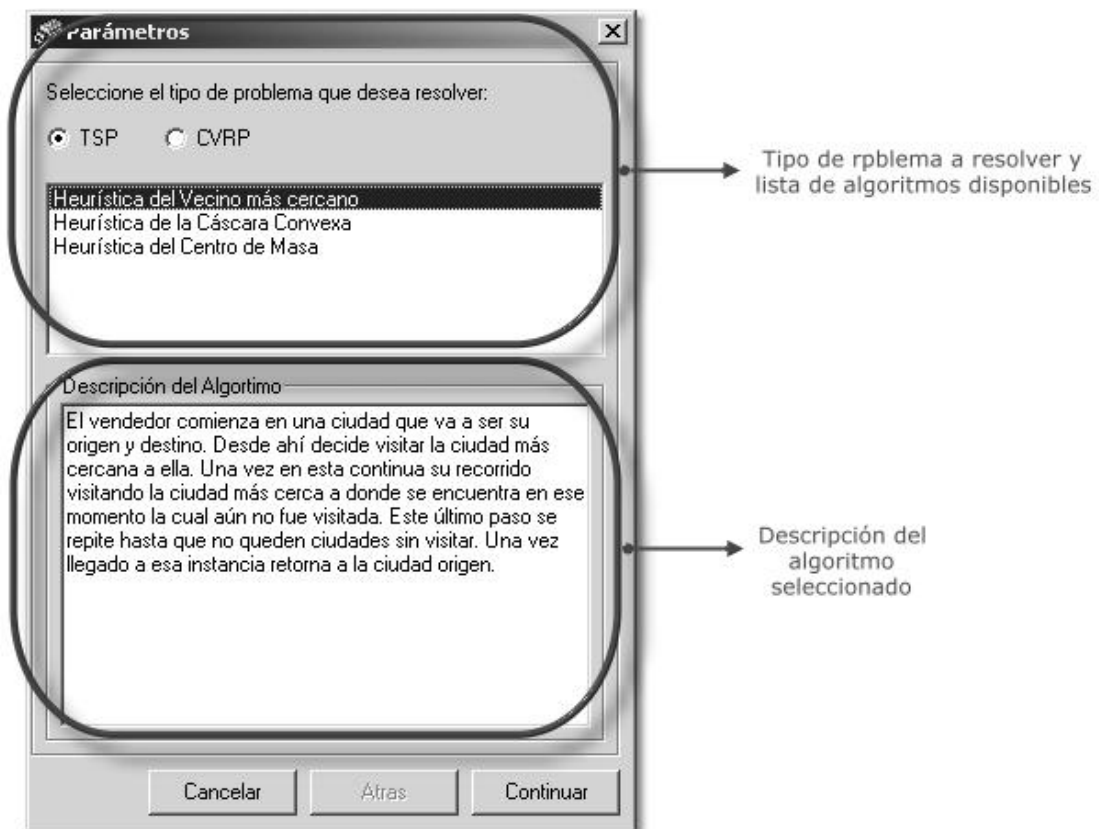
### Ejecutar un Algoritmo

Los algoritmos que están presentes en esta versión de la herramienta (Visual Tour Ver. 1.0) comprenden a los descritos en la documentación Nuevas Metaheurísticas basadas en la física para la resolución del TSP y del VRP (Ver. 2.0). En este documento se encuentran de forma detallada las descripciones de todos los criterios que se ofrecen para su uso en Visual Tour.

Para ejecutar un algoritmo debe seleccionar el comando Seleccionar Algoritmo de la barra de botones como muestra en la figura.



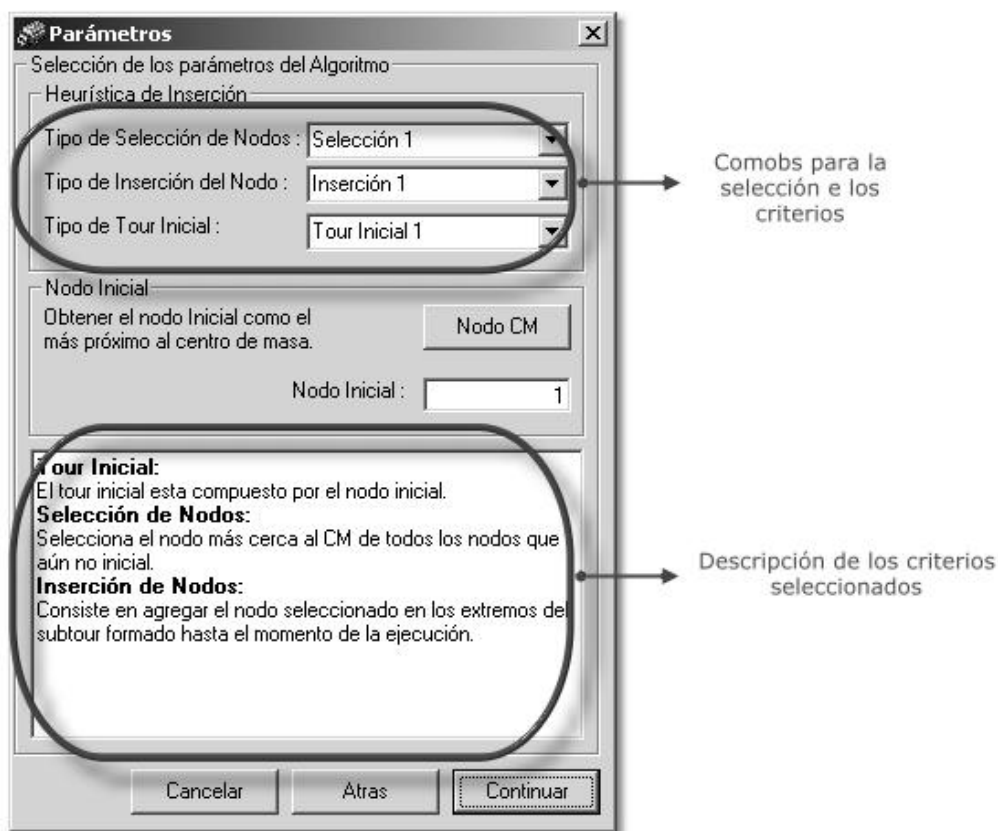
A continuación se desplegará una nueva ventana en la cual podrá seleccionar el tipo de problema que desea resolver.



Los dos problemas para los cuales existen algoritmos implementados son los correspondientes al TSP y al CVRP, para los cuales existe una lista de métodos que el usuario podrá seleccionar.

Una vez seleccionado el algoritmo se deberá presionar el botón de continuar que dará paso a la ventana que muestra la siguiente figura. Aquí el usuario podrá seleccionar los criterios que provee el algoritmo previamente seleccionado.

A continuación se puede ver la presentación estándar para la configuración de la ejecución de un algoritmo de TSP, más precisamente el de la Heurística del Centro de Masas.



Una vez seleccionados los parámetros correspondientes al algoritmo, se debe presionar el botón Continuar para que se proceda a la ejecución del mismo.

El tiempo de ejecución del algoritmo se desplegará en la barra de estado de la ventana de la aplicación. Una vez culminada la misma este se detendrá.

En cualquier momento de la selección es posible cancelar la acción de la misma o dirigirse al paso anterior utilizando los botones Cancelar o Atras respectivamente.

Cada algoritmo puede habilitar los criterios para los cuales este puede ser ejecutado, como deshabilitar aquellos que no sean compatibles con su modalidad. Inclusive es posible que algún algoritmo permita la configuración de un parámetro que sea propio de él, y solo será presentado cuando este sea seleccionado.

## Resultados

El resultado de la ejecución del algoritmo, va a estar compuesto por los caminos determinados junto con las distancias parciales entre cada nodo del mismo, la suma total de estas distancias, y una representación gráfica del recorrido determinado.

Los caminos determinados se desplegarán en una tabla que estará compuesta por los identificadores de los nodos que forman a este junto con la distancia que existe entre cada par de nodos sucesivos, en el o los caminos formados.

Esta tabla se encuentra bajo la pestaña de Rutas que se ubica en el extremo derecho de la ventana de la aplicación, como se muestra en la siguiente figura.

Inicio	Fin	Distancia
2	9	16,0312195418814
9	3	20,6155281280883
3	1	13,4536240470737
1	6	17
6	8	17,464249196573
8	5	55,9464029227975
5	4	18,3847763108502
4	7	30,8706980808663
7	10	3,16227766016838
10	2	77,0584193972339

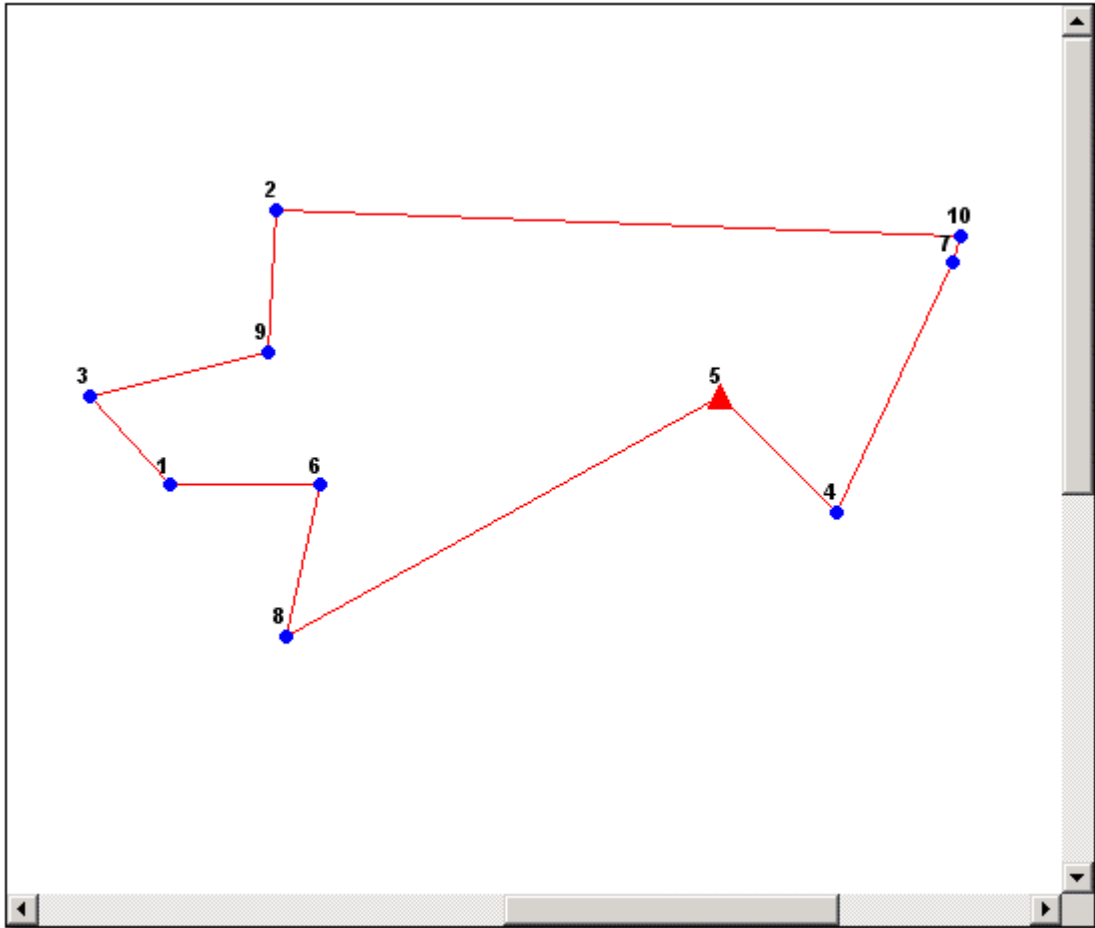
**Total : 269,987195285533**

Como se puede ver, en este caso el camino generado para una instancia del TSP, queda determinada por una tabla que indica por cada paso el nodo inicial y el final, junto con la distancia entre ambos. El camino determinado sería el formado por la concatenación de todos los caminos parciales de la tabla tomados desde el primer registro hasta el último.

Para el caso que se muestra en la figura el camino sería: **2-9-3-1-6-8-5-4-7-10-2**, en donde cabe destacar que el nodo inicial del camino es el identificado como 2.

El total de la distancia de todos los recorridos, se desplegará en la parte inferior de la tabla como muestra la figura de arriba.

La representación gráfica del recorrido se desplegará en la ventana de diseño. A continuación se muestra el grafo generado correspondiente a la tabla de resultados que mencionamos anteriormente.

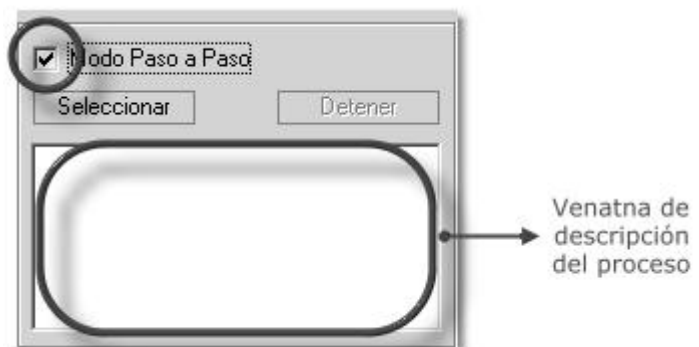


### Modo Paso A Paso

Bajo la Pestaña Datos Instancia del marco de la herramienta posicionado sobre la derecha de la misma, se encuentra el panel de comandos para la ejecución en modalidad Paso A Paso.

Esta modalidad permite ejecutar los algoritmos que se encuentren programados dentro de la lógica de la herramienta Visual Tour, bajo una modalidad en la cual la ejecución se irá efectuando a medida que el usuario lo indique.

Para utilizar esta modalidad primero se debe seleccionar el



check box del **Modo Paso a Paso** como se muestra en la figura de la izquierda. A partir de esto se habilitará el botón **Seleccionar** que permitirá la selección del algoritmo. La ventana que se desplegara será la correspondiente a la selección del mismo. Los pasos a seguir a partir de acá son los mismos que los de la sección de **Ejecutar un Algoritmo** descriptos en la sección anterior. Una vez hecho esto se visualizará el botón de **Comenzar** que permitirá el inicio de la ejecución del algoritmo.

Luego del inicio de la ejecución, el botón pasará a llamarse **Próximo paso**, y se deberá presionar tantas veces como pasos tenga el algoritmo que se esta ejecutando. A medida que se presione, se irá visualizando en la pantalla de diseño ejecución parcial del algoritmo, con una descripción del paso efectuado. En cualquier momento del proceso, el usuario podrá cancelar el mismo presionando el botón de **Detener**. Mientras se este en esta modalidad no se podrá insertar nuevos nodos, ni editar los ya existentes.

### Ejemplo

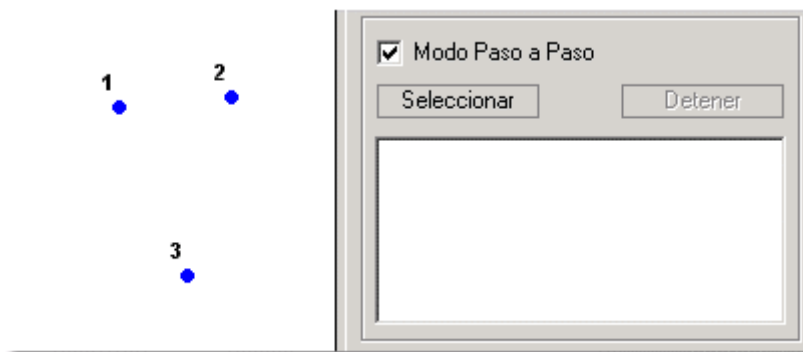
A modo de ejemplo se muestra un caso simple del proceso de ejecución de un algoritmo bajo esta modalidad. El problema a resolver consistirá en una instancia de TSP utilizando el algoritmo de la Heurística del Centro de Masas, con los siguientes parámetros:

Tipo de Selección de Nodos: **Selección 6**

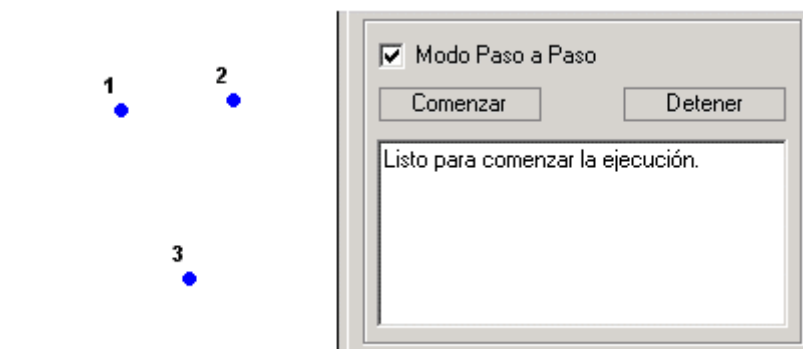
Tipo de inserción del Nodo: **Inserción 1**

Tipo de Tour Inicial: **Tour Inicial 1** (corresponde al nodo con Id. 1)

A continuación se puede ver el juego de datos que se utilizará para la ejecución.



En las siguientes figuras se muestra el proceso de ejecución a medida que se va presionando el botón de Próximo paso. El orden de aparición de las figuras corresponde al orden de ejecución.



The figure illustrates the step-by-step execution of a metaheuristic algorithm for the Traveling Salesman Problem (TSP) with three nodes (1, 2, 3). Each step is shown in a separate window with a status bar and a text area.

- Step 1:** The initial state shows three nodes (1, 2, 3) as blue dots. The status bar indicates "Modo Paso a Paso" is checked. The text area shows "El nodo inicial es: 1". Buttons for "Proximo paso" and "Detener" are visible.
- Step 2:** The center of mass of the visited nodes (1 and 2) is calculated, shown as an orange dot. The status bar indicates "Modo Paso a Paso" is checked. The text area shows "El centro de masa de los nodo no visitados es: 0". Buttons for "Proximo paso" and "Detener" are visible.
- Step 3:** The node closest to the center of mass is identified as node 2, shown as a green dot. The status bar indicates "Modo Paso a Paso" is checked. The text area shows "El nodo más cercano al centro de masa es: 2". Buttons for "Proximo paso" and "Detener" are visible.
- Step 4:** Node 2 is inserted into the sub-tour (1, 2), shown as a red line connecting nodes 1 and 2. The status bar indicates "Modo Paso a Paso" is checked. The text area shows "Inserto el nodo al subTour. 2 1". Buttons for "Proximo paso" and "Detener" are visible.

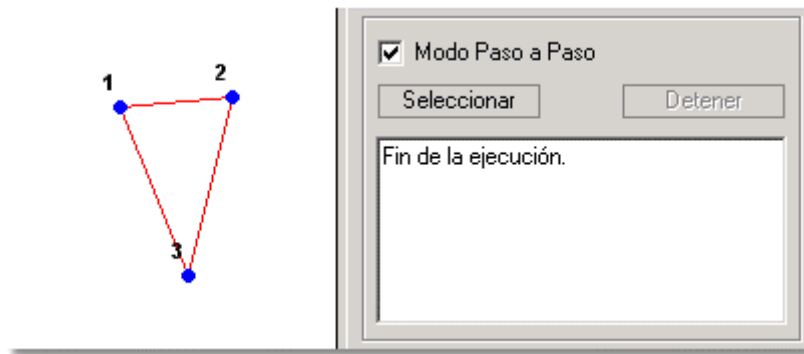
The diagram illustrates the insertion of node 3 into a subTour (1, 2) using a center-of-mass heuristic. The process is shown in four steps:

- Step 1:** Node 3 is blue. The subTour consists of nodes 1 and 2 connected by a red line.
- Step 2:** Node 3 is orange. The center of mass of the visited nodes (1 and 2) is calculated as 0.
- Step 3:** Node 3 is green. Node 3 is identified as the node closest to the center of mass.
- Step 4:** Node 3 is blue. Node 3 is inserted into the subTour, resulting in the subTour (1, 2, 3).

Each step includes a control panel with the following text:

- Modo Paso a Paso
- Proximo paso
- Detener
- Inserto el nodo al subTour. 2 1
- El centro de masa de los nodo no visitados es: 0
- El nodo más cercano al centro de masa es: 3
- Inserto el nodo al subTour. 2 1 3

Una vez finalizada la ejecución del algoritmo, la ventana de descripción de pasos desplegará el texto "Fin de la ejecución", indicando la finalización satisfactoria del proceso como se muestra en la figura a continuación.



La modalidad de Paso A Paso esta estrechamente ligada con la implementación del algoritmo, lo que quiere decir que es una funcionalidad la cual el programador del algoritmo debe proveer al mismo para que esta pueda ser utilizada una vez integrado esta a la herramienta. En el caso de los algoritmos que existen actualmente en esta versión de la herramienta (Visual Tour ver. 1.0), los algoritmos en su totalidad constan de esta funcionalidad.

## Reportes

Esta funcionalidad permite generar un reporte con los datos correspondientes a la ejecución de una instancia de datos. Para eso es necesario haber ejecutado uno de los algoritmos provistos sobre algún juego de datos.

El reporte generado es en formato HTML, y se generará a partir de un template de XSL sobre datos cargados en XML.

Información que contendrá el reporte:

- Datos pertinentes a la instancia
- Información correspondiente al algoritmo seleccionado
- Los nodos que la comprenden con sus coordenadas y sus respectivas demandas
- Los caminos generados a partir de la ejecución del algoritmo seleccionado
- Una imagen del sistema de nodos con las respectivas rutas que se formaron a partir del algoritmo seleccionado.

Para el caso del ejemplo utilizado en la modalidad **Paso A Paso** el diseño del reporte generado para el sería el siguiente.



## Reporte

### Información de la Instancia

**Nombre:** Nombre Instancia  
**Comentario:** Sin Comentario  
**Tipo:** TSP  
**Dimensión:** 3  
**Tipo de Peso:** TSP DIMENSION

### Información del algoritmo utilizado

**Tour Inicial:** El tour inicial esta compuesto por el nodo inicial.  
**Selección:** Selecciona el nodo más cerca al CM de todos los nodos que aún no inicial.  
**Inserción:** Consiste en agregar el nodo seleccionado en los extremos del subtour formado hasta el momento de la ejecución.

### Datos de las Rutas



ID Ruta	Distancia	Abasteció	Recorrido
1	42,697792544995	3	1, 2, 3

### Datos de los Nodos

Nodo ID	Coord. X	Coord. Y	Demanda
1	100	63	0
2	110	64	0
3	106	48	0

## Generación de Datos Aleatorios

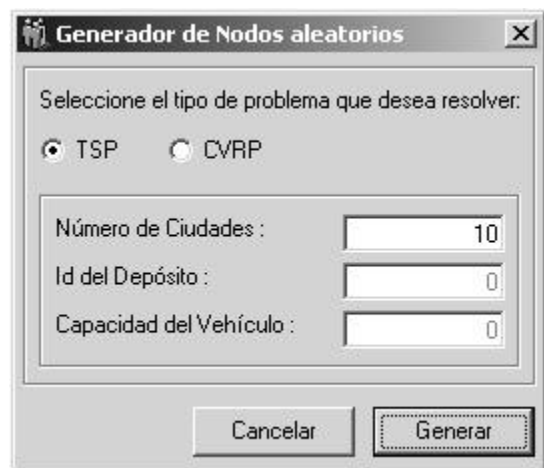
Es posible generar juegos de datos para la realización de pruebas sobre los algoritmos. Para ello solo se debe indicar el tipo de problema (TSP o VRP) para los cuales se desea que se generen los datos.

Para acceder a esta funcionalidad se debe presionar el comando de la Barra de Botones que aparece en la figura a continuación



Esta opción permite seleccionar diferentes parámetros en función del problema seleccionado:

- 1 Cantidad de Nodos a ser generados
- 2 TSP
  - a. Identificador del nodo inicial
- 3 VRP
  - a. Identificador del nodo que corresponderá al depósito. Capacidad del vehículo



La capacidad de carga que se le asigne al vehículo será determinante para la asignación de demandas a los nodos generados. Las mismas se determinarán a partir de valores aleatorios entre 1 y la capacidad seleccionada.

### **Limpiar Pantalla**

Esta función eliminará el contenido de la colección de nodos a su vez con la tabla de resultados, que se encuentren cargados en ese momento.

Para acceder a esta funcionalidad se debe presionar el comando de la **Barra de Botones** que aparece en la figura a continuación



## 13 Apendice 5: Algoritmos Genéticos

### 13.1 Introducción

Los algoritmos genéticos han aparecido recientemente en la historia de la resolución de problemas de optimización para los cuales, una solución exacta llevaría demasiado tiempo y / o recursos o simplemente no existen algoritmos exactos que permitan conocer una solución. Estos algoritmos tienen la característica de dar una solución tentativa con un costo menor que los algoritmos exactos (si es que los hay).

Los algoritmos genéticos abordan la resolución de problemas de optimización de complejidad elevada utilizando un mecanismo similar al que emplea la Naturaleza en el proceso de la selección natural: la reproducción, mediante la recombinación del material genético, para ir adaptándose a las condiciones del entorno. [\[IAG\]](#)

Aunque la aplicación de estos algoritmos en la actualidad no está muy difundida creemos que en un futuro sí, ya que el avance de la teoría se verá acompañado por el avance de la computación en lo que respecta a la capacidad de procesamiento. Además porque son un medio para resolver problemas de los cuales no sabemos mucho tras un mínimo análisis, o incluso una herramienta para realizar ese análisis.

### 13.2 Algoritmos genéticos vs. evolutivos

En distintos documentos se encuentran diferentes definiciones de ambos conceptos. Por mencionar algunos: [\[AGE\]](#)

- “los métodos que incorporan a un algoritmo genético conocimientos sobre el dominio se les denomina “evolutivos” mientras que el vocablo “genéticos” se reserva para los relacionados con los que no los incorporan”.
- “los algoritmos genéticos son una clase de algoritmos evolutivos”
- “algoritmos genéticos + estructuras de datos = programas evolutivos”
- “algoritmos genéticos se le denomina al conjunto de la teoría de los programas evolutivos”
- 

Dado que no existe, entonces, una clara definición o separación de ambos conceptos es que no los vamos a considerar como distintos. Nos referiremos a este tipo de algoritmos como algoritmos genéticos (AG) en general.

### 13.3 Descripción

En un paralelismo con su contrapartida biológica que obviamente es tan sólo esquemático, las soluciones tentativas del problema hacen el papel de los individuos sujetos a un proceso evolutivo. Una adecuada codificación (representación) simbólica de dichas soluciones cumplen a su vez la función del código genético. Las soluciones que se aproximan más a las que se puedan tener por óptimas se combinan entre sí (en similitud con el proceso reproductivo) para dar lugar a nuevas soluciones que heredan las características, tanto las buenas como las defectuosas, de aquellas de las que proceden. A esta combinación de individuos (padres) se le denomina cruzamiento y

genera uno o mas individuos hijos de la próxima generación. También se considera la creación de nuevos individuos a partir de un solo padre como una transformación del mismo. A esta manera de generar nuevos individuos se la denomina mutación. Los resultados emergen al cabo de una larga serie repetitiva de etapas (las generaciones) en las que los descendientes reemplazan a los progenitores. [\[AG1\]](#)

Para diseñar un algoritmo genético, entonces, se debe considerar una representación de las soluciones posibles en una cadena de símbolos. Estas cadenas representan el equivalente de los cromosomas en el modelo biológico (individuos). Se comienza con una población de soluciones posibles, que pueden ser generadas al azar o mediante algún método que produzca soluciones relativamente satisfactorias. A partir de este punto inicial, el algoritmo procede por etapas que ejecutan los siguientes pasos elementales:

1. Se evalúan todas las soluciones (individuos) de la población, con el fin de otorgar más probabilidades de cruzamiento a las más satisfactorias.
2. Mediante un mecanismo de selección, dando más oportunidades a las mejor evaluadas, se eligen las soluciones que han de cruzarse entre sí o mutarse para dar lugar a descendencia.
3. El cruce opera sobre las cadenas de los individuos de cada pareja de las soluciones elegidas como progenitores. A cada conjunto de progenitores se les aplica el operador de cruzamiento y se genera como salida el o los descendientes pertenecientes a la nueva generación.
4. La mutación opera sobre las cadenas de los individuos seleccionados para ello. A cada uno de ellos se les aplica el operador de mutación y se genera como salida el o los descendientes pertenecientes a la nueva generación.

Al repetirse la etapa descrita se van obteniendo soluciones que convergen hacia el óptimo de la función de evaluación empleada. Si la elección de la codificación del individuo, el criterio de selección, la función de cruzamiento y la función de mutación son acertadas, el proceso de convergencia resultará satisfactorio.

A modo de resumen, los algoritmos genéticos son técnicas de optimización que emulan los procesos de evolución natural. Estos algoritmos establecen primero una relación entre una solución concreta y su codificación; después se crea una población inicial de soluciones, y se la somete a procesos cíclicos de evaluación, selección, cruzamiento y mutación mejorando la calidad de las soluciones en las sucesivas generaciones hasta converger a una cierta solución.

## 13.4 Aplicación de un Algoritmo genético

Para utilizar un GA es necesario aportar ciertos componentes dependientes del problema a resolver, y exteriores al algoritmo en sí: [\[AGE\]](#)

1. La **representación** de cada solución tentativa (representación de individuos)
2. La **función de evaluación**. Se aplica a cada individuo para medir cuan factible es la solución.

La representación de las soluciones debe tener la capacidad de almacenar las buenas características de los individuos explorados. Esta capacidad se deberá de mantener en

las distintas generaciones que convergerán al óptimo buscado. El hecho de que estas características se mantengan (o se mejoren) de generación en generación es "responsabilidad" de las funciones de cruzamiento y / o mutación utilizadas para crear una nueva generación. Pero la capacidad de poder representar dichas características es "responsabilidad" de la codificación elegida.

La función de evaluación es la que (independientemente de que funciones de cruzamiento y / o mutación tengamos) me dará una medida de lo buena que es una solución. Esta función de evaluación se le aplicará a cada individuo de una generación y en base a ella se decidirá (con un criterio de selección) cuales serán los progenitores de la próxima generación. Si esta función no toma en cuenta alguna de las buenas características de un individuo es posible que a lo largo de las generaciones se extinga y esa característica no se vea en la solución final.

### 13.4.1 Restricciones de dominio

Se refieren al dominio de la función de evaluación. Es decir, si el resultado de una cruce o una mutación no genera un individuo que sea una potencial solución no le podrá ser aplicada la función de evaluación y por lo tanto no podrá participar en la selección de los progenitores para la próxima generación. [\[AGE\]](#)

Este problema se puede atacar de varias maneras:

1. **Usar una representación que no tenga restricciones.** En la codificación considerada no se puede representar un individuo que no sea una solución factible.
2. **Agregar un operador de reparación** después de los operadores de cruce y mutación para convertir los individuos no válidos en válidos.
3. **Asignarle un valor nulo a los candidatos que no sean válidos.** Esto se puede hacer con un filtro en el principio de la función de evaluación. El problema aparece si la probabilidad de que el AG genere individuos no válidos es alta. En ese caso se corre el riesgo de que la población útil sea muy pequeña y por lo tanto la posibilidad de que la variedad genética sea muy pobre. En el peor de los casos se podría extinguir la población válida de individuos.
4. **Operadores cerrados.** Si al proveerles soluciones válidas las funciones de cruce y de mutación solo pueden generar soluciones también válidas, y se inicializa la población con individuos válidos en su totalidad el problema queda solucionado. Generalmente es muy difícil encontrar operadores de cruzamiento y mutación cerrados logrando una variedad genética deseable. Por eso en la mayoría de los casos este método no se utiliza.

En general si es posible encontrar una representación a la cual no haya que agregarle restricciones (1) u operadores cerrados (4) que contengan las características deseables para el algoritmo se utilizan estos. Si no, se intenta con el punto 3. El problema es que la mayoría de las veces la probabilidad de que las funciones de cruzamiento y mutación generen individuos no válidos es alta. En otro caso se le agrega un operador de reparación a la salida de las funciones de cruzamiento y mutación.

### 13.4.2 Equilibrio entre la exploración y la explotación de soluciones

Al igual que en los procesos de selección natural, debe existir equilibrio entre la explotación de soluciones buenas (cruzamiento) y la búsqueda de nuevas posibilidades (mutación) para mantener una razonable diversidad en la población que evoluciona generación tras generación hasta alcanzar un punto de equilibrio que es o está muy cerca del óptimo absoluto del problema. [\[AG2\]](#) Si la exploración no es suficiente corremos el riesgo de quedarnos "estancados" en un máximo local. Este fenómeno es una patología de los algoritmos genéticos y se denomina "convergencia anticipada". Si la explotación no es la suficiente corremos el riesgo de no llegar al máximo buscado.

## 13.5 Aplicación de AG a TSP

El TSP es un problema que cumple con las características para ser resuelto con un algoritmo genético. Es un problema de optimización en el cual no se puede, en un tiempo razonable, explorar todas las soluciones posibles para determinar la óptima de manera exacta.

El TSP es un problema de los denominados NP-Complejos. El espacio de estados del problema crece de forma exponencial (o comparable a exponencial, como por ejemplo, factorial) ante incrementos lineales del número de elementos que intervienen en el problema.

TSP es, en el ámbito de los AG el ejemplo por excelencia de la utilidad de estos algoritmos ya que, logra en tiempos razonables soluciones óptimas o muy próximas a ellas.

### 13.5.1 Construcción del algoritmo

Existen muchas formas de implementar algoritmos genéticos para solucionar TSP.

Se enumerarán a continuación nueve items básicos en los que se puede componer la construcción del algoritmo genético: [\[AG2\]](#)

- **Representación de los individuos** Debe existir una relación explícita o implícita (que se pueda decodificar) entre una cadena de símbolos (individuo) y una solución posible del problema. En este caso, la representación elegida es la cadena de números que se corresponden con las ciudades a visitar. Para que el individuo sea factible (solución tentativa) debe corresponderse con un camino hamiltoniano.
- **Poblaciones y su tamaño.**- Los algoritmos genéticos pueden trabajar con una población única o varias poblaciones separadas (islas) y un sistema de migración, y pueden utilizar poblaciones de mayor o menor tamaño. En general, la diversidad aumenta con el número de islas y número total de individuos en la población, mientras que el costo de procesamiento aumenta también con esos dos factores. Se considerará una única población y un tamaño de población del orden de cinco a diez veces el número de ciudades.
- **Población inicial.**- El resultado final del proceso de optimización con algoritmos genéticos no es muy sensible a la calidad de las soluciones en la población inicial. Puede optarse por tomar soluciones aleatorias, aunque si se utiliza algún método heurístico para construir las soluciones iniciales se suele reducir el tiempo de convergencia hacia la solución óptima.

- **Evaluación.**- La selección natural opera bajo el principio de eliminar determinística o probabilísticamente a los individuos menos eficaces. Así pues, la evaluación es el elemento crítico de todo programa evolutivo que pretenda solucionar un problema de optimización complejo; para cada solución se debe poder calcular un costo asociado a la misma. En el caso TSP basta con definir el costo como la distancia total recorrida asociada a cada individuo.
- **Selección.**- Emulando la selección natural, los mejores individuos en cada generación (soluciones con menor costo) deben tener más probabilidades de sobrevivir para producir la siguiente generación. Se pueden eliminar los peores, utilizar una probabilidad de supervivencia inversamente proporcional al costo, etc. Se puede considerar también una selección elitista, asignándose una pequeña probabilidad de seleccionar adicionalmente el campeón absoluto (la solución de menor costo encontrada hasta el momento) aunque no pertenezca a la generación.
- **Cruzamiento.**- Es el que transfiere las características deseables de las soluciones encontradas de una generación a la siguiente. Los operadores de cruzamiento son aquellos que producen individuos viables de padres distintos viables; los hijos deben mantener alguna de las características de los padres y no degenerar en exceso en el proceso de cruzamiento.
- **Mutación.**- Es la que transfiere diversidad a la población. Los operadores de mutación alteran aleatoriamente los códigos genéticos (cadenas de símbolos que representan a los individuos) generando normalmente soluciones peores que las originales que son eliminadas probabilísticamente en el proceso de selección; sin embargo, en algunos casos la alteración genética es beneficiosa y produce un individuo mejorado cuya probabilidad de supervivencia aumenta extendiéndose rápidamente por la población.
- **Probabilidades de cruzamiento y mutación.**- En general se supone que la probabilidad de cruzamiento debe ser elevada ( $\geq 95\%$ ) y la de mutación reducida ( $\leq 5\%$ ) para mantener un buen equilibrio entre explotación y exploración de soluciones; sin embargo estas cifras son muy variables y dependen del problema y operadores utilizado. También se pueden considerar algoritmos adaptativos que vayan variando estas probabilidades durante la evolución de acuerdo a ciertas condiciones en los nuevos individuos.
- **Criterio de convergencia.**- Se considera una condición tal que cuando se cumpla se termine con la iteración y el mejor evaluado hasta el momento o de la última generación será la solución propuesta por el algoritmo. El criterio puede estar dado por el número de etapas, por el tiempo de ejecución o por la baja tasa de mejora de una generación con respecto a las anteriores.

### 13.5.2 Restricciones de dominio

La representación intuitiva de los individuos para el planteo de este problema es una cadena de números. Cada ciudad considerada tendrá un número asociado y el orden de la cadena dará el camino de la solución planteada por un individuo determinado.

En esta representación claramente se pueden "armar" cadenas no válidas de soluciones al problema. Por lo tanto utilizar una representación que no tenga restricciones no es el caso que consideraremos para solucionar las restricciones de dominio.

Encontrar operadores (de cruzamiento y mutación) cerrados en el dominio de soluciones no es sencillo manteniendo la diversidad genética. Esto es, que los operadores cerrados tienen una alta probabilidad de explotación pero no de exploración. El riesgo que se corre, entonces, es el de "convergencia anticipada". Por esto no consideraremos operadores cerrados para solucionar las restricciones de dominio.

Al asignarle un valor nulo a los candidatos que no sean válidos corremos un alto riesgo de extinción, ya que la probabilidad de que los operadores de cruzamiento y mutación generen individuos no válidos es alta.

Por lo tanto vamos a considerar, para solucionar el problema de las restricciones de dominio agregar un **operador de reparación**.

Imaginemos que todas las ciudades están conectadas con todas las demás. Si asignamos un número a cada ciudad, cualquier cadena formada por números no es una cadena válida. Por ejemplo, si tenemos 6 ciudades numeradas de la 1 a la 6.

153246 es una cadena válida, pero 121222 no es válida, ya que se repiten ciudades

Una solución a este problema es agregar un operador de reparación: dejamos que el algoritmo genere en cada posición un número cualquiera 1-6 pero interpretamos esos números de manera distinta según la posición de la cadena en la que se encuentren, por ejemplo, en la cadena: 136211 el primer dígito representa una ciudad del 1-6, es decir, la ciudad 1. El segundo dígito indica el número de ciudad, pero numeradas teniendo en cuenta sólo las ciudades que no se han utilizado ya. Como la ciudad 1 ya se ha usado, solo nos quedan las ciudades 2-3-4-5-6 y las numeramos así: 1-2-3-4-5. Como quedan valores disponibles (queda uno, el 6) asignamos esos valores restantes de forma cíclica, es decir, las ciudades restantes son 2-3-4-5-6 representadas por 1-2-3-4-5 y 6 como aparece un 3, ese valor representa la ciudad 4

Para el tercer dígito, hemos usado ya las ciudades 1 y 4 así que quedan libres las ciudades 2-3-5-6 las ciudades restantes son 2-3-5-6 representadas por 1-2-3-4 y 5-6 como el dato es un 6, hace referencia a la ciudad 3.

Con esto es evidente que el último de los datos es indiferente ya que no aporta información: solo queda una ciudad y ya se sabe a qué ciudad hay que ir.

Ahora supongamos que no todas las ciudades están conectadas con todas las demás. La solución es la misma: tener en cuenta en la codificación que el número almacenado en el gen representa una de las varias ciudades disponibles desde ese nodo, numeradas de forma circular. [AGTSP]



## 14 Bibliografía

Referencia	Información	Tema
[TSPLIB]	Librería de datos <a href="http://www.densis.fee.unicamp.br/~moscato/TSPBIB_home.html">http://www.densis.fee.unicamp.br/~moscato/TSPBIB_home.html</a>	Librería de datos
[MAT]	Matemáticas discreta y combinatoria – 3° edición Autor: Ralph P. Grimaldi Editorial: Addison-Wesley iberoamericana ISBN: 0-201-65376-1	Grafos y aplicación de Grafos a TSP
[FIS1]	Fundamentos de Física. Tercera edición Autor: Frank J. Blatt Editorial: Prentice Hall I.S.B.N.: 0-205-11784-8	Conceptos de Física
[FIS2]	Física Cuarta Edición Autores: Resnick, Halliday, Krane Editorial: CECSA I.S.B.N.: 0-471-80458-4	Conceptos de Física
[DICC]	Diccionario enciclopédico Editorial: Océano Grupo Editorial I.S.B.N.: 84-494-0188	Definiciones
[VRPTW]	" <i>Probabilistic Analyses and Practical Algorithms for the Vehicle Routing Problem with Time Windows</i> ". Julien Bramel y David Simchi-Levi September 1992 Revised: April 1993, July 1993	Información para TSP y VRP.
[TSPDT]	"A new Hybrid Heuristic for large Geometric Traveling Salesman Problems based on the Delaunay Trinagulation". Natalio Krasnogor, CeTAD, Universidad Nacional de La Plata, La Plata, Argentina. Pablo Moscazo, CeTAD, Universidad Nacional de La Plata, La Plata, Argentina. Michael G. Norman	Información para TSP y VRP.
[TSPAC]	" <i>Algoritmos basados en hormigas Inteligencia Articial Avanzada</i> " Dorigo, Colorni, Maniezzo, 1992. Página web: <a href="http://www.inf.utfsm.cl/~mcriff/IA-avanzada/hormigas-parte2.ps">http://www.inf.utfsm.cl/~mcriff/IA-avanzada/hormigas-parte2.ps</a> Último acceso: setiembre 2003	Información metaheurística para TSP. Algoritmo ACO.
[CH95]	" <i>The Vehicle Routing Problem and Local Search Metaheuristics</i> ", PhD thesis, Department of Engineering Science, The University of Auckland, 1995	Información TSP y VRP.
[RSV]	Routing and Scheduling of Vehicles and Crews. The State of the art. Lawrence Bodin, Bruce Golden, Arjang Assad, Michael Ball College of Business and management, University of Maryland at College Park, MD 20742 U.S.A.	Estado del arte, algoritmos y heurísticas para TSP y VRP.
[GCWW]	G. Clarke, W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points".	Algoritmo de ahorro de Clark y

	Operations Research, Vol 12, 568-581.	Wright
[TDEL]	"Heurísticas para el TSP-2D Euclídeo y Simétrico Basadas en la Triangulación de Delaunay y sus Subgrafos", Trabajo de Grado para la Licenciatura en Informática, UNLP, Abril 1998. N. Krasnogor. Directores: P. Moscato y G. Baum.	Triangulación de Delaunay
[DHMCE]	"Una Metaheurística Co-Evolutiva para el Problema del Viajante de Comercio" Diana Holstein, Pablo Moscazo Página web: <a href="http://www.densis.fee.unicamp.br/~moscato/papers/HolsteinThesis.pdf">http://www.densis.fee.unicamp.br/~moscato/papers/HolsteinThesis.pdf</a> Último acceso: setiembre 2003	Definición problemas NP.
[RSVSA]	<i>Routing And Scheduling Of Vehicles And Crews The state of Art</i> Lowrance Bodin, Bruce Golden, Arjang Assad and Micheal Ball College of Business and Management, University of Maryland at College Park.	Información para TSP y VRP.
[VRPTW]	"Probabilistic Analyses and Practical Algorithms for the Vehicle Routing Problem with Time Windows". Julien Bramel y and David Simchi-Levi September 1992 Revised: April 1993, July 1993	Información para TSP y VRP.
[TSPACO]	"Algoritmo ACO aplicado al TSP: Resumen de una experiencia práctica" Benjamín Arenas F., Alejandro Pavez S., Rodrigo Vidal K., Universidad Técnica Federico Santa María. Departamento de Informática.	Información metaheurística para TSP. Algoritmo ACO.
[CH95]	"The Vehicle Routing Problem and Local Search Metaheuristics", PhD thesis, Departament of Engineering Science, The University of Auckland, 1995	Información TSP y VRP.
[HIS]	<a href="http://www.math.princeton.edu/tsp/index.html">http://www.math.princeton.edu/tsp/index.html</a> <a href="http://www.math.princeton.edu/tsp/histmain.html">http://www.math.princeton.edu/tsp/histmain.html</a> Último acceso: setiembre 2003	Historia del TSP
[IAG]	<a href="http://personal1.iddeo.es/complexsystems/Geneticos.htm">http://personal1.iddeo.es/complexsystems/Geneticos.htm</a> Último acceso: setiembre 2003	Introducción algoritmos genéticos
[AGE]	<a href="http://indicart.com.ar/ga/home.html">http://indicart.com.ar/ga/home.html</a> <a href="http://indicart.com.ar/ga/epyga.htm">http://indicart.com.ar/ga/epyga.htm</a> Último acceso: setiembre 2003	Algoritmos genéticos y evolutivos
[AG1]	<a href="http://personal1.iddeo.es/complexsystems/Geneticos.htm">http://personal1.iddeo.es/complexsystems/Geneticos.htm</a> Último acceso: setiembre 2003	Algoritmos genéticos
[AG2]	<a href="http://ingenet.ulpgc.es/~ablesa/optimizacion/algoritmosgenet2.htm">http://ingenet.ulpgc.es/~ablesa/optimizacion/algoritmosgenet2.htm</a> Último acceso: setiembre 2003	Algoritmos genéticos
[AGTSP]	<a href="http://www.redcientifica.com/gaia/ce/ceno_c.htm">http://www.redcientifica.com/gaia/ce/ceno_c.htm</a> Último acceso: setiembre 2003	Algoritmos genéticos

		aplicados a TSP
[JEBOR]	Tomado de la página WEB del Imperial Collage: <a href="http://www.ms.ic.ac.uk/jeb/or/vrp.html">http://www.ms.ic.ac.uk/jeb/or/vrp.html</a> Último acceso: setiembre 2003	Información TSP y VRP
[SMAA]	" <i>TSP Algorithms in Action Animated Examples of Heuristics Algorithms</i> ". Stephan Martens <a href="http://itp.nat.uni-magdeburg.de/~mertens/TSP/index.html">http://itp.nat.uni-magdeburg.de/~mertens/TSP/index.html</a> Último acceso: setiembre 2003	Información TSP
[VRPNP]	Crossover and Diversity: A Study about GVR - <b>Pinhal de Marrocos</b> 3030 Coimbra, Portugal Jorge Tavares, Francisco B. Pereira, Penousal Machado, Ernesto Costa Página web: <a href="http://eden.dei.uc.pt/~jast/vrp/papers/vrp_adoro2003.pdf">http://eden.dei.uc.pt/~jast/vrp/papers/vrp_adoro2003.pdf</a> Último acceso: setiembre 2003	Definición VRP como problema NP-completo
[EDA]	Estructura de datos y algoritmos, Mark Allen Weiss Editorial Addison Wesley. ISBN 0-201-62571-7	Orden de algoritmo de búsqueda
[DSTSP]	Data strucutres of traveling salesman. M. L M. L. Fredman <sup>1</sup> , D. S. Johnson <sup>2</sup> , L. A. McGeoch <sup>3</sup> , G. Ostheimer <sup>4</sup> , Published version appears in J. ALGORITHMS 18 (1995), 432-479	Estructuras de datos para TSP