

Guión de prácticas 2. Busca Tabú

Nesta práctica aplicaremos o algoritmo de Busca Tabú ao problema do “viaxante de comercio”. A práctica divídese en dous apartados:

Algoritmo Tabú básico (6 puntos; especificación obrigatoria)

Neste apartado desenvolverase unha versión básica da busca Tabú. Todos deberedes utilizar o mesmo deseño do algoritmo. Este deseño inicial común ten algunhas diferenzas sobre o exemplo utilizado en clase de teoría. As súas características son as seguintes:

1. Representación

Usaremos unha **representación de orde** formada por secuencia dos índices das cidades. Por exemplo, se tomamos (sen perda de xeneralidade) como cidade de partida e fin a de índice 0, o percorrido pode representarse coma:

$$\{1\ 3\ 2\ 4\ 5\ 6\ 7\ 9\ 8\ \dots\}$$

Non é preciso incluír a cidade de partida e fin na representación final do percorrido, xa que sempre é a mesma. Trataremos o problema de $n=100$ cidades, polo que a lonxitude de cada solución é 99 enteiros.

2. Solución inicial

A solución inicial será **totalmente aleatoria**.

3. Xeración da veciñanza dunha solución

Utilizaremos o operador de **intercambio de dous elementos**, descrito na práctica 1 (busca local). A veciñanza explotarase completamente (salvo o tabú), polo que a estratexia de busca local vai ser “o mellor.”

4. Función de custe

A función de custe é a especificada na práctica 1.

5. Lista Tabú

A lista Tabú estará formada movementos de intercambio de índices $\{i, j\}$, onde se segue o criterio de que $i > j$ igual que na práctica 1. A xestión da LT é similar á indicada nas clases expositivas.

Estableceremos n como parámetro de tenencia, polo que o tamaño da lista tabú será de n elementos.

6. Reinicialización

No caso de que transcorran 100 iteracións consecutivas sen que mellore a **mellor solución S_{OPT} acadada ata ese momento** farase unha reinicialización desde S_{OPT} . Trátase polo tanto dunha estratexia de reinicialización **por intensificación**. No reinicio **non** se restaurará a lista tabú, para dar opción a visitar veciños non considerados anteriormente.

Guión de prácticas 2. Busca Tabú

7. Criterio de parada

Finalizaremos a busca cando se realicen 10.000 iteracións do algoritmo.

Algoritmo Tabú mellorado (4 puntos; especificación voluntaria)

Neste apartado o obxectivo é realizar melloras ao algoritmo obtido previamente. Poderá modificarse:

- A xeración da solución inicial (por exemplo, inicialización “greedy”, ...)
- A xestión da lista Tabú, coa inclusión dalgún criterio de aspiración (por exemplo, excluír da lista tabú unha solución se mellora a que é mellor ata ese momento)
- O operador de xeración da veciñanza: non considerar todos os pares de índices, cambiar o operador, ...
- Outras estratexias de reinicialización con intensificación: reiniciar dende unha lista das **NS** mellores solucións rexistradas e/ou restaurar a lista tabú.
- Cambiar a estratexia de **reinicialización** para dar **diversificación**, cunha estratexia de **memoria a longo prazo**, consistente nunha matriz simétrica *frec* que almacena o número de veces que cada par de cidades teñen sido consecutivas nalgunha das solucións aceptadas durante a busca. Modificarase o método de reinicialización para dotalo de maior diversidade. Para isto usase unha estratexia voraz sobre una matriz de distancias modificada que **penalice** os pares de cidades con **maior** frecuencia:

$$d(i, j) = d(i, j) + \mu \cdot (d_{\max} - d_{\min}) \cdot \frac{frec(i, j)}{frec_{\max}}$$

Pódese experimentar con diversos valores de μ . Pódese restaurar ou non a lista tabú na reinicialización.

- Utilizar un criterio de **oscilación estratéxica** na reinicialización, que alterne **intensificación e diversificación**.

Traza de execución e caso de uso

Estes dous requisitos son de atención obrigatoria:

1. O programa deberá poder **compilarse en liña de comandos**. No arquivo LEEME.txt debes incluír ao principio, **obrigatoriamente**, o **comando de compilación** que se debe usar para compilar en liña.
2. O programa deberá poder **executarse en liña de comandos**, coa sintaxe

a.out distancias.txt [aleatorios.txt]

Cando se utilicen os dous parámetros o programa deberá ler os números aleatorios que precise a partir dun arquivo de nome **aleatorios.txt**. Proporcionarase o arquivo de nome **aleatorios.txt**

Guión de prácticas 2. Busca Tabú

xunto cun caso de uso completo coa execución da metaheurística para eses números, no que se describirán os resultados a obter.

3. A saída por pantalla deberá ser **idéntica** (en contido e formato) á especificada na traza, para facilitar a detección de erros de implementación.

Entrega da práctica

A data límite de entrega será a indicada na aula virtual.

Será obrigatorio entregar o código fonte correspondente a cada unha das partes (habilitarase unha entrega diferente para cada parte). Cada entrega consistirá nun arquivo comprimido, e en publicar os resultados de **10 execucións diferentes do algoritmo** nunha folla de cálculo compartida (da que proporcionaremos o enlace mediante un anuncio na Aula Virtual). Ademais, xunto co código da especificación voluntaria se incluírá un documento de texto (máximo media páxina) no que se explique en qué consistiron as melloras introducidas e unha valoración persoal de porqué as consideras apropiadas para o problema.

Avaliación da práctica

Este traballo contará para a parte práctica da materia cunha ponderación do 25%. A avaliación do funcionamento da especificación obrigatoria realizarase mediante un test de autoavaliación sobre os resultados de execución da vosa implementación en diferentes casos (matrices de distancias, números aleatorios, ...) en diferentes etapas do proceso de busca.

A avaliación da especificación voluntaria puntuará:

- a porcentaxe de mellora conseguida sobre a especificación obrigatoria
- a xustificación da adecuación das melloras introducidas