

**NOMBRE:** Iñaki Diez Lambies

Disponemos de un conjunto de 12 cartas numeradas del 1 a 12. Hay que dividir las cartas en dos pilas, tal que la suma de los números de las cartas de la primera pila, más el producto de los números de las cartas de la segunda pila sea lo más cercano (por exceso o por defecto) a 700. Por ejemplo, si la Pila1 contiene las cartas del 1 al 11 y la Pila2 la carta 12, el valor resultante sería 78.

(8 puntos) Se pide resolver este problema mediante la técnica GRASP con fase de mejora. Concretamente hay que indicar clara y razonadamente:

- El modelo de las soluciones al problema, así como la función de evaluación.
- El proceso de decodificación para obtener la solución real.
- Los parámetros y estructuras de datos necesarias.
- La aplicación de dos iteraciones de la fase constructiva GRASP. Se pide un esquema de los pasos que seguiría el proceso de resolución, con los parámetros correspondientes y la solución
- Un posible proceso de mejora.

#### - Modelo y estructuras de datos necesarios

Para modelar este problema se utilizará un array de 12 valores enteros con valores 0 por defecto donde cada valor  $i$ -ésimo indica si la carta con  $valor=i+1$  se sitúa en la Pila1, en la Pila2 o en ninguna de momento. Por ejemplo, para el ejemplo dado donde la Pila1 contiene las cartas del 1 al 11 y la Pila2 la carta 12 se modelaría tal que:

Índices: 0 1 2 3 4 5 6 7 8 9 10 11  
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2]

El estado inicial sería:  
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Un estado intermedio podría ser:  
[0, 2, 1, 1, 2, 0, 0, 0, 1, 1, 2, 1]

#### - Función de evaluación

La función de evaluación de este problema sería tan simple como recorrer el array y sumar en un contador (iniciado a 0) los  $i+1$  que son igual a 1 y en otro (iniciado a 0 pero que pasa a 1 si existe un valor que multiplicar, para así evitar la multiplicación por 0 y al mismo tiempo evitar sumar 1 cuando no haya ninguna carta en la Pila2) multiplicar los igual a 2 para, posteriormente sumar ambos contadores entre sí. Una vez hecho esto, se deberá restar al valor resultante el valor objetivo (en nuestro caso 700) y realizar el valor absoluto de este resultado para ver cómo de cerca estamos de la solución final.

#### - Decodificación

Descodificar una solución es tan fácil como identificar los índices de la solución que están evaluados a 1 para saber las cartas (número de carta =  $i + 1$ ) que se encuentran en la Pila1 y posteriormente hacer lo equivalente para los igual a 2 para saber cuales se colocan en la Pila2.

### - Aplicación de dos iteraciones

Para simular el proceso aleatorio, se escogerá un proceso pseudoaleatorio donde siempre se seleccionará la carta con valor más bajo de índice en el array y, en caso de empate entre dos pilas, se escogerá la de mayor índice. Se ha escogido un valor de Alpha de 0.05 ya que se ha considerado que

#### 1. Primera iteración

$S = \{[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]\}$ , Alpha= 0.2

$C_{min} = 688$ ;  $C_{max} = 688 + (688*2+689*2+690*2+ \dots +698*2+699*2) * 0.2$  [Alpha] =  $688 + 16644 * 0.2 = 4016.8$

RCL={1.1, 1.2, 2.1, 2.2, 3.1, ..., 11.2, 12.1, 12.2}

Valor carta	Coste de ponerla en pila 1 (cx1)	Coste de ponerla en la pila 2 (cx2)
1	$Abs(1 - 700) = 699$	$Abs(1 - 700) = 699$
2	$Abs(2 - 700) = 698$	$Abs(2 - 700) = 698$
3	697	697
4	696	696
5	695	695
6	694	694
7	693	693
8	692	692
9	691	691
10	690	690
11	689	689
12	688	688

Escogemos pseudoaleatoriamente el valor 1.2 y continuamos.

#### 2. Segunda iteración

Haríamos los mismos pasos anteriormente realizados pero calculando en la tabla de fitness el valor de la solución teniendo en cuenta el coste actual ( en nuestro caso restaríamos 1 al valor de 700 dentro del valor absoluto).

#### - Posible caso de mejora

Un posible caso de mejora puede encontrarse en intentar intercambiar las cartas de un montón a otro para intentar conseguir un mejor valor de la función de fitness, como si se tratara de una “mutación” en el caso de los algoritmos genéticos.

(2 puntos) Explica razonadamente qué modificaciones serían necesarias en la fase constructiva GRASP si la primera pila debiera tener exactamente 10 cartas.

Para conseguir este objetivo podríamos hacer que, a la hora de calcular la función de fitness para los nodos que comiencen a incumplir esta restricción les podemos poner un valor muy alto, cercano a infinito, para que nunca sean escogidos. De forma que, por muy alto que sea el coste de poner la carta en la Pila2 en vez de la Pila1, siempre se escogerá la opción de la Pila2.