**Memoria Práctica 2**

Exploración en Espacios de Estados y Computación Evolutiva



**Autores:**

Autor 1

Autor 2

Autor 3

Autor 4

**Inteligencia Artificial 2023-2024**

Universidad Politécnica de Madrid

E.T.S. de Ingeniería en Sistemas Informáticos

Departamento de Sistemas Informáticos

# 1. Autores y trabajo realizado

Se deberá completar la tabla que aparece a continuación con la información de cada uno de los autores de la práctica. Se deberá indicar el porcentaje de dedicación estimado (si se considera que los 4 autores han dedicado el mismo tiempo/esfuerzo, será de 25% para cada uno). También se indicará el trabajo realizado por cada uno de los autores (código desarrollado en cada ejercicio, experimentación realizada, análisis de resultados, escritura de la memoria, etc).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nº Matricula | Nombre | Apellidos | Porcentaje dedicación | Trabajo realizado |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# 2. Ejercicio 1: Branch and Bound

## 2.1. Representación del problema

Indicar aquí cómo se han representado los **estados** y las **transiciones** del problema de planificación de tareas con recursos, expresándolo con fórmulas matemáticas si fuera necesario. Indicar también cómo se ha representado el estado inicial y final (o la función **solución**), así como la función de **coste**.

Indicar qué funciones se han desarrollado en el código para esta funcionalidad.

## 2.2. Implementación de Branch and Bound

Indicar qué funciones han sido necesarias desarrollar en el código para la implementación de este algoritmo.

## 2.3. Resultados

Indicar para cada problema si ha sido posible la ejecución de Branch and Bound, y mostrar en una tabla los resultados obtenidos a nivel de calidad (makespan de la solución obtenida) y velocidad (tiempo que tarda el algoritmo en encontrar la solución).

# 3. Ejercicio 2: A\*

## 3.1. Heurística(s)

Indicar aquí qué heurística(s) se ha(n) empleado, expresándolo con fórmulas matemáticas si fuera necesario, y justificar y demostrar formalmente si cumple(n) o no la condición de minorancia.

Indicar qué funciones se han desarrollado en el código para esta funcionalidad.

## 3.2. Implementación de A\*

Indicar qué funciones han sido necesarias desarrollar en el código para la implementación de este algoritmo.

## 3.3. Resultados

Indicar para cada problema si ha sido posible la ejecución de A\*, y mostrar en una tabla los resultados obtenidos a nivel de calidad (makespan de la solución obtenida) y velocidad (tiempo que tarda el algoritmo en encontrar la solución).

# 4. Ejercicio 3: Algoritmo genético básico

## 4.1. Representación y fitness

Indicar aquí qué codificación se ha usado (utilizando representaciones gráficas si fuera necesario) y qué función de fitness se ha empleado (expresándola con fórmulas matemáticas) para la resolución del problema de planificación de tareas con recursos.

Indicar qué funciones se han desarrollado en el código para esta funcionalidad.

## 4.2. Implementación del algoritmo genético básico

Indicar qué funciones han sido necesarias desarrollar en el código para la implementación de este algoritmo.

## 4.3. Resultados

Representar gráficamente la evolución generacional del algoritmo genético en función de la fitness y/o del genotipo, y discutir cómo se comportan la convergencia y la diversidad para este método.

Representar en una (o varias) tabla(s) los resultados obtenidos por el algoritmo genético a nivel de calidad y velocidad del algoritmo (recordar que estos algoritmos son estocásticos y una sola ejecución no es representativa). Describir el proceso realizado para la obtención de la tabla, y discutir si los resultados obtenidos son satisfactorios.

# 5. Ejercicio 4: Algoritmo genético avanzado

## 5.1. Representación y fitness

Indicar aquí qué codificación se ha usado (utilizando representaciones gráficas si fuera necesario) y qué función de fitness se ha empleado (expresándola con fórmulas matemáticas) para la resolución del problema de planificación de tareas con recursos utilizando el algoritmo genético avanzado.

Indicar qué funciones se han desarrollado en el código para esta funcionalidad.

## 5.2. Operadores genéticos

Indicar aquí qué operadores genéticos (generación, selección, cruce, mutación, criterio de parada, selección ambiental, etc.) se han usado e implementado para el algoritmo genético avanzado. Incluir todos aquellos que se hayan usado en el proceso aunque no formen parte del algoritmo final entregado en el código.

Indicar qué funciones han sido necesarias desarrollar en el código para la implementación de estos operadores.

## 5.3. Implementación del algoritmo genético avanzado

Indicar qué funciones han sido necesarias desarrollar en el código para la implementación de este algoritmo.

## 5.4. Experimentación y resultados

Indicar qué experimentos con distintos operadores e hiperparámetros se han llevado a cabo para definir y ajustar el algoritmo genético avanzado final.

Representar gráficamente la evolución generacional del algoritmo avanzado en función de la fitness y/o del genotipo, y discutir cómo se comportan la convergencia y la diversidad para este método.

Representar en una (o varias) tabla(s) los resultados obtenidos por los distintos algoritmos genéticos desarrollados en el proceso de experimentación a nivel de calidad y velocidad de los algoritmos. Describir el proceso realizado para la obtención de la tabla, y discutir si los resultados obtenidos son satisfactorios, y justificar la elección del algoritmo genético final que se ha devuelto en la función *exercise4*.

# 6. Conclusiones

Discutir los resultados obtenidos en los cuatro ejercicios, representando en una tabla una comparativa de los 4 métodos para los distintos problemas, y concluir qué método proporciona mejores resultados justificando la respuesta.