

El impacto de la asimetría en la resolución de problemas de distribución y rutasⁱ

Alejandro Rodríguez¹, Rubén Ruiz²

¹ Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Pza. Ferrándiz Carbonell, 2 03801 Alcoy, España arodriguez@doe.upv.es

² Grupo de Sistemas de Optimización Aplicada, Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Camino de Vera s/n, 46021 Valencia, España rruiz@eio.upv.es

Palabras clave: ATSP benchmark algoritmos asimétrico

1. Introducción y objetivos

El clásico problema del viajante de comercio (*TSP – Traveling Salesman Problem*) ha sido ampliamente estudiado en la literatura de Investigación Operativa. Esto se debe a varios motivos:

- Por un lado, su gran complejidad de resolución incluso para problemas aparentemente sencillos y de pequeñas dimensiones (la versión de decisión es un problema NP-Completo). El TSP es un gran reto para los investigadores. Desde hace décadas se ha dedicado un gran esfuerzo para desarrollar técnicas de resolución, procurando a cada paso mejorar los resultados obtenidos por técnicas anteriores: ya sea en términos de proximidad al óptimo, como en calidad de la solución frente a tiempos de computación necesarios.
- Por otro lado, es visible y de gran importancia la aplicación práctica del TSP en la resolución de problemas reales de dirección de operaciones y logística: secuenciación de operaciones, problemas de rutas, entre otros.

Sin embargo, al revisar la literatura científica y los algoritmos desarrollados, se detecta cierta contradicción entre ambos objetivos. Si bien muchos autores justifican la importancia de sus desarrollos en la aplicación práctica de los mismos, son pocos los que realmente tienen en consideración la particular complejidad y asimetría de la realidad, que en cambio, obvian y simplifican para testear sus heurísticas. Dicho de otro modo, la gran mayoría de los problemas reales de secuenciación y de rutas (por citar sólo estos dos tipos) requieren de estructuras de datos en forma de grafos dirigidos y asimétricos (*ATSP – Asymmetric Traveling Salesman Problem*), mientras que por el contrario, una abrumadora parte del esfuerzo y de la literatura se centra en el desarrollo de técnicas para la resolución del problema en entornos de grafos no dirigidos y simétricos (*symmetric TSP*). Existe un supuesto comúnmente aceptado de que un algoritmo o método que funcione bien para un entorno simétrico, lo hará también para un entorno asimétrico. En este trabajo pretendemos refutar este supuesto.

La justificación y el origen de este trabajo es precisamente analizar y cuantificar la importancia de la asimetría (propia de las redes de distribución reales) frente a los resultados obtenidos por las mejores técnicas y heurísticas de resolución conocidas para los problemas TSP y ATSP. ¿Son válidas las técnicas y heurísticas TSP en entornos asimétricos? ¿Cuál es el

impacto en términos de calidad de la solución (proximidad al óptimo) y tiempos de computación?.

2. Desarrollo del estudio

Para resolver las cuestiones anteriormente planteadas, este trabajo se ha organizado de la siguiente manera:

- En primer lugar, se hace una selección y una breve descripción de las principales técnicas y algoritmos seleccionados para este estudio: NN - *nearest neighbor algorithm* (1957), 2-Opt heuristic (1958), Lin and Kernighan (1973), Concorde TSP solver (2003), LKH Keld Helsgaun (1998).
- A continuación se describen los indicadores que se utilizarán en el estudio para valorar los resultados (valor de la función objetivo, distancia al óptimo y tiempo de computación). También se describe el ámbito de los experimentos y el procedimiento utilizado (generación de experimentos, *hardware* y *software* utilizado).
- Por último, se presentan y analizan los resultados obtenidos, haciendo una comparativa en función de las técnicas utilizadas, el tamaño del problema (número de nodos), la asimetría y su reformulación.

En las conclusiones de este trabajo se contesta a las cuestiones anteriormente planteadas, y se puede observar como algunas de las más reconocidas técnicas y heurísticas de resolución TSP se comportan de un modo ineficaz e ineficiente frente a entornos reales con asimetría. Así mismo, se reconoce y se mide la bondad de la heurística LKH para diferentes grados de tamaño y asimetría del problema.

2.1. Algoritmos para la resolución del TSP

En este apartado se describen brevemente los algoritmos seleccionados para realizar nuestro estudio comparativo. Para realizar esta selección, fue necesaria una revisión bibliográfica y del estado del arte. De la ingente cantidad de algoritmos que aparecen en la literatura se seleccionaron aquellos de mayor reconocimiento científico (número de referencias científicas, posición en *benchmarks* anteriores) y bondad de las soluciones obtenidas en instancias científicamente reconocidas (óptimo en problemas de grandes dimensiones). También se ha procurado recoger en la selección (de heurísticas) un espectro desde las técnicas más simples y clásicas hasta las más actuales y avanzadas (Rosenkrantz et al. 1977). Este estudio no pretende hacer una comparativa exhaustiva de todas las técnicas existentes, sino más bien cuantificar y validar las soluciones de los algoritmos más reconocidos frente a la asimetría de la matriz de distancias.

NN - *nearest neighbor algorithm* (1957)

El algoritmo del vecino más próximo (*nearest neighbor*) es una heurística voraz (*greedy*) sencilla. El algoritmo comienza con una ruta formada por una secuencia de nodos elegidos aleatoriamente, a continuación añade al último nodo de la ruta el nodo más cercano no incluido todavía. El algoritmo se detiene al completar todo el conjunto de nodos. Una extensión a este algoritmo repetir el proceso tomando como nodo inicial cada uno de los nodos de la matriz y devolver la mejor de las n rutas generadas (*repetitive nearest neighbor*). Esta heurística constructiva es realmente rápida y fácil de implementar. Podríamos decir que se trata de la heurística más sencilla de todas.

2-Opt heuristic (1958)

Esta técnica pertenece a un conjunto de técnicas sencillas de búsqueda local, cuyo objetivo es mejorar la solución inicialmente generada (Rego 2002). La idea que hay detrás de esta heurística es la de definir una estructura de vecindad para el conjunto de todas las rutas admisibles. Una ruta t' es vecina de otra ruta t si t' puede ser obtenida a partir de t mediante la sustitución de k arcos (borrado y reemplazo de k arcos factibles, movimiento k -Opt). En dicha estructura, la ruta puede ser mejorada iterativamente desde su solución hasta la mejor solución vecina, hasta que ya no sea posible mejorar más. El resultado es una ruta óptima local llamada *k-optimal*. En la práctica existen las variantes 2-Opt (Croes 1958) y 3-Opt (Lin 1965).

Lin and Kernighan (1973)

A diferencia de la técnica anterior, la heurística (Lin y Kernighan 1973) no usa siempre el mismo valor de k , sino que a cada movimiento k -Opt intenta encontrar el mejor valor de k . La heurística irá construyendo una secuencia de 2-Opt movimientos comprobando a cada paso donde detenerse. Entonces se tomará la secuencia que mayor mejora proporcione (equivalente a un movimiento k -Opt con k variable). El óptimo local se alcanza cuantos más movimientos se realicen. Si además se utiliza una técnica de marcha atrás o *backtracking*, se puede encontrar el óptimo, aunque para ello se requiere mucho tiempo de computación. Es por ello, por lo que sólo se permite cierto retroceso en el proceso de optimización que finalmente puede dar lugar a un óptimo local o a un óptimo global.

Concorde TSP solver (2003)

En realidad Concorde* es un *solver* para resolver el TSP. Fue desarrollado por David Applegate, Robert E. Bixby, Vašek Chvátal, y William J. Cook (2002). Está especialmente concebido para la resolución de problemas TSP simétricos. Concorde ha demostrado en multitud de instancias su capacidad para la resolución proporcionando el óptimo del problema. En mayo del 2004, por ejemplo, Concorde se utilizó para resolver óptimamente un problema TSP de 24.978 ciudades de Suecia. La mayor instancia de problema de viajante de comercio resuelta hasta el momento, consiste en una aplicación VLSI (método para la producción de chips personalizados mediante conexiones realizadas por láser) de 85.900 localizaciones que surgió en los *Bell Laboratories* a finales de 1980. Mediante Concorde se encontró la ruta más corta posible con una longitud de 142.382.261 unidades. A esta instancia se le conoce como pla85900 y está publicada en el repositorio de TSPLIB de Gerd Reinelt y en Applegate et al. (2006). En este problema, los costes fueron modelados como distancias euclidianas redondeadas al valor entero superior más cercano.

LKH Keld Helsgaun (1998)

LKH es una aplicación de la heurística Lin-Kernighan para resolver el problema de viajante de comercio. Diversos experimentos computacionales han demostrado que LKH es muy eficaz. Aunque el algoritmo LKH es heurístico, las soluciones óptimas se producen con una impresionante alta frecuencia (Helsgaun 2000). LKH ha producido soluciones óptimas para todos los problemas que sus investigadores han podido obtener, incluyendo la instancia pla85900. En la actualidad se está atacando un problema de 1.904.711 ciudades (1,9 millones) repartidas por todo el planeta. El 9 de agosto de 2006, Keld Helsgaun usando una variante de

* <http://www.tsp.gatech.edu/concorde.html>

su algoritmo heurístico LKH, consiguió encontrar una buena solución (próximo al óptimo) con un resultado de la función objetivo igual a 7.516.043.230. Recientemente, en noviembre de 2008 volvió a mejorar su marca con una distancia de 7.515.947.511*. LKH ostenta actualmente el récord de las instancias de problemas con óptimos desconocidos (*DIMACS TSP Challenge 2002*) cuyos tamaños van desde los 1.000 a los 10.000.000 nodos. Cabe mencionar que los autores de este artículo (Rodríguez, A. y Ruiz, R.) durante el desarrollo de los experimentos que se presentarán a continuación, detectaron y ayudaron a Keld Helsgaun a resolver algunos *bugs* (pequeños fallos de programación) en la última versión disponible de su algoritmo.

2.2. Indicadores y experimentos

En este apartado, se describen brevemente los indicadores que se han utilizado en este estudio. A continuación se explicará el conjunto de experimentos realizados y su procedimiento.

Indicadores

- Distancia total de la ruta: es el resultado de la función objetivo, todos los algoritmos tratan de minimizar la distancia total del recorrido. La solución óptima tendrá la menor distancia posible de todas las soluciones. Recordemos que el número de soluciones posibles en el TSP es igual al factorial del número de nodos. Es decir, un número exponencial.
- Incremento sobre el óptimo: se utilizará este indicador para medir la bondad de las soluciones. El incremento (expresado en porcentaje) es la diferencia de distancias $\frac{d_s - d^*}{d^*} \cdot 100$ siendo $d^* \leq d_s$. Si la solución es óptima el incremento será del 0%, a mayor incremento, peor será la calidad de la solución (más alejada del óptimo).
- Tiempo de computación: se calculará el promedio de los tiempos de computación de los experimentos para valorar la rapidez con la que se ejecuta el algoritmo. En este estudio se intenta valorar si la calidad de la solución compensa los tiempos de computación necesarios. Se espera encontrar a su vez, una relación entre los tiempos de computación y el tamaño del problema a resolver (número de nodos n).

Experimentos

Se generó un conjunto de experimentos (aleatoriamente) cuyo rango va desde problemas de 100 nodos hasta 500 nodos (con 50 nodos de intervalo). Para todos los casos se generaron dos variantes iniciales: matrices simétricas (donde $d_{ij} = d_{ji}$) y asimétricas (transformando la matriz simétrica para que no se cumpla la igualdad anterior). Así mismo, se generaron otras dos variantes: sin reformulación de la matriz, y con reformulación (véase más abajo). En definitiva se generó un máximo de 36 escenarios de experimentación diferentes (se eliminarán los no posibles). El propósito de estos escenarios es analizar las siguientes cuestiones: ¿cómo se comportan los diferentes algoritmos (en términos de calidad de la solución y tiempos de computación) frente al tamaño del problema? ¿y respecto a la simetría de la matriz? ¿si se transforma una matriz asimétrica en simétrica mediante su reformulación, se observa algún cambio en el comportamiento?.

* Para más información sobre este reto computacional visite <http://www.tsp.gatech.edu/world>

Reformulación de la matriz de distancias

Es útil en ocasiones, transformar la matriz de distancias D de un TSP en otra matriz diferente D' pero que proporciona la misma solución óptima. Esta transformación se puede realizar, por ejemplo, mediante el empleo de constantes multiplicativas y aditivas a los elementos de la matriz d_{ij} . Una transformación diferente consiste en la reformulación de una matriz asimétrica a una matriz simétrica. Esto se puede lograr duplicando el número de nodos (Jonker, R. y Volgenant, T. 1983). Tal y como se muestra en la ecuación (1), para cada nodo existente es necesario añadir otro nodo virtual.

$$\begin{pmatrix} 0 & d_{12} & d_{13} \\ d_{21} & 0 & d_{23} \\ d_{31} & d_{32} & 0 \end{pmatrix} \Leftrightarrow \begin{pmatrix} 0 & \infty & \infty & -\infty & d_{21} & d_{31} \\ \infty & 0 & \infty & d_{12} & -\infty & d_{32} \\ \infty & \infty & 0 & d_{13} & d_{23} & -\infty \\ -\infty & d_{12} & d_{13} & 0 & \infty & \infty \\ d_{21} & -\infty & d_{23} & \infty & 0 & \infty \\ d_{31} & d_{32} & -\infty & \infty & \infty & 0 \end{pmatrix} \quad (1)$$

Las distancias entre cada nodo y su correspondiente nodo virtual debe ser un valor muy pequeño ($-\infty$). Esto hará que en la ruta solución, aparezcan pares de nodos reales y virtuales. Las distancias reales serán usadas entre los nodos reales, y entre los nodos virtuales (distancias de ida al nodo, y de vuelta desde el nodo respectivamente). El resto de distancias se penalizan utilizando un valor lo suficientemente grande ($+\infty$). En la expresión (1) se puede observar una matriz asimétrica (a la izquierda) y una matriz simétrica equivalente (a la derecha). De este modo se puede resolver con técnicas TSP (por ejemplo Concorde) diseñadas pensando en matrices simétricas, problemas ATSP que en realidad tienen matrices asimétricas. Una vez resuelto el TSP, se deben eliminar de la ruta solución TSP los nodos virtuales para tener la verdadera solución al ATSP. Se debe tener en cuenta que en ocasiones la solución óptima ATSP es la ruta inversa al TSP (si los nodos virtuales aparecen antes que su correspondiente nodo real).

En este trabajo, se utiliza esta técnica de reformulación para comparar el comportamiento de las diferentes heurísticas, enfrentándolas a escenarios de matrices simétricas y asimétricas.

Proceso e infraestructura

Los experimentos fueron generados y resueltos con un *software* diseñado para tal efecto (Figura 1). Dicho *software* fue ejecutado en un ordenador con procesadores Intel Core 2 Duo 2,4 GHz y 2 GB de memoria RAM. Los algoritmos (especialmente Concorde y LKH que son librerías externas) fueron integrados en el software mediante *scripts*. Para algunas transformaciones de datos y el análisis estadístico de los resultados se utilizó *R Project*^{*}. *R* es un entorno de trabajo muy amplio y un lenguaje de programación para el cálculo estadístico y trabajo gráfico. Es un proyecto *GNU* similar al lenguaje *S* desarrollado en los *Bell Laboratories* (actualmente *Lucent Technologies*).

Esta infraestructura puede ser adaptada y utilizada fácilmente para generar diferentes experimentos y evaluar nuevos escenarios, por ejemplo para valorar otras heurísticas.

^{*} <http://www.r-project.org/>

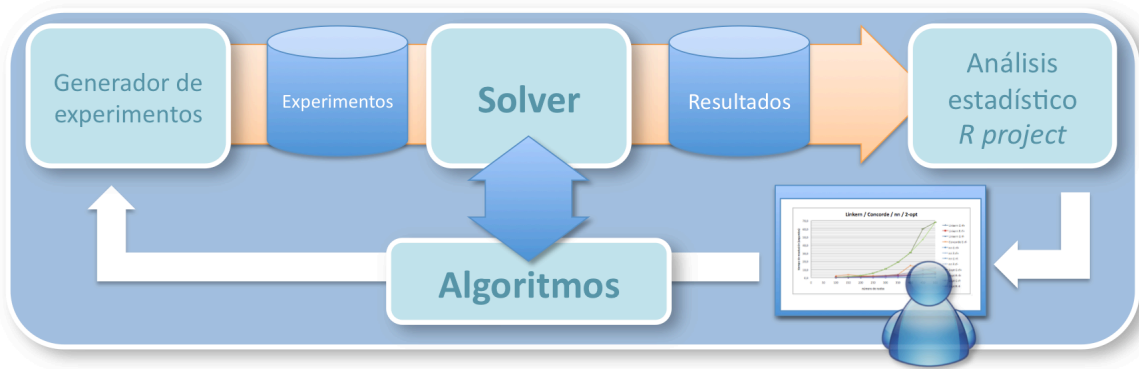


Figura 1. Proceso de generación, resolución y análisis de los experimentos.

3. Resultados y análisis

En este apartado se presentan y describen los principales resultados obtenidos en términos de tiempos de computación y calidad de la solución. Nótese que en la leyenda se indica la heurística utilizada, el tipo de matriz del problema (G simétrica, R asimétrica) y la reformulación (rf+ matriz reformulada, rf- matriz no reformulada).

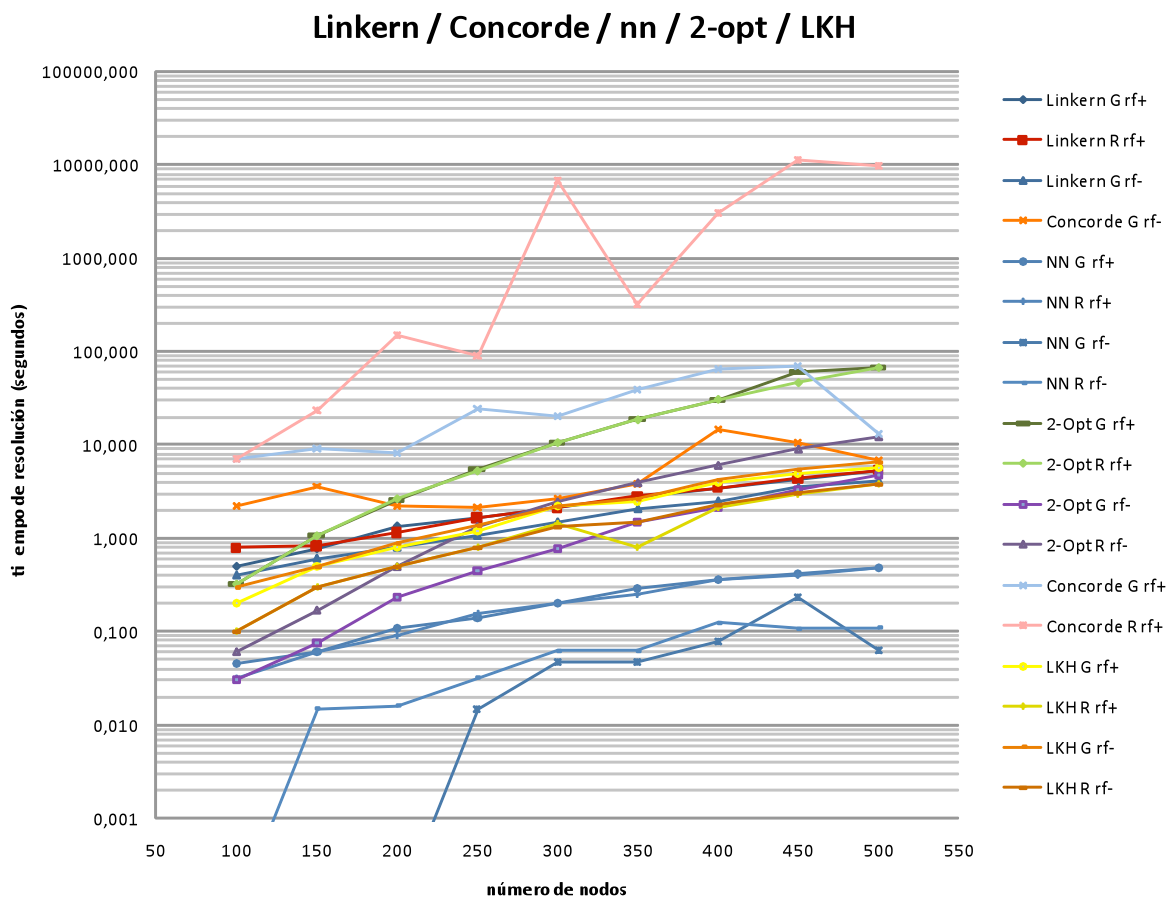


Figura 2. Tiempos de resolución (segundos) para cada experimento. Nótese la escala logarítmica.

Tiempos de computación

A continuación (Figura 2) se muestra la estadística de tiempos de resolución consumidos para cada escenario. Se puede observar cómo a medida que aumenta la complejidad del problema se requiere más tiempo de computación.

El tiempo de computación oscila entre los microsegundos hasta las 2,7 horas (para el peor de los escenarios). Se puede observar cómo la técnica más sencilla (NN - *nearest neighbor algorithm*) consume muy poco tiempo, frente a Concorde R rf+ en el otro extremo. Si se discriminan los tiempos de Concorde R rf+ (Figura 3), destaca el excelente comportamiento de LKH R rf- (3,8 seg.) frente a 2-Opt, e incluso frente a Concorde G rf- (7 seg).

En otras palabras LKH es una técnica muy rápida capaz de resolver problemas grandes y asimétricos ATSP sin necesidad de reformulación, en la mitad de tiempo en la que Concorde es capaz de encontrar el óptimo a un problema de tamaño equivalente pero simétrico TSP. Veamos ahora qué implicaciones tiene este estudio en términos de calidad de la solución.

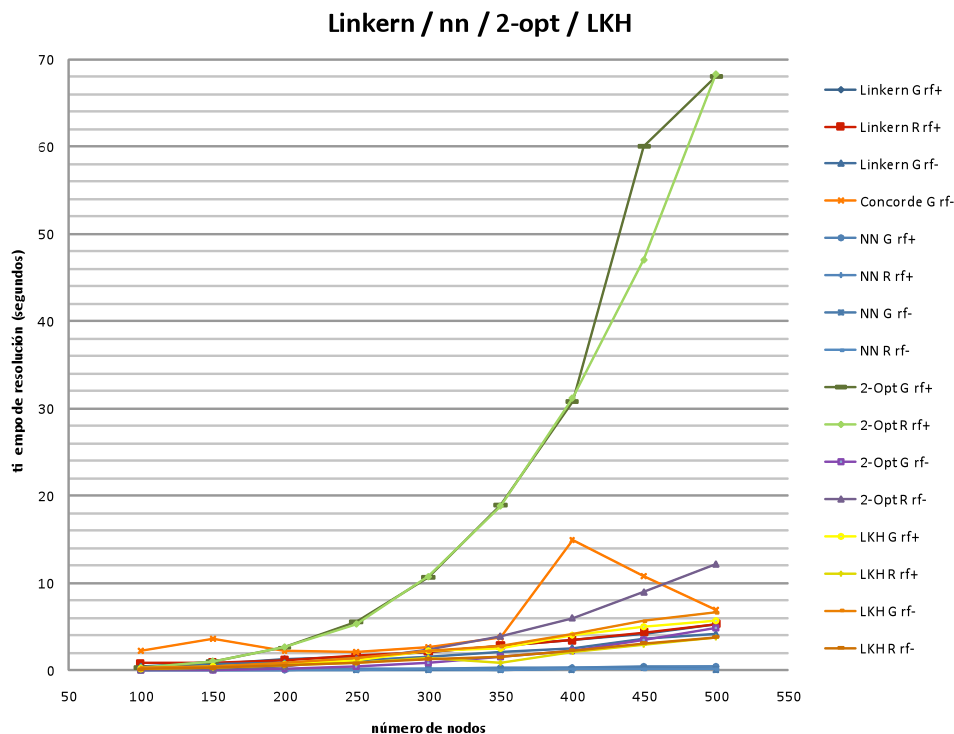


Figura 3. Tiempos de resolución (segundos) para cada experimento, excepto solver Concorde.

Calidad de la solución

Lo primero que hay que subrayar (Figura 4), es que Concorde es una técnica capaz de encontrar el óptimo para TSP (simétricos) en tiempos pequeños (del orden de menos de 10 segundos) para todos los casos. No así para los ATSP, aunque en estos casos Concorde también encuentra el óptimo en todos los experimentos (Figura 5), requiere de una reformulación (ATSP \Rightarrow TSP) y esto hace caer la eficiencia computacional de Concorde (que requiere de hasta 2,7 horas). En otras palabras, Concorde al afrontar un ATSP reformulado es del orden del 318% al 136.822% más lento. Concorde está diseñado pensando en problemas TSP simétricos, y la reformulación para poder resolver problemas ATSP es viable pero afecta

de manera terminal la eficiencia de Concorde (Figura 2). No ocurre así con LKH; que según nuestros experimentos, es una técnica extremadamente rápida ($< 3,8$ seg.) incluso con problemas grandes (LKH R rf- n500 en la Figura 3), que además se comporta muy bien en términos de calidad de la solución (Figuras 4 y 5). LKH R rf- demuestra que LKH es capaz de resolver ATSP sin reformular muy rápidamente, y que además proporciona soluciones muy próximas al óptimo ($< 0,5\%$) en todos los casos.

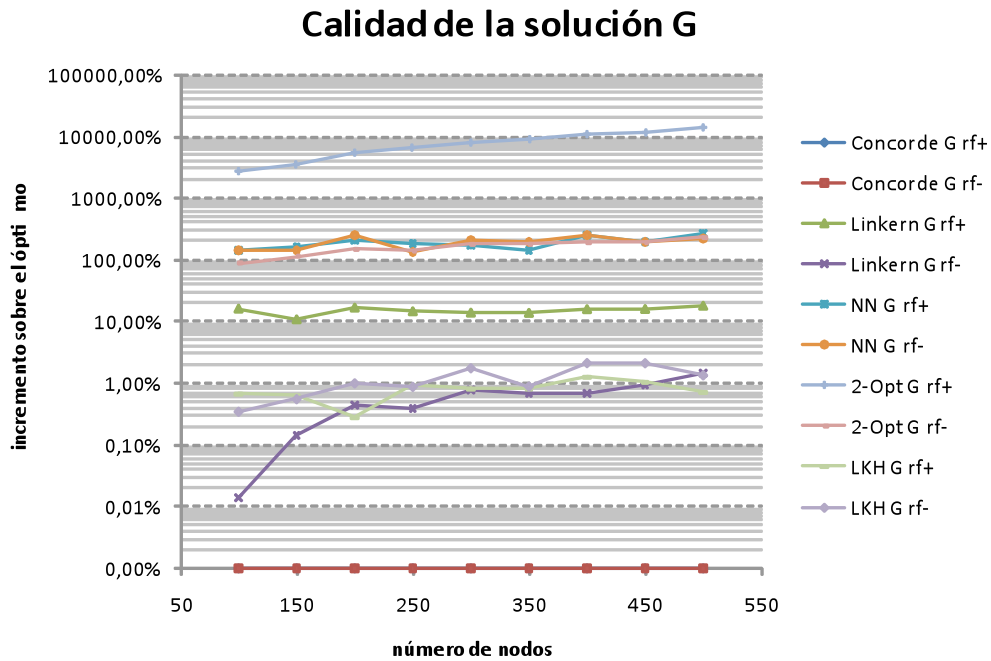


Figura 4. Calidad de las soluciones para cada experimento (matriz G simétricas).

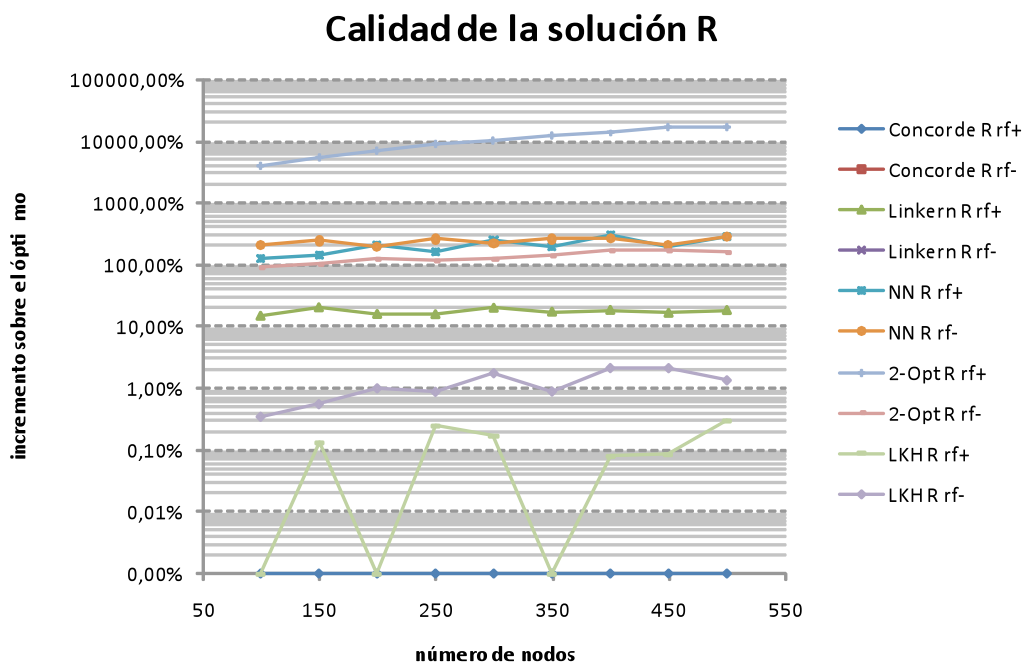


Figura 5. Calidad de las soluciones para cada experimento (matrices R asimétricas).

Otra de las conclusiones a las que hemos llegado, es que si bien las técnicas más sencillas (nn, 2-Opt) son muy rápidas, la calidad de las soluciones obtenidas en una primera pasada es deficiente. Linkern es una técnica muy rápida pero que sólo funciona con TSP.

4. Aplicación de los resultados en un problema real

Como apunte final, queremos señalar que las conclusiones obtenidas en este trabajo están siendo aplicadas en el desarrollo del *software* comercial llamado *RoutingMaps* para el cálculo y optimización de rutas y la gestión de flotas de vehículos (Rodríguez 2007).

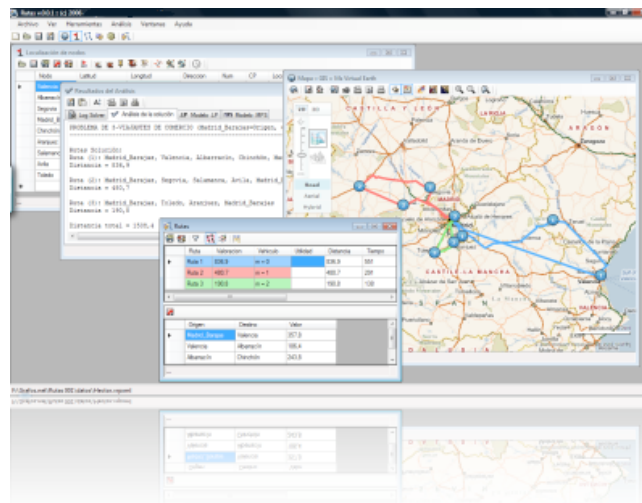


Figura 6. Pantalla del software *RoutingMaps* (<http://soa.iti.es>).

Estos experimentos están sirviendo para dotar al desarrollo de las mejores técnicas existentes en la literatura científica para la resolución de un espectro de problemas desde TSP a diferentes variantes CVRP, como se puede ver en un capítulo de libro de reciente publicación, (Ángel Juan, Javier Faulín, Rubén Ruiz y otros, 2009).

5. Conclusiones

En este artículo se ha presentado el resultado de un conjunto de experimentos con el objeto de averiguar el comportamiento de diferentes técnicas heurísticas para la resolución de TSP y ATSP. Nuestros experimentos indican que si bien Concorde es una técnica que proporciona el óptimo incluso en problemas grandes, y muy rápidamente, ésta deja de ser eficiente cuando el problema es asimétrico. Así mismo, si lo que se pretende es tener una técnica flexible, eficiente y con gran calidad en las soluciones, nuestros experimentos indican que LKH es la elección adecuada.

En la actualidad, estamos diseñando y ejecutando nuevos experimentos, extendiendo el ámbito de este estudio al contexto real de los problemas de rutas de vehículos VRP. Podemos adelantar que nuestros próximos experimentos estarán basados en matrices reales con diferente grado de asimetría (construidas no aleatoriamente sino a partir de información GIS y un conjunto de geo-localizaciones). Queremos extender el estudio para comprender en qué medida el territorio y las redes reales de comunicaciones (transporte por carretera) afectan a las técnicas antes mencionadas. Estamos así mismo investigando y desarrollando un *framework* que ayude a obtener las matrices de distancias reales a partir de un GIS de un modo muy rápido incluso para un gran número de localizaciones.

Referencias

- Applegate, D.; Cook, W.; Dash, S.; Rohe, A. (2002). Solution of a min-max vehicle routing problem. *INFORMS Journal on Computing* 14 (2): 132–143
- Applegate, D.; Cook, W.; Rohe, A. (2003). Chained Lin-Kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1): 82-92.
- Applegate, D.; L., Bixby, R. E.; Chvátal, V. y Cook, W. J. (2006). *The Traveling Salesman Problem: A computational study*. Princeton University Press.
- Croes, G.A. (1958). A method for solving traveling-salesman problems. *Operations Research*, 6(6): 791-812.
- Helsgaun, K. (1998). An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *Writings on Computer Science*. Roskilde University, No. 81.
- Helsgaun, K. (2000). An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic. *European Journal of Operational Research* 126 (1), 106-130.
- Jonker, R. y Volgenant, T. (1983). Transforming asymmetric into symmetric traveling salesman problems. *Operations Research Letters*, 2:161-163.
- Juan, A. A.; Faulín, J.; Ruiz, R.; Barrios, B.; Gilibert, M. y Vilajosana, X. (2009). Using Oriented Random Search to Provide a Set of Alternative Solutions to the Capacitated Vehicle Routing Problem. En Chinneck y otros, (editores) “Operations Research and Cyber-Infrastructure”. Springer.
- Lin, S. (1965). Computer solutions of the traveling-salesman-problem. *Bell System Technology Journal*, 44:2245-2269.
- Lin, S.; Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research*, 21, 498-516.
- Prim, R. (1957). Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389-1401.
- Rego, C. y Glover, F. (2002). Local search and metaheuristics. Gutin and Punnen. Capítulo 8, páginas 309-368.
- Rodríguez A. (2007). Integración de un SIG con modelos de cálculo y optimización de rutas de vehículos CVRP y software de gestión de flotas. *International Conference on Industrial Engineering and Industrial Management*. Universidad Politécnica de Madrid.
- Rosenkrantz, D. J.; Stearns, R. E. y Lewis, I. P. M. (1977). An analysis of several heuristics for the traveling salesman problem. *SIAM Journal on Computing*, 6(3):563-581.

ⁱ Este trabajo se deriva de la participación de sus autores en un proyecto de investigación financiado por el Ministerio de Ciencia e Innovación, con referencia IAP-020100-2008-11, titulado “OACS - Optimización Avanzada de la Cadena de Suministro”.