Modelos Bioinspirados y Heurísticas de Búsquedas
4º curso Grado Ingeniería en Informática
Área de Ciencias de la Computación e Inteligencia Artificial
Departamento de Tecnologías de la Información

# PRÁCTICA 2 (Versión 2022, 2.0)

# Algoritmos heurísticos no constructivos

# búsqueda VNS y genéticos

Los cambios respecto a V 1 están resaltados

**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 conjutno de ellas para a traves de operaciones de transformación ir mejorando las soluciones candidatas. Los algoritmos a estudiar son la búsqueda VNS y los genéticos: básico, multimodal y CHC. Deberán compararse con los resultados de la búsqueda local primer vecino de la práctica 1.

Se deberá ejecutar con al menos 2 semillas. Dada la complejidad computacional de los algoritmos se admitirá modificar los criterios de parada para que la ejecución del algoritmo no supere 30 minutos de ejecución en la experimentación final.

**VNS**

El algoritmo VNS que consideraremos se compone de los siguientes pasos:

1. Generar la solución actual aleatoria $S_{act}$ y hacer $k = 1$, $bl = 0$
2. Si ($k > k_{max}$) hacer $k = 1$.
3. Generar una solución vecina ($S_{vec}$) de $S_{act}$ con el operador de generación de vecino para el valor del parámetro $k$, $S_{vec} = N_k(S_{act})$.
4. Aplicar la Búsqueda Local sobre la solución $S_{vec}$, obteniendo $S'$. Hacer $bl = bl + 1$.
5. Si $S'$ es mejor que $S_{act}$, hacer $S_{act} = S'$ y $k = 1$. Si no, hacer $k = k + 1$.
6. Si ($bl < bl_{max}$) volver a 2. Si no, devolver $S_{act}$ y terminar.

Existen dos posibles implementaciones de la condición de parada:

- Estrictamente según los apuntes, cuando se llega a k > kmax
- O se sigue intentando un número de búsquedas locales aunque se haya llegado a kmax (la que está arriba descrita). Esta estrategia es una mejora propuesta ya que la sublista no genera siempre los mismos vecinos para la misma k, por lo que es posible que si logre avanzar aunque llegue a Kmax si intentamos una nueva mutacion fuerte y una búsqueda

*El operador de generación de vecino* hará uso de una *Sublista Aleatoria de Tamaño Fijo, donde se redistribuyen entre los distintos elementos de la sublista elementos de otras estaciones de la misma. Esto se puede realizar mediante n intercambios estaciones en pares dentro de la sublista l .*

Valores de los Parámetros

El algoritmo de búsqueda local será el prime mejor con los parámetros de la práctica 1. Se trabajará con $k_{max} = 4$, es decir, cinco entornos diferentes. El valor de $s$ (tamaño de la sublista) en el operador de generación de vecino definirá el tamaño del movimiento según el valor de $k$. Irá aumentándose de la siguiente forma:

- $k = 1$: Se aplica un tamaño de $s = 4$.
- $k = 2$: Se aplica un tamaño de $s = 8$.
- $k = 3$: Se aplica un tamaño de $s = 12$.
- $k = 4$: Se aplica un tamaño de $s = 16$.

# Algoritmos Genéticos

### Algoritmo Genético Básico

. El alumno puede elegir la combinación de operadores y modalidad que crea mas oportuna. Justifique la elección para obtener resultados adecuados según la teoría si es el caso:

- **Tipo**: Estacionario/Generacional (en dicho caso elite de 5 individuos)
- **Mutación** de 5% al 20% de los genes son cambiados de promedio en caca cruce, por cruce o por gen; es decir que se puede realizar mutación en cada cruce con esa probabilidad, o dejar sin mutar algunos cromosomas, pero el porcentaje medio debe ser el decidido. Misma configuración que el operador de vecindad en búsqueda local.
- **Población inicial**: de 15 a 30 individuos
- **Cruce**: basado en corte en dos puntos. Se selecciona dos posiciones aleatorias y se generan dos hijo con la combinación de las partes de los padres que determinan estos puntos. **Este operador es crítico:**
- **1) Si es muy disruptivo hijos (muy diferentes de los padres) , no generará buenos fitness . Se puede limitar la longitud del trozo copiado para hacerlo menos disruptivo.**

  Tb es posible implementar un cambio de N posiciones , ya que la contigüidad de dos estaciones no significa nada en nuestra representación y el cruce uniforme y en dos puntos es similar.

- **2) Debe generar cromosomas válidos. Para ello debe desarrollarse una estrategia que mantenga los individuos por encima de 205 plazas.**

  Otra forma de tratar los individuos por debajo de este límite es asignándoles un valor alto, de esta manera entran en la población, pero con pocas posibilidades de prosperar. Esto sólo es razonable si el número de inválidos que ocurren es relativamente bajo.

- **Cromosoma**: Misma representación
- **Fitness** se va a ampliar sobre la práctica 1 aplicando un coste al número de plazas, para evitar que mediante un aumento de las mismas se consigan valores no realistas. Las plazas se multiplican por un factor de coste que deberá estimar el alumno para que evoluciones de manera que las plazas se acerquen a los 200-220.

- **Selección**: Torneo (K= % de la población, con un mínimo de 3 individuos), Ruleta Proporcional. Crowding o aleatorio (sólo si estacionario)
- **Reemplazo**: (sólo si estacionario), torneo (K= % de la población, con un mínimo de 3 individuos) /aleatorio.

**Algoritmo Genético CHC**

Aunque el algoritmo CHC fue concebido para cromosomas con codificación binaria, existen versiones para su uso con cromosomas con codificación en vector. El cálculo de la distancia de Hamming se realizará teniendo en cuenta cuantos genes 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.

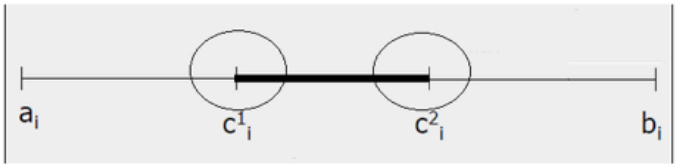
La población debe ser menor o igual a la elegida en el genético básico. En el arranque, los valores de un cromosoma corresponden al mejor individuo de la generación anterior, y el resto serán aleatorios.

Se pueden poner más copias del mejor individuo para acelerar el proceso de reinicio si este es muy lento tras la primera reinicialización

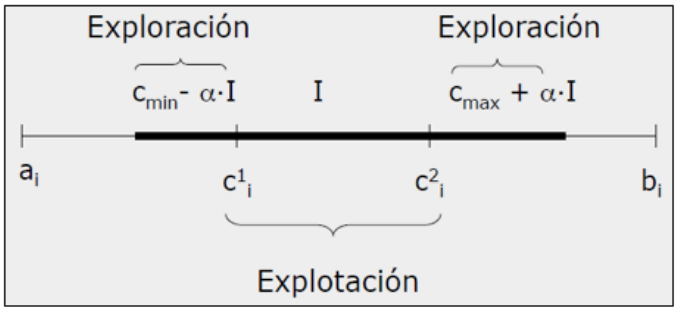
Se puede utilizar el mismo cruce que en el básico, pero hay que añadir artificialmente la mutación, ya que éste cruce no cambia los números de las estaciones para una implementación mas realista habría que utilizar uno de los siguientes cruces:

## Cruces

**Parent Centered**




**BLX-Alfa**




Los cruces en CHC se hacen cambiando exactamente la mitad de los elementos **QUE SON DISTINTOS en ambos padres y luego aplicando Parent Centered (una mutación gausiana del valor intercambiado) o BLX-Alpha (que genera un valor para cada hijo aplicando el algoritmo descrito en los apuntes) .**

**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**.(distancia Hamming)**
- Tamaño de la población menor o igual al genético básico.
- Probabilidad de cruce 0.8. (si no se cruza se copia como hijo )
- ~~Probabilidad de mutación por gen 0.01-0.05.~~
- Mantener misma mutación que en el algoritmo básico.

El valor del radio tanto para el secuencial como para el clearing es crítico y debe ser propuesto por el alumno para que las soluciones finales sean suficientemente diferentes.
Un radio muy pequeño no afectará casi nunca a los individuos al realizar el clearing o adaptar el fitness en el conjunto de soluciones y un valor muy alto hará que todos estén penalizados.

En clearing es conveniente dejar un número de generaciones entre operaciones de aclarado, este parámetro denominado P, en el el generacional P iteraciones y en el estacionario p*Población/2 ya que una población equivalente se genera cada ese número de cruces.
El kappa (numero de representantes de un nicho tiene que ser >= 2, para evitar que en un clearing quede sólo un elemento si los demás están en el mismo nicho y así poder generar el resto de los individuos hasta rellenar la población cruzando los que hayan quedado del aclarado.

**Metodología de Comparación**

El alumno tendrá que contabilizar el número de evaluaciones (llamadas realizadas a la función de coste) producidas por los distintos algoritmos, que será empleado como una medida adicional de comparación de su calidad.

A partir de la experimentación efectuada, se construirá una tabla global de resultados con la estructura mostrada en la Tabla mostrada.. La columna etiquetada con *Mej* indica el valor de la mejor solución encontrada, la columna σ refiere a la desviación típica y la columna etiquetada con *Ev y Ev. Mejor* indica respectivamente el número medio y mínimo de evaluaciones realizadas por el algoritmo en las cinco ejecuciones (salvo en el caso del algoritmo *Greedy* que sólo se ejecuta una vez).

| **Algoritmo** | **Ev.Medias** | **Ev. Mejor** | **σ EV** | **Mejor Fitness** | **Media fitness** | **σ fitness** |
|---|---|---|---|---|---|---|
| Busqueda local | | | | | | |
| Búsqueda VNS | | | | | | |
| G. Básico | | | | | | |
| G. Multimodal | | | | | | |
| CHC | | | | | | |

A partir de los datos mostrados en estas tablas, el alumno realizará un **análisis de los resultados obtenidos**, **que influirá de forma decisiva en la calificación de la práctica**. En dicho análisis, se deben comparar, para cada instancia, las distintas variantes de los algoritmos en términos de: número de evaluaciones, mejor resultado individual obtenido y mejor resultado medio (robustez del algoritmo).

Las prácticas se realizarán individualmente.

**Fecha y Método de Entrega;**

El día 16 de Mayo durante la sesión de prácticas. Debe entregar 1 fichero comprimido ZIP, que contenga:

- Documento DOC (MS Word) ó PDF con las tablas de resultados y análisis.
- Ficheros de código fuente completo ejecutable utilizado.
- Scripts, si los ha utilizado.
- Se realizará una defensa de la práctica explicando cada una de las decisiones tomadas en el código. La evaluación tendrá en cuenta:

**Nivel Básico (4):**

- Corrección del algoritmo
- Utilización de la terminología apropiada (la utilizada en el material de clase)

**Nivel intermedio (5-6):**

- Análisis e interpretación sobre la capacidad de exploración/explotación de cada algoritmo relacionado con los resultados obtenidos.
- Gráficas y ejemplos concretos comparados.

**Nivel Alto (7-8)**

- Mejoras en el rendimiento de la aplicación
- Resultados obtenidos razonablemente cercanos al optimo
- Análisis de la estadística de los resultados. Algoritmos más estables. Desviación típica.

**Nivel Avanzado (9-10):**

- Función de simulación que muestre la evolución.
- Guardado en Fichero
- Función de fitness mejorada
- Solución final dibujada

Permanezca atento a posibles nuevas versiones mejoradas de este documento.