



Modelos Bioinspirados y Heurísticas de Búsqueda 4 Curso Grado Ingeniería en Informática

Área de Ciencias de la Computación e
Inteligencia Artificial
Departamento de
Tecnologías de la Información

PRÁCTICA 3 (Versión 2021, 1.0)

Algoritmos heurísticos no constructivos

Genéticos

Objetivos

El objetivo de esta práctica es estudiar el funcionamiento de los algoritmos heurísticos no constructivos. Este tipo de algoritmos parten de una solución inicial o un conjunto de ellas para a través de operaciones de transformación ir mejorando las soluciones candidatas. Los algoritmos a estudiar son los evolutivos: genético simple, multimodal y CHC. Para ello, se requerirá que el alumno implemente estos algoritmos, para resolver el *Problema del Viajante de Comercio* (TSP). El comportamiento de los algoritmos implementados deberá compararse con un *Algoritmo Greedy*.

Enunciado de la práctica

El Problema del Viajante de Comercio (TSP) es uno de los problemas de optimización combinatoria más conocidos. En su formulación más general, dadas una serie de ciudades, el objetivo consiste en encontrar el circuito de menor coste que parta de una ciudad concreta, pase por todas las demás una sola vez y retorne a la ciudad de origen.

En nuestro caso trabajaremos con tres instancias del problema (más una opcional), obtenidas de la biblioteca TSPLIB, todas ellas correspondientes al TSP simétrico (misma distancia o coste independientemente de si es en un sentido u otro). Serán las siguientes:

- St70: Tamaño 70 ciudades. Coste de la solución óptima: 675
- Ch130: Tamaño 130 ciudades. Coste de la solución óptima: 6.110
- A280: Tamaño 280 ciudades. Coste de la solución óptima: 2.579
- Opcionalmente: Vm1748: Tamaño 1748 ciudades.

Todos estos ficheros presentan el mismo formato, una lista con dos valores para cada ciudad que representan sus coordenadas en el plano. Para componer la matriz de costes se deberá calcular la *distancia euclídea* entre cada par de ciudades (i, j) . Los costos obtenidos han de ser números enteros, es decir, no se considerarán

decimales. Así, la matriz de distancias se calcularía, por ejemplo en lenguaje C, de la siguiente forma:

```
xd = x[i] - x[j];  
yd = y[i] - y[j];  
dij = rint( sqrt( xd*xd + yd*yd ) );
```

donde `rint` es la función de redondeo y `sqrt` es la raíz cuadrada.

A continuación, mostramos los valores de los parámetros a considerar en la experimentación de cada uno de estos algoritmos (parámetros, soluciones iniciales, condiciones de parada, etc).

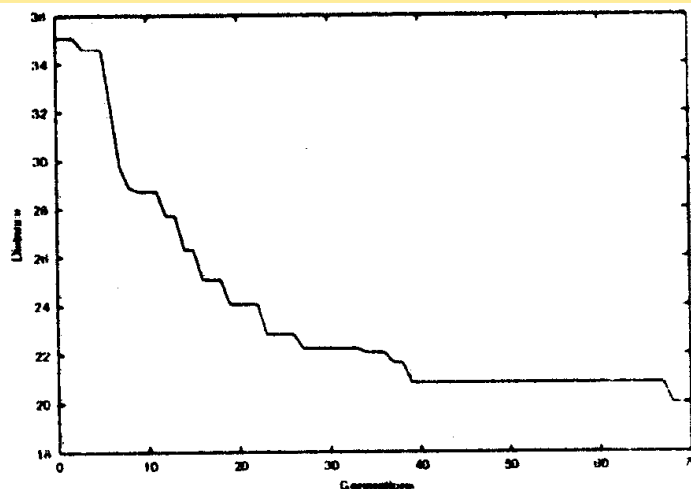
Algoritmos Genéticos

Cada algoritmo debe ejecutarse al 5 veces para cada subconjunto de datos, es decir, una vez para cada semilla con cada uno de los conjuntos de datos con semilla conocida y publicada en las tablas.

Algoritmo Genético Básico

El alumno puede elegir la combinación de operadores y modalidad que crea mas oportuna, explicando la elección para obtener resultados adecuados:

- Estacionario/Generacional (en dicho caso elite de 10% de la población).
- Mutación: Sublista de **S elementos**. Hacer un estudio sobre los resultados teniendo en cuenta parámetros de convergencia. La convergencia de la población se puede observar sobre el valor medio, mínimo y máximo de los individuos respecto al número de generaciones y la solución alcanzada. Realizar gráficos.

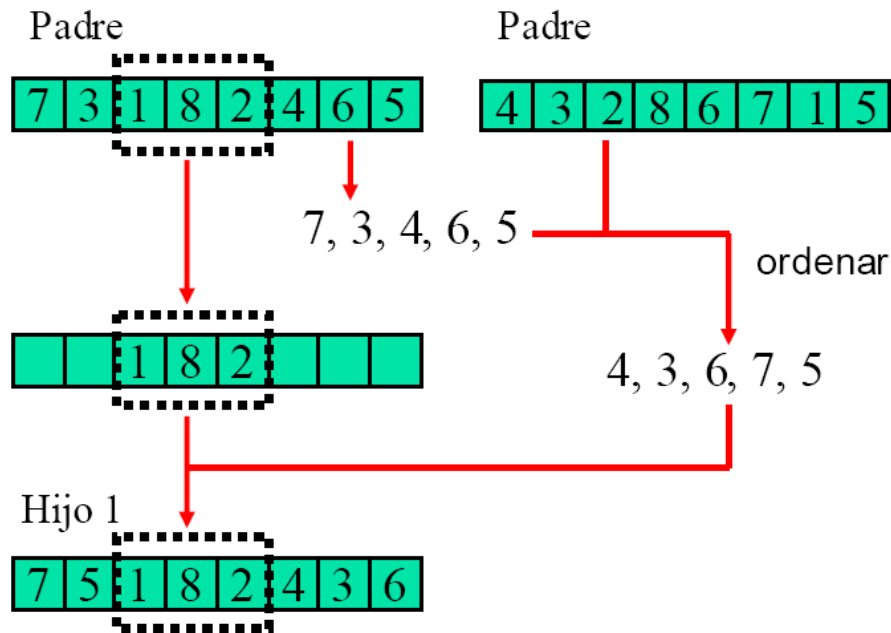


- Población inicial: 30 a 100 individuos. Estudio razonado según similitud de población. Se debe comparar el porcentaje de genes iguales (misma ciudad en misma posición) durante la ejecución del mismo. Para ello se compara el mejor individuo con todos los demás y se muestra en un gráfico la media junto con el valor del mejor individuo de la población.

- Cruce: basado en Orden OX . Se puede implementar el standard o la alternativa a OX en la que se comienza a rellenar en el hijo en la misma posición del 2º padre. Siguiendo esta estrategia, el hijo resultante del ejemplo siguiente quedaría:

43182675

Ejemplo OX standard:



- Cromosoma: Se recomienda utilizar la misma representación y fitness que en la búsqueda local.
- Selección: Torneo ($K = \text{un \% de la población}$) o Ruleta Proporcional. Justificar K
- Reemplazo: (en el caso de estacionario), torneo o crowding determinístico.
- Criterio de parada: 1000 cruces sin mejora(estacionario) o su equivalente en generacional ($1000 \cdot 2 / \text{población}$).

Todos los parámetros numéricos deben ser justificados con al menos 3 ejecuciones sobre Ch130 para cada valor del parámetro (al menos tres valores). Los parámetros no numéricos pueden ser evaluados o bien justificados según lo mostrado en los apuntes sobre la presión selectiva de los operadores selección-reemplazo.

Algoritmo Genético CHC

Aunque el algoritmo CHC fue concebido para cromosomas con codificación binaria, existen versiones para su uso con cromosomas con codificación de orden. El cálculo de la distancia de Hamming se realizará teniendo en cuenta cuantos flujos difieren entre sí (cuantas posiciones difieren). Sólo aquellas cadenas con una distancia (mayor del umbral) serán combinados. El umbral se inicializará a $L/4$ siendo L la longitud de la cadena o cromosoma. Cuando ningún descendiente es insertado en la nueva población el umbral se reduce a 1.

En la fase de recombinación no se aplica ningún proceso de mutación. En su lugar, cuando la población converge o el proceso de búsqueda deja de progresar adecuadamente (el umbral de cruce llega a 0 y no se generan nuevos descendientes), la población se reiniciará. El cromosoma que represente la mejor solución hasta ese momento se utilizará como patrón para generar la nueva población (copiándose), y el resto se inicializarán de forma aleatoria.

La población debe ser menor o igual a la elegida en el genético básico. En el arranque, los valores de un cromosoma corresponden al mejor individuo de la generación anterior, y el resto serán aleatorios.

Haga distintas pruebas para determinar en cada problema, cuántas iteraciones del algoritmo CHC deben ejecutarse.

Algoritmo Genético Multimodal

Considerando como partida el AG Básico, debe implementar un AG multimodal espacial mediante el método de secuencial (5 nichos) o clearing, determinando un radio adecuado basado en la distancia de hamming.

- Donde $d(i,j)$ es la distancia entre las soluciones.
- Tamaño de la población menor o igual al genético básico.
- Probabilidad de cruce 0.8. (si no se cruza se copia como hijo) o implementar elitismo
- Probabilidad de mutación: la calculada en el AG Básico.

Algoritmo de comparación: *Greedy*

Para efectuar la comparativa de resultados, se debe usar el algoritmo *Greedy de la práctica 1*.

	<i>Caso 1</i>		<i>Caso 2</i>		<i>Caso 3</i>	
	<i>Coste</i>	<i>Ev.</i>	<i>Coste</i>	<i>Ev.</i>	<i>Coste</i>	<i>Ev.</i>
Ejecución 1	x	x	x	x	x	x
Ejecución 2	x	x	x	x	x	x
Ejecución 3	x	x	x	x	x	x
Ejecución 4	x	x	x	x	x	x
Ejecución 5	x	x	x	x	x	x
Media	x	x	x	x	x	x
Desv. Típ.	x	x	x	x	x	x

Tabla 1.1

Finalmente, se construirá una tabla global de resultados con la estructura mostrada en la Tabla 1.2. En dicha tabla figura el coste de la solución óptima, el mejor, la media, el peor y la varianza.

Resultados que deben aportarse

El comportamiento de los algoritmos implementados deberá compararse entre sí y con las técnicas *Greedy* y Búsqueda Local de la Práctica 1 de la misma forma que la que se ha realizado en la práctica 2.

A partir de los datos mostrados en estas tablas, el alumno realizará un análisis de los resultados obtenidos, que influirá en la calificación de la práctica. En dicho análisis, se deben comparar, para cada instancia, los distintos algoritmos en términos de eficiencia, calidad de la solución, número de evaluaciones, mejor resultado individual obtenido y mejor resultado medio (robustez del algoritmo), etc.

Se deben mostrar en una gráfica la diferencia entre los valores iniciales de alguna ejecución representativa para cada algoritmo (la misma para todos) y los finales.

Las prácticas se realizarán **individualmente**. Se realizará control de copias con herramientas antiplagio. En caso de detección de copia se suspenderá a ambos alumnos (en el caso de ser ambos alumnos actuales de la asignatura) la totalidad de la asignatura y se delegará en la comisión de docencia el estudio del caso. Si es lícito usar software de prácticas de otras asignaturas, librerías de internet etc..., que cubran partes genéricas como estructuras de datos. En dicho caso habrá que referenciar **claramente en la documentación y código el origen del código reutilizado**.

Fecha y Método de Entrega;

El día 5 de junio durante la sesión de prácticas: Código fuente y documentación (análisis), con una extensión máxima de 4 folios y una pequeña presentación de 2 minutos con los resultados resumidos y las conclusiones del análisis respecto a las elecciones realizadas. Tenga en cuenta que el análisis que realice, cuenta tanto como las implementaciones.

Es **Obligatorio** solicitar al menos una revisión previa de la práctica en clase para evitar desviaciones graves del cometido de la misma. En ella se revisarán las estructuras básicas y los fallos detectados en ellas sólo indicarán la necesidad de repetir la revisión hasta una semana antes de la entrega. Si no se realiza o se corrigen los fallos antes del día de la entrega, en caso de errores graves de concepto puede suponer no superar la práctica.

BIBLIOGRAFIA

[Grefenstette, J., Gopal, R., Rosmaita, B., & Van Gucht, D. \(1985, July\). Genetic algorithms for the traveling salesman problem. In Proceedings of the first International Conference on Genetic Algorithms and their Applications \(Vol. 160, No. 168, pp. 160-168\). Lawrence Erlbaum.](#)