

# Técnicas heurísticas bio-inspiradas en la optimización

## TAREA 3

*Fecha de entrega: viernes 12 de octubre 2018*

Dr. Eric Rincón

Trimestre 18-O

**Observación 1:** La tarea deberá realizarse de forma individual incluyendo: a) un archivo en formato pdf con la descripción de los resultados obtenidos, b) el código fuente de los programas realizados para resolver los ejercicios. La tarea deberá enviarse en un archivo zip cuyo nombre siga la siguiente estructura: Tarea\_3\_Nombre\_Apellido.

**Observación 2:** A partir de esta tarea se deberán realizar la menos 30 corridas con los algoritmos desarrollados, por lo cual es necesario que defina las 30 semillas que empleará para iniciar el generador de números aleatorios en cada una de las corridas. Para realizar la selección de las 30 semillas no es necesario que siga algún procedimiento específico, pero deberá usar las mismas durante el resto del curso.

### Ejercicio 1

Deberá presentar un resumen de al menos una cuartilla explicando los puntos más importantes del artículo: Stelios H. Zanakis, James R. Evans, (1981). *Heuristic "Optimization": Why, When, and How to Use It*. Interfaces 11(5):84-91. <http://dx.doi.org/10.1287/inte.11.5.84>

### Ejercicio 2

Escriba un programa para generar soluciones al problema del sudoku. El programa deberá cumplir con los siguientes puntos:

- En todo momento respetará los valores predefinidos para el sudoku que se está resolviendo.
- Generará una solución inicial de forma aleatoria.
- Generará soluciones vecinas que serán aceptadas, o rechazadas, de acuerdo al criterio *first improvement* de la técnica *hill climbing*.
- Se empleará la misma función objetivo dada en el artículo: Sergio G. de los Cobos Silva, Miguel A. Gutiérrez Andrade, Jhon G. Close, (2012). *Solución al acertijo del SUDOKU con recocido simulado*. ContactoS 84:31-39. <http://www2.izt.uam.mx/newpage/contactos/download/contactos84.pdf>

Para este problema se considerarán dos vecindarios diferentes.

- **Vecindario 1:** Para un sub-cuadrado, elegido de forma aleatoria, se generará una permutación aleatoria de los números 1,...,9 que sustituirá a los valores de la solución actual.
- **Vecindario 2:** Para un sub-cuadrado, elegido de forma aleatoria, se seleccionarán, de forma aleatoria, dos valores numéricos que permutarán sus posiciones.

Para evaluar el desempeño de su algoritmo, resolverá con ambos vecindarios las instancias **Easy a**, **Medium a**, **Hard a** y **Hardb**, que encontrará en la página <http://lipas.uwasa.fi/~timan/sudoku/>.

Para cada instancia a resolver, deberá:

1. Efectuar al menos 30 corridas definitivas de su algoritmo, para determinar datos estadísticos sobre las corridas (mejor/peor solución obtenida; promedio y desviación estándar calculadas sobre las 30 corridas; cualquier elemento que le parezca relevante).
2. Comparar los resultados obtenidos por los 2 vecindarios propuestos.

El reporte de su trabajo deberá ser similar a un artículo para un evento especializado en el cual presentaría los resultados de su investigación. Por lo tanto, deberá incluir:

- Una definición breve sobre el problema (variables, restricciones, función objetivo elegida, codificación, ...).
- Una presentación breve del paradigma de resolución utilizado, en este caso *hill climbing*.
- Una descripción de los procesos internos del método que debieron ser ajustados para resolver el problema propuesto en esta tarea (y cualquier información que le parezca relevante sobre su implementación).
- Un resumen de los resultados (resultados definitivos y datos estadísticos sobre las instancias pedidas), comparando el desempeño de los vecindarios propuestos. Deberá incluir la mejor y la peor solución devuelta por el algoritmo, indicando su costo.

## Ejercicio 3

Deberá implementar la estrategia *hill climbing (first improvement)* para realizar un proceso de optimización en las funciones de prueba indicadas en la **Tarea 2**, para 5 dimensiones. El algoritmo deberá:

1. Generar soluciones iniciales aleatorias distribuidas de forma uniforme en todo el rango de la función.
2. Se emplearán los siguientes vecindarios.
  - (a) **Vecindario 1:** Considere que la solución actual está dada por  $(x_1, \dots, x_n)$ . Seleccione una variable al azar,  $x_i$ , y genere un valor aleatorio,  $r \in (-1, 1)$ . La solución vecina mantendrá las mismas coordenadas que la solución actual excepto por  $x_i$  que ahora estará dada por  $x_i + r$ .
  - (b) **Vecindario 2:** Considere que la solución actual está dada por  $(x_1, \dots, x_n)$ . Genere  $n$  valores aleatorios,  $r_1, \dots, r_n \in (-0.5, 0.5)$ . La solución vecina estará dada por  $(x_1 + r_1, \dots, x_n + r_n)$ .

Para cada instancia a resolver, deberá:

1. Efectuar al menos 30 corridas definitivas de su algoritmo, para determinar datos estadísticos sobre las corridas (mejor/peor solución obtenida; promedio y desviación estándar calculadas sobre las 30 corridas; cualquier elemento que le parezca relevante).
2. Comparar los resultados obtenidos por los 2 vecindarios propuestos.

El reporte de su trabajo deberá ser similar a un artículo para un evento especializado en el cual presentaría los resultados de su investigación. Por lo tanto, deberá incluir:

- Una definición breve sobre el problema (variables, restricciones, codificación, ...).
- Una presentación breve del paradigma de resolución utilizado, en este caso *hill climbing*.
- Una descripción de los procesos internos del método que debieron ser ajustados para resolver el problema propuesto en esta tarea (y cualquier información que le parezca relevante sobre su implementación).
- Un resumen de los resultados (resultados definitivos y datos estadísticos sobre las instancias pedidas), comparando el desempeño de los vecindarios propuestos. Deberá incluir la mejor y la peor solución devuelta por el algoritmo, indicando su costo.