

# Tarea 1

Luis Fernando Torres Hernández

Técnicas de optimización heurísticas

November 07, 2024

# Implementación de Búsqueda Tabú para Clustering

En esta práctica se desarrolló una implementación de la metaheurística de búsqueda tabú con el objetivo de realizar un análisis de clustering sobre un conjunto de datos botánicos. El conjunto de datos utilizado corresponde a mediciones morfológicas de 150 especímenes pertenecientes a tres especies distintas del género Iris.

Se utilizaron las mediciones morfológicas de longitud y ancho de sépalos y pétalos para caracterizar cada espécimen. La búsqueda tabú se empleó para inicializar los centroides de los clusters y para refinar de forma iterativa una partición inicial aleatoria.

## Metodología

Se implementó un algoritmo de clustering jerárquico utilizando la metaheurística de búsqueda tabú. Los parámetros utilizados fueron los siguientes:

- Lista tabú: Longitud fija de 7 elementos.
- Vecindario: Se generaron 20 vecinos en cada iteración, se observaron mejores resultados al aumentar la vecindad de 15 a 20.
- Función objetivo: Distancia Manhattan entre cada punto y su centroide asignado.
- Criterio de aspiración: Aceptación de soluciones que están en tabú si esta era mejor que la mejor solución encontrada hasta el momento.

## Resultados

Para evaluar el impacto del número de iteraciones en el rendimiento del algoritmo, se realizaron 50 ejecuciones independientes para cada uno de los siguientes valores de iteraciones: 50, 100, 200 y 500. Para cada ejecución, se registró el valor de la función objetivo al finalizar la búsqueda.

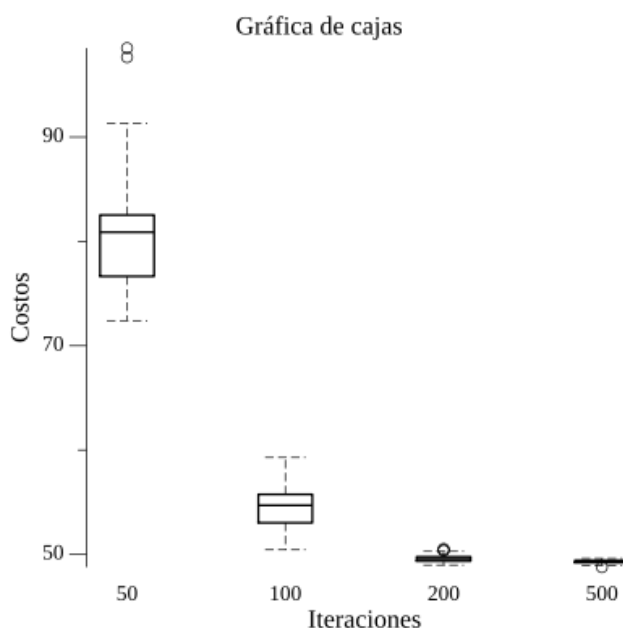


Figura 1: Diagrama de cajas

Los resultados muestran una tendencia clara: a medida que aumenta el número de iteraciones, el valor promedio de la función objetivo disminuye, indicando una mejora en la calidad de las soluciones. Sin embargo, el diagrama de cajas (Figura 1) revela que la variabilidad entre las ejecuciones también disminuye a medida que aumenta el número de iteraciones, sugiriendo una convergencia hacia una solución óptima.

Sin embargo, no se observaron diferencias significativas entre las soluciones obtenidas con 200 y 500 iteraciones, lo que sugiere que un número de iteraciones de 200 puede ser suficiente para obtener soluciones de alta calidad en este problema.

La variabilidad y los valores atípicos observados en los resultados pueden atribuirse en parte a la naturaleza estocástica de la búsqueda tabú y a la influencia de la inicialización aleatoria. A pesar de esta variabilidad, los resultados obtenidos son consistentes con la hipótesis de que un mayor número de iteraciones permite explorar un espacio de soluciones más amplio y encontrar soluciones de mejor calidad.»

## Compilación y ejecución del código

Para compilar y ejecutar el proyecto, es necesario tener instalado Go (Golang) versión 1.22 o superior. Los siguientes comandos deben ejecutarse desde el directorio raíz del proyecto:

Compilación:

```
go get
go build
```

El primer comando (**go get**) descargará las dependencias necesarias para el proyecto. El segundo comando (**go build**) compilará el código fuente y generará un ejecutable llamado **tarea1** si el sistema operativo es tipo Unix o **tarea1.exe** en Windows.

Ejecución:

- En sistemas tipo Unix

```
./tarea1
```

- En Windows

```
tarea1.exe
```

Al ejecutar el programa, se realizarán 200 ejecuciones independientes del algoritmo de búsqueda tabú. Al finalizar, se mostrará en la consola la mejor solución encontrada entre todas las ejecuciones. Además, se generará un archivo PNG llamado **boxplot.png**, que contiene un diagrama de cajas que muestra la distribución de los valores de la función objetivo obtenidos en cada ejecución.