UNIVERSIDAD DE GRANADA
E.T.S.I. INFORMÁTICA Y TELECOMUNICACIÓN

Departamento de Ciencias de la
Computación e Inteligencia Artificial

Metaheurísticas

http://sci2s.ugr.es/graduateCourses/Metaheuristicas
https://decsai.ugr.es

Guión de Prácticas

Práctica 3.a:
Búsquedas por Trayectorias para el Problema de la Máxima Diversidad

Curso 2018-2019

Tercer Curso del Grado en Ingeniería Informática

Práctica 3.a

Búsquedas por Trayectorias para el Problema de la Máxima Diversidad

1. Objetivos

El objetivo de esta práctica es estudiar el funcionamiento de las *Técnicas de Búsqueda basadas en Trayectorias* (tanto simples como múltiples) en la resolución del problema de la máxima diversidad (MDP) descrito en las transparencias del Seminario 2. Para ello, se requerirá que el estudiante adapte las siguientes técnicas metaheurísticas a dicho problema:

- Enfriamiento Simulado (ES).
- Búsqueda Multiarranque Básica (BMB).
- GRASP.
- Búsqueda Local Reiterada (ILS).
- Hibridación de ILS y ES (ILS-ES).

El estudiante deberá comparar los resultados obtenidos con las estimaciones existentes para el valor de los óptimos de una serie de casos del problema, así como con el algoritmo *greedy* básico y la búsqueda local (BL) descritos en el Seminario 2 y desarrollados en la Práctica 1.a.

La práctica se evalúa sobre un total de **3,5 puntos**, distribuidos de la siguiente forma: ES (1 punto), BMB (0,25 puntos), GRASP (1,25 puntos), ILS (0,75 puntos) e ILS-ES (0,25 puntos).

La fecha límite de entrega será el **Sábado 8 de Junio de 2019** antes de las 23:59 horas. La entrega de la práctica se realizará por internet a través del espacio de la asignatura en PRADO.

2. Trabajo a Realizar

El estudiante podrá desarrollar los algoritmos de la práctica siguiendo la modalidad que desee: trabajando con cualquiera de los frameworks de metaheurísticas

estudiados en el Seminario 1, implementándolos a partir del código C proporcionado en la web de la asignatura o considerando cualquier código disponible en Internet.

Los métodos desarrollados serán ejecutados sobre una serie de casos del problema. Se realizará un estudio comparativo de los resultados obtenidos y se analizará el comportamiento de cada algoritmo en base a dichos resultados. **Este análisis influirá decisivamente en la calificación final de la práctica**.

En las secciones siguientes se describen los aspectos relacionados con cada algoritmo a desarrollar y las tablas de resultados a obtener. Los casos del problema y los estadísticos de calidad (*Desv* y *Tiempo*) serán los mismos que en la Práctica 1.a (véase las Secciones 3 y 4 de dicho guión de prácticas).

3. Componentes de los Algoritmos

Los algoritmos de esta práctica tienen en común las siguientes componentes:

- *Esquema de representación*: Se seguirá la representación en forma de un conjunto *Sel*={s_1, …, s_m} que almacena los *m* elementos seleccionados de entre los *n* elementos del conjunto *S*.

- *Esquema de generación de vecinos empleado en los algoritmos de búsqueda por trayectorias simples*: Se empleará el movimiento de intercambio *Int(Sel,i,j)* en el que se escoge un elemento y se intercambia por otro que no estuviera seleccionado. Será obligatorio emplear la **factorización** de la función objetivo en todos los casos.

  A continuación veremos las particularidades de cada algoritmo.

3.1. Enfriamiento Simulado

Algoritmo

Se ha de emplear un algoritmo ES con las siguientes componentes:

- *Esquema de enfriamiento*: Se empleará el esquema de Cauchy modificado:

$$T_{k+1} = \frac{T_k}{1+\beta \cdot T_k} \qquad ; \qquad \beta = \frac{T_0 - T_f}{M \cdot T_0 \cdot T_f}$$

  donde M es el número de enfriamientos (iteraciones) a realizar, T_0 es la temperatura inicial y T_f la temperatura final que tendrá un valor cercano a cero[1].

- *Operador de Vecino y exploración del entorno para L(T)*: En cada iteración del bucle interno *L(T)*, se aplicará un único movimiento *Int(Sel,i,j)* para generar una

[1] **NOTA**: Si el estudiante observa que este esquema de enfriamiento enfría demasiado rápido, puede sustituirlo por un esquema proporcional: $T_{k+1} = \leftarrow \alpha \cdot T_k$ con $\alpha \in [0.9, 0.99]$.

única solución vecina que será comparada con la solución actual. Se escogerán aleatoriamente los elementos s_i y s_j a intercambiar.

- *Condición de enfriamiento L(T)*: Se enfriará la temperatura, finalizando la iteración actual, bien cuando se haya generado un número máximo de vecinos *máx_vecinos* (independientemente de si han sido o no aceptados) o bien cuando se haya aceptado un número máximo de los vecinos generados *máx_éxitos*.

- *Condición de parada*: El algoritmo finalizará bien cuando haya alcanzado el número máximo de evaluaciones prefijado o bien cuando el número de éxitos en el enfriamiento actual sea igual a 0.

Valores de los parámetros y ejecuciones

La temperatura inicial se calculará en función de la siguiente fórmula:

$$T_0 = \frac{\mu \cdot C(S_0)}{-\ln(\phi)}$$

donde T_0 es la temperatura inicial, $C(S_0)$ es el coste de la solución inicial y $\phi \in [0,1]$ es la probabilidad de aceptar una solución un μ por 1 peor que la inicial. En las ejecuciones se considerará $\phi=\mu=0,3$. La temperatura final T_f se fijará a 10^{-3} (*¡comprobando siempre que sea menor que la inicial!*).

Los parámetros que definen el bucle interno *L(T)* tomarán valor *máx_vecinos*=10·*n* (tamaño del caso del problema) y *máx_éxitos=0.1*máx_vecinos*. **El número máximo de evaluaciones será 50000**. Por lo tanto, el número de iteraciones (enfriamientos) M del algoritmo ES será igual a 50000/*máx_vecinos* [2].

3.2. Búsqueda Multiarranque Básica

Algoritmo

El algoritmo BMB consistirá simplemente en generar un determinado número de soluciones aleatorias iniciales y optimizar cada una de ellas con el algoritmo de BL considerado. Se devolverá la mejor solución encontrada en todo el proceso.

Valores de los parámetros y ejecuciones

Se realizará una única ejecución de la BMB sobre cada caso del problema. En dicha ejecución, se realizarán 25 iteraciones, es decir, se generarán 25 soluciones iniciales aleatorias y se aplicará la BL sobre cada una de ellas. Cada aplicación de la BL finalizará bien cuando no se encuentre mejora en todo el entorno o bien cuando se hayan realizado **50000 evaluaciones**.

[2] **NOTA 1**: Es posible que se realicen menos de 50000 evaluaciones durante la ejecución debido a la condición de enfriamiento cuando se alcanzan *máx_éxitos*.
NOTA 2: Puede que con *máx_vecinos*=10·*n* se generen demasiados vecinos para cada enfriamiento. El estudiante puede probar también con *máx_vecinos*=5·*n* o directamente *n*.

3.3. GRASP

Algoritmo

El algoritmo GRASP constará de dos componentes: construcción de soluciones *greedy* probabilísticas y optimización de las mismas mediante el algoritmo de BL. El algoritmo *greedy* probabilístico a considerar es el explicado en las transparencias del Seminario 4. Una vez generada cada solución *greedy* probabilística inicial se aplicará el algoritmo de BL sobre ella. Finalmente, se devolverá la mejor solución encontrada.

Valores de los parámetros y ejecuciones

Se realizará una única ejecución del algoritmo GRASP sobre cada caso del problema. En dicha ejecución, se realizarán 25 iteraciones, es decir, se generarán 25 soluciones *greedy* probabilísticas y se aplicará la BL sobre cada una de ellas. El parámetro α que determina el umbral de tolerancia de calidad para la construcción de la lista restringida de candidatos tomará valor 0.3. Cada aplicación de la BL finalizará bien cuando no se encuentre mejora en todo el entorno o bien cuando se hayan realizado **50000 evaluaciones**.

3.4. Búsqueda Local Reiterada (ILS)

Algoritmo

El algoritmo ILS consistirá en generar una solución inicial aleatoria y aplicar el algoritmo de BL sobre ella. Una vez obtenida la solución optimizada, se estudiará si es mejor que la mejor solución encontrada hasta el momento y se realizará una mutación sobre la mejor de estas dos, volviendo a aplicar el algoritmo de BL sobre esta solución mutada. Este proceso se repetirá un determinado número de veces, devolviéndose la mejor solución encontrada en toda la ejecución. Por tanto, se seguirá el *criterio del mejor* como criterio de aceptación de la ILS.

Tal y como se describe en las transparencias del Seminario 4, el operador de mutación de ILS estará basado en un operador de vecino para representación de orden que provoque un cambio más brusco en la solución actual que el considerado en la BL. Para ello, usaremos el operador de modificación por sublista aleatoria de tamaño fijo t. Este proceso consiste en generar aleatoriamente una posición i de inicio de la sublista y reasignar aleatoriamente el orden de los nodos existentes entre esa posición i y la posición $i+t$.

Valores de los parámetros y ejecuciones

Se realizará una única ejecución del algoritmo ILS sobre cada caso del problema. En dicha ejecución, se realizarán 25 iteraciones, es decir, se aplicará 25 veces el algoritmo de BL, la primera vez sobre una solución inicial aleatoria y las 24 restantes sobre soluciones mutadas. Cada aplicación de la BL finalizará bien cuando no se encuentre mejora en todo el entorno o bien cuando se hayan realizado **50000 evaluaciones**. Se usará un valor de $t=0.1\cdot m$ para el número de elementos seleccionados distintos a modificar en la mutación.

3.5. Algoritmo Híbrido ILS-ES

Algoritmo

El algoritmo ILS-ES tendrá la misma composición que el ILS estándar, con la única diferencia que el algoritmo de búsqueda por trayectorias simples considerado para refinar las soluciones iniciales será el ES de la Sección 3.1 en lugar de la BL empleado hasta ahora.

Valores de los parámetros y ejecuciones

Se realizará una única ejecución del algoritmo ILS-ES sobre cada caso del problema. En dicha ejecución, se realizarán 25 iteraciones, igual que en la ILS estándar y se optimizarán usando el algoritmo ES con los parámetros indicados en la Sección 3.1.

4. Tablas de Resultados a Obtener

Se diseñará una tabla para cada algoritmo (*Greedy*, BL, ES, BMB, GRASP, ILS e ILS-ES) donde se recojan los resultados de la ejecución de dicho algoritmo al conjunto de casos del problema. Tendrá la misma estructura que la Tabla 5.1 del guión de la Práctica 1.a.

Finalmente, se construirá una tabla de resultados global que recoja los resultados medios de calidad y tiempo para todos los algoritmos considerados, tal como se muestra en la tabla 4.1. Aunque en la tabla que sirve de ejemplo se han incluido todos los algoritmos considerados en esta práctica, naturalmente sólo se incluirán los que se hayan desarrollado. Los resultados del Greedy y de la BL corresponden a los de la Práctica 1.a.

Tabla 4.1: Resultados globales en el MDP

Algoritmo	**Desv**	**Tiempo**
Greedy	x	x
BL	x	x
ES	x	x
BMB	x	x
GRASP	x	x
ILS	x	x
ILS-ES	x	X

A partir de los datos mostrados en estas tablas, el estudiante realizará un análisis de los resultados obtenidos, que influirá significativamente en la calificación de la práctica. En dicho análisis se deben comparar los distintos algoritmos en términos de calidad de las soluciones y tiempo requerido para obtenerlas. En este caso concreto, se pueden comparar las técnicas basadas en trayectorias simples (BL y ES) con las de trayectorias múltiples (BMB, GRASP e ILS), teniendo en cuenta que las segundas

tienen ventaja al ejecutar el optimizador de trayectoria simple varias veces. Por otro lado, se puede analizar también el comportamiento de los algoritmos en algunos de los casos individuales que presenten un comportamiento más destacado.

5. Documentación y Ficheros a Entregar

Además de la documentación detallada en la Sección 6 del guión de la Práctica 1.a, en lo referente al punto d) se incluirá, al menos, la siguiente información:

1. Esquema de representación de soluciones empleado.
2. Descripción en pseudocódigo de la función objetivo.
3. Descripción en pseudocódigo del proceso de generación de soluciones aleatorias (usado en la BMB y en la ILS).

En lo que respecta al punto e), se incluirá la siguiente información:

1. Descripción en pseudocódigo del esquema de búsqueda seguido por cada algoritmo (ES, BMB, GRASP, e ILS).
2. Además se detallarán, al menos, las siguientes componentes particulares de cada algoritmo:
   a) Para el algoritmo ES, descripción en pseudocódigo del cálculo de la temperatura inicial y del esquema de enfriamiento.
   b) Para el algoritmo de BL, descripción en pseudocódigo del método de creación de la lista de candidatos, el de exploración del entorno, el operador de generación de vecino y su factorización.
   c) Para el algoritmo GRASP, descripción en pseudocódigo del mecanismo de generación de soluciones *greedy* probabilísticas, indicando el modo de obtención de la lista restringida de candidatos.
   d) Para el algoritmo ILS, descripción en pseudocódigo del operador de mutación empleado.

Como recomendación, el apartado 4 debería describirse en un máximo de dos páginas. En el apartado 5, el número total de páginas para describir cada algoritmo (incluyendo el pseudocódigo del esquema de búsqueda y de las componentes particulares) sería de una página para ES, BMB e ILS, y de dos páginas para GRASP.

Se recuerda que **la documentación nunca deberá incluir listado total o parcial del código fuente en caso de haberlo implementado**.

En lo referente al **desarrollo de la práctica**, se seguirán los mismos criterios descritos en la Sección 6 del guión de la Práctica 1.a. El **método de evaluación** será el descrito en la Sección 7 de dicho guión.