Tarea 2 - Representación de Soluciones

Ejercicio 1. Representación binaria para números reales

Existen diferentes métodos para codificar números reales, por ejemplo: números de punto flotante, representación binaria, representación de punto fijo y código de gray.

a) Describe e implementa el algoritmo de representación binaria para mapear (codificar y decodificar) números naturales. Menciona cuántos números son representables con ***m*** bits.

   Por ejemplo, la función (método) puede tener la siguiente firma:
   int [ ] codifica_aux(int n, int nBit)
   - n es el número natural a codificar en binario
   - nBit es la cantidad de bits a utilizar (m)

b) Si se requiere una representación uniforme de números reales en el intervalo [a, b]
   ¿Cómo se generaliza la representación binaria para mapear números reales?, ¿Cuál es la máxima precisión?

   Sugerencia. Considera lo siguiente:
   Se tiene una partición uniforme del intervalo [a, b], por ejemplo:

   $$10 = x_0 < x_1 < x_2 < ... < x_n = 20$$

   Si hacemos n = 10, tenemos una precisión de 0 dígitos decimales (ya que x_i = 10 + i, es decir, la partición uniforme sólo consideraría números enteros).

   En cambio, si n = 100, tenemos una precisión de 1 dígito (p. e. x_25 = 10 + 25(0.1) = 12.5), pero con n=100 no es posible representar 12.52.*

   ¿Qué relación hay entre n, el número de bits y la precisión de la representación?

c) Implementa un algoritmo que codifique números reales en una representación binaria, considerando una partición uniforme sobre un intervalo [a, b], utilizando m bits.
   * La implementación debe usar la función del inciso a).

   Por ejemplo, la función (método) puede tener la siguiente firma:
   int [ ] codifica(double x, int nBit , double a, double b)
   - x representa es el número real que se desea codificar en binario
   - nBit es el número de bits a utilizar (el tamaño que debe tener el arreglo que se devuelve)
   - a, b son los valores del intervalo

d) Implementa un algoritmo para decodificar los vectores de bits como un número real.

   Por ejemplo, la función (método) puede tener la siguiente firma:
   double decodifica(int x_cod[ ], <int nBits>, double a, double b)

   - int x_cod [ ] : es un arreglo que representa el vector de bits que representa la codificación binaria de la solución.
   - int nBits : es el número de bits que se utilizará para representar un número real (corresponde con el tamaño del vector x_cod; en algunos lenguajes este parámetro podría ser opcional).
   - a,b : Se debe cumplir que el número devuelto es un valor en el intervalo [a, b]

e) Implementa las funciones necesarias para codificar y decodificar vectores de números reales

   Por ejemplo, la función (método) puede tener la siguiente firma:
   int [ ] codifica(double x[ ], <int dim_x>, <int nBits>, double a, double b)
   double [ ] decodifica(int x_cod[ ], int dim x, <int nBits>, double a, double b)

Ejercicio 2. Búsqueda por escalada

El problema de COLORACIÓN en grafos consiste en encontrar el mínimo número de colores que se requieren para asignar a cada vértice un color, de manera que dos vértices adyacentes no tengan el mismo color.

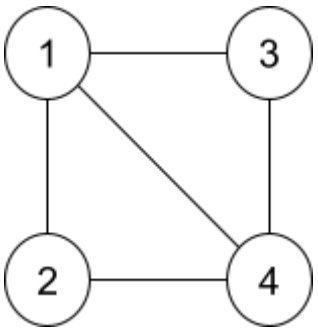
Considera el siguiente esquema de codificación de ejemplares.

a) Implementa la lectura de archivos para leer información de ejemplares (instancias) del problema. Los archivos seguirán el siguiente esquema de codificación:
   - El archivo puede empezar con comentarios, que son líneas que empiezan con el carácter c (y deben ser ignoradas).
   - Justo después de los comentarios, estará la línea “p edge nVertices nAristas” que indica que es un ejemplar que tiene un total de nVertices número de vértices, y nAristas número de aristas.
   - A continuación, siguen nAristas líneas, de la forma “e x y” que indican la lista de aristas. x, y son los índices de los vértices correspondientes. (los índices de los vértices siempre empiezan en 1).

Ejemplo:
c Archivo: prueba1.col
p edge 4 5
e 1 2
e 1 3
e 1 4

e 2 4
e 3 4

El ejemplar correspondiente sería el siguiente:



b) Describe e implementa un esquema de representación de soluciones
Justifica la elección del tipo de representación. Describe las características del esquema de representación de soluciones propuesto (tamaño del espacio de búsqueda, directa o indirecta, lineal o no lineal, tipo de mapeo, factibilidad de soluciones, ¿representación completa?, etc)

c) Describe e implementa una función de evaluación para las soluciones.

d) Describe e implementa un generador de soluciones aleatorias.

e) Describe e implementa una función u operador de vecindad, acorde al tipo de representación implementado en los incisos anteriores.

f) Propón y codifica algunos ejemplares de prueba y prueba tu implementación.
En el reporte deberás agregar:

- Nombre del ejemplar de prueba (archivo)
- Representación gráfica del ejemplar
- Ejemplo de solución aleatoria generada
- Evaluación de la solución aleatoria
- Ejemplo de la aplicación del operador (o función) de vecindad

* Agrega al menos un ejemplar que sea suficientemente corto para poder representarlo en una gráfica pequeña, con al menos 5 vértices y 6 aristas.
** Agrega al menos un ejemplar, suficientemente grande (al menos 10 vértices y 15 aristas), que solo esté representado en archivos. Deberás adjuntar el archivo que codifica el ejemplar, y un archivo que codifique la solución.

Ejercicio 3. Preguntas de repaso

1. Menciona algún ejemplo de representación de soluciones con codificación indirecta.

2. Para cada una de los siguientes tipos de representaciones, proponer un problema de optimización combinatoria e ilustrar la representación de soluciones con un ejemplar concreto; deben ser problemas diferentes a los vistos en clase.
    a. Codificación Binaria
    b. Vector de valores discretos
    c. Permutaciones

    En cada caso se debe indicar claramente:
    - El espacio de búsqueda
    - La función objetivo
    - Tamaño del espacio de búsqueda
    - Ejemplar concreto del problema
    - Ejemplo de una solución, codificada con la representación correspondiente.
    - ¿Qué tipo de mapeo induce la representación?

Consideraciones Generales

- En el reporte deberán incluir algún comentario / discusión sobre la representación utilizada en cada uno de los problemas.

El entregable para esta tarea deberá ser un archivo zip con la siguiente estructura:

+ # cuenta / <--- Nombre de la carpeta
  * El número de cuenta puede ser el de cualquiera de los integrantes del equipo
    - src / <--- carpeta con el código fuente de su implementación
    - output / <--- carpeta con ejemplos de las soluciones de prueba generadas

    - README.txt <--- Archivo con instrucciones para compilar y ejecutar Se debe incluir el comando para ejecutar un ejemplo de cada inciso
    - makefile <-- [Opcional]
    - reporteT2.pdf <--- Reporte de la Tarea
    - ejecuciones.csv <-- [Opcional] Hoja de Cálculo con la información de las ejecuciones realizadas.

El reporte (reporteT2.pdf) deberá incluir al menos:
- Nombre completo
- Título y número de Tarea
- Respuestas a los ejercicios planteados
- Comentarios / Conclusiones

El formato para el reporte es libre. Pueden usar word, latex, o cualquier otro procesador; es obligatorio que se incluya el archivo pdf. No hay extensión mínima ni máxima, aunque se sugiere considerar un reporte entre 3-5 páginas, pero deben incluir las respuestas / comentarios que se piden en cada uno de los ejercicios.