

Una ayuda a la aplicación de técnicas heurísticas de optimización

Jose Antonio Lozano Alonso, Francisco Javier Echarte Ayerra

Dept. de Ciencias de la Computación e Inteligencia Artificial

Universidad del País Vasco/Euskal Herriko Unibertsitatea

e-mails: lozano@si.ehu.es

laudix@infonegocio.com

Resumen

El presente artículo presenta una herramienta de ayuda a la implementación y experimentación con técnicas heurísticas de optimización. En concreto, las técnicas de optimización a las que da soporte son: el algoritmo de enfriamiento estadístico, los algoritmos genéticos y la búsqueda tabú.

Al mismo tiempo que se exponen las principales características de la herramienta, se ilustrará el uso que se hace de la misma en la asignatura “Optimización. Técnicas Avanzadas”.

1. Introducción, Situación y Motivación

La asignatura “Optimización. Técnicas Avanzadas” [1] es una asignatura optativa de segundo ciclo, dentro del plan de estudios de Ingeniero en Informática en la Universidad del País Vasco. Dicha asignatura consta en el plan de estudios con una carga lectiva de seis créditos (cuatro teóricos y dos prácticos). El número de alumnos matriculados en la asignatura oscila entre veinte y treinta y cinco. El primer coautor de la presente ponencia viene impartiendo esta asignatura durante los cuatro últimos años.

Los contenidos teóricos de la asignatura se centran en la resolución de problemas de optimización combinatoria [3] mediante técnicas heurísticas. Las más destacables son: búsqueda local, enfriamiento estadístico (*simulated annealing* (a veces citado como: temple, recocido simulado, o enfriamiento simulado)), búsqueda tabú y algoritmos genéticos [2].

La parte práctica de la asignatura consiste en la resolución de un problema de optimización combinatoria mediante el uso de los algoritmos vistos en clase y, variaciones de los mismos específicas para el problema concreto a resolver.

Éstos son problemas clásicos de optimización combinatoria como: el problema del agente viajero, el problema de la mochila, el problema de la partición del grafo, etc.

Una de las dificultades a la que se enfrentaban los alumnos al realizar la parte práctica de la asignatura era que, la implementación de los algoritmos presentados en las clases teóricas no es sencilla. Esto implicaba que una mayor parte del tiempo lo invertían en la implementación de uno de los algoritmos, quedando muy poco tiempo para la comparación de las soluciones obtenidas con diferentes conjuntos de parámetros y de componentes para dicho algoritmo, y por supuesto no pudiendo llevar a cabo una comparación experimental con diferentes métodos.

Dado que los algoritmos son conceptualmente fáciles de entender, la implementación en este caso, no aportaba ningún conocimiento extra a los alumnos.

Una solución a esta situación es la utilización de paquetes de software estándar que tuviesen implementados los algoritmos. Sin embargo, no existe actualmente ninguna herramienta que implemente todos los algoritmos, y menos en lengua española. Esto suponía que los alumnos debían familiarizarse con más de un paquete, lo que implica nuevamente un gran esfuerzo.

Para solucionar dicha dificultad hemos elaborado una herramienta que supone una ayuda a la aplicación de estos algoritmos: recoge todos los algoritmos vistos en clase bajo un mismo entorno; tiene implementados un conjunto de componentes y de opciones que permiten la comparación experimental de diferentes algoritmos; y además es abierta en el sentido de que es posible añadir nuevos componentes al sistema, de forma sencilla.

El resto del artículo está organizado de la siguiente manera: la sección 2 describe en líneas generales la herramienta. La sección 3 trata sobre las posibilidades de aumentar el VOpt, dedicándose la sección 4 a explicar el entorno. La sección 5 apunta las limitaciones del sistema y se termina apuntando las conclusiones en la sección final.

2. VOpt: Una herramienta de ayuda a la aplicación de técnicas heurísticas de optimización

2.1 Aspectos generales y metodología de utilización básica

VOpt (Visual Optimization) [4] es una herramienta de ayuda a la aplicación y experimentación con técnicas heurísticas de optimización. Dicha herramienta se ha centrado en tres algoritmos heurísticos (los más utilizados atendiendo a la literatura del área): el algoritmo de enfriamiento estadístico, los algoritmos genéticos y la búsqueda tabú.

La herramienta codifica los componentes fundamentales de estas tres técnicas de optimización. De esta forma, el estudiante, una vez elegido el algoritmo a utilizar, decide dentro del conjunto de componentes disponibles aquellos que más le interesen. Esto permite que el usuario concentre su atención en la implementación de la función de coste o función objetivo del problema de optimización a resolver.

Una vez codificado el procedimiento que implementa la función de coste y elegidos los componentes del algoritmo, el sistema genera código C. Si se dispone de un compilador de C en el ordenador es posible compilarlo desde el entorno de VOpt, en caso contrario el código C generado se puede trasladar a cualquier otra máquina que disponga de un compilador y compilarlo en dicha máquina.

Si la compilación se realizó dentro del entorno, el sistema devuelve los errores de compilación si es que estos existían (los asociados a la codificación de la función objetivo). Finalmente, se dispondrá de un fichero ejecutable. Dicho fichero puede ser ejecutado desde el sistema y los datos de salida son almacenados en un archivo que gestiona VOpt.

Dado que estos algoritmos utilizan de alguna forma la aleatoriedad, los resultados variarán de una ejecución a otra, por lo que resulta interesante realizar varias ejecuciones si es que se quiere obtener alguna información de valor. Por esta razón, el sistema ofrece también algunas funciones para calcular estadísticas sencillas utilizando los ficheros de salida.

2.2 Componentes básicos de los métodos de optimización

Un elemento común a todos los algoritmos de optimización heurísticos es la codificación de la solución. Cada solución a un problema de optimización dado debe codificarse de una forma concreta que permita almacenarlo en el ordenador.

La herramienta VOpt dispone por defecto de las siguientes codificaciones: lista binaria, lista de reales, lista de enteros y permutaciones. Cada una de estas codificaciones dispone de dos inicializaciones diferentes, una aleatoria y otra a partir de un fichero de datos.

El algoritmo de enfriamiento estadístico dispone del siguiente conjunto de componentes: sistema de vecinos, longitud de las cadenas de Markov, función de reducción de la temperatura y criterio de terminación. Además de los componentes anteriores el sistema proporciona al usuario la posibilidad de establecer el valor inicial del parámetro temperatura.

En relación a los algoritmos genéticos los componentes configurables son: elección de la población inicial, operadores de selección, cruce y mutación y el criterio de parada. Los parámetros a los que se debe asignar valor son: probabilidad de cruce, probabilidad de mutación, tamaño de la población y longitud del individuo.

La búsqueda tabú dispone de los siguientes componentes: sistema de vecinos, criterio de parada, criterios de aspiración. Los parámetros que deben ser establecidos son: longitud de la lista tabú, tiempo que un atributo permanece tabú, longitud de la intensificación y de la diversificación.

3. VOpt: Un sistema abierto

Una de las características a destacar del sistema es que es una herramienta abierta. Es decir, es posible incorporar de forma sencilla nuevos componentes.

Esta característica es fundamental si trabajamos con algoritmos de optimización. Cada problema tiene sus propias características, y por lo tanto puede ocurrir que los componentes que ofrece por defecto el sistema no sean los más adecuados. Por ejemplo es posible que necesitemos una codificación diferente.

Para crear nuevos componentes el sistema dispone de un conjunto de ayudas. Por ejemplo, en el caso de una nueva codificación, una vez definida la misma es necesario codificar también las funciones de creación, inicialización, copia y destrucción. A la hora de crear las funciones anteriores, una de las ayudas que proporciona el sistema es la cabecera de dicha función.

Sin embargo no sólo es posible crear componentes, sino también modificar los ya existentes. Por ejemplo, pueden existir situaciones donde sea interesante que un algoritmo genético utilice sólo mutación y no cruce. Cada algoritmo viene dado por una plantilla, luego en este caso hay que construir una nueva plantilla. La construcción de las plantillas se facilita mediante la utilización de directivas, que hacen que algunos componentes de la plantilla sean obligatorios mientras que otros sean opcionales.

4. VOpt: Un entorno amigable

Otra característica a destacar de la herramienta VOpt es su entorno. Proporciona un interfaz gráfico con menús y ventanas de tipo Windows.

El acceso a las diferentes funcionalidades se puede realizar de diferentes formas permitiendo al usuario utilizar la que más le convenga en cada momento.

El sistema tiene integrado una ayuda donde además de contener información acerca de cada una de sus funcionalidades y componentes, proporciona una introducción a los métodos de optimización que se implementan. Además cada componente tiene asociada una descripción que aparece en la pantalla cuando se pincha con el ratón en dicho elemento, de esta forma el usuario

puede conocer que representa el componente que se va a añadir al algoritmo que se quiere construir.

La figura 1 muestra un posible estado del interfaz. La información que refleja el entorno nos indica que estamos codificando un algoritmo de enfriamiento estadístico para un problema de minimización. El usuario se encuentra eligiendo una codificación para las soluciones, teniendo seleccionada (TlistaBinaria) una lista de ceros y unos. Obsérvese también como se muestra el código C de dicha componente.

5. VOpt: Puntos débiles

VOpt también tiene algunos puntos que son susceptibles de mejora. El principal problema es que en la actualidad, únicamente se ejecuta de forma correcta bajo Windows NT. El sistema fue desarrollado por el segundo coautor de este escrito, como proyecto fin de carrera en el año 1999, y desde entonces la tecnología ha superado el sistema operativo Windows NT. Por supuesto no se dispone de ningún soporte técnico, ni está prevista ninguna actualización.

En relación con las capacidades de la herramienta en sí, hay aspectos que podrían ser mejorados. Por ejemplo, las herramientas de que se dispone para analizar los resultados de diferentes ejecuciones son muy básicas, y en la mayoría de los casos hay que utilizar otros programas auxiliares (Excel, SPSS, etc.).

Otro punto débil es el control de las ejecuciones y de los datos que se generan. Por ejemplo, el establecimiento de los parámetros para estos algoritmos se realiza en la mayoría de los casos tras experimentar con ciertos conjuntos de ellos. Una persona experta en la aplicación de estos algoritmos es capaz de descubrir fallos u obtener mejores conjuntos de parámetros simplemente observando una traza de la ejecución del algoritmo (por ejemplo en el caso del algoritmo de enfriamiento estadístico, la información que debería proporcionar la traza es: valor del parámetro temperatura, solución actual, solución vecina elegida, número de aceptados con el valor de t fijo y mejor solución conseguida hasta el momento). Desafortunadamente el sistema actual no permite la obtención de este tipo de trazas.

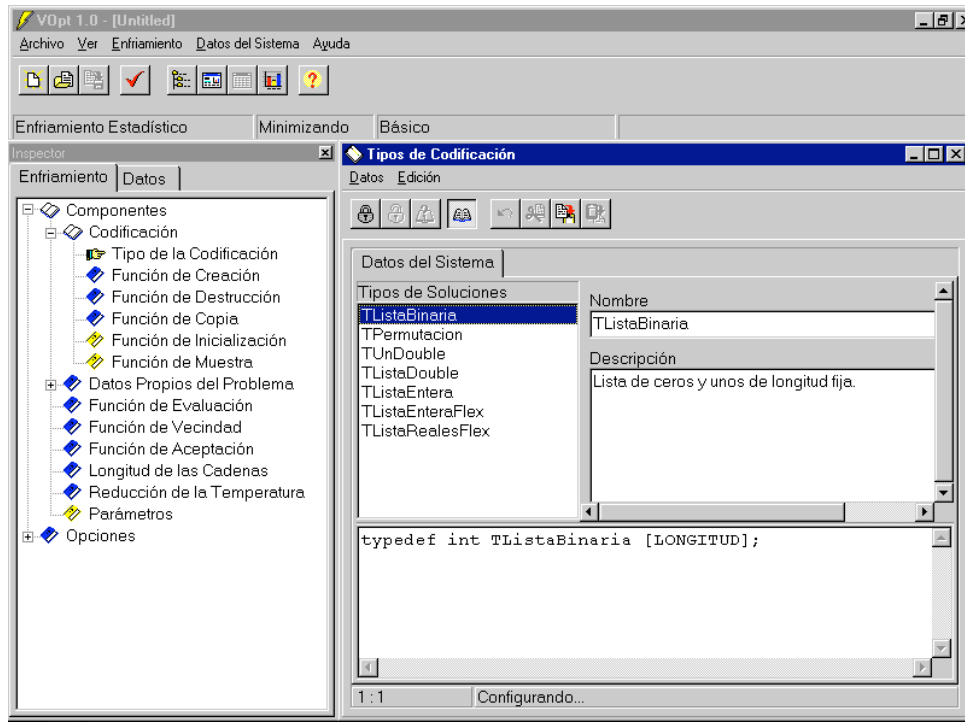


Figura 1. Ejemplo de uso del entorno VOpt

6. Conclusiones

En este escrito hemos presentado el software VOpt. Este software constituye una ayuda en la aplicación y experimentación con algoritmos heurísticos de optimización. El entorno proporciona la oportunidad de que el usuario se centre en la implementación de la función objetivo obviando los detalles de la implementación del algoritmo de optimización. Al mismo tiempo el sistema aún bajo un mismo entorno tres técnicas que optimización hasta ahora dispersas en paquetes diferentes. Finalmente destacar que el sistema es un sistema abierto que permite añadir nuevos componentes de forma sencilla.

La utilización de VOpt en la asignatura ha sido muy satisfactoria. Los alumnos han realizado experimentos con varias técnicas de optimización, adquiriendo más conocimiento experimental acerca de su funcionamiento.

Agradecimientos

Los autores desean agradecer las sugerencias de la profesora Susana Rodríguez.

Referencias

- [1] *Guía Docente de la Facultad de Informática 2001-2002*. Universidad del País Vasco. San Sebastián.
- [2] C. Reeves. *Modern heuristic techniques for combinatorial optimization*. Blackwell scientific publications, 1993.
- [3] C. Papadimitriou y K. Steiglitz. *Combinatorial optimization: Algorithms and complexity*. Prentice-Hall, 1982.
- [4] F. Echarte. *Entorno de ayuda a la aplicación de técnicas de optimización*. Proyecto fin de carrera. Fac. Informática UPV-EHU, 1999.