Tema 2. Algoritmos Genéticos. Ejercicios

Abdelmalik Moujahid, Iñaki Inza y Pedro Larrañaga
Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad del País Vasco–Euskal Herriko Unibertsitatea

Diseñar algoritmos genéticos (función de coste, representación de individuos, operadores de cruce y mutación) para los siguientes problemas:

1. *El problema del agente viajero*
   Dada una colección finita de ciudades, determinar la gira de mínimo coste, visitando cada ciudad exactamente una vez y volviendo al punto de partida.
   Sea $n \geq 3$, y $C = (c_{ij})$ una matriz $M(n, n)$ de números reales positivos, encontrar la permutación cíclica π de los enteros de 1 a n que minimice $\sum_{i=1}^{n-1} c_{\pi(i)\pi(i+1)} + c_{\pi(n)\pi(1)}$

2. *El problema de la mochila* $0-1$
   Dado un conjunto finito de items, cada uno de los cuales tiene asocido un peso y una ganancia, seleccionar el subconjunto de items a incluir en una mochila –capaz de soportar un peso máximo finito– cuya inclusión proporcione una ganancia máxima.
   Sean n items y una mochila capaz de soportar un peso máximo C. Denotamos por b_j el beneficio obtenido al introducir el item j en la mochila, mientras que w_j denota el peso asociado a dicho item j. Se trata de seleccionar un subconjuto de items que maximicen $Z = \sum_{j=1}^{n} b_j x_j$, sujeto a la restricción $\sum_{j=1}^{n} w_j x_j \leq C$, siendo
   $$x_j = \begin{cases} 1 & \text{si el item } j \text{ seleccionado} \\ 0 & \text{si el item } j \text{ no seleccionado} \end{cases}$$

3. *El problema de las n reinas*
   En un hipótetico tablero de ajedrez $n \times n$, posicionar n reinas, de tal manera que ninguna ataque al resto.

4. *Problema de agrupamiento I*
   Agrupar N números en k grupos disjuntos minimizando la suma de las diferencias entre los grupos.

5. *Problema de agrupamiento II*
   Disponer los N^2 primeros números naturales en una matriz cuadrada $M(N, N)$ de tal manera que las sumas tanto por filas como por columnas coincidan.

6. *Problema de emplazamiento de bloques rectangulares*
   Dado un conjunto de n bloques rectangulares de distintas anchuras y alturas, se trata de encontrar un emplazamiento –asignación de los centros de los bloques rectangulares a puntos de un espacio cartesiano bidimensional– de tal forma que no haya solapamientos entre los bloques rectangulares, y se minimice la función de coste $f = A + \lambda C$, donde A es el área del rectángulo que engloba todos los bloques rectangulares, λ es una constante positiva y C un término de conectividad $C = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} d_{ij}$, siendo d_{ij} la distancia entre los centros de

los bloques rectangulares y w_{ij} el coste relacionado al unir el bloque rectangular i-ésimo con el bloque rectangular j-ésimo.

7. *Particionamientos de un grafo. Problema "max-cut". Problema "min-cut".*
Dado un grafo $G = (V, E)$ sin ciclos, donde cada arista lleva asociada un peso entero positivo, se trata de encontrar una partición de V en dos conjuntos disjuntos V_0 y V_1 de tal manera que la suma de los pesos de las aristas que tienen un extremo en V_0 y el otro extremo en V_1, sea máximo (*problema max-cut*) o mínimo (*problema min-cut*).

8. *Problema del conjunto de vértices independientes.*
Dado un grafo $G = (V, E)$, se trata de encontrar el denominado conjunto maximal de vértices independientes, es decir el conjunto $VIM \subset V$ de mayor cardinalidad para el cual ninguna de sus aristas se encuentre en E ($\forall u, v \in VIM \Longrightarrow \{u, v\} \notin E$).

9. *El problema de la satisfabilidad. SAT*

   El problema de la satisfactibilidad (SAT) es un paradigma de la NP-completitud. SAT se basa en un conjunto de n variables Booleanas, y una función Booleana, f, tal que:

$$f : B^n \longrightarrow B$$

   siendo $B = \{0, 1\}$. La cuestión es conocer si existe una asignación de las n variables $\mathbf{x} = \{x_1, \ldots, x_n\} \in B^n$ tal que $f(\mathbf{x}) = 1$. En caso de que tal asignación exista el problema se dice satisfacible, mientras que en caso contrario el problema se dice no satisfacible. La función $f(\mathbf{x})$ está en forma normal conjuntiva, es decir:

$$f(\mathbf{x}) = c_1(\mathbf{x}) \wedge c_2(\mathbf{x}) \wedge \ldots \wedge c_m(\mathbf{x})$$

   donde cada una de las m cláusulas es una disyunción de literales, siendo un literal una variable o su negación.

   Los problemas k-SAT se refieren a aquellos problemas SAT en los cuales las m cláusulas están constituidas por disyunciones de k literales. En la formulación MAXSAT la función objetivo se define como el número de cláusulas satisfechas. Es decir

$$f(\mathbf{x}) = c_1(\mathbf{x}) + c_2(\mathbf{x}) + \ldots + c_m(\mathbf{x})$$

   donde

$$c_i(\mathbf{x}) = \begin{cases} 1 & \text{si } la\ i\text{-ésima cláusula se satisface} \\ 0 & \text{en otro caso} \end{cases}$$

   Desarrollar un algoritmo genético para el problema 3-SAT en su formulación MAXSAT.