

UN ALGORITMO DE TEMPLADO SIMULADO PARA UN PROBLEMA DE OPTIMIZACIÓN DE RUTAS DE VEHÍCULOS: UNA INTERFAZ GRÁFICA EN JAVA

Roque Martín Guitián de Frutos¹ y Balbina Virginia Casas Méndez²

¹Departamento de Aplicaciones Informáticas. Complejo Hospitalario Universitario de Santiago de Compostela (CHUS), Travesía da Choupana s/n, 15706 Santiago de Compostela. e-mail: roqueguitian@hotmail.com

²Departamento de Estadística e Investigación Operativa. Facultad de Matemáticas, Universidad de Santiago de Compostela, Rúa Lope Gómez de Marzoa s/n, Campus Sur, 15782 Santiago de Compostela. e-mail: balbina.casas.mendez@usc.es

RESUMEN

Este trabajo considera el problema de minimizar el coste de llevar a cabo las rutas de los vehículos utilizados por una cooperativa agrícola real que distribuye la alimentación animal entre los socios. Debido a que resolver el modelo exacto es computacionalmente oneroso, nos proponemos implementar algoritmos heurísticos para reducir el tiempo de cálculo. Para este fin, primero se obtiene una solución inicial a través de una heurística de inserción. El algoritmo funciona por construcción sucesiva de caminos hasta que son atendidos todos los pedidos realizados por los socios. En segundo lugar, se diseña un algoritmo de templado simulado que trabaja en la solución inicial para obtener posibles mejoras. Una dificultad de este enfoque radica en la forma de obtención de soluciones de vecinos con el objetivo de llegar a soluciones óptimas. Con el fin de implementar un software eficiente, se han diseñado y programado siete maneras de obtención de soluciones vecinas y que son intercaladas al azar. El tiempo de ejecución y las iteraciones máximas deben ser especificados por el usuario de acuerdo con sus necesidades. Por último, hemos construido una interfaz gráfica. Con ello, el usuario puede interactuar con el sistema rápida y fácilmente. Mostramos las peculiaridades de la aplicación informática desarrollada y algunos de los resultados numéricos obtenidos con ella.

Palabras e frases clave: optimización de rutas de vehículos, algoritmo de templado simulado, interfaz gráfica en Java

1. INTRODUCCIÓN

La logística de los transportes es una temática que gana importancia conforme las sociedades avanzan en su desarrollo industrial. La utilización de métodos de computación para calcular rutas y trayectos óptimos o cercanos a los óptimos puede suponer ahorros destacables. Es por ello que gobiernos y empresas se ven cada vez más interesados en desarrollar sistemas que ayuden a planificar tránsitos, minimizando costes y proporcionando flexibilidad. A menudo la gestión del transporte sigue un patrón centralizado en el que una flota de vehículos distribuye las mercancías desde un almacén central hacia un conjunto de receptores diseminados en distintas localidades. Conforme la cantidad de vehículos, entregas, mercancías o receptores crece, también lo hacen las dificultades para concretar una planificación que minimice los costes.

Este trabajo está motivado por un problema real enfocado en una compañía de comestibles para ganado. AIRA es una cooperativa agraria localizada en Taboada, una villa gallega situada a medio camino entre las ciudades de Lugo y Orense. En ella se producen cuatro tipos diferentes de alimento que se sirven entre sus clientes. La empresa cuenta con alrededor de 1500 ganaderos

(clientes) diseminados en 60 municipios colindantes, cubriendo un área bastante amplia como se ve en Figura 1. Los ganaderos realizan sus pedidos para los diferentes tipos de alimento, órdenes que bien pueden ser urgentes o disponer de un tiempo determinado de entrega. En el caso de órdenes urgentes, el plazo se reduce a tan sólo un día y éstas suponen un mayor precio para el cliente. Habitualmente los ganaderos emiten pedidos una o dos veces al mes y en condiciones normales solamente necesitan un tipo de alimento.

La distribución de la mercancía se realiza mediante la flota de camiones disponibles en la cooperativa y cada conductor es remunerado en función de la distancia recorrida y la carga transportada. Los camiones tienen diversos tamaños y por cuestiones legales disponen de un límite de masa transportable y una distancia máxima a recorrer cada día. Los vehículos se encuentran compartimentalizados en varias tolvas (de tres a cinco) que a su vez tienen distintas capacidades. Es más, por razones técnicas, cada tolva tan sólo puede contener un tipo de alimento y no puede ser compartida entre ganaderos. Más aún, dado el tamaño y características de cada camión algunos de ellos no pueden circular por determinadas vías o acceder a ciertas explotaciones por lo que para cada vehículo existe una lista propia de clientes que no pueden ser servidos.

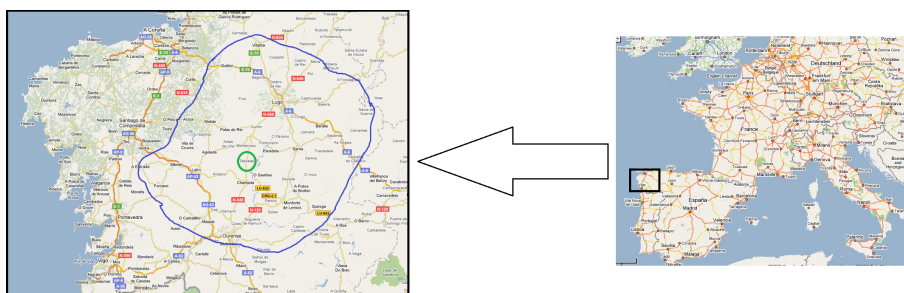


Figura 1: Zona de influencia de la cooperativa AIRA.

El propósito de este estudio es facilitar a la cooperativa una herramienta que de manera automática proponga aquellas rutas para cada camión que satisfagan las restricciones y minimicen los costes de transporte. Hemos de decir que estos costes de transporte se desglosan en una componente fija para cada camión más una cantidad proporcional a la distancia recorrida y la cantidad de alimento cargado en cada ruta.

Por el momento, un grupo de investigación de Ingeniería Agraria de la Universidad de Santiago de Compostela (USC) ha diseñado un sistema global de posicionamiento (GPS) que permite monitorizar las distintas rutas de los camiones. No obstante los gestores de la cooperativa esperan poder mejorar el sistema en los años venideros. Por tanto, este trabajo de enrutado es parte de un proyecto mayor que trata de automatizar todo el proceso de distribución de la compañía.

Este sistema GPS utilizado actualmente proporciona toda la información geográfica que emplearemos para resolver el problema. En cuanto a los datos que se manejan, conocemos las capacidades de cada tolva, las demandas de los diferentes clientes y la urgencia de los pedidos. También sabemos si un camión es capaz de acceder a cada granja y su carga máxima autorizada. Por otra parte, la cooperativa nos proporcionó toda la información relativa al precio de cada descarga y el coste variable del transporte de una determinada cantidad de alimento para una distancia dada. La Figura 2 muestra una imagen de las instalaciones y camiones de la cooperativa AIRA.

Los Problemas de Ruteo de Vehículos (VRP) como el que nos ocupa son problemas de optimización combinatoria. Este tipo de problemas tratan de minimizar o maximizar una función objetivo de varias variables definida sobre un conjunto discreto. El interés que reviste el área no es puramente práctico. Los VRP pertenecen en su mayoría a la clase de complejidad NP-completo. La motivación académica por resolverlos radica en que no es posible construir algoritmos que en tiempo polinomial resuelvan cualquier instancia del problema.



Figura 2: Instalaciones de la cooperativa AIRA.

2. IMPLEMENTACIÓN DE METODOLOGÍAS HEURÍSTICAS Y METAHEURÍSTICAS

El problema de la cooperativa AIRA puede ser formulado mediante un modelo de programación matemática (cf. Álvarez Mozos et al. (2011)) que se puede escribir en cualquier lenguaje de programación. Después de eso, podemos utilizar algún algoritmo que proporcione una solución exacta. Este tipo de metodologías identifican la mejor solución a costa de grandes costes computacionales; rechazamos este enfoque considerando que ejemplos con sólo dos camiones y seis socios requerían tiempos de ejecución de más tres horas del servidor de optimización NEOS. De esta manera, se optó por un enfoque híbrido, que utiliza en primer lugar un algoritmo heurístico de inserción para obtener una solución inicial en un tiempo de ejecución razonable (cf. Clarke and Wright (1964) y Mole and Jameson (1976)). A continuación, se realizan procesos de mejora de la solución inicial mediante el uso de un metaheurístico de templado simulado (véase Kirkpatrick, Gelatt, and Vecchi (1983) y Nunes de Castro (2006)).

Ruiz et al. (2006) también se inspira en un problema real de otra cooperativa española que fabrica alimentos para animales. Su enfoque genera en un primer paso todas las rutas posibles por medio de un algoritmo de enumeración implícita, y en segundo lugar, utiliza un modelo de programación entera que selecciona las rutas óptimas sin considerar todas las restricciones del problema. Por otro lado, en Álvarez Mozos et al. (2011) se diseña y prueba un algoritmo tabú. En el presente trabajo abordamos la tarea de adaptar y probar el metaheurístico de templado simulado, con la peculiaridad frente a otros metaheurísticos y en general, de exigir un menor coste computacional.

2.1. ALGORITMO HEURÍSTICO PARA OBTENER UNA SOLUCIÓN INICIAL

Nace con el objetivo de calcular una solución inicial al problema en un tiempo de ejecución corto (segundos). Si bien no se trata de una solución óptima sí que se exige un resultado de calidad con el que poder iterar en futuras optimizaciones. Partimos de la implementación de la estructura de datos del problema con todos los atributos definidos en el enunciado proporcionado

por la cooperativa. Sobre dicha implementación, el algoritmo desarrollará progresivamente rutas de camiones que sirven a clientes hasta quedar todos los pedidos planificados. El pseudocódigo que sigue este algoritmo se explica a continuación.

1. La programación de cada viaje comienza con la elección de un cliente no servido (cliente semilla) y un camión inicial (camión semilla). Si no hay clientes para servir, el algoritmo termina. Si no hay ningún camión que pueda servir a algún cliente, se entiende que se ha planificado el trabajo diario para todos los camiones y se reinicia el programa (con vehículos vacíos) al día siguiente para atender a los clientes pendientes. Si se han completado 365 días de programación y todavía hay clientes sin servir, termina el algoritmo.
2. Se construye el camino que parte del repositorio central, visita al cliente semilla y vuelve a la compañía. De la manera más eficiente se cargan las tolvas del camión en el viaje de ida. Si ningún cliente puede ser insertado en este paso, el camión ha completado su programación diaria y volvemos al paso 1 para definir más rutas de otros camiones.
3. Se analizan todos los clientes que no han sido atendidos y las posibilidades de inserción que tienen en cada subtrazado del camino. El cliente y la posición se escogen garantizando el menor coste en la inserción.
4. Se inserta en el viaje al cliente con la posición obtenida. Las tolvas se cargan en todos los subtrazados que preceden a esta posición y se comprueba que las restricciones del problema no se violaron (en cuyo caso se vuelve al paso 3 desechando este cliente). En el caso de que una orden asignada no quepa completamente en un vehículo, el pedido se divide en dos, uno se carga en el espacio libre del camión y se considera servido y el otro no. Si no es posible la inserción de nuevos clientes en este trayecto, se regresa al paso 2 para generar un trayecto nuevo para este camión.

El usuario puede elegir la estrategia de elección de los clientes semilla entre las que se muestran a continuación.

1. El más alejado de la fábrica.
2. Cliente con más carga de trabajo.
3. Cliente aleatorio.

El usuario también puede definir la estrategia de elección del camión utilizado en cada caso, entre las que se muestran a continuación.

1. El camión con menor kilometraje.
2. El camión de mayor capacidad.
3. Camión aleatorio.

2.2. METAHEURÍSTICO DE TEMPLADO SIMULADO

Las metodologías de optimización metaheurísticas ofrecen la posibilidad de calcular soluciones que si bien no tienen porqué ser óptimas, son muy buenas para cualquier tipo de VRP. Estos algoritmos parten normalmente de un resultado inicial muy simple que mediante la exploración de la región factible del problema es progresivamente optimizado.

El algoritmo trabaja sobre la solución inicial buscando posibles mejoras que son gestionadas y optimizadas en posteriores iteraciones. Mediante una amplia exploración de la región factible se logran diseñar soluciones óptimas o cercanas a las mismas. La estructura de esta metodología es la siguiente.

1. Repetidamente el algoritmo tratará de buscar una solución vecina a la actual empleando uno de los métodos aleatorios implementados. La ejecución finaliza en el caso de que se alcancen las iteraciones o tiempo máximos permitidos o si no es posible encontrar ninguna solución vecina. En ese caso se devuelve el resultado procesado con menores costes.

- Obtenida la solución vecina, se evalúa en función de sus costes asociados y el valor actual de un parámetro T (denominado temperatura, que va disminuyendo a lo largo de las iteraciones) del que depende la probabilidad de aceptar una solución vecina con coste superior a la actual. Si es aceptada pasará a ser la nueva solución mientras que en caso contrario el resultado será desechado.
- Se actualiza el valor de T si procede y se regresa al paso 1.

La dificultad de esta metodología radica en la forma de obtener soluciones vecinas. El empleo de un sistema de búsqueda excesivamente simple puede dar lugar a algoritmos que no exploran detalladamente toda la región factible. El rango de soluciones alcanzables en este caso es muy reducido y con ello lo es la posibilidad de encontrar resultados óptimos. Con el objetivo de implementar un software eficiente se han diseñado y programado siete formas de obtener soluciones vecinas que se intercalan de manera aleatoria. Sólo de esta manera tendremos la posibilidad de explorar una parte suficientemente amplia de la región de soluciones posibles pudiendo localizar el óptimo global si el tiempo de ejecución lo permite.

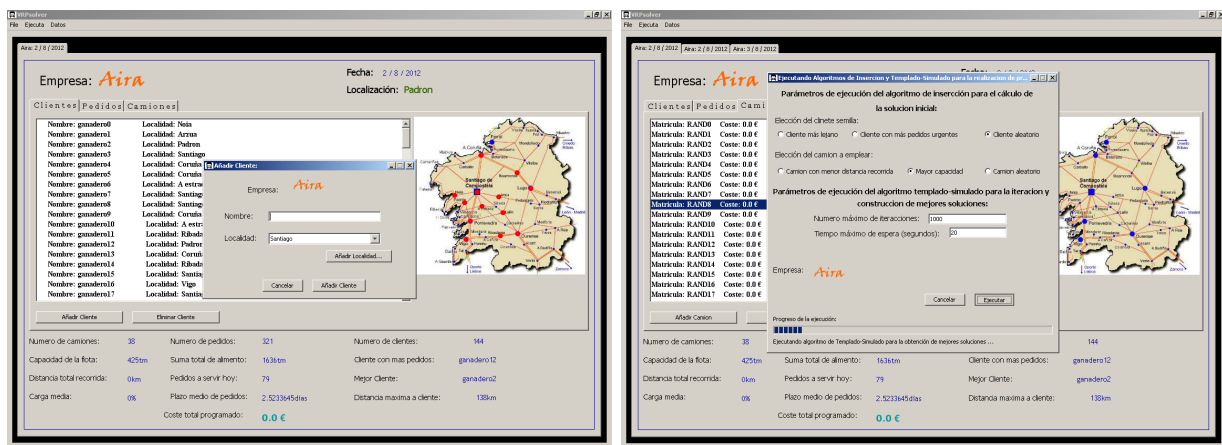


Figura 3: Creación de un nuevo cliente.

Figura 4: Ejecución de los algoritmos de resolución.

2.3. INTERFAZ GRÁFICA

Para el mejor acceso a los datos y a los resultados de los problemas es necesario el diseño de una interfaz gráfica que, de forma sencilla e intuitiva, permita al usuario sacar el máximo partido a las funcionalidades del software. Mediante los diagramas UML correspondientes se ha modelado la interacción del usuario a alto nivel en el manejo de las funcionalidades del sistema; punto de partida para el diseño de la interfaz en la que destacan los siguientes componentes:

- La aplicación ha de permitir la gestión de la estructura de datos del problema, comprendiendo la creación, acceso y eliminación de todas aquellas entidades propias del VRP que se esté diseñando.
- Se implementan los componentes que permiten la introducción de parámetros y la ejecución de los algoritmos que resuelven el problema definido.
- La visualización de resultados incluye el acceso a los trayectos que son solución y al estado de las tolvas de todos los vehículos en cada instante. La gran cantidad de trayectos, rutas y camiones hace imprescindible el diseño de una interfaz gráfica usable que permita el acceso a dicha información de la forma más simple y rápida posible.
- Se han implementado algunas funcionalidades extra como la generación de problemas aleatorios o la exportación de resultados a documentos PDF.

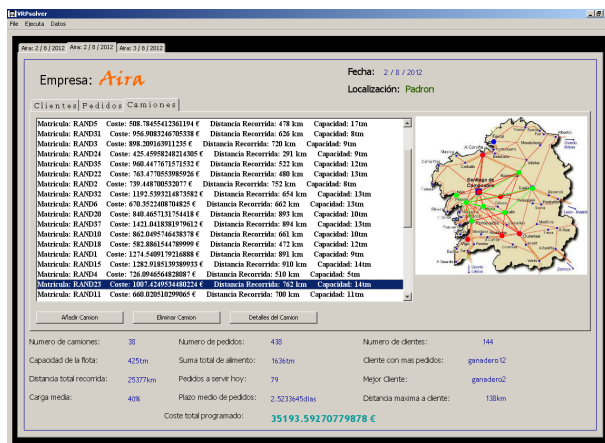


Figura 5: Consulta de los trayectos de un camión.

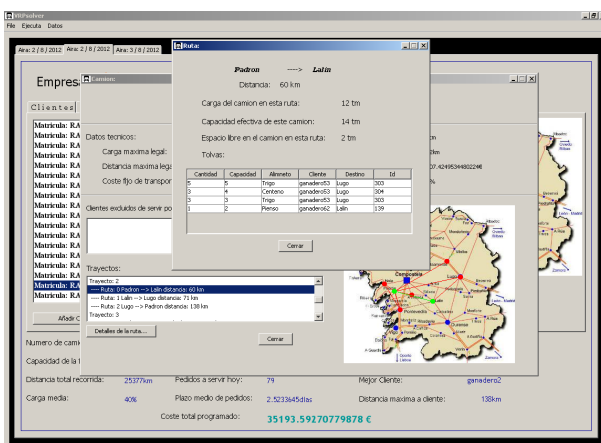


Figura 6: Consulta del estado de las tolvas en una ruta.

3. PRUEBAS Y RESULTADOS

Con la colaboración de la cooperativa agraria AIRA que ha cedido información de casos propios, se han podido generar ejemplos de problemas reales acordes con el enunciado propuesto. En este apartado se trabaja con uno de ellos a fin de ilustrar las características de funcionamiento del software.

El ejemplo comprende una compañía localizada en el municipio de Taboada con una flota de dos camiones con cinco tolvas cada uno. La clientela alcanza los 21 ganaderos diseminados por la geografía gallega. Los clientes emiten un total de 22 pedidos urgentes de un único alimento. Se proporcionan archivos con la matriz de distancias, las características de carga y tolvas de los camiones, los datos de los pedidos, etc.

En primer lugar es necesario definir la estructura de datos mediante la interfaz gráfica para sobre ellos ejecutar los algoritmos implementados. La metodología heurística de inserción nos proporcionará una solución inicial y puede funcionar bajo dos parámetros diferentes: la forma de elección del cliente semilla y del camión inicial. En la Tabla 1 se indican las soluciones de la función objetivo resultado de la ejecución de este algoritmo con cada uno de sus posibles parámetros.

Camión\cliente	Más lejano	Más urgente	Aleatorio
Menos distancia	1008	1008	1029
Más capacidad	1007	1007	1054
Aleatorio	1005	1010	1040

Tabla 1: Resultados reales del algoritmo heurístico de inserción.

Posteriormente se ha ejecutado sobre estas soluciones iniciales la metodología de optimización metaheurística templado simulado implementada. Para cada situación se han ejecutado un total

de 50000 iteraciones o se han empleado 300 segundos, siendo el factor restrictivo el tiempo en la mayoría de los casos. La Tabla 2 muestra las soluciones mejoradas mediante este proceso para cada uno de los resultados anteriores.

Camión\cliente	Más lejano	Más urgente	Aleatorio
Menos distancia	788	775	838
Más capacidad	921	992	848
Aleatorio	917	992	842

Tabla 2: Resultados reales del algoritmo metaheurístico templado simulado.

Tras el proceso de optimización, las soluciones ofrecen resultados que suponen un ahorro medio del 13,8% que puede llegar a alcanzar el 23% en los mejores casos. Son valores interesantes tratándose de una actividad de logística de transportes.

A la vista de los resultados se puede destacar la homogeneidad de las soluciones iniciales entre las que hay muy poca variabilidad. Ello fortalece la idea de que empleando cualquier método de selección de cliente semilla o camión inicial, las soluciones que se producen son de calidad siendo su ejecución instantánea. El funcionamiento del algoritmo de optimización durante un tiempo moderado (5 minutos) genera mejoras en todas las situaciones anteriores y muchas de estas mejoras comprenden cantidades considerables. Ello justifica la posibilidad de emplear de este tipo de metodologías en la resolución de VRPs.

4. CONCLUSIONES

Este trabajo nace de la necesidad de resolver un problema de enrutado de vehículos real en un contexto real. El estudio pormenorizado del mismo junto con el consiguiente análisis bibliográfico permiten concretar las particularidades del dilema que nos ocupa. La investigación operativa ofrece para este tipo de casos un amplio abanico de metodologías que aportan ideas para su resolución. Su comparación nos permite realizar la elección justificada de aquellos algoritmos empleados. La construcción de una aplicación informática que implemente estos procedimientos junto con el análisis exitoso de los resultados obtenidos justifican la realización de este trabajo. En líneas generales se puede decir que la herramienta ha sido creada exitosamente pues satisface plenamente los objetivos para los que fue diseñada. Entre dichas metas podemos destacar la buena planificación de tránsitos de mercancías, el estudio e investigación de problemas de logística de transportes junto con sus técnicas de resolución, la construcción de un entorno gráfico que da soporte a la herramienta y la acreditación de unos requisitos mínimos de robustez, eficiencia y usabilidad. Dada la naturaleza de este trabajo cabe destacar su elevado carácter abierto de cara a modificaciones o ampliaciones. A continuación se indican algunas de estas mejoras a desarrollar en futuras versiones: adaptación de la aplicación a contextos similares, implementación de alternativas de resolución, integración con dispositivos de localización, ...

REFERENCIAS

- Álvarez-Mozos, M, Carpenle, L., Casas-Méndez, B and Mosquera-Rodríguez, M. (2011) Logistic management in a livestock feed cooperative: model, heuristics and case study. Proceedings of the RSME Conference on Transfer and Industrial Mathematics. Servicio de Publicaciones e Intercambio Científico de la Universidad de Santiago de Compostela, 151–157.
- Clarke, G. and Wright, W. (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12, 568-581.
- Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983) Optimization by simulated annealing. *Science*, 220, 671-680.
- Mole, R. H. and Jameson, S. R. (1976) A sequential route-building algorithm employing a generalised savings criterion. *Operational Research Quarterly*, 27, 503-511.

Nunes de Castro, L. (2006) Fundamentals of natural computing: basic concepts, algorithms, and applications. Section 3.3 Hill climbing and simulated annealing. Chapman Hall/CRC. Computer Information Science Series.

Ruiz, R., Maroto, C., and Alcaraz, J. (2006) A decision support system for a real vehicle routing problem. European Journal of Operational Research, 153, 593–606.