



Universidad de Castilla-La Mancha  
Escuela Superior de Ingeniería Informática

**Trabajo Fin de Grado**  
Grado en Ingeniería Informática  
Computación

**Optimización Multiobjetivo de la asignación  
docente en el Grado de Ingeniería Informática  
para evitar superposiciones y satisfacer  
preferencias de horarios de los estudiantes**

*José Ángel Serrano Pardo*

Junio, 2024





## **TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática

Computación

# **Optimización Multiobjetivo de la asignación docente en el Grado de Ingeniería Informática para evitar superposiciones y satisfacer preferencias de horarios de los estudiantes**

**Autor:** José Ángel Serrano Pardo

**Tutor:** José Miguel Puerta Callejón

Junio, 2024



*Aquí va la dedicatoria que cada cual  
quiera escribir. El ancho se controla  
manualmente*



## Declaración de autoría

Yo, José Ángel Serrano Pardo , con DNI 49800390P, declaro que soy el único autor del trabajo fin de grado titulado “Optimización Multiobjetivo de la asignación docente en el Grado de Ingeniería Informática para evitar superposiciones y satisfacer preferencias de horarios de los estudiantes”, que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual, y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a ... de ... de 20 ...

Fdo.: José Ángel Serrano Pardo





## Resumen

Los algoritmos genéticos son una rama de la inteligencia artificial donde se simula el proceso de la evolución biológica, se usan métodos de búsqueda heurística que utilizan procesos de selección, cruzamiento y mutación para generar soluciones óptimas o cercanas a las óptimas a problemas complejos que son difíciles de resolver mediante técnicas convencionales. A través de iteraciones sucesivas, una población inicial de soluciones potenciales evoluciona hacia una solución óptima o cercana, adaptándose al entorno definido por una función fitness específica. En este estudio, se adapta un algoritmo genético para abordar la optimización multiobjetivo en la asignación de grupos para las matriculas en el Grado de Ingeniería Informática de la Universidad de Castilla-La Mancha. El desafío incluye múltiples objetivos, como maximizar la satisfacción de los alumnos, minimizar el número de solapes entre los horarios de las matriculas de un alumno, maximizar la cohesión de grupos de teoría y práctica para asignaturas del mismo curso en los alumnos y maximizar el equilibrio del número de alumnos en los grupos de teoría y practicas en cada asignatura. El algoritmo genético diseñado utiliza una codificación específica para representar las variables del problema y operadores genéticos adaptados para mantener la diversidad genética en la población, permitiendo así una exploración efectiva del espacio de soluciones.



## Agradecimientos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



## Índice general

---

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introducción</b>                                    | <b>1</b>  |
| 1.1      | Motivación   | 1         |
| 1.2      | Objetivos  | 2         |
| 1.3      | Competencias.  | 2         |
| 1.4      | Estructura del documento                               | 3         |
| <b>2</b> | <b>Estado del arte</b>                                 | <b>5</b>  |
| 2.1      | Escuela superior de ingeniería informática de Albacete | 5         |
| 2.2      | Metaheurísticas evolutivas                             | 10        |
| <b>3</b> | <b>Desarrollo</b>                                      | <b>11</b> |
| 3.1      | Ficheros Iniciales                                     | 11        |
| 3.2      | Modelización del problema                              | 12        |
| 3.2.1    | <i>Representación de los individuos</i>                | 12        |
| 3.2.2    | <i>Objetivos</i>                                       | 13        |
| 3.2.3    | <i>Función de evaluación</i>                           | 17        |
| 3.2.4    | <i>Población Inicial</i>                               | 18        |
| 3.2.5    | <i>Selección</i>                                       | 18        |
| 3.2.6    | <i>Cruce</i>   | 19        |
| 3.2.7    | <i>Mutación</i>  | 21        |
| <b>4</b> | <b>Experimentación</b>                                 | <b>23</b> |
| <b>5</b> | <b>Conclusiones</b>                                    | <b>25</b> |
|          | <b>Referencia bibliográfica</b>                        | <b>27</b> |

---



## Índice de figuras

---

|     |  |    |
|-----|--|----|
| 2.1 | Módulos que forman el grado de Ingeniería Informática . . . . .            | 6  |
| 2.2 | Asignaturas módulo Formación Básica para la Ingeniería . . . . .           | 7  |
| 2.3 | Asignaturas módulo Formación Común a la rama de ingeniería Informática     | 7  |
| 2.4 | Horario 1º cuatrimestre Grupo 1ºA . . . . .                                | 9  |
| 2.5 | Horario 1º cuatrimestre Grupo 2ºA . . . . .                                | 9  |
| 3.1 | Representación de un individuo en el problema de asignación de matrículas. | 13 |





## Índice de tablas

---



## Índice de algoritmos

---



## Índice de listados de código

---



# 1. Introducción

---

## 1.1. Motivación

La gestión de matriculas para asignar grupos es uno de los procesos esenciales y más complejos en cualquier universidad ya que actualmente existe una gran cantidad de alumnos y no todos los grupos son iguales, ya que tienen distintos horarios, siendo algunos por la mañana y otros por la tarde, además de que cada alumno tiene preferencias distintas ya sea porque prefiere un horario temprano o porque le complementa mejor con las asignaturas que tiene u otras actividades. Actualmente los estudiantes analizan muy bien los horarios de las asignaturas antes del proceso de matriculación para calcular cual sería su configuración óptima y tener el menor número de solapes de horarios y menor pérdida de tiempo entre clases, aunque no todos consiguen matricularse en sus grupos ideales debido a que los alumnos no realizan la matrícula a la vez, sino que se hace en distintos días y primero la realizan los estudiantes con mayor nota(referencia), esto puede causar frustración en los alumnos que no consiguen matricularse en el grupo que querían debido a que no quedaban plazas pudiendo afectar a su rendimiento académico u organización personal. Después de realizar las matriculas se tiene que seleccionar grupo de prácticas, en este caso si se realiza al mismo tiempo para todos los alumnos por lo que suele ser complicado obtener el grupo que se desea añadiendo más dificultad a los alumnos en su tarea de conseguir un horario óptimo. Teniendo en cuenta que los grupos tienen que tener una cantidad de alumnos limitada y es imposible satisfacer todas las preferencias de los alumnos de forma individual. Por eso nace la idea de desarrollar un sistema con inteligencia artificial que recibiendo las matriculas que han realizado los alumnos, usando procesos de optimización se haga una asignación de grupos de forma que se intente favorecer a los alumnos teniendo el menor número de solapes de horarios en asignaturas de un alumno y reducir el tiempo muerto entre clases para cada alumno, todo esto manteniendo el equilibrio de alumnos por grupo en las asignaturas e intentado mantener cierto parecido con la configuración original.

---

## 1.2. Objetivos

1. Analizar el problema de la asignación docente:
  - Identificar y analizar los principales factores y restricciones que afectan a la asignación docente, como las preferencias de los alumnos, las restricciones horarias, las plazas limitadas en los grupos de teoría y practicas.
2. Diseñar un modelo de optimización multiobjetivo:
  - Desarrollar un modelo matemático que represente el problema de la asignación docente, donde este explicados todos los parámetros a optimizar y sus restricciones.
3. Implementar un algoritmo genético adaptado:
  - Configurar e implementar un algoritmo genético específico para resolver el modelo de optimización, incluyendo la codificación de soluciones, la función de fitness, y los operadores de selección, cruzamiento y mutación.
4. Realizar experimentos y evaluar el rendimiento del algoritmo:
  - Diseñar y llevar a cabo una batería de experimentos para probar el algoritmo genético con distintas configuraciones y en escenarios diferentes comparando el rendimientos y la evolución del tiempo de ejecución.
5. Analizar los resultados y validar el modelo:
  - Evaluar los resultados obtenidos, analizando la eficacia y eficiencia del algoritmo genético en la optimización de la asignación docente usando varias métricas para determinar cual es la mejor configuración que puede tener este algoritmo genético para obtener la mejor solución.

## 1.3. Competencias

En este proyecto se han trabajado las siguientes competencias.

- Comunes a la rama de la informática
  - **[CO6]** Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
  - **[CO7]** Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados para la resolución de un problema.
  - **[CO14]** Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.



- **[CO15]** Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.
- Tecnología específica. Computación.
  - **[CM3]** Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
  - **[CM4]** Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.
  - **[CM5]** Capacidad para adquirir, obtener, formalizar y representar el conocimiento humano en una forma computable para la resolución de problemas mediante un sistema informático en cualquier ámbito de aplicación, particularmente los relacionados con aspectos de computación, percepción y actuación en ambientes o entornos inteligentes.

## 1.4. Estructura del documento

La estructura de los capítulos de la memoria es la siguiente.

- Introducción
  - Presenta el contexto y la motivación del estudio, describe los desafíos de la asignación docente y establece los objetivos generales y específicos del trabajo.
- Estado del arte
  - Revisa el contexto del Grado Universitario en Ingeniería Informática, el funcionamiento del sistema de matrículas y la organización de los grupos. Proporciona una visión general de las metaheurísticas evolutivas y una revisión detallada de los algoritmos genéticos y su aplicación en problemas similares.
- Desarrollo
  - Detalla el diseño del algoritmo genético, incluyendo la representación de soluciones, la función de aptitud y los operadores genéticos utilizados. Describe la implementación del algoritmo en un entorno de desarrollo específico.
- Experimentación
  - Describe la configuración experimental, los escenarios de prueba, los métodos de evaluación y presenta los resultados obtenidos. Incluye una discusión sobre la eficacia del algoritmo

---

## ■ Conclusiones

- Resume los hallazgos principales del estudio, discute la contribución del trabajo y su impacto en la mejora de la asignación docente. Propone posibles mejoras al algoritmo y sugiere direcciones para futuras investigaciones.

## 2. Estado del arte

---

En este capítulo se describe el funcionamiento de las matriculas en la Escuela Superior de Ingeniería Informática de Albacete y el tipo de algoritmos genéticos que se pueden usar para optimizar la asignación de grupos en las matriculas de los estudiantes.

### 2.1. Escuela superior de ingeniería informática de Albacete

Las titulaciones en la escuela superior de ingeniería informática de Albacete comienzan en el año 1985 con los estudios de diplomado en informática, mas tarde en el curso académico 1992-93 se convierten en los títulos de Ingeniería Técnica de Sistemas e Ingeniería Técnica de Gestión de 3 años cada uno para terminar incluyendo la titulación de Ingeniería Informática de 5 años de duración en el curso 1990-19. Estas 3 titulaciones continuaron hasta que en el curso 2010-11 se implanto el actual grado de Ingeniería Informática adaptándose al espacio europeo de educación superior [European Commission, 2024] y sustituyendo a las 3 titulaciones anteriores. Además en la escuela se imparten varios títulos de postgrado como el master universitario en ingeniería información y el doctorado en tecnologías informáticas avanzadas aparte de otros títulos propios como el máster en sistemas informáticos embarcados y el máster en ciberseguridad y seguridad de la información.

Este trabajo se centra sobre el actual grado de Ingeniería Informática, por lo que vamos a ver como está organizado su plan de estudios y que tipos de asignaturas tienen y como funcionan las matriculas para este grado y como afectan para nuestro problema, toda esta información la vamos a ver en la web oficial de la escuela [Escuela Superior de Ingeniería Informática de Albacete, 2024].

En la web nos centraremos en el apartado que existe para el grado, dentro de ese apartado hay 2 secciones que son las que más nos interesan, la primera es la sección que nos habla del plan de estudios, lo primero que podemos encontrar es un bloque donde se agrupan todos los módulos que tiene el grado y los créditos que aporta cada módulo.

---

|            |   |                         |                      |                                  |
|------------|---|-------------------------|----------------------|----------------------------------|
| ECTS<br>12 | Trabajo Fin de Grado                                |                         |                      |                                  |
| ECTS<br>24 | Optatividad   |                         |                      |                                  |
| ECTS<br>48 | Computación   | Ing. de<br>Computadores | Ing. del<br>Software | Tecnologías de la<br>Información |
| ECTS<br>96 | Formación Común a la rama de Ingeniería Informática |                         |                      |                                  |
| ECTS<br>60 | Formación Básica para la Ingeniería                 |                         |                      |                                  |

**Figura 2.1:** Módulos que forman el grado de Ingeniería Informática

Tenemos los módulos de Formación Básica para la Ingeniería y Formación Común a la rama de ingeniería Informática, estos módulos son obligatorios para todos los alumnos, luego tenemos el tercer módulo que está dividido en las 4 menciones que tiene el grado, es obligatorio que se aprueben todas las asignaturas de una mención para completar el grado, por último tenemos el módulo de optatividad que está formado por distintas asignaturas de libre elección y el módulo del trabajo fin de grado. De todos estos módulos tenemos 2 que tienen más relevancia y son los 2 primeros, esto es porque el bloque de asignaturas que forman estos módulos son las asignaturas que tienen más de un grupo, esto es un punto crucial ya que el resto de asignaturas al tener un solo grupo no se podrán modificar los grupos de las matriculas de esas asignaturas y por tanto no pueden ser optimizadas pero si que influirán en la optimización del resto de asignaturas.





| Módulo I                      | Materia                                   | Asignatura y Guía Docente Electrónica |   |   | ECTS |
|-------------------------------|---|---------------------------------------|---|---|------|
| Formación Básica<br>(60 ECTS) | Fundamentos Matemáticos de la Informática | Álgebra y Matemática Discreta         |  |  | 6    |
|                               |   | Cálculo y Métodos Numéricos           |  |  | 6    |
|                               |   | Estadística                           |  |  | 6    |
|                               |   | Lógica                                |  |  | 6    |
|                               | Fundamentos Físicos de la Informática     | Fundamentos Físicos de la Informática |  |  | 6    |
|                               | Ingeniería de Computadores                | Estructura de Computadores            |  |  | 6    |
|                               |   | Tecnología de Computadores            |  |  | 6    |
|                               | Programación                              | Fundamentos de Programación I         |  |  | 6    |
|                               |   | Fundamentos de Programación II        |  |  | 6    |
|                               | Gestión de las Organizaciones             | Fundamentos de Gestión Empresarial    |  |  | 6    |

Figura 2.2: Asignaturas módulo Formación Básica para la Ingeniería

| Módulo II                                   | Materia  | Asignatura y Guía Docente Electrónica    |   |   | ECTS |
|---|--|--|---|---|------|
| Común a la Rama de Informática<br>(96 ECTS) | Ingeniería de Computadores   | Arquitectura de Computadores             |  |  | 6    |
|   |  | Organización de Computadores             |  |  | 6    |
|   | Programación   | Estructura de Datos                      |  |  | 6    |
|   |  | Metodología de la Programación           |  |  | 6    |
|   |  | Programación Concurrente y Tiempo Real   |  |  | 6    |
|   | Ética, Legislación y Profesión   | Aspectos Profesionales de la Informática |  |  | 6    |
|   | Sistemas Operativos, Sistemas Distribuidos y Redes                       | Redes de Computadores I                  |  |  | 6    |
|   |  | Redes de Computadores II                 |  |  | 6    |
|   |  | Sistemas Distribuidos                    |  |  | 6    |
|   |  | Sistemas Operativos I                    |  |  | 6    |
|   | Ingeniería del Software, Sistemas de Información y Sistemas Inteligentes | Bases de Datos                           |  |  | 6    |
|   |  | Ingeniería del Software I                |  |  | 6    |
|   |  | Ingeniería del Software II               |  |  | 6    |
|   |  | Interacción Persona-Ordenador I          |  |  | 6    |
|   |  | Sistemas de Información                  |  |  | 6    |
|   |  | Sistemas Inteligentes                    |  |  | 6    |

Figura 2.3: Asignaturas módulo Formación Común a la rama de ingeniería Informática

---

En estas imágenes tenemos las asignaturas que forman los 2 bloques obligatorios para todos los alumnos y que tienen más de un grupo, estas asignaturas son todas las asignaturas del 1º y 2º curso y las asignaturas del primer cuatrimestre del 3º curso. La otra sección de la web[Escuela Superior de Ingeniería Informática de Albacete, 2024] que nos va dar una información vital para este trabajo es la sección de horarios. En esta sección están todos los horarios divididos entre asignaturas del 1º y 2º cuatrimestre, con esto sabremos que franja horaria tiene cada asignatura en cada grupo, esta será una información muy importante para uno de nuestros objetivos, los solapes, pero lo veremos más adelante.

Una vez sabemos esto podemos explicar como funciona el sistema de matrículas en la escuela, el alumno tendrá asignado un día y hora para poder realizar su matrícula desde la web de la secretaría virtual de la universidad, ahí tendrá que elegir en que asignaturas quiere matricularse y en las que tienen más de un grupo en que grupo quiere estar. En los casos donde el alumno solo tiene que matricularse en asignaturas del mismo curso en cada cuatrimestre no existirá mucho problema en elegir grupo para las asignaturas que lo necesiten, un detalle importante es que en este proceso cuando se elige grupo es solo para la clase de teoría, para elegir subgrupo de prácticas se hará posteriormente a través de la página de la asignatura en el campus virtual. En cambio un alumno que tenga asignaturas de distintos cursos que tienen varios grupos si le es relevante la selección de grupos, ya que dependiendo de los grupos de teoría y subgrupos de práctica que seleccione puede que tenga solape de horarios y no pueda asistir a todas las clases. Por ejemplo un alumno se quiere matricular en la asignatura Fundamentos de Programación I del 1º curso y en la asignatura Organización de computadores del 2º curso, si en la asignatura de primero se matricula en el grupo a y quiere matricularse también en el grupo a de la asignatura del 2º curso tiene que tener cuidado con el subgrupo de prácticas, ya que si selecciona el primero tendrá un solape de horarios y no podría asistir a las 2 clases.

| 1º GRUPO A (10) - 1 Semestre |  |   | 2023-24  | Primer Cuatrimestre                                  |   |  |
|------------------------------|--|---|--|--|---|--|
|                              | Lunes  | Martes  | Miércoles  | Jueves   | Viernes                                   |  |
| 8:15<br>9:45                 | Fundamentos de Gestión Empresarial<br>AULA 1.11        | Fundamentos de Programación I<br>AULA 1.11      | Fundamentos Físicos de la Informática<br>AULA 1.11 | Tecnología de Computadores<br>AULA 1.11              | Cálculo y Métodos Numéricos<br>AULA 1.11  |  |
| 9:45<br>11:15                | Cálculo y Métodos Numéricos<br>AULA 1.11               | Fundamentos de Gestión Empresarial<br>AULA 1.11 | Fundamentos de Programación I<br>AULA 1.11         | Fundamentos Físicos de la Informática<br>AULA 1.11   | Tecnología de Computadores<br>AULA 1.11   |  |
| 11:35<br>13:05               | Fundamentos Físicos de la Informática<br>Lab. Física 2 | Tecnología de Computadores<br>Elect.Digit. 1    | Cálculo y Métodos Numéricos<br>SOFTW. 4            | Fundamentos de Gestión Empresarial<br>Aula Multiusos | Fundamentos de Programación I<br>SOFTW. 7 |  |
| 16:00<br>17:30               | Fundamentos Físicos de la Informática<br>Lab. Física 2 | Tecnología de Computadores<br>Elect.Digit. 1    | Cálculo y Métodos Numéricos<br>SOFTW. 4            | Fundamentos de Gestión Empresarial<br>Aula Multiusos | Fundamentos de Programación I<br>SOFTW. 7 |  |

Figura 2.4: Horario 1º cuatrimestre Grupo 1ºA

| 2º GRUPO A (10) - 1 Semestre |   |  | 2023-24                                   | Primer Cuatrimestre                   |   |  |
|------------------------------|---|--|---|---------------------------------------|---|--|
|                              | Lunes                                     | Martes                                 | Miércoles                                 | Jueves                                | Viernes                                   |  |
| 9:45<br>11:15                | Estructura de Datos<br>SOFTW. 4           | Lógica<br>Aula Multiusos               | Organización de Computadores<br>SOFTW. 7  | Ingeniería del Software I<br>SOFTW. 7 | Sistemas Operativos I<br>SOFTW. 4         |  |
| 11:35<br>13:05               | Ingeniería del Software I<br>AULA 1.11    | Sistemas Operativos I<br>AULA 1.11     | Estructura de Datos<br>AULA 1.11          | Lógica<br>AULA 1.11                   | Organización de Computadores<br>AULA 1.11 |  |
| 13:05<br>14:35               | Organización de Computadores<br>AULA 1.11 | Ingeniería del Software I<br>AULA 1.11 | Sistemas Operativos I<br>AULA 1.11        | Estructura de Datos<br>AULA 1.11      | Lógica<br>AULA 1.11                       |  |
| 17:30<br>19:00               | Estructura de Datos<br>SOFTW. 4           | Lógica<br>Aula Multiusos               | Organización de Computadores<br>AULA 1.11 | Ingeniería del Software I<br>SOFTW. 7 | Sistemas Operativos I<br>SOFTW. 4         |  |

Figura 2.5: Horario 1º cuatrimestre Grupo 2ºA

---

Cuando todos los alumnos hayan realizado este proceso según sus preferencias y las plazas disponibles en cada grupo de teoría y subgrupo de practicas tendremos una configuración inicial para usar como punto de partida en el algoritmo genético.

## 2.2. Metaheurísticas evolutivas

Las metaheurísticas son técnicas de optimización de búsqueda basadas en algoritmos heurísticos que se utilizan para resolver problemas complejos en los que el espacio de búsqueda es muy grande o desconocido



## 3. Desarrollo

---

En este capítulo se explica el proceso de desarrollo del algoritmo genético, los ficheros iniciales, la modelización matemática del problema, ficheros de salida y las distintas implementaciones que se han hecho.

### 3.1. Ficheros Iniciales

Antes de empezar cualquier desarrollo vamos a analizar los ficheros iniciales, explicar su origen y ver como nos puede ayudar su contenido al desarrollo del algoritmo genético.

Los ficheros iniciales son los siguientes:

- *asignaturas.xlsx* En este fichero encontraremos todo el listado de asignaturas que hay disponibles en el grado de ingeniería informática
  - **COD.ASIG:** Identificador único de la asignatura.
  - **NOMBRE ASIGNATURA:** Nombre completo de la asignatura.
  - **TECNOLOGÍA:** Indica que tipo de asignatura es, obligatoria, optativa o si pertenece a alguna intensificación.
  - **CUATRIMESTRE:** Cuatrimestre en el que se imparte la asignatura.
- *horarios.xlsx* En este fichero estará todos los horarios de todos los grupos de teoría y práctica para cada asignatura
  - **CUATRIMESTRE:** Cuatrimestre en el que se imparte la asignatura.
  - **CURSO:** Curso en el que se imparte la asignatura.
  - **GRUPO:** Grupo al que pertenece este horario.
  - **ID GRUPO:** Identificador numérico del curso.
  - **DÍA:** Día en el que se imparte la asignatura.
  - **HORARIO:** Horario en el que se imparte la asignatura.
  - **BLOQUE:** Indica al bloque al que pertenece la asignatura.

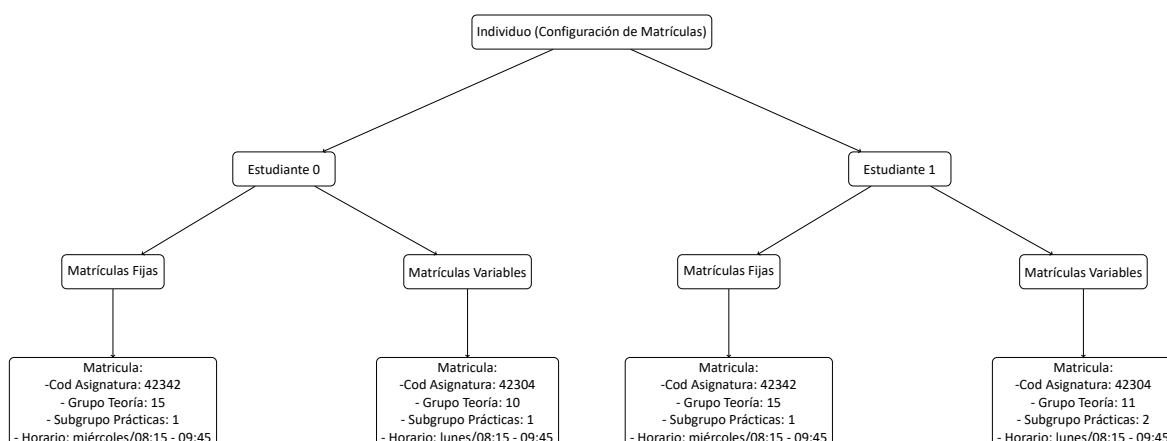
- 
- **ASIGNATURA:** Nombre completo de la asignatura.
  - **CODIGO:** Identificador único de la asignatura.
  - **TEORÍA/PRÁCTICA:** Indica si el horario es de un grupo de teoría o practicas.
- *matriculas.xlsx* En este fichero tendremos todas las matriculas del curso 2020-2023 en el grado de ingeniería informática en Albacete, los datos de los están anonimizados.
- **AÑO:** Año lectivo de la matrícula.
  - **PLAN:** Plan docente al que pertenece la matrícula.
  - **CODIGO:** Identificador único de la asignatura.
  - **ASIGNATURA:** Nombre completo de la asignatura.
  - **DNI:** Dni anonimizado del estudiante al que pertenece la matricula.
  - **ALUMNO:** Nombre anonimizado del estudiante al que pertenece la matricula.
  - **GRUPO:** Grupo de teoría al que pertenece la matricula.
  - **GP:** Subgrupo de prácticas al que pertenece la matricula.

## 3.2. Modelización del problema

El siguiente paso es como podemos modelizar nuestro problema para poder aplicarlo en un algoritmo genético, necesitamos definir los individuos que formaran cada población, cual serán los objetivos a optimizar, como podemos calcularlos e integrarlos en una función de evaluación y definir como funcionarán los mecanismos genéticos de selección, cruce y mutación, además también hay que definir una condición de parada.

### 3.2.1. Representación de los individuos

En este problema cada individuo representara una configuración para todas las matriculas de cada alumno, cada individuo tendrá una lista de alumnos, cada alumno tendrá dos listas, una de matriculas fijas, las cuales son de grupo único y no pueden ser modificadas y otra de matriculas variables, que son de asignaturas que poseen más de un grupo y pueden ser modificadas en cada generación, en cada matricula estará reflejado el grupo de teoría, subgrupo de práctica y su horario. En esta representación los genes que podrán ser modificados serán el grupo de teoría y subgrupo de práctica de las matrículas variables, el grupo de teoría podrá tomar los valores de 10, 11 y 12 en asignaturas del 1º curso, 10 u 11 en asignaturas de los demás cursos y el subgrupo de prácticas podrá ser 1 o 2.



**Figura 3.1:** Representación de un individuo en el problema de asignación de matrículas.

### 3.2.2. Objetivos

Una vez hemos representado los individuos del problema tenemos que definir cuáles serán los objetivos a optimizar y cómo podemos calcularlos, en este caso tendremos 5 objetivos que se describen a continuación:

#### 1. Número de solapes (minimizar):

Representan la suma de solapes que tiene cada alumno de horarios de diferentes asignaturas, podemos calcularlo de la siguiente manera:

Por cada alumno calcularemos los solapes que tiene en asignaturas del mismo cuatrimestre, esto es importante ya que entre asignaturas de distinto cuatrimestre no importa que tengan el mismo horario.

$$\text{Solapes}_{\text{cuatrimestre}} = \sum_{\substack{i=1 \\ c_i \in Q}}^{n-1} \sum_{\substack{j=i+1 \\ c_j \in Q}}^n \text{intersección}(C_i, C_j)$$

**Donde:**

- $C_i$  y  $C_j$  son asignaturas específicas en las que está matriculado el estudiante.
- $Q$  representa el conjunto de asignaturas de ese cuatrimestre en las que está matriculado el estudiante.
- $n$  es el número total de asignaturas en las que está matriculado el estudiante en ese cuatrimestre.
- La función  $\text{intersección}(C_i, C_j)$  devuelve 1 si las asignaturas  $C_i$  y  $C_j$  se solapan en el tiempo y 0 en caso contrario.

Una vez que tenemos esta definición para calcular los solapes totales por cuatrimestre que tiene un estudiante haremos la suma para saber el total de solapes de un estudiante:

---


$$\text{Solapes}_{\text{estudiante}} = \text{Solapes}_{\text{cuatrimestre}_1} + \text{Solapes}_{\text{cuatrimestre}_2}$$

Para los solapes totales hacemos la suma de los solapes de cada estudiante.

$$\text{Solapes Totales} = \sum_{i=1}^n \text{Solapes}_{\text{estudiante}_n}$$

**Donde:**

- $n$  es el número de estudiantes que tiene el individuo.

## 2. Equilibrio de alumnos por clase (maximizar):

Este objetivo representa la equidad de estudiantes en los distintos grupos de teoría y subgrupos de prácticas en las asignaturas, esto significa que es el más importante, ya que el resto de objetivos están pensados exclusivamente para la comodidad de los estudiantes y facilitarles un buen horario lectivo, por lo que pueden cumplirse en mayor o menor medida, pero si no se cumple este objetivo sería muy difícil realizar las clase debido a la capacidad de las aulas, aparte de que para el alumno también es más incomodo para la resolución de dudas.

Calcularemos el equilibrio de cada asignatura obligatoria, para ello seguimos los siguientes pasos.

### 2.1 Número ideal de estudiantes por grupo de teoría

Para la asignatura  $i$  con  $N_i$  estudiantes matriculados y  $G_i$  grupos, el número ideal de estudiantes por grupo es  $N_{ideal_i} = N_i/G_i$

### 2.2 Proporción de equilibrio por grupo principal

Para cada grupo principal  $g$  de la asignatura  $i$ , el equilibrio se calcula restando la cantidad ideal de estudiantes  $N_{ideal_i}$  al número real de estudiantes  $n_{g_i}$  en ese grupo. Esta diferencia se normaliza dividiéndola por  $N_{ideal_i}$  y el resultado se toma en valor absoluto para evitar resultados negativos. Finalmente, se resta esta proporción de 1 para obtener una medida del equilibrio, donde valores cercanos a 1 significa que el número de estudiantes en ese grupo está cerca del valor ideal y valores menores que existe mayor desequilibrio:

$$E_{g_i} = 1 - \left| \frac{n_{g_i} - N_{ideal_i}}{N_{ideal_i}} \right|$$

### 2.3 Número ideal de estudiantes por subgrupo de prácticas

Será el número ideal de alumnos en el grupo principal  $N_{ideal_i}$  entre 2

$$N_{ideal_{sg}} = \frac{N_{ideal_i}}{2}$$

### 2.4 Proporción de equilibrio por subgrupo de prácticas

Para cada subgrupo de prácticas  $s_g$  de la asignatura  $i$ , el equilibrio se calcula restando la cantidad ideal de estudiantes  $N_{ideal_{sg}}$  al número real de

estudiantes  $n_{sg_i}$  en ese grupo. Esta diferencia se normaliza dividiéndola por  $N_{ideal_{sg}}$  y el resultado se toma en valor absoluto para evitar resultados negativos. Finalmente, se resta esta proporción de 1 para obtener una medida del equilibrio, donde valores cercanos a 1 significa que el número de estudiantes en ese subgrupo está cerca del valor ideal y valores menores que existe mayor desequilibrio:

$$E_{sg_i} = 1 - \left| \frac{n_{sg_i} - N_{ideal_{sg}}}{N_{ideal_{sg}}} \right|$$

### 2.5 Promedio de equilibrio en la asignatura

Para calcular el equilibrio total de cada asignatura  $i$  se calcula con el promedio de las proporciones de todos los grupos de teoría y subgrupos de prácticas.

$$E_{asignatura_i} = \frac{\sum_{g=1}^G D_{grupo\ principal_{g_i}} + \sum_{s=1}^{S_g} D_{subgrupo_{sg}}}{G + S_g}$$

**Donde:**

- $G$  es el número de grupos principales de teoría de la asignatura.
- $S_g$  es el número de subgrupos de practicas de la asignatura.

Una vez que tenemos el equilibrio de cada asignatura obligatoria podemos calcular el equilibrio total del individuo de la siguiente manera.

$$E_{total} = \frac{1}{A} \sum_{a=1}^A E_{asignatura_a}$$

**Donde:**

- $A$  es el número de asignaturas obligatorias.

### 3. Tasa de cohesión de teoría por curso para un estudiante (maximizar):

Con este objetivo se representa que un alumno este matriculado en el mismo grupo de teoría para asignaturas del mismo curso, con esto se trata de evitar los tiempos muertos, ademas también puede favorecer a reducir los solapes entre horarios de asignaturas del mismo curso, lo podemos calcular de la siguiente manera.

Sea  $S$  el conjunto total de estudiantes, y para cada estudiante  $s \in S$ , tenemos  $C_s$  como el conjunto de cursos en los que está matriculado. Para cada curso  $c \in C_s$ , tenemos un conjunto de asignaturas  $A_{s,c}$  en las que el estudiante está matriculado dentro de ese curso. Además, para cada asignatura  $a \in A_{s,c}$ , el estudiante está asignado a un grupo de teoría  $g_{s,a}$ .

La tasa de cohesión por curso para un estudiante  $s$  en un curso  $c$ , denotada como  $T_{s,c}$  se calcula como la proporción del grupo más frecuente en el que el estudiante está

---

matriculado dentro de ese curso, respecto al total de asignaturas del curso en las que está matriculado. Esto es:

$$T_{s,c} = \frac{\max(\text{freq}(g_{s,a}))}{|A_{s,c}|} \quad \forall a \in A_{s,c}$$

La tasa de cohesión promedio para un estudiante  $s$ , que toma en cuenta todos los cursos en los que está matriculado, se calcula como el promedio de las tasas de cohesión por curso de ese estudiante

$$T_s = \frac{1}{|C_s|} \sum_{c \in C_s} T_{s,c}$$

Finalmente, la tasa de cohesión general para todos los estudiantes de un individuo se obtiene como el promedio de las tasas de cohesión de todos los estudiantes:

$$T = \frac{1}{|S|} \sum_{s \in S} T_s$$

**4. Tasa de cohesión prácticas por curso para un estudiante (maximizar):**

Este objetivo representa lo mismo que el anterior pero esta vez con los grupos de prácticas.

Sea  $S$  el conjunto total de estudiantes, y para cada estudiante  $s \in S$ , tenemos  $C_s$  como el conjunto de cursos en los que está matriculado. Para cada curso  $c \in C_s$ , tenemos un conjunto de asignaturas  $A_{s,c}$  en las que el estudiante está matriculado dentro de ese curso. Además, para cada asignatura  $a \in A_{s,c}$ , el estudiante está asignado a un subgrupo de practicas  $sg_{s,a}$ .

La tasa de cohesión por curso para un estudiante  $s$  en un curso  $c$ , denotada como  $T_{s,c}$  se calcula como la proporción del grupo de practicas más frecuente en el que el estudiante está matriculado dentro de ese curso, respecto al total de asignaturas del curso en las que está matriculado. Esto es:

$$TP_{s,c} = \frac{\max(\text{freq}(sg_{s,a}))}{|A_{s,c}|} \quad \forall a \in A_{s,c}$$

La tasa de cohesión promedio para un estudiante  $s$ , que toma en cuenta todos los cursos en los que está matriculado, se calcula como el promedio de las tasas de cohesión por curso de ese estudiante

$$TP_s = \frac{1}{|C_s|} \sum_{c \in C_s} TP_{s,c}$$

Finalmente, la tasa de cohesión general para todos los estudiantes de un individuo se obtiene como el promedio de las tasas de cohesión de todos los estudiantes:

$$TP = \frac{1}{|S|} \sum_{s \in S} TP_s$$


---

#### 5. Tasa coincidencias configuración inicial (maximizar):

En uno de nuestros ficheros iniciales tenemos una configuración inicial de matriculas que han hecho los alumnos, uno de nuestros objetivos será maximizar el parecido de la solución con la configuración inicial de los alumnos. Lo podemos calcular de la siguiente manera:

Para cada estudiante se calcula la proporción de coincidencias que tienen los grupos de teoría y practica de sus matriculas del individuo respecto a los que hay en la configuración inicial.

$$TC_a = \frac{\text{Numero de coincidencias}_a}{\text{Numero maximo de coincidencias}_a}$$

Después calculamos el promedio de todos los estudiantes

$$TC = \frac{1}{A} \sum_{a=1}^A TC_a$$

### 3.2.3. Función de evaluación

Una vez tenemos representados los objetivos a optimizar y como los vamos a medir, tenemos que definir una función objetivo, esta función nos dirá como de buena es una solución, una de las primeras cosas que tenemos que hacer para crear una función de evaluación es tener nuestros objetivos en la misma escala, en este caso 3 de los 4 objetivos están en una escala de 0 a 1, todos excepto el número de solapes que puede ser cualquier número entero positivo, por lo que habría que normalizarlo para que este en la misma escala que el resto de objetivos, uno de los métodos podría ser dividir el número de solapes del individuo entre el número de solapes máximo, pero saber cual podría ser el número de solapes máximo es muy difícil, por lo que hay que buscar otra solución, lo que se hará es lo siguiente, al tener una configuración de matriculas inicial se calculara su número de solapes, y ese valor lo usaremos como una simulación del número de solapes máximos, así podremos dividir los solapes de cada individuo y tenerlos en una escala cercana al resto de objetivos, es verdad que si algún objetivo tuviese un número de solapes mayor al de la configuración inicial tendría un valor mayor que 1, pero eso haría que ese individuo tuviese una función de evaluación muy mala. Una vez que tenemos todos los objetivos en una escala similar toca unirlos en una función de evaluación, para este problema se creara una función que se tenga que maximizar, a mayor valor mejor sera el individuo, la función sera una operación ponderada donde los objetivos a minimizar estarán restando y los objetivos a maximizar estarán sumando.

$$f(i) = -w_1 * \text{Solapes Totales} + w_2 * E_{total} + w_3 * T + w_4 * TP + w_5 * TC$$

La asignación de los pesos intentara transmitir a la función de evaluación como de importante es cada objetivo, antes se ha explicado la importancia del equilibrio del número de alumnos por grupo, por lo que el peso  $w_3$  tendra que tener un valor elevado, en este problema el valor de los pesos sera fijo, y todas las pruebas se harán con los siguientes valores.

- 
- $w_1 = 0.1$
  - $w_2 = 0.1$
  - $w_3 = 0.55$
  - $w_4 = 0.1$
  - $w_5 = 0.15$

### 3.2.4. Población Inicial

El siguiente paso es crear nuestra población inicial, el número de elementos de la población inicial ira indicado por un hiperparámetro, para crear cada individuo crearemos un clon de la configuración inicial ya que contiene en que asignaturas se quiere matricular cada alumno, el siguiente paso sera asignar a cada asignatura en la lista de asignaturas variables de cada alumno un grupo aleatorio de teoría y prácticas, lo que se hará es para cada alumno asignar un grupo aleatorio por curso, esto quiere decir que los individuos de la población inicial estarán matriculados en el mismo grupo en asignaturas del mismo curso, el grupo de practicas sera aleatorio para todas las asignaturas sin importar el curso, esta decisión se ha tomado ya que si cada asignatura tuviese un grupo de teoría totalmente aleatorio serian soluciones muy caóticas, lo que implicaría tener que hacer muchas generaciones para encontrar una buena solución e incluso podría provocar caer en un optimo local, por eso es mejor tener individuos en la población inicial con cierto sentido y generación a generación con pequeños cambios acercarse a soluciones mejores.

### 3.2.5. Selección

El proceso de selección juega un papel muy importante puesto que se encarga de decidir qué individuos van a pasar directamente a la generación siguiente o serán padres de los futuros descendientes.[Flores et al., 2022] Existen muchos métodos de selección, en este problema implementaremos tres métodos diferentes con los que realizaremos experimentos para determinar cual funciona mejor con este problema, los métodos implementados son:

- Basada en fitness  
En este método cada individuo tendrá una probabilidad de ser elegido directamente proporcional al fitness, para calcular la probabilidad que tiene cada individuo se puede crear una distribución de probabilidad usando la formula referenciada en[Flores et al., 2022], de forma que la probabilidad de cada individuo es:

$$p_s(i) = \frac{f(i)}{\sum_{i=1}^N f(i)}$$

El problema de este método es que si existe un individuo muy bueno va a tener muchas probabilidades de ser seleccionado, acaparando la población y evitando la diversidad genética lo que supondrá no hacer una exploración de espacio de soluciones efectiva y pudiendo ocasionar un convergencia prematura y teniendo como resultado un optimo local.



- Basada en rango

Con este método se trata de atenuar la presión selectiva del método anterior[Flores et al., 2022]. La idea es crear un ranking de individuos según su fitness, donde mayor fitness ocupan mejores posiciones en El ranking, y con una formula asignar nuevas probabilidades sin tener en cuenta el fitness y solo teniendo en cuenta la posición en el ranking, teniendo una población de  $N$  individuos y que cada individuo ocupa una posición  $i$  en el ranking, podemos calcular las nuevas probabilidades selección con la siguiente formula[Flores et al., 2022]

$$p_s(i) = \frac{2(N - i + 1)}{N^2 + N}$$

Con esta formula los elementos con mejor fitness todavía tendrán mayor probabilidad de ser seleccionados pero se ha reducido bastante, de esta forma los elementos con peor fitness también tendrán posibilidades de ser seleccionados con lo que se tiene mayor diversidad genética y ofrece una mejor exploración del espacio de soluciones.

- Selección por torneo

Intenta disminuir todavía más la presión selectiva[Flores et al., 2022], este método depende de un tamaño de torneo  $k$  que se indique, el funcionamiento es el siguiente, se eligen aleatoriamente  $k$  elementos y nos quedamos con el de mayor fitness. Nótese que cada individuo intervendrá (en media) en  $k$  torneos. A mayor  $k$  mayor presión selectiva[Flores et al., 2022]. Este método depende totalmente del tamaño del torneo y cambia drásticamente su afectación. Un tamaño de torneo pequeño da muchas más oportunidades a los individuos con peor fitness de forma que se aumenta la diversidad genética, esto hace que tarde más en converger pero reduce los óptimos locales, en cambio un tamaño de torneo grande aumenta la presión selectiva acelerando la convergencia pero con mayor riesgo a óptimos locales. Puede que el mejor funcionamiento de este método sea un tamaño donde se ofrezca una gran exploración del espacio de soluciones pero explotando y mejorando los mejores individuos.

### 3.2.6. Cruce

Es el primer operador de variación que se aplica, a partir de dos padres seleccionados a través de alguno de los métodos anteriores se cruzaran para generar hijos nuevos, normalmente el cruce se suele realizar bajo una probabilidad bastante grande, en este problema se hará siempre, con este operador estamos mezclando los estudiantes de dos individuos, nunca se introduce material genético nuevo, por lo que al hacer esto se está explorando el espacio de soluciones para intentar encontrar una solución mejor a partir de otras soluciones buenas.

Es importante recalcar que el cruce se hará a nivel de estudiante, cada individuo en la población tiene una lista de estudiantes, y cada estudiante tiene a su vez una lista de matriculas variables y matriculas fijas, nuestro objetivo será optimizar los grupos de teoría y subgrupos de prácticas de cada matricula variable. Tenemos dos enfoques de como realizar el cruce para este problema:

- 
1. **Cruzar las matrículas variables individualmente:** Para cada estudiante, se pueden intercambiar las matrículas variables entre los padres, lo que significa que algunas provengan del padre 1 y otras del padre 2.
  2. **Cruzar la configuración completa de cada estudiante:** De esta manera se selecciona la configuración completa de matrículas de un padre u otro para cada estudiante de forma independiente. Es decir, para cada estudiante se decide si hereda toda la configuración del padre 1 o del padre 2.

La segunda opción, donde se hereda la configuración completa del estudiante, se considera más adecuada en este problema. Al cruzar individualmente las matrículas se corre el riesgo de desordenar demasiado el material genético, lo que podría ocasionar dificultades en la convergencia de una buena solución. Al heredar la configuración completa de cada estudiante, se realiza una exploración más controlada del espacio de soluciones, lo que facilita modificaciones graduales a través del operador de mutación.

Una vez explicado el esquema del cruce, tenemos que pasar a su implementación. En los algoritmos genéticos existen distintos tipos de operadores de cruce, cada uno tiene unas características y ventajas. En este trabajo se utilizarán tres variantes que se explican a continuación:

1. **Un punto:** En este caso en cada generación se seleccionará un punto de manera aleatoria en la lista de alumnos para cada cruce, de forma que los segmentos de los padres antes de ese punto se mantienen, mientras que los segmentos posteriores se intercambian. Este es uno de los métodos más sencillos, su principal desventaja es que las combinaciones genéticas están limitadas por la ubicación del único punto de cruce.
2. **Varios puntos:** Este método es una evolución del anterior, ahora seleccionaremos dos puntos de corte en cada cruce, las secciones de la lista de alumnos ubicadas entre estos dos puntos se intercambian entre los padres, el resto de segmentos se quedan igual. Vemos que ganamos la ventaja de aumentar la diversidad genética en comparación con el método anterior, lo que nos permite explorar un espacio de búsqueda más amplio, aunque puede ser insuficiente si los genes importantes están muy dispersos.
3. **Uniforme:** Este cruce tiene una forma más aleatoria, ya que se basa en una máscara de cruce formada por bits que indica para cada alumno de la lista si proviene del padre 1 o del padre 2, esta máscara será aleatoria en cada cruce. Este método es el que mayor diversidad genética proporciona ya que cada alumno puede provenir de cualquiera de los padres sin seguir una estructura predeterminada. Este método es muy recomendable cuando se tiene que explorar un espacio de soluciones muy amplio, además combina muy bien con mutaciones de baja probabilidad ya que al combinar configuraciones completas de alumnos se harán pequeños cambios que hagan llegar a una buena solución.

### 3.2.7. Mutación

Este sera el segundo operador de variación que se use, se aplicara a los hijos generados en el proceso de cruce, consiste en introducir pequeñas modificaciones aleatorias en el individuo, en este caso la mutación cambiara el grupo de teoría o prácticas en las matriculas variables de los alumnos, al igual que el operador de cruce, el de mutación también ayuda a explorar el espacio de soluciones, pero de manera diferente, ya que el operador de cruce combina el material genético existente entre los individuos, el operador de mutación introduce nuevo material genético que no estaba presente en los padres. Esto da la capacidad al algoritmo genético de explorar áreas del espacio de búsqueda que no hubieran sido accesibles a través del cruce.

---

## 4. Experimentación

---

---

## 5. Conclusiones

---

---



## Referencia bibliográfica

---

[Escuela Superior de Ingeniería Informática de Albacete, 2024] Escuela Superior de Ingeniería Informática de Albacete (2024). Página web oficial de la escuela.

[European Commission, 2024] European Commission (2024). Inclusive and connected higher education - bologna process.

[Flores et al., 2022] Flores, M., Gámez, J., and García, I. (2022). *Algoritmos Genéticos*. Departamento de Sistemas Informáticos, UCLM - Albacete. Disponible en formato PDF.