



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Estadística e Investigación Operativa
Aplicadas y Calidad

Propuesta de formulación multiobjetivo para el Problema
de Formación de Equipos en el aula y resolución mediante
un algoritmo evolutivo

Trabajo Fin de Máster

Máster Universitario en Ingeniería de Análisis de Datos, Mejora de
Procesos y Toma de Decisiones

AUTOR/A: Arronategi Pujana, Aitor

Tutor/a: Sánchez Anguix, Víctor

Cotutor/a: Alberola Oltra, Juan Miguel

CURSO ACADÉMICO: 2023/2024

Resumen

Este proyecto se enfoca en abordar el "Team Formation Problem" (TFP) o Problema de Formación de Equipos (PFE), un desafío recurrente en la literatura académica. Específicamente, se busca proponer una nueva formulación para el TFP en el entorno universitario. Se desarrolla un modelo de optimización multiobjetivo que busca maximizar la heterogeneidad dentro de los grupos de estudiantes en cuanto al nivel de aptitud técnica, perfiles psicológicos y balance de género, así como establecer una partición en equipos que sean lo más homogénea posibles entre sí. Para resolver este desafío, se diseñará un algoritmo evolutivo multiobjetivo conocido, denominado Non-Sorted Dominated Sorting Genetic Algorithm (NSGA-II). La mayor aportación de este proyecto son los operadores de cruce y mutación personalizados que se utilizarán en el NSGA-II, operadores que se han diseñado específicamente para adaptarse a la formulación propuesta. Principalmente, el estudio pretende analizar si la implementación de dichos operadores concluye en mejores resultados, comparándolo con un algoritmo NSGA-II con operadores estándar y con otros enfoques heurísticos utilizados en formulaciones TFP similares. Debido a que el algoritmo a emplear trabaja con poblaciones de soluciones, se hace uso del hipervolumen como medida de calidad multidimensional para comparar los diferentes resultados obtenidos. Como se buscaba en un principio, los resultados indican que los operadores de cruce y mutación personalizados obtuvieron mejores resultados para distintos tipos de clase.

Palabras clave: Formación de equipos, Optimización Multiobjetivo, Aptitudes técnicas, Perfiles psicológicos, Balance de género, NSGA-II, Mutación, Cruce.

Abstract

This project focuses on addressing the "Team Formation Problem" (TFP), a recurrent challenge in the academic literature. Specifically, it seeks to propose a new formulation for the TFP in the university environment. A multi-objective optimization model is developed that seeks to maximize heterogeneity within student groups in terms of technical aptitude level, psychological profiles and gender balance, as well as to establish a partition into teams that are as homogeneous as possible among themselves. In order to solve this challenge, a known multi-objective evolutionary algorithm, called Non-Sorted Dominated Sorting Genetic Algorithm (NSGA-II), will be designed. The major contribution of this project is the custom crossover and mutation operators that will be used in NSGA-II, operators that have been specifically designed to fit the proposed formulation. Mainly, the study aims to analyze whether the implementation of such operators leads to better results, comparing it with an NSGA-II algorithm with standard operators and with other heuristic approaches used in similar TFP formulations. Since the algorithm to be employed works with populations of solutions, the hypervolume is used as a multidimensional quality measure to compare the different results obtained. As originally searched, the results indicate that the custom crossover and mutation operators achieved better results for different class types.

Key words: Team Formation, Multi-objective, Optimization, Technical Skills, Psychological Profiles, Gender Balance, NSGA-II, Mutation, Crossover.

Resum

Aquest projecte se centra a abordar el "Team Formation Problem" (TFP) o Problema de Formació d'Equips (PFE), un repte recurrent en la literatura acadèmica. Específicament, es busca proposar una nova formulació per al TFP en l'entorn universitari. Es desenvolupa un model d'optimització multiobjectiu que cerca maximitzar l'heterogeneïtat dins dels grups d'estudiants pel que fa al nivell d'aptitud tècnica, perfils psicològics i balanç de gènere, així com establir una partició en equips que siguen el més homogeni possible entre si. Per a resoldre aquest repte, es dissenyarà un algoritme evolutiu multiobjectiu conegut, denominat Non-Sorted Dominated Sorting Genetic Algorithm (NSGA-II). La major aportació d'aquest projecte són els operadors de creuament i mutació personalitzats que s'utilitzaran en l'NSGA-II, operadors que s'han dissenyat específicament per a adaptar-se a la formulació proposada. Principalment, l'estudi pretén analitzar si la implementació d'aquests operadors conclou en millors resultats, comparant-lo amb un algoritme NSGA-II amb operadors estàndard i amb altres enfocaments heurístics utilitzats en formulacions TFP similars. A causa que l'algoritme a emprar treballa amb poblacions de solucions, es fa ús de l'hipervolum com a mesura de qualitat multidimensional per a comparar els diferents resultats obtinguts. Com es buscava en un principi, els resultats indiquen que els operadors de creuament i mutació personalitzats van obtenir millors resultats per a diferents tipus de classe.

Paraules clau: Formació d'equips, Optimització Multiobjectiu, Aptituds tècniques, Perfils psicològics, Balanç de gènere, NSGA-II, Mutació, Creuament.

Contenido

1	INTRODUCCIÓN	6
2	OBJETIVOS	7
3	MARCO TEÓRICO	8
3.1	Optimización	8
3.2	Heurísticas y Metaheurísticas	10
3.3	Algoritmos genéticos	11
3.3.1	Enfoques monobjetivo y multiobjetivo en algoritmos genéticos	15
3.3.2	Algoritmos genéticos multiobjetivo	17
3.4	NSGA-II: Non-Sorted Genetic Algorithm II	21
3.4.1	Clasificación por Frentes no dominadas veloz	22
3.4.2	Clasificación por distancia de agrupamiento	23
3.4.3	Operador <i>crowded-comparison</i> para el reemplazamiento	24
3.5	Medición de la calidad de las poblaciones de individuos	25
4	REVISIÓN BIBLIOGRÁFICA	27
4.1	El problema de formación de equipos	27
4.1.1	Interacción entre las personas y la tecnología en el proceso de formación de equipos	27
4.1.2	Equipos optimizados en el aula	29
4.2	Revisión generalizada sobre la formación de equipos optimizados académicos	29
4.2.1	Revisión sobre el TFP académico con enfoque monobjetivo	30
4.2.2	Limitaciones comunes en la bibliografía TFP académico monobjetivo	32
4.2.3	Revisión del TFP Académico con enfoque multiobjetivo	34
4.3	Aportación del proyecto al estado del arte	36
5	DESCRIPCIÓN DEL PROBLEMA	37
5.1	Definiciones	37
5.2	Modelo de optimización	39
5.3	Formalización de las funciones objetivo	39

5.3.1	Medición de la aptitud técnica	39
5.3.2	Medición de la divergencia de psicológica	41
5.3.3	Medición del balance de género	42
6	DISEÑO E IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO	44
6.1	Codificación de las variables y soluciones	44
6.2	Operador de selección	45
6.3	Operador de cruce personalizado	46
6.4	Operador de mutación personalizado	49
6.5	Software utilizado	50
7	EXPERIMENTACIÓN	52
7.1	Creación de instancias	52
7.2	Puesta a punto	54
7.2.1	Definición de hiperparámetro	54
7.2.2	Optimización bayesiana de hiperparametros	56
7.2.3	Empleo de la optimización bayesiana para la búsqueda de los valores de la tasa de cruce y la tasa de mutación	58
7.3	Comparación con otros métodos	61
7.3.1	Objetivo principal del experimento	61
7.3.2	Métodos a comparar	61
7.4	Diseño de los experimentos	64
7.5	Análisis de las soluciones	66
8	CONCLUSIONES	69
9	BIBLIOGRAFÍA	70

1 Introducción

La educación ha mejorado notablemente gracias al desarrollo de diversos enfoques y tecnologías. El cambio de paradigma respecto al aprendizaje es muy notable en los últimos años, donde cada vez las instituciones se enfocan más en brindar un aprendizaje colaborativo, dejando de lado el aprendizaje individual en el que se basaba la educación del siglo pasado (Laal & Laal, 2012). Este cambio puede haberse producido por la versatilidad y capacidad de respuesta que proporciona el trabajo en equipo a situaciones que implican la toma de decisiones y la resolución de problemas complejos, siendo evidente que la exposición más temprana de los alumnos a este tipo de situaciones mejorara la adaptabilidad a cualquier tipo de problema que se puedan encontrar en su futuro, tanto personal como laboral (Gallagher, Stepien, & Rosenthal, 1992).

El presente proyecto se enfoca en el proceso de formación automatizada de equipos académicos, donde en los últimos años se han realizado multitud de investigaciones enfocándose en lo que se llaman Problemas de Formación de Equipos (PFE) o TFP (por sus siglas en inglés). A pesar de las numerosas investigaciones realizadas, el TFP sigue siendo un problema persistente, especialmente en entornos académicos en los que deben tenerse en cuenta diversos conjuntos de habilidades, perfiles psicológicos, factores demográficos e interacciones interpersonales para un rendimiento óptimo del equipo.

Este Trabajo Fin de Máster pretende contribuir al debate actual sobre el TFP. El proyecto trata de introducir una formulación novedosa para el TFP, integrando técnicas de optimización multiobjetivo para mejorar la composición de los equipos. Este se enfoca en maximizar lo que se denomina en el ámbito de TFP como “**Intra-Heterogeneidad**” o Heterogeneidad del intraequipo, la diversidad de características, tales como habilidades, conocimientos, personalidades, perspectivas y género entre los miembros de un mismo equipo. La “Intra-Heterogeneidad”, busca balancear la complementariedad y la diversidad interna del equipo para optimizar su rendimiento y adaptabilidad a diversas tareas y desafíos.

Al considerar como objetivos clave la intra-heterogeneidad de las aptitudes técnicas, los perfiles psicológicos y el equilibrio entre hombres y mujeres, el proyecto pretende a su vez optimizar la homogeneidad entre equipos en el entorno de las aulas, denominada “**Inter Homogeneidad**” o Homogeneidad interequipo.

En resumen, el contenido de la memoria se ha estructurado de la siguiente manera:

1. **Introducción:** Se ha presentado a grandes rasgos el tema a tratar en este TFM, contextualizando el problema y exponiendo las características que se tomaran en cuenta en el proceso de formación de equipos.
2. **Objetivos:** Se expondrán los principales objetivos perseguidos en este proyecto.
3. **Marco Teórico:** Se ahondará en los conceptos teóricos esenciales para comprender la manera en la que se va a enfrentar a la resolución del problema.
4. **Revisión bibliográfica:** Se presentarán los trabajos contribuyentes al estado del arte, así como la aportación del trabajo actual a este.

5. **Descripción del problema:** Se profundizará en el aspecto matemático del problema, definiendo las formulaciones empleadas para afrontar el problema y las funciones objetivo.
6. **Diseño e implementación del algoritmo genético:** Tras la conceptualización del problema, se expondrá la adaptación al ámbito computacional, explicando operadores novedosos que se aplican en este tipo de técnicas y mencionando las herramientas informáticas utilizadas para resolver el problema.
7. **Experimentación:** Consecuentemente, mediante la puesta a punto del algoritmo, el análisis comparativo y la validación empírica se muestran en esta sección los principales resultados del algoritmo propuesto para afrontar el problema.
8. **Conclusiones:** La investigación finalizara con las principales conclusiones obtenidas en el transcurso del trabajo, así como futuras mejoras que se puedan aportar al problema en cuestión.

2 Objetivos

Los principales objetivos de este Trabajo Fin de Master (TFM) son:

- Presentar un algoritmo capaz de abordar problemas TFP en el ámbito universitario, donde las clases suelen ser de tamaño mediano (40 a 60 alumnos por aula) para los cuales los métodos exactos no logran obtener resultados en plazos razonables (Candel, Sanchez-Anguix, Alberola, Julián, & Botti, 2023).
- Proponer una alternativa novedosa a los algoritmos de formación de equipos optimizados existentes.
- Proponer nuevos operadores evolutivos adaptados al problema presente, con el objetivo de mejorar el rendimiento del NSGA-II con operadores evolutivos clásicos empleados anteriormente en problemas TFP.
- Demostrar la superioridad técnica del algoritmo multiobjetivo propuesto frente a otras heurísticas utilizadas en problemas TFP similares.
- Comprobar que el rendimiento del algoritmo propuesto es mayor en diferentes tipos de aula con diferentes distribuciones de características estudiantiles.

3 Marco teórico

En este apartado se presentarán los principales conceptos necesarios para contextualizar el proyecto a desarrollar. Para empezar, se describirán los puntos más importantes de la optimización de problemas matemáticos (sección 3.1). Tras ello, se presentará el concepto de los métodos aproximados para resolver los problemas de optimización (sección 3.2), con el objetivo de enfocarse posteriormente en los algoritmos genéticos (GA, por sus siglas en Inglés), su enfoque multiobjetivo y dar una clasificación general de los GA multiobjetivo (sección 3.3). Tras la presentación de los conceptos principales para entender el algoritmo a utilizar en el trabajo, el NSGA-II, se ahondará en su funcionamiento y los mecanismos principales que utiliza (sección 3.4).

3.1 Optimización

En las ciencias empíricas, la optimización (también, optimización o programación matemáticas) es la selección del mejor elemento (con respecto a algún criterio) de un conjunto de elementos disponibles. La investigación operativa es uno de los campos de la matemática en cuyas bases funciona la optimización (The Nature of Mathematical Programming, s.f.).

Según (Winston & Goldberg, 2004), suelen seguirse los siguientes pasos a la hora de modelar escenarios en los que se debe optimizar las decisiones tomadas en una organización (Candel, Sanchez-Anguix, Alberola, Julián, & Botti, 2023):

1. **Formulación del problema a resolver**, el cual estará relacionado con alguna de las metas a cumplir por la organización.
2. **Observación del sistema**. Esta fase comprende la recopilación de datos del sistema en observación para la estimación de parámetros que utilizará el modelo de optimización en las fases siguientes.
3. **Formulación de un modelo matemático de optimización para el problema**. Una formulación que trata de representar matemáticamente los factores asociados al problema a resolver.
4. **Verificación del modelo**. Se trata de una fase que puede darse en múltiples ocasiones, ya que el modelo planteado puede no recoger algún comportamiento del proceso o en general presentar fallos en su definición.
5. **Selección de una alternativa que se ajuste**. Una vez se ha definido el modelo de optimización, se personaliza con los datos de los parámetros obtenidos y se resuelve el problema asociado, devolviendo la mejor decisión encontrada. Esta parte requiere del uso de programas informáticos o librerías para resolver problemas matemáticos, denominados *solvers*. Un *solver* se encargaría de tomar el modelo matemático y devolver la solución al problema, así como posibles análisis sobre ella.
6. **Presentación de los resultados**. Se pueden crear informes explicativos de la respuesta obtenida por el algoritmo de optimización, y discutir si estas son coherentes con el juicio experto. En caso de no ser correctas se estaría ante

problemas de cómo se formula o modela el problema, o bien en la propia recogida de los datos utilizados en las estimaciones.

7. **Implementación y evaluación de recomendaciones.** El sistema puede ponerse en producción una vez cumple con las expectativas de la organización, aunque debe ser constantemente monitorizado ya que con el paso del tiempo los parámetros que lo conforman (y que dependen de los datos del sistema) pueden cambiar.

Centrándose en la base de la formulación matemática de un problema de optimización, los siguientes elementos forman la parte esencial de un modelo de optimización (Sánchez Anguix):

- **Parámetros:** equivalente a los datos de entrada del problema, y los que se estiman inicialmente. Representan, generalmente, valores o hechos sobre los que no se tiene poder de decisión.
- **Variables:** representan las decisiones que se quieren efectuar en el problema.
- **Función objetivo:** función matemática que expresa qué es lo que se desea optimizar (ya sea maximizar o minimizar) y que depende de las variables y los parámetros definidos. En otras palabras, indica el rendimiento del sistema que se desea crear de forma cuantitativa, y es lo que se emplea para evaluar la calidad de una posible solución al problema.
- **Restricciones:** definen qué relaciones existen entre las variables además de restringir los posibles valores que pueden tomar ellas. Por tanto, determinan los hechos que se deben cumplir en una solución válida al problema.

En el contexto de las distintas formulaciones matemáticas existentes, se recalcan los modelos de programación entera lineal. En estos modelos, todas las variables de decisión cobran valores enteros y la función objetivo y restricciones son lineales. Además, se debe cumplir que los valores de las variables de decisión no sean negativos (restricción de no negatividad) y los coeficientes del modelo sean deterministas. Su definición matemática general sería la siguiente:

Z : Función objetivo,

X_i : Variables,

$\langle c, a, b \rangle$: parámetros

$$\max Z = \sum_{i=1}^n c_i X_i$$

Sujeto a:

$$\sum_{i=1}^n a_{1,i} X_i \leq (\geq)(=) b_1$$

...

$$\sum_{i=1}^n a_{m,i} X_i \leq (\geq)(=) b,$$

Con:

$$X_i \geq 0, X_i \in Z$$

Al principio, un problema TFP puede parecer algorítmicamente sencillo, pero se vuelve más complejo a medida que aumenta el número de estudiantes y/o las características de los estudiantes y de los grupos. La relación entre las características de los alumnos y la expansión combinatoria del número de alumnos hace que la tarea en cuestión sea un problema de optimización combinatoria NP-duro, es decir, un problema cuya solución óptima no puede ser encontrada por ningún algoritmo en tiempo polinómico (Huxham, 2000).

Finalmente, En cuanto a los algoritmos de optimización, podríamos destacar la clasificación dada según la calidad de la solución obtenida. Si el algoritmo se encarga de garantizar la optimalidad de las soluciones, entonces, se trata de un método exacto, cuyo principal inconveniente es el tiempo que puede llegar a tardar cuando la talla de los problemas crece. Como se ha mencionado en los objetivos, el presente proyecto aborda la resolución de problemas de formación de equipos en aulas de gran tamaño. Tomando en cuenta que resolver problemas cuya complejidad no es polinómica conlleva elevados costes computacionales, los métodos exactos en los cuales se exploran todas las soluciones del problema no son factibles computacionalmente. Para reducir este tiempo, es necesario aplicar algoritmos aproximados como las heurísticas o las metaheurísticas para buscar soluciones viables en un tiempo adecuado (Candel, Sanchez-Anguix, Alberola, Julián, & Botti, 2023) (Graf, 2006).

3.2 Heurísticas y Metaheurísticas

Los heurísticos son procedimientos que utilizan conocimientos previos sobre un problema y técnicas aplicables a su solución, encontrando así soluciones viables con un menor coste computacional. El principal problema de los heurísticos es que no aseguran encontrar el resultado óptimo, aunque dada la complejidad computacional de muchos problemas TFP, es recomendable aplicar métodos aproximados con los que obtener resultados infra óptimos pero que garanticen soluciones viables en tiempos reducidos.

Dentro del abanico de las heurísticas, existen las denominadas metaheurísticas, que, en cambio, comprende estrategias genéricas (no creadas específicamente para el problema en cuestión) para mejorar los procedimientos heurísticos (Graf, 2006).

Una metaheurística guía a una heurística subordinada utilizando conceptos derivados de la inteligencia artificial, las ciencias biológicas, matemáticas, naturales y físicas para mejorar su rendimiento (Osman & Kelly, 1996). Aunque una heurística no garantiza encontrar una solución óptima, tiende a encontrar soluciones cerca de él, siempre que este bien diseñada y dependiendo del tipo de problema.

Existen infinidad de metaheurísticas, basándose muchas de ellas en reglas o procedimientos de la naturaleza, como pueden ser los algoritmos evolutivos, el proceso

de liberación de feromonas por parte de las hormigas a la hora de encontrar recursos para guiar a otras (algoritmo de colonia de hormigas), el algoritmo que imita el proceso de recocido del acero para mejorar sus propiedades (*Simulated annealing*), etc (Game, 2020). Concretamente, en el presente trabajo se aplica un algoritmo genético, uno de los algoritmos más utilizados por su eficacia para resolver problemas NP-duro.

3.3 Algoritmos genéticos

Los algoritmos genéticos (GA, por sus siglas en inglés) pueden clasificarse dentro del grupo de los algoritmos evolutivos, un grupo de metaheurísticas que inspiran su funcionamiento en la teoría de la evolución de Darwin.

En los algoritmos evolutivos, se tiene en cuenta que los seres que conforman las especies o poblaciones de individuos tienen rasgos (determinados por los genes) que pueden traspasarse a generaciones posteriores, y la existencia de recursos limitados en la naturaleza hará que prevalezcan aquellos grupos que mejor consigan adaptarse al medio (Darwin, evolución y selección natural, s.f.). Análogamente, al aparearse dos individuos de una misma especie dan luz a nuevos individuos que comparten cierta información genética de sus predecesores. Si esta información nueva resulta valiosa para cumplir con la mencionada selección natural, la población que comparte este tipo de características tenderá a crecer conforme suceden las generaciones o ciclos evolutivos. No obstante, puede ocurrir que alguno de los descendientes presente algún tipo de alteración en su material genético ya sea por alguna imperfección hereditaria, ciertos cambios en el ADN, u otras razones, ayudando esto a generar diversidad en la población. El traspaso de características como éstas depende de factores como la selección hecha por los predecesores o si estos genes son dominantes o no.

Aplicando esto a la lógica computacional (Holland, 1992), se presenta los recursos y procedimientos utilizados en los algoritmos genéticos para aplicar esta teoría a un problema de optimización:

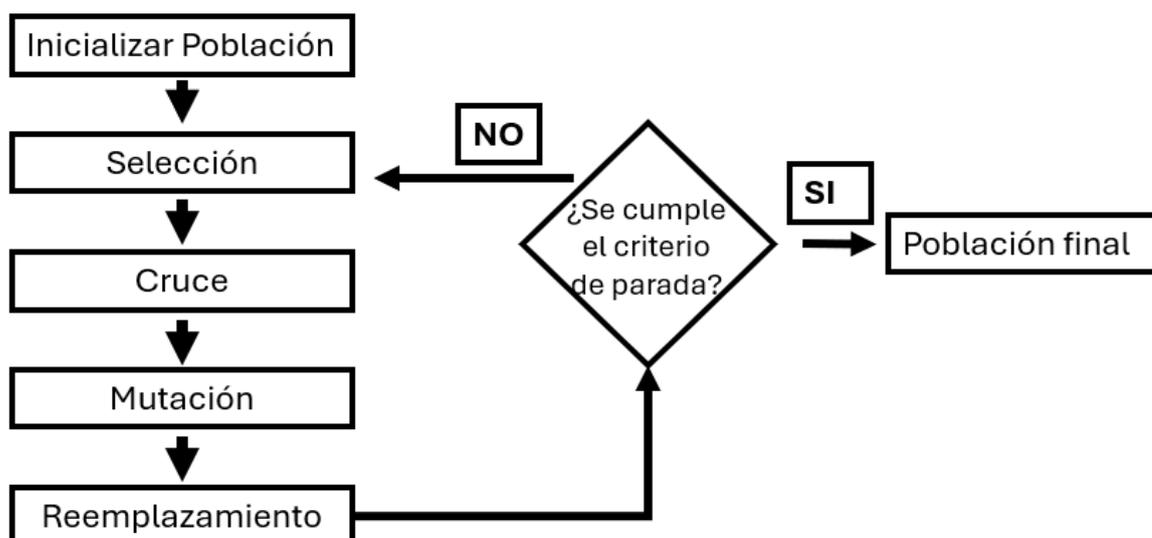


Figura 1: Esquema general de un algoritmo genético (Holland, 1992)

Como se observa en la Figura 1, un algoritmo genético emplea una población de soluciones que mantiene en todo momento donde se aplican los denominados operadores genéticos. Estos operadores combinan la exploración y explotación del espacio de búsqueda, mediante la selección, el cruce, la mutación y el Reemplazamiento.

A grandes rasgos, el operador de selección tiene la función de elegir los individuos que tomaran parte en la reproducción de nuevos descendientes. El operador de cruce es el mecanismo que permite la mezcla de material genético de los padres la cual transmitirán a los hijos. El operador de mutación proporciona un pequeño elemento de aleatoriedad en la vecindad (entorno) de los individuos mediante la alteración de su material genético. Por último, mediante el reemplazamiento de la población se decide cuáles son los individuos que van a continuar en la siguiente generación (Holland, 1992).

A continuación, se procede a detallar la función y utilidad de cada uno de los componentes y operadores que habilitan el funcionamiento de los algoritmos genéticos:

Individuo

Un individuo es una solución posible del problema y la representación genética de este se llama cromosoma. Como es lógico, estos cromosomas se pueden representar de maneras diferentes desde una codificación binaria, mediante números enteros, números reales, etc.

Los cromosomas contienen genes, los cuales guardan información que da identidad a esa solución en concreto. En el contexto de los problemas de optimización, los genes vendrían a representar las variables del problema. La situación real en la que se da una solución, se llama fenotipo mientras que el genotipo es la manera en la que se decide representar dicha solución. Se ilustra un ejemplo a continuación:

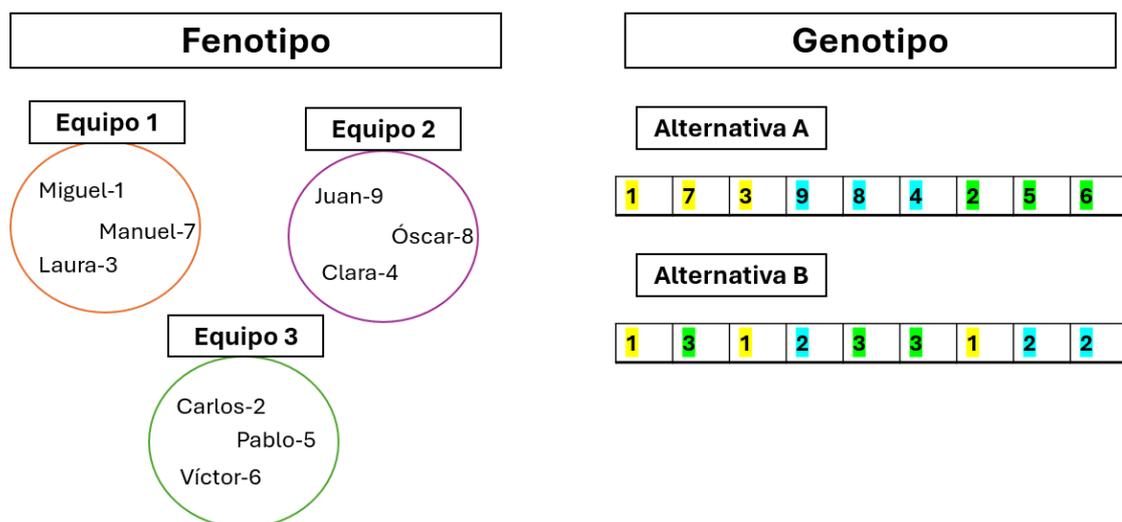


Figura 2: Fenotipo vs Genotipo [Elaboración propia]

En la Figura 4 se observa, se muestran 2 alternativas diferentes para representar los cromosomas de las soluciones. En la alternativa A, se expone la solución como una permutación de los identificadores de los alumnos, donde la posición de cada gen representa en que equipo están los alumnos. En la alternativa B, la posición de cada gen

representa el identificador asociado a cada alumno y los valores de los genes expresan el equipo asociado al alumno.

Evaluación fitness del individuo

El segundo paso consiste en evaluar cada cromosoma con el objetivo de medir la calidad de una función dada. En este caso utilizando las funciones objetivo para dicha labor. Con el resultado de esta evaluación se puede calcular la aptitud de la solución y cada cromosoma almacena un valor de fitness. En el problema que se aborda, la aptitud de la función es simplemente el valor de la función en un punto dado.

Generación de la población inicial

Para ejecutar un GA, primero es necesario inicializar la población de posibles soluciones. La estrategia más habitual consiste en generar soluciones aleatorizadas, aunque también se puede utilizar una heurística aleatorizada para obtener una población inicial más sofisticada. El tamaño de la población es el número de individuos de dicha población.

Selección

El operador de selección se enfoca en elegir a los individuos que pasarán a las operaciones genéticas según una estrategia dada. En los métodos de selección más comunes el objetivo es asegurar que las soluciones más prometedoras tengan una mayor probabilidad de contribuir a la generación de descendientes, con la esperanza de que estos descendientes hereden características favorables.

Los principales tipos de selección son la Selección mediante Ruleta y la Selección por Torneo (Shi-Jie(Gary) & Li, 2004). En la Selección mediante Ruleta, cada individuo tiene una probabilidad de ser seleccionado como padre, que es proporcional al valor de su fitness como individuo. La Selección por Torneo divide la población en grupos de n individuos y el mejor individuo del grupo sobrevive (midiéndose la calidad del individuo mediante su fitness). El tamaño de n se denomina presión de selección y su valor suele ser igual a 2, llamándose en ese caso Selección por Torneo Binario.

Cruce

El operador de cruce tiene como objetivo mezclar el material genético de dos soluciones actuales en la población, denominados padres (normalmente suelen ser 2 individuos) para explorar el espacio de búsqueda en busca de una nueva solución prometedoras, denominada hija/os. En cada generación, a cada par de padres se le aplica una probabilidad de cruce introduciendo así cierta aleatoriedad a este proceso nativamente estocástico. En los algoritmos genéticos, normalmente esta probabilidad suele rondar alrededor del 80-95%.

Como ejemplo se muestra un operador de cruce muy utilizado en la bibliografía de problemas combinatorios con representación cromosómica de tipo permutación, como es el caso de los problemas que se abarcan en este proyecto. Este frecuente operador se denomina *Cut and Crossfill*: (Eiben & Smith, 2003).

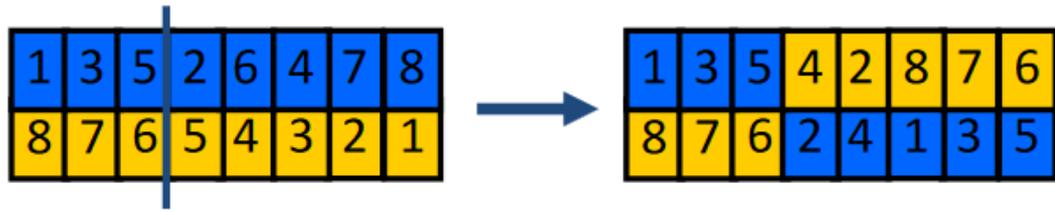


Figura 3: Operador de cruce Cut and Crossfill [Elaboración propia]

En la Figura 3, el procedimiento que sigue este operador es el siguiente:

- Elegir punto de cruce aleatorio.
- Copiar las primeras partes en los hijos.
- Crear la segunda parte insertando valores del otro padre, en el orden que aparecen, empezando después del punto de cruce y omitiendo los valores que ya están en el hijo.

Mutación

La mutación consiste en agregar cierta variación genética en los hijos generados mediante el proceso de cruce. Su objetivo es crear cierto cambio en ellos con el objetivo de buscar otros espacios de búsqueda para no quedarse estancados en mínimos locales. Al igual que en el cruce, se le añade cierta aleatoriedad mediante lo que se denomina probabilidad de mutación, la cual se aplica antes de cada proceso mutante en los descendientes para saber si mutarlos o no. Al contrario que en el cruce, la probabilidad de mutación suele ser bastante baja, alrededor del 2-10%.

En los problemas combinatorios como el TFP, un operador muy utilizado para representaciones de tipo permutación suele ser la *Inverse Mutation* o Mutación inversa. Se puede observar un ejemplo de dicho operador en la siguiente figura (Chen, 2011):

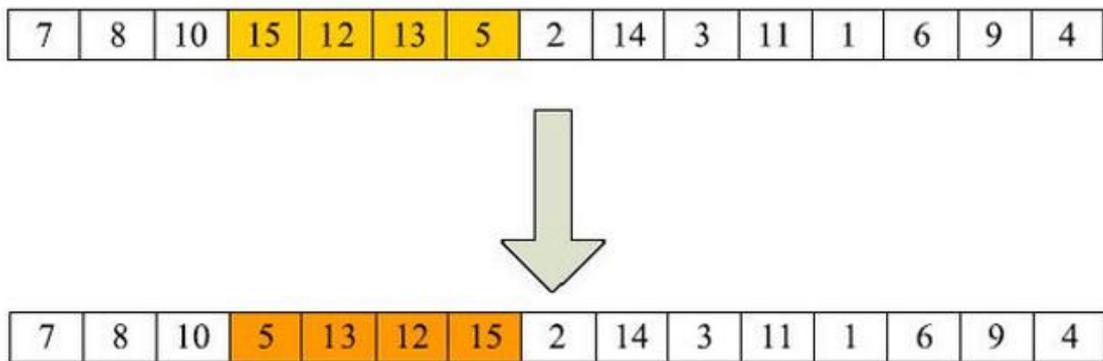


Ilustración 1: ejemplo del operador mutación inversiva (Chen, 2011)

Como se observa en el ejemplo, la Mutación inversa se basa en escoger 2 índices superior e inferior (siempre con una diferencia >1 entre ellos) e invertir el orden de los genes incluidos entre esos dos índices. Así, obtenemos una nueva solución que ha sido modificada en parte y permite explorar nuevos espacios de búsqueda.

Reemplazamiento

El reemplazamiento se basa en elegir cuáles serán los individuos de la población que pasarán a formar parte de la siguiente generación, una vez aplicados los principales operadores evolutivos y obtenidos los individuos evolucionados a partir de ellos. En cierta forma, comparte características con el operador de selección, siendo los principales métodos el Reemplazamiento mediante Ruleta, Reemplazamiento por Torneo, Reemplazamiento Proporcional al Fitness, etc (Holland, 1992) (Baker, 1985).

Una vez comprendidas las pautas básicas de un GA, se procede a analizar la evolución que ha sufrido estos tipos de algoritmos. En un principio, Los algoritmos genéticos fueron ideados para funcionar con una función objetivo. Sin embargo, Los problemas con objetivos múltiples surgen de forma natural en la mayoría de las disciplinas y su solución ha sido un reto para los investigadores durante mucho tiempo (Carlos A., Gary B., & David A., 2007). En el ámbito de los GA ocurre igualmente, y es en 1984 donde (Dunn, 1964) propuso el primer GA con enfoque multiobjetivo. Mediante esta presentación, en la siguiente subsección se profundiza en que se basa el enfoque multiobjetivo en los GA y sus ventajas principales frente el enfoque monobjetivo.

3.3.1 Enfoques monobjetivo y multiobjetivo en algoritmos genéticos

Como en muchos otros problemas, en los TFP se consideran comúnmente varios objetivos. Para convertir estos objetivos en uno solo, la solución más común en la bibliografía TFP es utilizar el método de suma ponderada.

En el método de las sumas ponderadas la idea es asociar a cada función objetivo con un coeficiente de peso y minimizar/maximizar la suma ponderada de las funciones objetivo (Zadeh, 1963). De esta manera, las múltiples funciones objetivo se transforman en una sola función objetivo. Los coeficientes de peso w_i son números reales tales que $w_i \geq 0$ para todo $i = 1, \dots, k$ y se supone que los pesos están normalizados, es decir $\sum w_i = 1$.

Sin embargo, uno de los inconvenientes más evidentes de este enfoque es la ardua tarea de determinar los valores de ponderación adecuados, lo que supone un reto de enorme importancia para los responsables de la decisión. En particular, combinaciones dispares de ponderaciones pueden dar lugar a soluciones idénticas, lo que agrava la complejidad del proceso de toma de decisiones. Además, al intentar combinar funciones objetivo-conflictivas en una sola, aplicando un enfoque monobjetivo, puede perderse información y, dependiendo de la naturaleza del problema, el rendimiento del algoritmo se ve perjudicado (Miranda, 2020).

Es aquí donde entran en juego los algoritmos multiobjetivo. La principal diferencia entre un problema de objetivo único y uno multiobjetivo es que las funciones objetivo para más de un objetivo constituyen un espacio multidimensional. Las soluciones óptimas, en este caso, se definen guiándose por el concepto de Dominancia (Deb & Jain, 2014).

Dada alguna función multiobjetivo f , cuyos componentes corresponden a i objetivos, una solución x domina a una solución y si (Ali Babar & Kitchenham, 2007):

$$\forall i \in \{1, \dots, M\}: f_i(x) \geq f_i(y)$$

y

$$\exists i f_i(x) > f_i(y)$$

Como el espacio objetivo es multidimensional, el planteamiento para hallar soluciones óptimas no se basa en encontrar un único punto, sino un conjunto de puntos no dominados. Se define como “Frente no dominado” a un conjunto de soluciones las cuales no se dominan unas a otras. En la siguiente figura sea aprecia un ejemplo sencillo a modo ilustrativo:

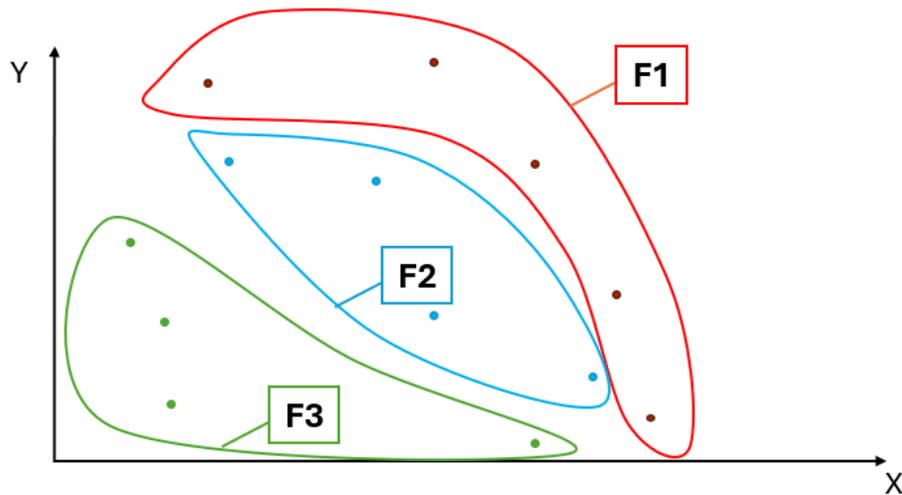


Figura 4: Ejemplo de 3 frentes no dominados para un problema con 2 funciones objetivo [Elaboración propia]

En la Figura 4, se puede observar un espacio de soluciones de un problema con dos funciones objetivo el cual se quiere maximizar en ambos sentidos. Una propiedad de este conjunto de puntos es que no es posible mejorar un objetivo sin empeorar otro. Debido a esta compensación, encontrar un conjunto diverso de puntos es crucial para tomar una decisión final precisa. De este modo, se obtiene una manera de clasificar jerárquicamente las soluciones dadas en este ámbito de varias dimensiones.

Ligado al concepto de soluciones no dominadas, cualquier solución que no sea dominada por un individuo del espacio de soluciones se denomina una solución globalmente Pareto-óptima (Carlos A., Gary B., & David A., 2007). Consecuentemente, se denomina Frente de Pareto (FP) al conjunto de soluciones que son globalmente Pareto-óptimas. La Figura 5 muestra el Frente de Pareto en distintas direcciones de direcciones de optimización:

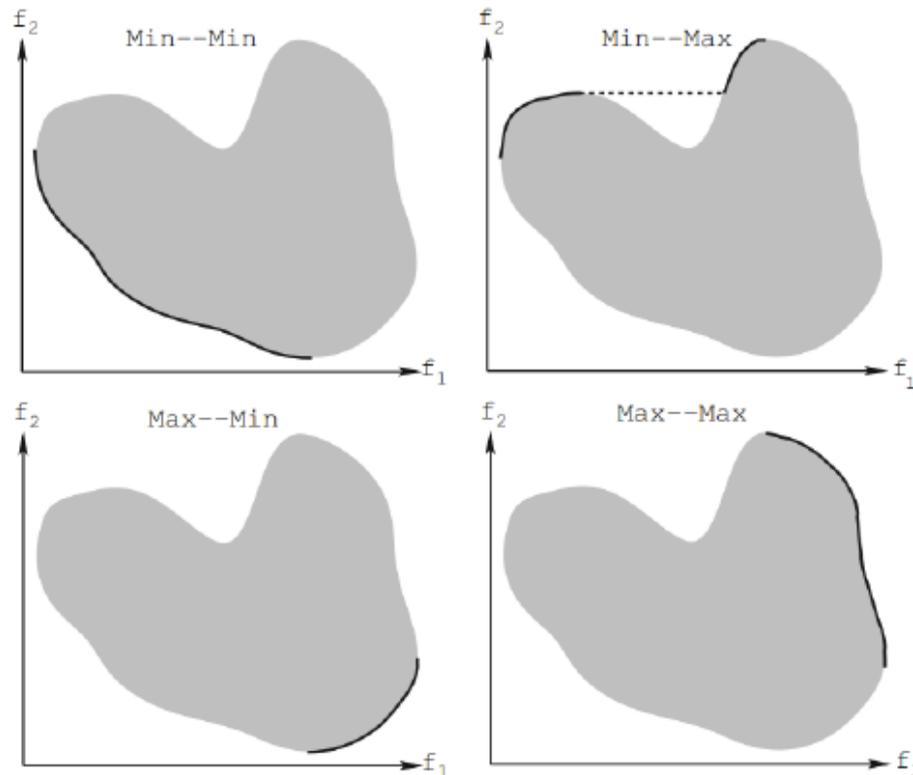


Figura 5: Conjuntos de soluciones Pareto óptimas en distintos tipos de problemas (ResearchGate)

Encontrar el Frente de Pareto de un problema suele ser bastante difícil. Por lo tanto, aproximaciones razonablemente cercanas suelen ser aceptables con un tiempo computacional limitado (Carlos A., Gary B., & David A., 2007).

3.3.2 Algoritmos genéticos multiobjetivo

Los algoritmos genéticos multiobjetivo se basan principalmente en la idea ofrecida por Goldberg (Miller & Goldberg, 1995). Este sugirió combinar el uso de la clasificación por frentes no dominados y la selección de padres mediante diferentes estrategias, enfocándose en intentar acercar una población hacia el frente de Pareto en un problema de optimización multiobjetivo.

Los 3 puntos clave en el funcionamiento de los algoritmos multiobjetivo son los siguientes (Carlos A., Gary B., & David A., 2007):

1. **Guiar a la población actual hacia el FP:** El primer concepto que comparten los distintos GA consiste en encontrar el conjunto de soluciones de la población que son Pareto no dominadas por el resto de la población. A estas soluciones se les asigna el rango más alto y se eliminan de la lista. A partir de la población restante, se determina otro conjunto de soluciones no dominadas por Pareto y se les asigna el siguiente rango más alto. Hay diferentes métodos a la hora de asignar los rangos a los individuos de una población según el concepto de dominancia:
 - Por rango de dominancia: ¿Cuántos individuos dominan a un individuo?

- Por recuento de dominancia: ¿Cuántos individuos domina un individuo?
- -Por profundidad de dominancia: ¿En qué "frente" se encuentra un individuo?

En la figura 6, se muestra un ejemplo de cada método de asignación según el concepto de dominancia para un problema con dos funciones objetivo de maximización, ilustrando el nivel de cada individuo mediante un número.

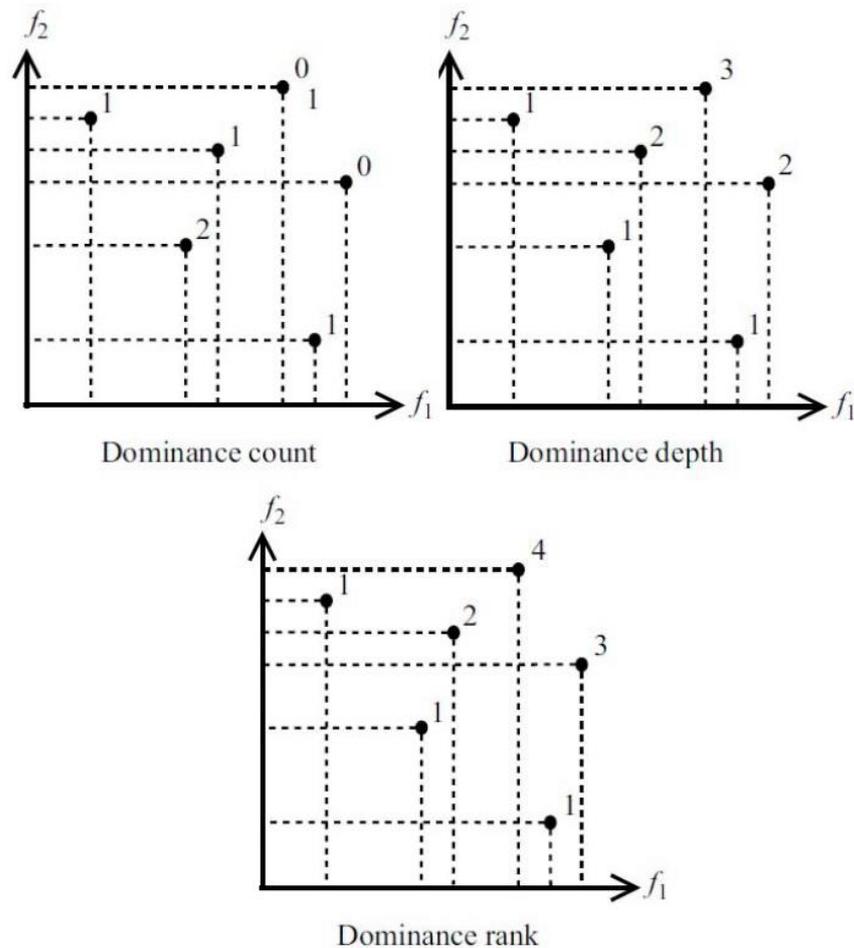


Figura 6: Ejemplo de clasificación por recuento de dominancia (izq.), clasificación por profundidad de dominancia (der.) y clasificación por rango de dominancia (abajo) (ResearchGate).

2. **Preservar la diversidad de las soluciones en el FP actual:** Este proceso continúa hasta que la población está debidamente clasificada. Goldberg también sugirió el uso de algún tipo de técnica de dispersión para evitar que el GA converja a un único punto en el frente. Un mecanismo de dispersión permite al GA mantener individuos debidamente dispersos a lo largo de toda la frontera no dominada (Schaffer J. D., 1985). Para ello hay varias técnicas utilizadas en la literatura, entre ellas se encuentra el enfoque de *Fitness sharing/Nitching*, el cual se va a emplear en el algoritmo que se empleara en este proyecto.

En un principio, este enfoque recopilaba individuos en nichos o vecindarios según su cercanía en el espacio de fitness. El tamaño de nicho se controla mediante un parámetro llamado radio de nicho (se pueden usar diferentes topologías para definir el nicho; rejillas, circunferencias, etc.). Se basa en contar cuántas soluciones se encuentran dentro del mismo nicho, disminuyendo proporcionalmente la aptitud cuanto más individuo se encuentren en el mismo (Deb & Goldberg, 1989).

3. **Aplicar los operadores genéticos convencionales:** Tras este proceso de clasificación, viene la fase evolutiva donde se le aplica a la población los principales operadores evolutivos (Selección, Cruce, Mutación y Reemplazamiento), repitiendo el mismo proceso durante las siguientes generaciones.

Una vez definido a rasgos generales la idea principal de los GA multiobjetivo, se presentan los principales algoritmos genéticos multiobjetivo, los cuales han sentado las bases para formar muchos de los algoritmos de este tipo que se conocen hoy en día (Carlos A., Gary B., & David A., 2007):

- **Vector Evaluated GA (VEGA)** (Schaffer J. D., 1985): Aunque no se basa en el concepto de dominancia de Pareto cabe mencionarlo por ser el predecesor de el termino MOEA (Multiobjective Evolutionary Algorithm). En este enfoque una fracción de cada población sucesiva se selecciona basándose en el rendimiento de objetivos separados. Para un problema con k objetivos, se generarían k subpoblaciones de tamaño $\frac{M}{k}$ cada una (suponiendo una población total de M). Cada subpoblación utiliza sólo una de las k funciones objetivo para la asignación de la aptitud. Mediante un operador de selección proporcional al fitness se escogen las parejas a cruzar y a continuación, estas subpoblaciones se mezclan para obtener una nueva población de tamaño M , sobre la que el GA aplicaría los operadores de cruce y mutación de la forma habitual.
- **Multi-objective Genetic Algorithm [MOGA]:** En (Fonseca & Fleming, 1999) propusieron una técnica denominada MOGA, en la que el ranking de dominancia se establece mediante el método de “por rango de dominancia”. Además, se le aplica a cada individuo un contador de *niching*. Mediante estos dos parámetros se proporciona un fitness a cada individuo para tras ello aplicar a la población actual los operadores genéticos habituales, repitiendo el proceso a lo largo de las generaciones.
- **Niched-Pareto Genetic Algorithm (NPGA):** (Miller & Goldberg, 1995) propuso un MOEA por selección de torneo, basado también en la dominancia de Pareto por rango de dominancia. Se basa principalmente en que dos individuos elegidos al azar se comparan con un subconjunto de toda la población. Si uno de ellos está dominado (por los individuos elegidos al azar de la población) y el otro no, gana el individuo no dominado. Cuando ambos competidores están dominados o no (es decir, hay un empate), el resultado del torneo se decide mediante *fitness-sharing*.
- **The Multiobjective Messy Genetic Algorithm** (Lamont, 2000) (MOMGA): Se basa en el algoritmo conocido como *Messy GA* pero en su versión multiobjetivo. Al igual

que los dos tipos de GA anteriores, el método de clasificación de los individuos es por rango de dominancia. A diferencia de las representaciones tradicionales de cromosomas de longitud fija, MOMGA utiliza un esquema de codificación desordenado en el que los cromosomas pueden tener longitudes variables e incluir información explícita sobre las posiciones de los genes. Esta codificación permite una representación más flexible de las soluciones, permitiendo al algoritmo ajustar y evolucionar dinámicamente la estructura de las soluciones potenciales para adaptarse mejor al panorama del problema.

- **Algoritmo micro genético multiobjetivo** (Coello Coello & Toscano Pulido, 2001): es un GA con una población pequeña y un proceso de reinicialización. En primer lugar, se genera una población aleatoria. Esta población aleatoria alimenta la memoria de población, que se divide en dos partes: una parte reemplazable y otra no reemplazable. La parte no sustituible de la memoria nunca cambia durante toda la ejecución y está destinada a proporcionar la diversidad necesaria para el algoritmo. En cambio, la parte reemplazable experimenta cambios después de cada ciclo del micro-GA. Este algoritmo también clasifica a sus individuos según el rango de dominancia.
- **Non-Dominated Sorting Algorithm (NSGA)** (Srinivas & Deb, 1994): Este algoritmo NSGA se basa en varias capas de clasificación de los individuos. Antes de realizar la selección de individuos que formaran parte de la reproducción, la población se clasifica en función de frentes no dominados, haciendo uso del método por profundidad de dominancia: Todos los individuos no dominados se clasifican en una categoría (con un valor de fitness ficticio igual para todos los miembros de ese frente, que es proporcional al tamaño de la población, para proporcionar un potencial reproductivo igual a estos individuos).

Para mantener la diversidad de la población, estos individuos clasificados “comparten” sus valores ficticios de fitness mediante el operador de *fitness-sharing*, degradando el fitness de los individuos que se ubican dentro del mismo nicho. Este operador hace uso de un parámetro llamado radio de nicho σ_{share} . A continuación, se ignora este grupo de individuos clasificados y se considera otro frente de individuos no dominados. Tras clasificar toda la población, se aplica un operador de selección estocástica proporcional al fitness-ficticio y se les aplican los operadores genéticos convencionales a los individuos seleccionados.

Se hace especial énfasis en el algoritmo NSGA original, puesto que el presente proyecto emplea una versión mejorada de este mismo, llamado NSGA-II, explicado a continuación.

3.4 NSGA-II: Non-Sorted Genetic Algorithm II

Como se va a ver en el apartado 4.2.3, una revisión exhaustiva indica que NSGA-II es el algoritmo más utilizado para los problemas de optimización combinatoria multiobjetivo siendo un método estándar para problemas de coalición como lo es el TFP (Verma, Pant, & Snasel, 2021).

Este algoritmo se desarrolló con el objetivo de lidiar con los principales problemas que acarrea el algoritmo NSGA original, que eran los siguientes:

- Necesidad de especificar el parámetro del *fitness sharing*: El rendimiento del método depende en gran medida del valor que se le atribuya al parámetro σ_{share} . Este parámetro suele ser fijado por el usuario, aunque existen algunas directrices (Deb & Goldberg, 1989).
- Falta de elitismo: El elitismo es una estrategia de mantenimiento y promoción de las mejores soluciones encontradas durante el proceso evolutivo. El elitismo puede acelerar significativamente el rendimiento del GA, que también puede ayudar a prevenir la pérdida de buenas soluciones una vez encontradas (Rudolph, 2001) (Zitzler, Deb, & Thiele, 2000)
- Gran complejidad computacional para la clasificación no dominada: Dado que cada solución debe compararse con todas las demás soluciones de la población, la complejidad global del enfoque de la función de reparto es $O(n^2)$ (Deb & Jain, 2014)

El primer problema se solucionó mediante el operador de diversidad llamado *Crowding distance*, explicado más adelante en esta misma sección.

La falta de elitismo se elimina mediante la conservación de las mejores soluciones de la generación anterior, juntándolos con los nuevos hijos generados y aplicando el reemplazamiento de la población sobre todos los individuos (antiguos y nuevos).

Respecto al tercer inconveniente, el algoritmo NSGA-II lo intenta solucionar mediante lo que se denominó *Clasificación No-Dominada Veloz*. Este método de clasificación se explicará un poco más adelante en esta misma sección.

A Grosso modo, como se puede observar en la Figura 7, este metaheurístico sigue las siguientes rutinas como método de evolución a resultados óptimos:

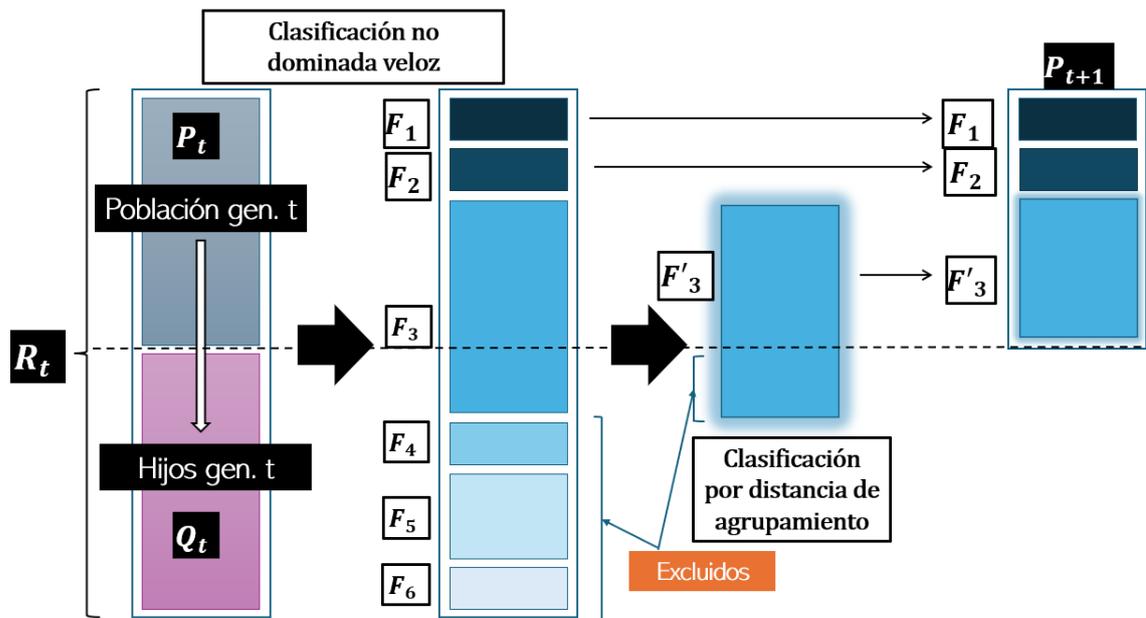


Figura 7: Bucle principal del algoritmo NSGA-II (Elaboración propia)

1. Como en cualquier Algoritmo genético, aplica operaciones de selección, cruce y mutación en la población presente (P_t) de tamaño N , obteniendo una nueva población R_t de tamaño $2N$.
2. utiliza un proceso de reemplazamiento no dominado, donde las soluciones se ordenan en frentes no dominados.
3. Se seleccionan N individuos de la población R_t . En el último frente a seleccionar, los individuos se ordenan según el operador de distancia de agrupamiento, excluyendo a los individuos que tiendan a agruparse con el objetivo de conservar la diversidad de soluciones.

3.4.1 Clasificación por Frentes no dominadas veloz

En el bucle principal de NSGA-II El primer frente no dominado es el conjunto de Pareto de la población actual (F_1), el segundo frente no dominado es el conjunto de Pareto de la población actual cuando se excluye el primer frente no dominado (F_2), y así sucesivamente (Deb, Pratap, Agarwal, & Meyarivan, 2002).

En la siguiente figura se presenta el pseudocódigo que describe el proceso de ordenación en frentes no dominados:

1	H	El cruce hace uso de dos padres para generar un hijo (H).
2	C_m y $C_{m'}$	Individuos seleccionados para el cruce.
3	$Q = \forall t_i \cup \forall t_{i'}; t_i \in C_m, t_{i'} \in C_{m'}$	Q es el conjunto de los equipos pertenecientes a C_m y $C_{m'}$.
4	$\vec{F}_i = \langle f_{1i}, f_{2i}, f_{3i} \rangle,$ $\vec{F}_{i'} = \langle f_{1i'}, f_{2i'}, f_{3i'} \rangle$	Fitness multiobjetivo \vec{F}_i y $\vec{F}_{i'}$ respectivos a dos equipos cualquiera del conjunto Q.
5	$f_i = \frac{1}{3}f_{1m} \times \frac{1}{3}f_{2m} \times \frac{1}{3}f_{3m}, t_i \in C_m$ $f_{i'} = \frac{1}{3}f_{1m'} \times \frac{1}{3}f_{2m'} \times \frac{1}{3}f_{3m'}, t_{i'} \in C_{m'}$	Se convierten los fitness t_i y $t_{i'}$ de en mono-objetivo mediante la suma ponderada equitativa de las 3 dimensiones ($f_m, f_{m'}$).
6	Para cada equipo en Q:	
	$p_j = \frac{f_j}{\sum_{f \in Q} f}$	Se calcula la probabilidad de un equipo j perteneciente a Q de ser seleccionado para pertenecer al hijo siendo dicha probabilidad proporcional a su fitness.
7	Se transforman dichas probabilidades. para que estas sean acumulativas p'_j , en el intervalo [0,1].	
8	Mientras H no tenga la misma longitud que los padres C_m y $C_{m'}$:	
9	Se calcula un numero aleatorio (a) en el intervalo [0,1].	
10	Se selecciona el equipo asociado al intervalo donde caiga el numero a.	
11	El equipo seleccionado no puede volver a ser elegido.	
12	Todos los alumnos repetidos en H son reemplazados aleatoriamente por un alumno que no se encuentre en dicha solución.	

Figura 8: Pseudocódigo de la clasificación no-dominada veloz (Deb, Pratap, Agarwal, & Meyarivan, 2002).

El pseudocódigo de la Figura 8 se puede resumir de la siguiente manera:

- Se recorren todos los pares de soluciones posibles, se lleva un registro de las soluciones dominadas por cada solución (S_p), así como un contador de cuántas soluciones la dominan (n_p).
- Para obtener el primer frente no dominado (F_1), se recorre la población para encontrar todas las soluciones con $n_p = 0$. Para encontrar los otros frentes, simplemente se itera a través de todas las listas (S_p) de soluciones dominadas que se mantienen para el frente anterior, y para todas estas soluciones se disminuye su contador ($n_q = n_q - 1$).
- Si algún contador (n_q) llega a 0, forma parte del siguiente frente (F_{i+1}) debido a que todo lo que lo dominaba estaba en un frente superior.

Una vez la población actual (R_t) se clasifica en sus frentes no dominados correspondientes, se aplica el operador de distancia de agrupamiento:

3.4.2 Clasificación por distancia de agrupamiento

Dado que el tamaño total de la población de R_t es mayor que el tamaño de población esperado, algunos frentes podrían no ser incluidos en P_{t+1} . Además, para introducir cierta diversidad entre las soluciones no dominadas se hace uso de la Clasificación por Distancias de Agrupamiento (CDA), que se utiliza en la selección por torneo binario y durante la fase de reducción de la población.

En la fase de reducción de R_t , todos los frentes que no puedan ser incluidos simplemente se eliminan. Cuando se considera el último frente permitido, puede que existan más soluciones en el último frente que espacios restantes en P_{t+1} . Para seleccionar qué soluciones sobrevivirán, se aplica el CDA. La selección tiene como objetivo preservar la diversidad de la población mediante la medición de la densidad de soluciones que rodean una solución en particular. La idea es cuantificar la relevancia de una solución para mantener la diversidad. (Ali Babar & Kitchenham, 2007).

El enfoque utilizado para obtener una estimación de la densidad de soluciones que rodean a una solución concreta en la población se denomina enfoque del vecino más cercano (Carlos A., Gary B., & David A., 2007). En la Figura 9 se aprecia un ejemplo de cálculo de la distancia de agrupamiento:

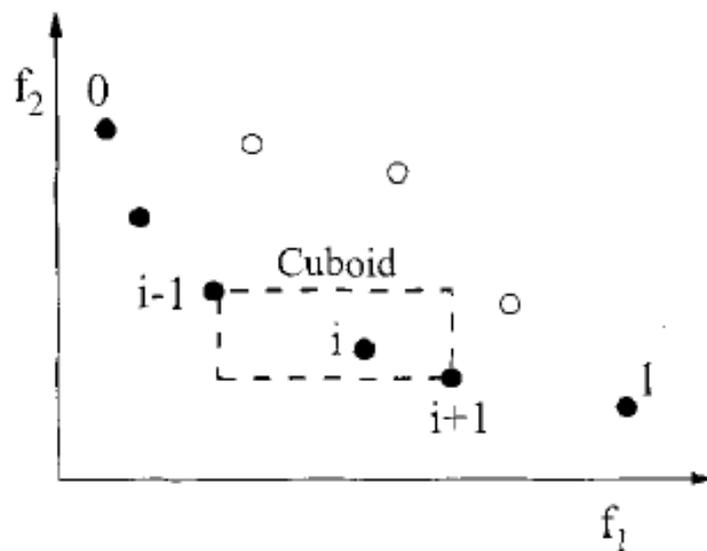


Figura 9: Cálculo de la distancia de agrupamiento (Deb, Pratap, Agarwal, & Meyarivan, 2002)

Se mide la distancia media de dos puntos a cada lado de este punto a lo largo de cada uno de los objetivos. Esta cantidad sirve para estimar el perímetro del cuboide formado por los vértices de los vecinos más próximos (distancia de aglomeración). En la Figura 9, la distancia de agrupamiento de la solución i en su parte frontal (marcada con círculos sólidos) es la longitud lateral media del cuboide (mostrado con un recuadro discontinuo). Para calcular la magnitud de esta métrica para cada solución, se calcula las distancias euclídeas multidimensionales de cada solución respecto a las demás soluciones y se multiplican estas distancias para obtener la métrica deseada.

3.4.3 Operador *crowded-comparison* para el reemplazamiento

Mediante la clasificación por frentes no dominados y la CDA podemos definir el operador *crowded-comparison*, que combinará las medidas de rango de no dominación y distancia de agrupamiento en una medida general que se puede utilizar en cualquier solución dentro de una población NSGA-II., podemos definir el operador *crowded-comparison* como:

- Si dos soluciones no tienen el mismo rango de no dominación, la que tiene el rango de no dominación más bajo domina a la otra.
- Si dos soluciones tienen el mismo rango de no dominación, entonces la que tiene la mayor distancia de agrupamiento domina a la otra.

Mediante este operador obtenemos las soluciones que pasaran a formar parte de la población de la siguiente generación, denominado como P_{t+1} en el esquema de Figura 7.

3.5 Medición de la calidad de las poblaciones de individuos

Hasta ahora, se ha definido la medición de la calidad de un individuo de un problema multiobjetivo, incluyendo la comparación entre individuos mediante el término “dominancia”. Sin embargo, la mayoría de los algoritmos genéticos multiobjetivo trabajan por medio de poblaciones de individuos.

Estas poblaciones de soluciones reúnen una multitud de individuos (soluciones del problema), cada uno con su fitness particular. Debido a esto, la primera duda que surge es, ¿Cómo se puede comparar una población de soluciones obtenida con un algoritmo con otra obtenida mediante otro algoritmo?

De especial interés en este contexto es el indicador de hipervolumen (Zitzler & Thiele, 1999) ya que es el único indicador de calidad conocido que es totalmente sensible a la dominancia de Pareto, es decir, siempre que un conjunto de soluciones domina a otro conjunto, tiene un valor del indicador de hipervolumen más alto que el segundo conjunto.

El indicador de calidad poblacional Hipervolumen, es una métrica de rendimiento para indicar la calidad de un conjunto de aproximación no dominado. Suponiendo un problema de minimización, el hipervolumen mide el espacio cubierto por un conjunto de individuos no dominados en el espacio \mathbb{R}_d , según un punto de referencia dado, siendo d el número de objetivos.

Como ejemplo, se toma un problema bi objetivo de minimización. En la siguiente figura se ilustra la aportación individual de cada solución (punto) a la medida del hipervolumen para dicho problema:

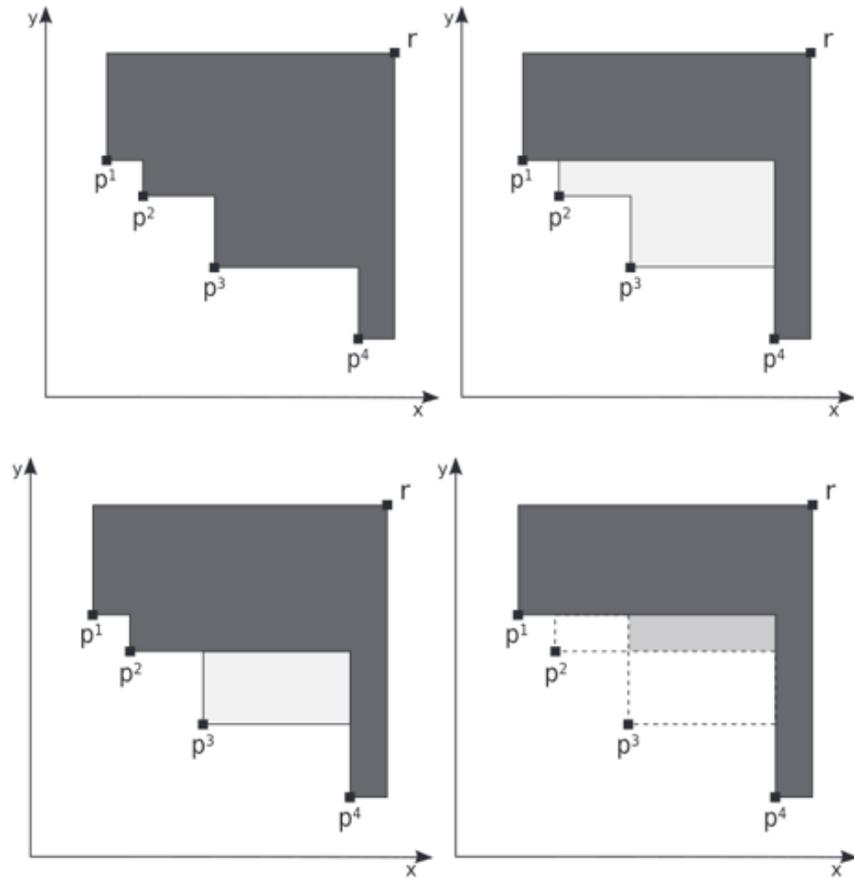


Figura 10: Ejemplo de aportación individual a la métrica de hipervolumen (Zitzler & Thiele, 1999)

En la Figura 10, se puede observar, en el gráfico de arriba a la izquierda, que el hipervolumen en ese caso es la suma del área cubierta (según el punto de referencia r) por cada punto en el espacio \mathbb{R}_2 . En los demás gráficos, se aprecian diferentes ejemplos de aportaciones individuales de cada solución al hipervolumen.

Tomando en cuenta que en el caso que se aborda se van a normalizar los fitness en todas las dimensiones para las soluciones obtenidas y que el problema a abordar contiene 3 dimensiones, el punto de referencia que se va a tomar en este trabajo es $r = [1,1,1]$, ya que el problema, aunque sea de maximización, se va a afrontar como un problema de minimización mediante la conversión al negativo de todas las funciones objetivo. Este punto se puede tomar como el fitness que tomaría la solución ideal del problema.

4 Revisión bibliográfica

La revisión bibliográfica se ha estructurado de tal forma que, primero se introduce el problema que se va a abordar y se presenta una clasificación general de los métodos de formación de equipos para tras ello centrarse en los problemas de formación de equipos optimizados (sección 4.1). Una vez conceptualizado el problema, se realizará una revisión bibliográfica sobre los principales artículos que tratan el tema (sección 4.2), para finalizar con los aportes principales del proyecto al estado del arte (sección 4.3).

4.1 El problema de formación de equipos

La conformación de grupos es un aspecto fundamental en la naturaleza humana, ya que establece límites y fomenta la interacción entre sus miembros. En particular, los grupos desempeñan un papel central en nuestras vidas y resulta difícil concebir la existencia humana sin la presencia de un grupo (Johnson, 2004). Dichos grupos se forman cuando dos o más personas interactúan e influyen mutuamente en sus discusiones con el propósito de aprender, resolver problemas o comprender más plenamente cualquier asunto en cuestión.

Formar un equipo es un desafío complicado para líderes, empleados, educadores, estudiantes, jugadores y muchas otras personas. Requiere explorar varias combinaciones de equipos posibles, recopilar información sobre las características y relaciones sociales de los miembros, y adaptarse a las condiciones dadas en un contexto social específico (Liu, Joy, & Griffiths, 2013). Además, no siempre resulta evidente encontrar las combinaciones de equipo más eficientes dentro de un grupo de personas.

En lo que respecta a los equipos, distintas estructuras conducirán a resultados diversos en la misma tarea muchas veces. Algunos equipos podrían ser más eficientes que otros, únicamente debido a la combinación de características de sus integrantes (Campion, Medsker, & Higgs, 1993). Elegir correctamente a los miembros del equipo probablemente implicaría cierto equilibrio entre habilidades técnicas y sociales, para lograr con éxito el objetivo para el cual se formó el equipo al mismo tiempo que se fomenta la satisfacción de sus miembros.

4.1.1 Interacción entre las personas y la tecnología en el proceso de formación de equipos

Primero de todo, resulta interesante estudiar la interacción entre los humanos y la tecnología a la hora de formar los equipos. El artículo (Gómez-Zará, 2020) es un documento de investigación que presenta un sistema de clasificación completo para las tecnologías que ayudan a la formación de equipos.

Dichos autores, ofrecen una taxonomía que se enfoca en el modo en que los sistemas sociotécnicos facilitan la formación de equipos basándose en las interacciones entre sistemas y usuarios. Según esta taxonomía los sistemas de formación de equipos deben tener en cuenta:

• **Contribución de los usuarios:** Hasta qué punto los sistemas conceden a los usuarios el control de sus preferencias de compañeros de equipo, pertenencia y composición final del equipo.

• **Participación de los usuarios:** El número de usuarios a los que el sistema permite participar en el proceso de formación de equipos.

Interseccionando estas dos dimensiones, se sugieren 4 tipos de sistemas en las que la tecnología ayuda a la formación de equipos que se presentan a continuación:

Equipos optimizados: Es la tipología de equipo a trabajar en el presente trabajo. Se caracterizan por una baja contribución y participación. Un usuario participa en el proceso de formación del equipo proporcionando los datos de entrada requeridos por el sistema. Este usuario no controla el proceso de formación de equipos, solamente sirve como proveedor de los datos. Una vez obtenidos los datos, un algoritmo será el responsable de formar los equipos sujetos a ciertas restricciones y objetivo/s.

Equipos supervisados: Se caracterizan por una alta contribución, pero solamente con la participación de un individuo autorizado. Los sistemas se emplean para apoyar las decisiones de los usuarios, simular combinaciones de equipos o conseguir más miembros para sus equipos. En última instancia, el usuario toma las decisiones finales basándose en los resultados de los sistemas.

Equipos autoformados: En este cuadrante, la contribución y la participación de los usuarios son altas. La composición final de los equipos no se conoce hasta que no se ha completado la fase de formación del equipo, ya que la pertenencia depende de las elecciones secuenciales realizadas por los usuarios.

Equipos aumentados: En este cuadrante, la contribución de los usuarios es baja, pero su participación en el proceso de formación de equipos es alta. Un sistema que ayuda a los usuarios a encontrar a los compañeros de equipo más adecuados da como resultado en "augmented teams" o equipos aumentados.

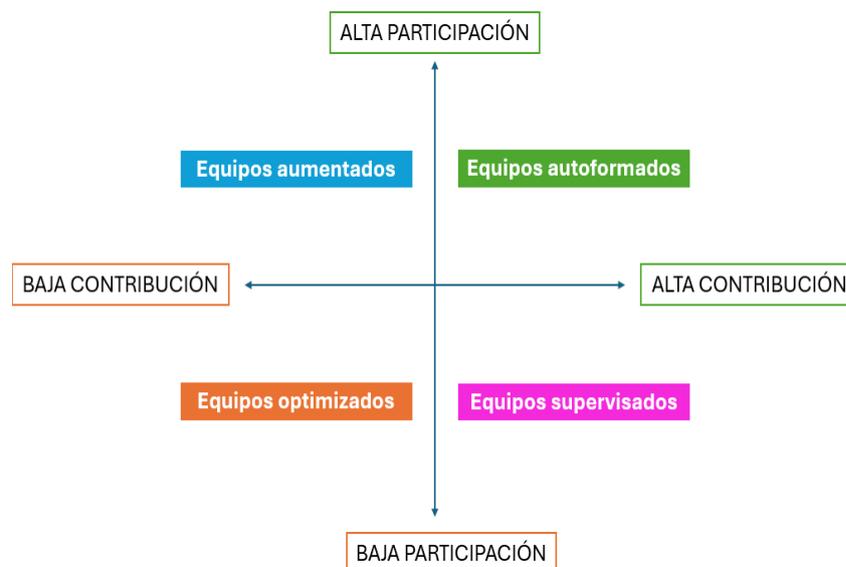


Figura 11: Tipos de equipos según la interacción entre el sistema y el usuario (Gómez-Zará, 2020)

En la Figura 11 se puede observar los niveles en las 2 dimensiones (participación y contribución) y como cada uno de los 4 equipos descritos pertenece a los cuadrantes generados por los dos ejes.

4.1.2 Equipos optimizados en el aula

El presente trabajo se enfoca en la formación automática de los equipos estudiantiles (equipos optimizados) mediante la creación de modelos y algoritmos que permiten formar los equipos de una forma rápida y lo más efectiva u óptima posible según un criterio dado.

Concretamente, la formación de equipos en el aula se diferencia de la formación de equipos genérica en que, en el ámbito académico, todos los individuos participantes deben formar parte de al menos un equipo.

Los problemas de formación de equipos se pueden clasificar dentro de lo que se conoce como *Coalition Structure Generation Problem (CSG)*. Principalmente, el problema de crear estructuras de coalición consiste en obtener una partición de un conjunto finito de individuos en coaliciones disjuntas de forma que se logre maximizar el incentivo obtenido por la actuación de éstas o, en su defecto, minimizar el coste derivado de dichas actividades (Candel, Sanchez-Anguix, Alberola, Julián, & Botti, 2023).

Una diferencia con el CSG clásico es que en el CSG clásico cualquier tamaño de equipo es válido mientras que en nuestro caso no tiene por qué ser así. Según esto, la creación de equipos en el aula (problema en el que se basa este trabajo) puede verse como una instancia del problema de CSG, pues se busca obtener la mejor organización de los alumnos para llevar a cabo los proyectos que se les plantean (Talal Rahwan, 2015).

El proceso comienza recopilando información sobre los estudiantes, como sus habilidades, preferencias, historial académico y datos demográficos. Luego, se utilizan algoritmos de optimización para formar grupos que maximicen la diversidad de habilidades y minimicen posibles discrepancias o conflictos entre los integrantes. Estos algoritmos pueden considerar varios criterios, como la homogeneidad de habilidades, la diversidad de antecedentes o la distribución equitativa de estudiantes con características específicas.

A continuación, se presenta una revisión general de los artículos científicos que se basan en problemas de formación de equipos optimizados, diferenciando entre los artículos con un enfoque monobjetivo del problema y los que utilizan un enfoque multiobjetivo.

4.2 Revisión generalizada sobre la formación de equipos optimizados académicos

Cabe mencionar, que este capítulo se ha estructurado de tal forma que los primeros 8 artículos presentados se basan en enfoques monobjetivo del problema TFP en el aula. Tras esta primera revisión monobjetivo, se van a exponer los principales problemas del enfoque monobjetivo y como el enfoque multiobjetivo intenta mejorar esas limitaciones.

Consecuentemente se pasará a presentar los principales artículos que aportan dicho enfoque multiobjetivo para el TFP académico, y se finalizara explicando que aportación brinda el presente proyecto a la bibliografía actual.

4.2.1 Revisión sobre el TFP académico con enfoque monobjetivo

Una vez explicada las diferentes maneras de formar equipos, las taxonomías sobre los algoritmos que se emplean de forma general y que características individuales se toman en cuenta, resulta interesante realizar una revisión literaria más concreta, enfocándose en la formación de grupos para el aprendizaje colaborativo en el aula como se hace en (Odo, Masthoff, & Beacham, 2019). En esta revisión se responden cuatro preguntas con la intención de realizar una clasificación de la literatura ya creada sobre la formación de equipos. Las presentes cuestiones son las siguientes:

1. ¿Tienen en consideración los profesores la Inter-homogeneidad e intra-heterogeneidad a la hora de formar los grupos?
2. ¿Qué características individuales son un criterio importante para formar los grupos?
3. ¿Los individuos incluidos en los experimentos son alumnos reales o son casos simulados?
4. ¿Cuáles son las diferentes técnicas utilizadas en la formación de equipos automatizado?

En (Ali Babar & Kitchenham, 2007), se propone una heurística basada en un algoritmo genético para la formación de grupos intra-homogeneos en cuanto a las características de los estudiantes en el aula, donde solamente se toman en cuenta los rasgos de personalidad del “Big Five Inventory”. Consecuentemente, al considerar que los grupos sean intra-homogeneos se descarta la inter-homogeneidad entre sí. Teniendo en cuenta, además, las siguientes restricciones para el tamaño de los grupos: debe ser al menos dos; no puede exceder la mitad del número total de estudiantes si es par; no puede exceder la mitad más uno del número total de estudiantes si es impar. Para realizar el experimento se hace uso de 2 grupo de estudiantes de dos asignaturas diferentes donde en cada asignatura se dividen en un grupo de control y otro experimental. Los grupos experimentales se formaron en base al algoritmo propuesto y los de control según las preferencias de los alumnos. Cabe mencionar que el método propuesto benefició los resultados académicos de los estudiantes.

(Ma, y otros, 2022) presenta una propuesta de un algoritmo exacto mediante CPLEX para la formación de grupos colaborativos en el aula basados en roles según la metodología E-CARGO considerando grupos liderados por un miembro. Para ello, toma en cuenta 3 tipos de restricciones: 1) Restricción para evitar conflictos entre los integrantes, 2) Restricción para considerar la diversidad de género dentro del grupo y 3). restricciones para controlar el tamaño del equipo de aprendizaje. Para las pruebas experimentales se hizo uso de un grupo de 51 estudiantes de ingeniería de Software en la *Hunan Normal*

University, donde se comparó el método exacto mediante CPLEX con varios métodos heurísticos como AG, métodos Greedy, etc. Finalmente, se observó que los experimentos muestran que el método de optimización es efectivo para obtener soluciones óptimas al mismo tiempo que garantiza los mejores resultados generales de aprendizaje para todos los estudiantes.

El artículo (Andrejczuk, Bistaffa, Blum, Rodríguez-Aguilar, & Sierra, 2019), contribuye notablemente en el campo de estudio con la introducción de Las competencias, la personalidad y el género de los miembros del equipo como factores clave en la formación de equipos heterogéneos en cuanto a su función de equipo. Tiene en cuenta que a la hora de componer los equipos cada alumno tenga asignada al menos una competencia y que cada competencia este asignada exactamente a un único alumno. El objetivo de este modelo reside en obtener una partición de equipos en los que los equipos sean lo más homogéneos (en cuanto a nivel de competencia) posibles entre sí y que todos ellos estén lo más cerca posible de la competencia requerida para dicha tarea. Cabe destacar que, compara un algoritmo exacto con uno heurístico con datos reales para ver si existen diferencias significativas entre las funciones objetivos obtenidas entre los dos métodos, concluyendo en que no se encontraron diferencias significativas en el rendimiento de los algoritmos (en términos relativos) al comparar entre los distintos tipos de tareas. Esto sugiere que las técnicas desarrolladas son altamente robustas.

En cuanto a plataformas de generación de equipos colaborativos (Ugarte, Aranzabal, Arruarte, & Larrañaga, 2022) presenta *Talsor*, una plataforma para la formación automática de equipos heterogéneos en cuanto a la conducta de los integrantes, aunque el objetivo final es obtener una partición de equipos lo más homogénea posible entre sí en cuanto a la tendencia de conducta. Para ello, previamente se emplea la teoría de roles de Belbin para asignar un rol predominante a cada individuo. *Talsor* permite jugar con el número y tamaño de los equipos, aunque la única restricción implementada respecto al tamaño de los grupos es que todos deben de ser iguales. Es importante destacar que esta plataforma se basa en un algoritmo genético para formar los equipos y que para los experimentos se hizo uso de estudiantes reales de la UPV/EHU. Los resultados experimentales fueron positivos, demostrando que los equipos de estudiantes logran un rendimiento significativamente superior en comparación con los casos en los que los estudiantes forman sus propios equipos.

Aunque el artículo (Shi-Jie(Gary) & Li, 2004) se enfoque en la formación de equipos multifuncionales en el ámbito de la ingeniería, tiene una notable aportación debido a que este modelo incluye 3 columnas fundamentales en la idiosincrasia de un equipo: El conocimiento multifuncional de los miembros, la capacidad de cada individuo de trabajar en equipo y la relación intrapersonal donde para descubrir el perfil de personalidad de cada integrante se utiliza una escala tipo Myers-Briggs. Para el experimento, se utiliza una compañía de 30 trabajadores donde se hace uso de un modelo de optimización exacto para conseguir los grupos que contengan a los trabajadores con mayor capacidad de trabajo en equipo y mejor relación personal. Los resultados ayudan a establecer un equipo multifuncional eficiente. Con esto. la gerencia puede organizar equipos con miembros que

no solo tienen experiencia en su área funcional, sino que también tienen habilidades de trabajo en equipo y disfrutan de una buena relación de trabajo con sus compañeros.

En cuanto a las encuestas de autopercepción de Belbin, estas no suelen incluir la retroalimentación de los compañeros para actualizar dichos roles individuales. Estas encuestas de retroalimentación se ven implementadas en el artículo (Juan M. Alberola, 2016). Dicha investigación implementa el aprendizaje bayesiano para mejorar la adecuación de roles para la formación de equipos heterogéneos en el aula. Para ello hace uso de una Inteligencia Artificial (IA) iterativa, donde los primeros grupos de varios estudiantes reales del grado de Turismo de la UPV se generan aleatoriamente basándose en los problemas de generación de estructuras de coalición. Por un lado, los resultados muestran que los estudiantes percibieron un mayor grado de cooperación y coordinación en los equipos propuestos por la herramienta. Por otro lado, el experimento también proporciona un mayor apoyo para que los estudiantes perciban las actitudes de sus compañeros de equipo como positivas.

El artículo (J. -M. Flores-Parra, 2018) también hace uso de la teoría de Belbin para la formación de equipos colaborativos heterogéneos con distintos roles, pero esta vez se enfoca el modelo como un Análisis de Redes Sociales (SNA), donde se emplea un algoritmo de Análisis de Redes (Grafos). El experimento se realizó sobre varios estudiantes de una Universidad de México donde se compararon los resultados ofrecidos por el algoritmo respecto a los grupos generados según el criterio del profesor. Los resultados obtenidos demuestran que, el profesor, teniendo en cuenta las comunidades, pudo tomar la decisión final de qué equipos formar. Según su experiencia, pudo ajustar los equipos de acuerdo con la teoría de Belbin. Al contrastar los grupos formados por el profesor con los resultados encontrados utilizando SNA, se demostró que todavía es necesario mejorar estos algoritmos para encontrar mejores equipos de trabajo en el futuro.

La mayoría de los autores hasta la fecha, realizan sus experimentos solamente con un grupo de alumnos y no comparan las soluciones obtenidas con una amplia gama de aproximaciones. Este no es el caso de (Yannibelli & Amandi, 2012), este artículo aborda el problema de formar equipos de aprendizaje colaborativo teniendo en cuenta los roles de los estudiantes y haciendo hincapié en la formación de equipos equilibrados según los roles de sus miembros basándose en la teoría de Belbin. Para ello, 10 conjuntos de datos y compara en cada uno de ellos el algoritmo genético propuesto con otros algoritmos como la búsqueda exhaustiva y el método aleatoria para verificar su efectividad. Mediante los resultados, se observó que para cada uno de los 10 conjuntos de datos el algoritmo obtuvo soluciones de muy buena calidad, logrando una aptitud media mayor de 8,5 en una escala sobre 10, todo esto con tiempo de ejecución medio más que aceptable siendo este mayor que 100 segundos y menor que 800 segundos.

4.2.2 Limitaciones comunes en la bibliografía TFP académico monobjetivo

En general, al analizar los diferentes artículos sobre algoritmos de formación de equipos en el aula, se pueden identificar algunas limitaciones comunes en la mayoría de ellos. Estas limitaciones sugieren áreas en las que se podría mejorar la investigación futura

y proporcionan oportunidades para desarrollar enfoques más completos y efectivos en la formación de equipos colaborativos. A continuación, se resumen las principales deficiencias que se observan en los artículos revisados:

1. Pocos conjuntos de datos: La mayoría de los artículos revisados utilizan un número limitado de conjuntos de datos en sus pruebas experimentales. Esto podría limitar la generalización de los resultados y la comprensión de cómo los algoritmos de formación de equipos funcionan en diferentes contextos educativos.

2. Falta de consideración de aptitudes técnicas: Muchos de los artículos revisados se centran en características individuales como rasgos de personalidad o roles, pero no incluyen consideraciones sobre las aptitudes técnicas de los estudiantes. La inclusión de habilidades técnicas relevantes para la materia específica en la formación de equipos podría ser crucial para lograr un aprendizaje colaborativo más efectivo y equilibrado.

3. Ausencia de comparaciones exhaustivas: A menudo, los artículos revisados no comparan de manera exhaustiva los enfoques heurísticos o metaheurísticos utilizados con otros métodos exactos o aproximados. Esta comparación permitiría evaluar la eficacia y eficiencia relativa de diferentes algoritmos en la formación de equipos. Sería útil para los investigadores incluir una evaluación comparativa más completa de diferentes enfoques en futuros estudios.

4. La mayoría de los trabajos académicos abogan por tratar el problema como un problema de un solo objetivo, aplicando ponderaciones a los objetivos y fusionándolos en un objetivo único.

Sin embargo, a pesar de las ventajas del método multiobjetivo (ver sección 3.3.1), persiste en la literatura una llamativa ausencia de este tipo de enfoques, y la mayoría de los trabajos optan por un conjunto dispar de factores o aspectos. Este enfoque se considera más adecuado en escenarios en los que los responsables de la toma de decisiones carecen de criterios predefinidos para asignar pesos o prioridades entre varios objetivos (López-Trujillo, Rosete-Suárez, André-Ampuero, & Infante-Abreu, 2020).

En resumen, los artículos revisados presentan valiosos enfoques y contribuciones en el campo de la formación de equipos colaborativos en el aula. Sin embargo, es importante reconocer las limitaciones comunes que se observan en estos estudios, como el uso de conjuntos de datos limitados, la falta de consideración de aptitudes técnicas, la falta de comparaciones exhaustivas y la mayoría le dan un enfoque monobjetivo con las desventajas citadas anteriormente. Además, la integración de rasgos personales y relaciones interpersonales podría ser un aspecto clave a considerar en futuras investigaciones. Abordar estas limitaciones podría conducir a un mayor desarrollo de algoritmos más completos y efectivos para la formación de equipos en entornos educativos.

El paradigma de la formación de equipos ha experimentado un cambio significativo hacia la adopción de estrategias multiobjetivo, que reconocen y abordan la naturaleza multifacética y a menudo contradictoria de los objetivos a optimizar. Esta transición hacia

un enfoque multiobjetivo se sustenta en la premisa de que las decisiones relacionadas con la formación de equipos no deben basarse únicamente en la optimización de un solo criterio, sino que deben considerar simultáneamente múltiples objetivos y restricciones. A continuación, se va a realizar un pequeño resumen de los principales trabajos publicados que han aplicado el denominado enfoque multiobjetivo.

4.2.3 Revisión del TFP Académico con enfoque multiobjetivo

Dentro del marco de optimización multiobjetivo para problemas TFP académicos, los tres trabajos publicados consisten en buscar estrategia de búsqueda versátil, potente y rentable, capaz de agrupar cualquier número dado de alumnos precalificados con múltiples características en cualquier número de grupos de aprendizaje inter homogéneos e intra heterogéneos óptimos.

La primera publicación que aplica un enfoque multiobjetivo para el problema de formación de equipos en el aula data del año 2018 (Garshasbi, Mohammadi, Graf, & Garshasbi, 2019). En este artículo se aborda el problema TFP con el objetivo de agrupar a los alumnos en 4 grupos de aprendizaje de 10 miembros cada uno, con un nivel predefinido de homogeneidad intergrupala y heterogeneidad intragrupal. Para precalificar a los alumnos se tuvieron en cuenta cinco características de diferentes tipos y variaciones. Se comparan los resultados entre un enfoque monoobjetivo mediante un algoritmo evolutivo simple y un algoritmo evolutivo multiobjetivo NSGA-II, haciendo uso de operadores de cruce y mutación convencionales, como lo son el *Order crossover* y el *Swap mutation*, respectivamente. La mayor aportación de este trabajo es que demuestra que para el problema que propone, el algoritmo NSGA-II supera con creces los valores preestablecidos para la intra-homogeneidad y la inter-heterogeneidad, mientras que el enfoque mono objetivo algunos de los grupos propuestos no pudieron cumplir los objetivos preestablecidos.

En (Miranda, 2020), se toman 3 objetivos a maximizar: la inter-heterogeneidad, proponiendo una mayor igualdad entre grupos distintos; la intra-heterogeneidad promoviendo la diversidad de estudiantes (estudiantes con características y aptitudes distintas) dentro de cada grupo y maximizar la empatía, de modo que los miembros de un grupo tengan afinidad entre sí. Las características que se consideraron fueron: género, habilidades de programación, habilidades de interpretación, niveles de conocimiento, aptitudes comunicativas y capacidad de liderazgo. Se hace uso del algoritmo evolutivo NSGA-II con operadores de cruce y mutación convencionales, el cual se compara con un método exhaustivo, método aleatorio y con un algoritmo de enjambre multiobjetivo con el propósito de demostrar la superioridad computacional del NSGA-II. Finalmente, El método propuesto muestra un rendimiento similar al enfoque exhaustivo en instancias de tamaños de clase inferiores (<25) pero con menos tiempo computacional. En cuanto a instancias con tamaños de clase superiores (>30), el algoritmo NSGA-II obtiene resultados significativamente mejores comparándolo con los demás métodos contemplados.

El último artículo referente al enfoque multiobjetivo para un TFP es (Casotti & Krohling, 2023). Este trabajo da un enfoque bi-objetivo, intentando maximizar la cohesión

entre los integrantes de los equipos a la vez que minimiza la discordancia entre los integrantes. Para resolver el problema, se proponen diferentes algoritmos multiobjetivo: NSGA-II, U-NSGA-III una versión modificada para problemas con múltiples objetivos (>3), SMS-EMOA (S-Metric Selection Evolutionary Multi-Objective Algorithm) y SPEA2 (Strength Pareto Evolutionary Algorithm 2). Se realizan varios experimentos con 7 instancias diferentes, con tamaños de 10 a 200 individuos, donde cada una de las soluciones se compara mediante la métrica de hipervolumen. En cuanto a los resultados, no hay un claro vencedor en términos generales. Esto afirma que no hay un algoritmo de optimización que se adapte mejor que las demás en cualquier situación, si no que la elección de la herramienta a utilizar depende de las características específicas de cada instancia (Wolpert & Macready, 1997).

Para finalizar con la revisión bibliográfica, se concede una tabla donde se resumen las características principales de los trabajos mencionados:

Tabla 1: Resumen de los artículos revisados

Referencia	Inter Homo	Características	Real vs simul.	Técnica	Mono vs Multi
(Ali Babar & Kitchenham, 2007)	Sí	Psicológicas	Real	GA	Monobj.
(Ma, y otros, 2022)	No ¹	Psicológicas y género	Real	Exacto, GA y Greedy.	Monobj.
(Andrejczuk, Bistaffa, Blum, Rodríguez-Aguilar, & Sierra, 2019)	Sí	Psicológicas, técnicas y género	Simulado	Exacto y Heurístico.	Monobj.
(44)	Sí	Psicológicas	Real	GA	Monobj.
(Shi-Jie(Gary) & Li, 2004)	No	Psicológicas	Real	Exacto	Monobj.
(Juan M. Alberola, 2016)	Intra-Heterogéneos	Psicológicas	Real	IA iterativa	Monobj.
(J. -M. Flores-Parra, 2018)	Intra-Heterogéneos	Psicológicas	Real	Análisis de Redes	Monobj.
(Yannibelli & Amandi, 2012)	Intra-Heterogéneos	Psicológicas	Simulado	Exacto y GA	Monobj.
(Garshasbi, Mohammadi, Graf, & Garshasbi, 2019)	Sí	Psicológicas, técnicas y género	Simulado	GA	Multiobj.
(Miranda, 2020)	Sí	Psicológicas, técnicas y género	Real	GA	Multiobj.
(Casotti & Krohling, 2023).	No	Psicológicas	Simulado	GA,SMS, SPEA	Multiobj

La segunda columna de la Tabla 1 hace referencia a si los artículos en cuestión consideran como objetivo obtener una partición de equipos inter-heterogenea e intra-homogenea. La tercera columna toma en consideración las características a tener en

¹ Considera la metodología E-CARGO basada en grupos por roles, por lo que se podría considerar la intra-heterogeneidad pero no la inter-homogeneidad.

cuenta para la formación de los grupos. En cuanto a la cuarta columna, se menciona si los datos de los alumnos con los que se realizaron los experimentos eran reales o simulados. Las dos últimas columnas hacen referencia a los métodos propuestos para afrontar el problema y el enfoque elegido, respectivamente.

En definitiva, se aprecia como los algoritmos evolutivos, específicamente los algoritmos genéticos como el NSGA-II, presentan una buena alternativa a la hora de resolver problemas tipo TFP multiobjetivo. Por una parte, por su sencillez, flexibilidad, versatilidad y gran potencial para tratar una gran variedad de problemas, los algoritmos genéticos son las técnicas de optimización más populares y se aplican ampliamente en diferentes campos de estudio. Además, El NSGA-II ha logrado un rendimiento mucho mejor en comparación con otros optimizadores multiobjetivo restringidos en problemas (Miranda, 2020), lo cual demuestra porque esta herramienta es tan citada en los artículos revisados.

Como se observa en los artículos (Garshasbi, Mohammadi, Graf, & Garshasbi, 2019) y (Miranda, 2020), dos funciones objetivo a maximizar son la inter-homogeneidad y la intra-heterogeneidad. El problema principal de estos es que, para conseguirlo, tratan cada característica con una escala numérica para poder adaptarla indiscriminadamente en el cálculo de las distancias euclídeas entre los alumnos utilizando sus características. En algunos casos, esto puede no resultar muy apropiado, por la escala nominal o binaria que presentan algunas características como podrían ser la personalidad o el género, respectivamente. Con una formulación y tratamiento individual de las características objetivo según su naturaleza y escala se podrían lograr mejores resultados.

4.3 Aportación del proyecto al estado del arte

Tomando en consideración los artículos revisados en el estado del arte, las aportaciones principales de este proyecto son:

- Consolidar en un único modelo multiobjetivo ((Andrejczuk, Bistaffa, Blum, Rodríguez-Aguilar, & Sierra, 2019), también lo hace, pero aporta un enfoque monobjetivo) los tres atributos fundamentales que delinear el óptimo desempeño de un equipo: el equilibrio entre habilidades técnicas, perfiles psicológicos y diversidad de género.
- Al contrario que en (Miranda, 2020), las funciones objetivo se adaptan a la naturaleza y escala de cada característica a considerar, impulsando la inter-homogeneidad e intra-heterogeneidad entre los equipos creados a partir de estas 3 características.
- Proponer operadores de cruce y mutación personalizados, funcionales y adaptables para otros problemas TFP.
- Demostrar la adaptabilidad del modelo propuesto a diferentes tamaños de clase y distintas distribuciones de características estudiantiles.
- Asimismo, se busca exhibir otros métodos menos utilizados en problemas TFP para la optimización de los hiperparámetros del modelo, confirmando así la efectividad de técnicas bayesianas de optimización para ciertos hiperparámetros relaciones con el algoritmo a utilizar.

5 Descripción del problema

En el presente apartado se va a exponer el modelo matemático desarrollado junto a varias definiciones para poder formular el problema TFP propuesto. Primero de todo, se comenzará con varias definiciones previas (Sección 5.1) para tras ello continuar con la formulación matemática del problema (Sección 5.2). Por último, la sección 5.3 se focalizará en la definición y formalización de las funciones objetivo.

Primero de todo, cabe destacar que, mediante el presente modelo, en cada una de las 3 características o dimensiones de los alumnos (Aptitudes técnicas, Personalidad y balance de género) que se van a tomar en cuenta a la hora de formar los equipos, se va a buscar un balance en cuanto a las aptitudes técnicas y la proporción de género. En cuanto a la distribución de la personalidad, en esta dimensión se intentará construir equipos lo más heterogéneos posibles. En términos generales, se puede decir que se buscan formar equipos intra-heterogeneos.

Por otra parte, La propuesta de formación de equipos en el aula se basa en un enfoque integral que considera las habilidades técnicas y teóricas de los estudiantes, así como sus perfiles psicológicos. Al evaluar estas dimensiones, buscamos crear equipos equilibrados y efectivos que se complementen entre sí. Un aspecto fundamental de este modelo es garantizar el balance de género en la conformación de equipos. Esto no solo promueve la diversidad de perspectivas, sino que también desafía los estereotipos de género y crea un ambiente inclusivo. En términos generales, se buscará una partición justa de los alumnos, para ello, se utilizará el termino equipos inter-homogeneos o partición homogénea en cuanto a las características de los equipos.

5.1 Definiciones

Definición 1

$S = \{s_1, s_2, \dots, s_n\}$: Conjunto de estudiantes en una clase de n individuos.

Definición 2

$s_j = \langle r_j, K_j, g_j \rangle$: cada estudiante es una tupla que contiene los siguientes 3 parámetros.

r_j : es un parámetro $\in [0,1]$ relacionado con las aptitudes técnicas del estudiante.

K_j : es una tupla de personalidad. Ver sección 5.3.2 *Medición de la divergencia de psicológica*.

g_j : género del estudiante $\in \{ hombre, mujer \}$.

Definición 3

$t_i \subset S$: subconjunto de estudiantes denominado equipo.
Siendo $|t_i|$ el tamaño del equipo.

Los equipos son disjuntos. Esto es, un alumno solamente puede pertenecer a un solo grupo.

A pesar de la posibilidad de que el modelo desarrollado pudiera manejar con precisión el proceso de formación de grupos inter-homogéneos e intra-heterogéneos considerando diferentes números de alumnos en grupos distintos, en este trabajo se puso en práctica la agrupación de miembros en grupos de aprendizaje del mismo tamaño. Además, los tamaños de los equipos se fijan previamente:

Definición 4

l : tamaño de equipo requerido para obtener una solución factible.

Según (Gaviola, Atanda, Howard, & Yuet, 2019) describe el tamaño ideal de los equipos como de cuatro a siete personas para promover una mayor participación de los estudiantes y mejorar la experiencia global, por lo que en nuestro caso el tamaño de los equipos se ha prefijado a 4 ($l = 4$).

Podemos presentar el conjunto de todos los equipos candidatos tomando en cuenta las definiciones previamente expuestas:

Definición 5

$T = \{t_1, t_2, \dots, t_i\}$: Conjunto de todos los equipos factibles. Siendo T un subconjunto de todas las combinaciones posibles de S.

Definición 6

$C_m \subset T$ Una solución C_m es una partición de un conjunto S de estudiantes en grupos t_i el cual puede ser una solución factible del problema. Se define un conjunto factible como $\forall i, j (i \neq j), t_i \cap t_j = \emptyset; \cup t_j = S, \forall t_j \in C_m$. El tamaño de C, $|C|$ esta restringido al número de alumnos que se requiere en cada equipo, en este caso 4.

Definición 7

$D = \{C_1, C_2, \dots, C_m\}$ Espacio de todas las soluciones factibles C_m del problema.

5.2 Modelo de optimización

Variables

δ_i : variable binaria $\begin{cases} 1 & \text{el equipo } i \text{ se va a utilizar} \\ 0 & \text{si no} \end{cases}$, $i = \{1, 2, \dots, 4!(n - 4)\}$

siendo n el tamaño de la clase.

Funciones objetivo

$f_1(C_m): D \rightarrow \mathbb{R}$: función de aptitud técnica correspondiente a la partición C_m .

$f_2(C_m): D \rightarrow \mathbb{R}$: función de índice de Personalidad correspondiente a la partición C_m .

$f_3(C_m): D \rightarrow \mathbb{R}$: función balance de género correspondiente a la partición C_m .

$$\text{Max } f_1(C_m), f_2(C_m), f_3(C_m) \quad (1)$$

Restricciones:

$$\sum_{t_i \in T, s_j \in t_i} \delta_i = 1; \quad j = 1, \dots, n \quad (2)$$

Mediante (2) se asegura que cada alumno solamente puede estar representado en uno y solo uno de los equipos de una solución factible C_m . Debido a esto, una solución factible C_m va a estar compuesto por una cantidad de equipos t_i igual a $|C|$ donde el numero de equipos de la partición se calcula mediante la división entre el numero de estudiantes de esa aula $|S|$ entre el tamaño de los grupos $|t_i|$: $|C| = \frac{|S|}{|t_i|}$. En este caso, se ha fijado $|t_i| = 4$; $|C| = \frac{|S|}{4}$.

5.3 Formalización de las funciones objetivo

La medición de las 3 funciones objetivo que se quieren maximizar se realiza a 2 escalas. Primero, para cada función, se explicará la metodología empleada para el cálculo de las aptitudes para un equipo t_i , y posteriormente se procederá a explicar el cálculo de cada aptitud para una partición C_m , teniendo en cuenta las aptitudes de los equipos que conforman esa partición.

5.3.1 Medición de la aptitud técnica

Según (Bekele, 2006), combinar estudiantes con distintos niveles de rendimiento académico impacta positivamente tanto en el aprendizaje de los estudiantes de bajo rendimiento como en los estudiantes con altos grados de rendimiento académico. Considerando lo anterior, se define un grupo balanceado en cuanto al nivel técnico como aquel donde sus integrantes tienen niveles muy diversos en cuanto a conocimiento técnico y rendimiento.

Con este enfoque, se presenta el vector R , el cual recopila los parámetros r_j asociados a el rendimiento académico de cada uno de los estudiantes del aula:

Definición 8

$R = \langle r_1, r_2, \dots, r_n \rangle$ Es un vector que contiene el rendimiento técnico de los n estudiantes del aula. También se define $R(s_j)_{s_j \in t_i}$ como el vector que contiene las aptitudes técnicas de los 4 estudiantes s_j del grupo t_i .

En este contexto, un grupo balanceado técnicamente debe incluir alumnos con puntuaciones técnicas bajas, medias y altas. Según (Bekele, 2006), esto se puede medir mediante un índice que denomina “Goodness of Heterogenity” o “Bondad de heterogeneidad”.

La métrica que evalúa la calidad de la heterogeneidad (GH) se basa en la idea de que, en un grupo con diversidad suficiente, una vez que se han identificado las puntuaciones más altas y bajas de los estudiantes, se espera que las puntuaciones de los demás estudiantes se encuentren en el punto intermedio entre estos extremos. En esta situación, la discrepancia absoluta entre la diferencia promedio (AD) y las puntuaciones de los demás estudiantes es mínima (Bekele, 2006).

Para calcular la métrica de GH para un equipo, se sigue este procedimiento. Tomamos AD como el promedio de las puntuaciones técnicas más altas y bajas de los estudiantes en el grupo i -ésimo.

$$AD(t_i) = \frac{\max(R(s_j))_{s_j \in t_i} - \min(R(s_j))_{s_j \in t_i}}{2} ; t_i \in C_m \tag{8}$$

y,

$$GH(t_i) = \frac{\max(R(s_j))_{s_j \in t_i} - \min(R(s_j))_{s_j \in t_i}}{1 + \sum_j |AD_i - r_{(j)}_{s_j \in t_i}|} ; t_i \in C_m \tag{9}$$

Donde $r_{(j)}_{s_j \in t_i}$, son las puntuaciones tecnicas de los dos estudiantes del grupo t_i , excluyendo a los alumnos con la puntuación más baja del grupo y el alumno con puntuación mas alta.

Solamente queda definir el nivel de aptitud técnica a nivel de partición o conjunto total de equipos formados, donde se busca la homogeneidad entre los grupos en términos de $GH(t_i)$. Con dicho objetivo, se hace uso del producto Bernoulli-Nash (Nash, 1950) para medir el valor sinérgico de una partición de equipos, lo que da valores más altos a las particiones justas en comparación con otras funciones como la suma:

$$AT(C_m) = \prod_{i=1}^i GH(t_i); t_i \in C_m,$$

5.3.2 Medición de la divergencia de psicológica

Tomando como referencia el artículo (Sánchez-Anguix, Alberola, Del Val, Palomares, & Teruel, 2023), se observa como en los experimentos realizados, los alumnos se sienten más satisfechos con las experiencia y dinámica del equipo cuando se emplea la teoría MBTI para formar los grupos ante la teoría de roles de Belbin. Sin embargo, la teoría MBTI ha podido resultar más eficaz puesto que la composición de los perfiles psicológicos dentro de la clase estaba más equilibrada según MBTI que la teoría de Belbin. Aun así, teniendo en cuenta que es más probable que se la composición dentro de MBTI sea más equilibrada, será este el índice a utilizar para medir el nivel de divergencia respecto a la personalidad de los equipos en este modelo.

El indicador Myers-Briggs o inventario tipológico de Myers-Briggs, –MBTI por sus siglas en inglés– es un cuestionario que evalúa cómo las personas perciben el entorno y toman decisiones (Briggs Myers, . McCaulley, Quenk, & Hammer, 1985).

La prueba utiliza cuatro escalas, llamadas dicotomías, definidas como pares opuestos entre ocho categorías:

- Introversión-Extraversión: Cómo enfocan su atención u obtienen su energía.
- Sensación-Intuición: Cómo perciben o toman la información.
- Pensamiento-Sentimiento: Cómo prefieren tomar decisiones.
- Juicio-Percepción: Cómo se orientan hacia el mundo exterior.

Cada categoría es simbolizada por una letra, de modo que el resultado se muestra con una combinación de cuatro letras, de entre las dieciséis posibles, que pretende definir la personalidad del sujeto.

El propósito de aplicar este índice es lograr heterogeneidad en el equipo en relación con las distintas mediciones empleadas en el test Myer-Briggs. El test MBTI asigna un tipo distinto para cada una de las cuatro dimensiones. Dado que cada dimensión tiene exactamente dos rasgos opuestos, podemos afirmar que la dimensión k tiene dos posibles rasgos asignados, conocidos como κ_1 y κ_2 , respectivamente. Se representa a continuación el vector de personalidad asociado a cada estudiante j :

Definición 9

$K_j = \langle k_1, k_2, k_3, k_4 \rangle$ Es un vector asignado a un alumno que contiene el rasgo asignado en cada una de las 4 dimensiones de personalidad definidas por el test MBTI.

Establecemos una función binaria $\gamma_k(s_j, k_l)$ que devuelve 1 cuando al estudiante s_j se le ha asignado el rasgo k_l a través del test MBTI en la dimensión k . Posteriormente, definimos la puntuación obtenida por un equipo t_i en una dimensión k de la siguiente manera:

Definición 10

Índice de divergencia psicológica por perfil k
MTBI: P_k

$$P_k(t_i) = \begin{cases} 0 & \text{si } \exists k_l, \sum_{s_j \in t_i} \gamma_k(s_j, k_l) = |t_i| \\ 1 & \text{si } \exists k_l, \sum_{s_j \in t_i} \gamma_k(s_j, k_l) = 1 \\ 2 & \text{si } \exists k_l, \text{ en otro caso} \end{cases}$$

Tras obtener el índice para cada perfil dentro de los 4 posibles perfiles de MBTI, aunamos todos estos en el índice de divergencia psicológica o índice de Personalidad, definido a continuación:

Definición 11

Índice de Personalidad: $P(t_i)$

$$P(t_i) = \frac{1}{8} \times \sum_{k=1}^4 P_k(t_i)$$

La función devuelve un valor de 1 cuando existe una diversidad de rasgos en cada dimensión del MBTI, y devuelve un valor de 0 cuando todos los miembros del equipo tienen la misma personalidad según el MBTI, llevando a cabo el propósito de intentar conseguir equipos con una composición heterogénea a lo que respecta el MBTI.

Siguiendo el ejemplo de la aptitud técnica a nivel de aula, para el nivel de divergencia psicológica se utiliza el mismo método, multiplicando el $P(t_i)$ de todos los equipos para obtener la medida global:

$$AT(C_m) = \prod_{i=1}^i P(t_i)$$

5.3.3 Medición del balance de género

Por último, tomaremos de referencia el artículo (Bekele, 2006), donde se calcula el nivel de balance de género de un equipo de la siguiente manera:

Definición 12

$BE(t_i) = \sin(\pi h(t_i))$: Es la función que calcula el balance absoluto de género dentro de un equipo propuesto por (Shi-Jie(Gary) & Li, 2004).

mide la preferencia por el balance de género. La función $\mu(t_i) = \frac{m(t_i)}{m(t_i)+h(t_i)}$ da como resultado la proporción de mujeres en un equipo considerando el número de mujeres ($m(t_i)$) y hombres ($h(t_i)$). Un equipo t_i está perfectamente equilibrado en términos de género si $m(t_i) = h(t_i)$, y, por lo tanto, $\mu(t_i) = 1/2$ y $\sin(\mu(t_i)) = 1$. Se Observa que cuando el número de mujeres y hombres es igual, se sigue que $\mu(t_i) = 1/2$ y $\sin(\mu(t_i)) = 1$. En este caso, decimos que un equipo está perfectamente equilibrado.

No obstante, muchas veces se da el caso de que la composición de las clases en cuanto a género en muchas carreras universitarias no está equilibrada, tomando función $\mu(t_i)$ en estos casos valores cerca del 0 o cerca del 1. Es por esto por lo que, se propone modificar la propuesta de (S. Liu, 2013) para lograr que la proporción de cada género en un equipo se asemeje lo máximo posible a la proporción de género en el aula:

Definición 13

$$BE(t_i) = 1 - \frac{|\mu_{aula} - \mu(t_i)|}{\max(\mu_{aula}, 1 - \mu_{aula})}$$

Es la función que calcula el balance de género según la proporción global del aula dentro de un equipo propuesto.

En esta función, estamos calculando la diferencia absoluta entre la proporción de género en el equipo $\mu(t_i)$ y la proporción de género en el aula μ_{aula} . Luego, dividimos esta diferencia por el máximo entre la proporción en el aula global y su complemento $(1-\mu_{aula})$. Esto escalará la diferencia de manera que el valor máximo de diferencia posible entre la proporción del aula y del grupo devuelva un valor de 0 en la función. Si las proporciones son idénticas, la función devolverá un valor de 1.

A nivel global, aplicaremos el mismo método que en las otras dos dimensiones:

Definición 14

$$BE(C_m) = \prod_{i=1}^i BE(t_i)$$

Es la función que calcula el balance de género para una partición factible de equipos C_m

6 Diseño e implementación del algoritmo genético

En esta sección se presenta como se ha adaptado el problema para poder desarrollar y obtener soluciones mediante el algoritmo Genético NSGA-II. En primer lugar, la sección 6.1 comprende la adaptación de la formulación matemática del modelo (variables, funciones objetivo, restricciones) al ámbito de la programación. Esto es, como se adapta el modelo matemático y el algoritmo genético para obtener soluciones viables para el problema TFP propuesto. En los apartados 6.2, 6.3 y 6.4 se explica el funcionamiento de los operadores de selección, cruce y mutación propuestos para el algoritmo NSGA-II. Por último, en el apartado 6.5 se referencian las principales librerías de Python² utilizadas para el desarrollo del proyecto.

6.1 Codificación de las variables y soluciones

Todos los Algoritmos Genéticos se inicializan codificando las variables de entrada en una estructura similar a un cromosoma que se asemeja a una solución potencial para el problema en estudio. Los cromosomas, como cadenas bien organizadas, están compuestos por genes conectados de manera inteligente y transfiriendo la información genética. Cada cromosoma, como un genotipo, representa virtualmente un objeto físico como su correspondiente fenotipo (Garshasbi, Mohammadi, Graf, & Garshasbi, 2019).

Dado que agrupar un número dado de aprendices en un número predefinido de grupos óptimos de aprendizaje inter-homogéneos e intra-heterogéneos es el foco principal de este estudio, se diseña una estructura similar a un cromosoma apropiada en la que cada aprendiz ocupa una posición única (es decir, un gen). Mediante la siguiente figura se representa un ejemplo de una clase de 40 alumnos para una solución (cromosoma):

s40	s12	s01	s25	s07	s29	s08	s15	...	s23	s39	s37	s19
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Figura 12: Representación de un cromosoma [Elaboración propia]

A este tipo de codificación *Order-Based Encoding* (Sen & Dutta, 2000), puesto que se basa en una permutación de una lista de números enteros. Cada gen representa el índice de un alumno, los cuales se dividen en equipos de cuatro (cada uno representado con un color diferente).

Teniendo en cuenta la estructura de almacenamiento de datos definida, el objetivo principal en la formación óptima de grupos es manipular el orden/posición de los estudiantes en los cromosomas. A cada aprendiz se le permite ocupar solo una posición dentro del cromosoma en cualquier iteración y puede cambiar la posición basándose en las decisiones dictadas por los operadores genéticos. (Garshasbi, Mohammadi, Graf, & Garshasbi, 2019).

² Python, <https://www.python.org/doc/>, accedido el 17 de junio de 2024

Es importante recalcar que, teniendo en cuenta la definición de las funciones objetivo que se van a explicar en la siguiente sección, el orden/posición de los alumnos en un equipo previamente formado es irrelevante. En la siguiente figura podemos observar que el “Equipo A” y el “equipo B” van a resultar en los mismos fitness en las 3 dimensiones definidas.

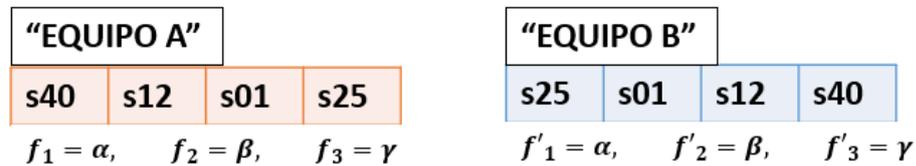


Figura 13: Ejemplo orden/posición intragrupo [Elaboración propia]

Consecuentemente, a nivel solución (C_m), el orden/posición de los distintos equipos también es irrelevante, exponiendo un ejemplo similar al de la anterior figura mediante la Figura 13:

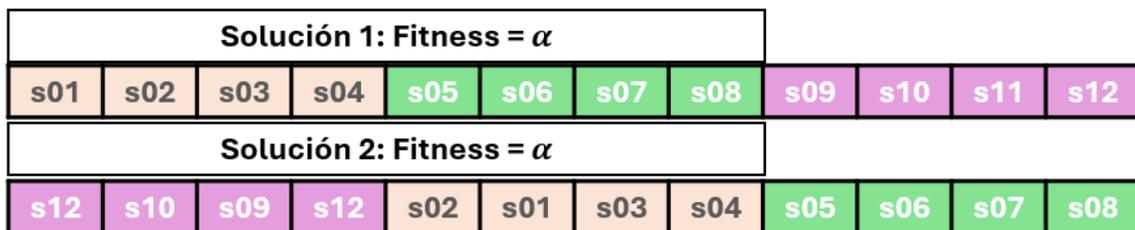


Figura 14: Ejemplo de dos particiones con el mismo fitness [Elaboración propia]

Como se aprecia, la posición de los alumnos a nivel equipo es irrelevante como lo es la posición de los equipos a nivel solución, puesto que la solución 1 obtiene el mismo Fitness que la solución 2.

Este tipo de codificación tiene ciertos inconvenientes como pueden ser las redundancias en las distintas soluciones de la población como los ejemplos que se han expuesto. Sin embargo, se ha optado por dicha codificación puesto que es cómoda para trabajar con ella y permite una fácil y diversa aplicación de los operadores de cruce y mutación debido a que Este tipo de codificación permite el uso de operadores sencillos de permutación, obteniendo siempre soluciones que son factibles (en el caso general) (Sen & Dutta, 2000).

6.2 Operador de selección

El operador de selección el cual será responsable de elegir que individuos de la población pasaran a formar nuevos individuos mediante el cruce es el Ranking mediante Torneo binario, originalmente propuesto en (Deb, Pratap, Agarwal, & Meyarivan, 2002).

Como los individuos del grupo de cruce son aquellos cuyos genes heredará la siguiente generación, es deseable que el grupo de cruce esté compuesto por individuos "buenos", de ahí viene el término “elitista”. Un mecanismo de selección en los GA es simplemente un proceso que favorece la selección de los mejores individuos de la población para el pool de apareamiento. La presión de selección es el grado en que se

favorece a los mejores individuos: cuanto mayor es la presión de selección, más se favorece a los mejores individuos (Miller & Goldberg, 1995).

Esta presión de selección impulsa al GA a mejorar la aptitud de la población en las generaciones sucesivas. La tasa de convergencia de un AG viene determinada en gran medida por la presión de selección, ya que, a mayor presión de selección, mayor tasa de convergencia. Sin embargo, si la presión de selección es demasiado baja, la tasa de convergencia será lenta y el AG tardará innecesariamente más tiempo en encontrar la solución óptima. Si la presión de selección es demasiado alta, aumenta la probabilidad de que el AG converja prematuramente a una solución incorrecta (subóptima) (Miller & Goldberg, 1995).

La selección por torneo binario proporciona presión de selección mediante la comparación a pares de la población. Este operador se basa en lo siguiente:

1. Mientras que el tamaño de la población no sea $2N$:
2. Se escogen aleatoriamente dos individuos de la población actual.
3. El mejor individuo según el operador *Crowded-comparison* pasa a formar parte del grupo de cruce

Dicho grupo de cruce, formado por los ganadores del torneo, tiene una aptitud media superior a la aptitud media de la población. Esta diferencia de aptitud proporciona la presión de selección, que impulsa al GA a mejorar la aptitud de cada generación sucesiva (Deb, Pratap, Agarwal, & Meyarivan, 2002).

6.3 Operador de cruce personalizado

Como bien se ha mencionado anteriormente, el operador de cruce se aplica para conseguir resultados prometedores en la población de soluciones combinando y alterando el código genético de las soluciones poblacionales.

Con el objetivo de obtener resultados superiores a los de un algoritmo NSGA-II estándar, se propone un operador de cruce personalizado. Este operador pretende adaptarse a la estructura y aprovechar la naturaleza de la codificación elegida para representar los individuos (cromosoma).

Teniendo en cuenta que las soluciones son una lista permutada de los índices de los alumnos de clase y que, la posición relativa que toma un gen en la solución no es relevante ni tampoco lo son los genes adyacentes a este, realmente lo importante es que alumnos comparten el mismo equipo, sin tomar en cuenta su posición relativa dentro de este. Por esta misma razón, el cruce debería de considerar la preservación de los equipos con una mayor aptitud, para así conseguir hijos potencialmente mejores.

Aun así, resulta interesante introducir cierta aleatoriedad en este proceso, para que el operador no solamente elija los mejores equipos de ambos padres, lo que podría resultar en soluciones ciertamente sesgadas. Para ello, se propone elegir los equipos a heredar mediante la introducción de un operador **de Selección por ruleta proporcional al fitness**. Asignando así una probabilidad de elección proporcional al fitness de los equipos candidatos.

Antes de presentar el pseudocódigo para el operador de cruce personalizado, se anticipan ciertas definiciones necesarias para el entendimiento de este:

- H : El cruce hace uso de dos padres para generar un hijo (H).
- C_m y $C_{m'}$: Individuos seleccionados para el cruce.
- $Q = \forall t_i \cup \forall t_{i'}; t_i \in C_m, t_{i'} \in C_{m'}$: es el conjunto de los equipos pertenecientes a C_m y $C_{m'}$.
- $\vec{F}_i = \langle f_{1i}, f_{2i}, f_{3i} \rangle$ y $\vec{F}_{i'} = \langle f_{1i'}, f_{2i'}, f_{3i'} \rangle$: Fitness multiobjetivo \vec{F}_i y $\vec{F}_{i'}$ respectivos a dos equipos cualquiera del conjunto Q .

Tomando en consideración todos los aspectos mencionados anteriormente, se presenta el pseudocódigo para el operador de cruce personalizado que se ha desarrollado:

- 1 $f_i = \frac{1}{3}f_{1m} \times \frac{1}{3}f_{2m} \times \frac{1}{3}f_{3m}, t_i \in C_m$
 $f_{i'} = \frac{1}{3}f_{1m'} \times \frac{1}{3}f_{2m'} \times \frac{1}{3}f_{3m'}, t_{i'} \in C_{m'}$

Se convierten los fitness t_i y $t_{i'}$ de en mono-objetivo mediante la suma ponderada equitativa de las 3 dimensiones ($f_m, f_{m'}$).
- 2 Para cada equipo en Q :

$$p_j = \frac{f_j}{\sum_{f \in Q} f}$$

Se calcula la probabilidad de un equipo j perteneciente a Q de ser seleccionado para pertenecer al hijo siendo dicha probabilidad proporcional a su fitness.
- 3 Se transforman dichas probabilidades. para que estas sean acumulativas p'_j , en el intervalo $[0,1]$.
- 4 Mientras H no tenga la misma longitud que los padres C_m y $C_{m'}$:
- 5 Se calcula un numero aleatorio (α) en el intervalo $[0,1]$.
- 6 Se selecciona el equipo asociado al intervalo donde caiga el numero α .
- 7 El equipo seleccionado no puede volver a ser elegido.
- 8 Todos los alumnos repetidos en H son reemplazados aleatoriamente por un alumno que no se encuentre en dicha solución.

Figura 15: Pseudocódigo del operador de cruce personalizado

En la siguiente figura se muestra un ejemplo de el funcionamiento del operador de cruce:

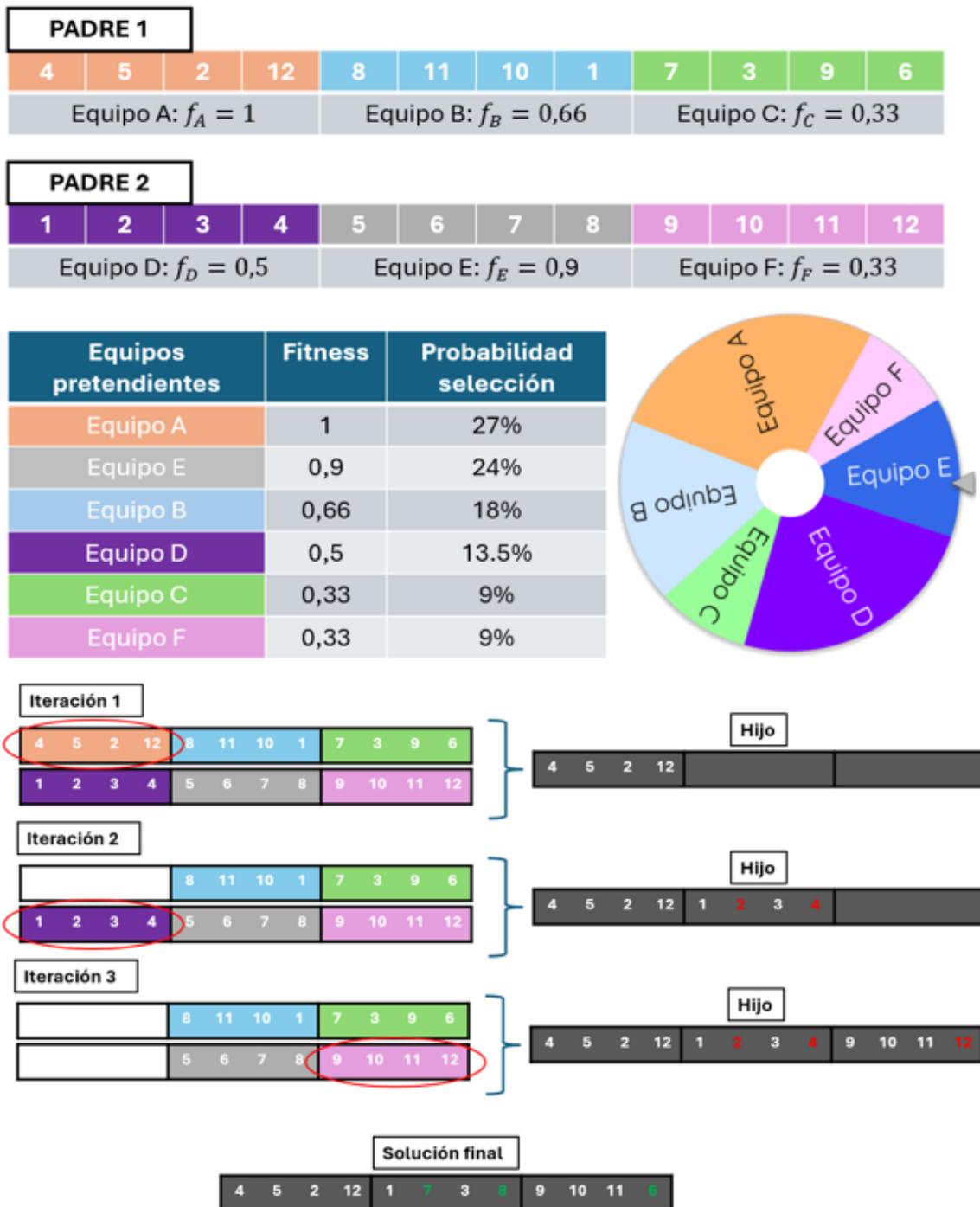


Figura 16: Ejemplo del proceso de cruce personalizado

En Figura 16, se identifican dos individuos denominados “Padre 1” y “Padre 2”, los cuales representan dos formas distintas de repartir en equipos de 4 integrantes una clase de 12 alumnos. Dichos padres servirán como ejemplo explicativo del operador de cruce personalizado.

En el Padre 1, los equipos se identifican con las letras “A”, “B” y “C”, mientras que, en el Padre 2, los equipos se identifican con las letras “D”, “E” y “F”. Cada equipo tiene su fitness asociado, el cual se denomina f_x , siendo x la letra identificativa de cada equipo.

Según dicho fitness, a cada equipo se le adjudica una probabilidad de pertenencia a la solución, la cual es proporcional a su fitness entre el total de todos los fitness.

En la iteración 1 observamos que el equipo “A” es el primer equipo elegido para formar parte de la solución. Esto no es muy sorprendente puesto que era el equipo con mayor probabilidad de aparición.

En la iteración 2 y 3 se escogen los equipos “D” y “F”. Como se aprecia, en estos dos nuevos equipos introducidos en los hijos hay alumnos que aparecen una segunda vez, los cuales se muestran en rojo (2,4 y 12). Para que la solución sea válida estos alumnos repetidos se cambian aleatoriamente por el índice de otros alumnos que no estén presentes en la permutación, como son (7,8 y 6) respectivamente.

Mediante este simple ejemplo de una clase de 12 alumnos, se muestra el funcionamiento del operador de cruce personalizado para obtener un hijo mediante el cruce de 2 alumnos, obteniendo una nueva solución factible.

6.4 Operador de mutación personalizado

El operador de mutación aumenta la diversidad de las soluciones y ayuda al algoritmo a escapar de los mínimos locales al cambiar uno o más genes de un cromosoma por valores aleatorios. La mutación se realiza mediante un hiperparámetro α , llamada probabilidad de mutación. Para cada solución de la población se genera un valor aleatorio $r \in [0, 1]$: $r < \alpha$, entonces se realiza la mutación (Esgario, Egias da Silva, & Krohling, 2019).

Al igual que el cruce, con intención de mejorar la capacidad de convergencia del algoritmo NSGA-II, se propone un operador de mutación que aproveche la estructura por equipos. El operador de mutación consiste en los siguiente:

- 1 Se elige un número entero β aleatorio, tal que $1 < \beta < N_{equipos}$.
- 2 Se escogen β equipos al azar los cuales se eliminan de la solución.
- 3 Se genera una permutación aleatoria haciendo uso de los alumnos pertenecientes a los equipos seleccionados.
- 4 Se introduce esta permutación aleatoria a la solución.

Figura 17: Pseudocódigo del operador de mutación personalizado

Se ilustra a continuación un ejemplo para facilitar la comprensión del proceso de mutación:

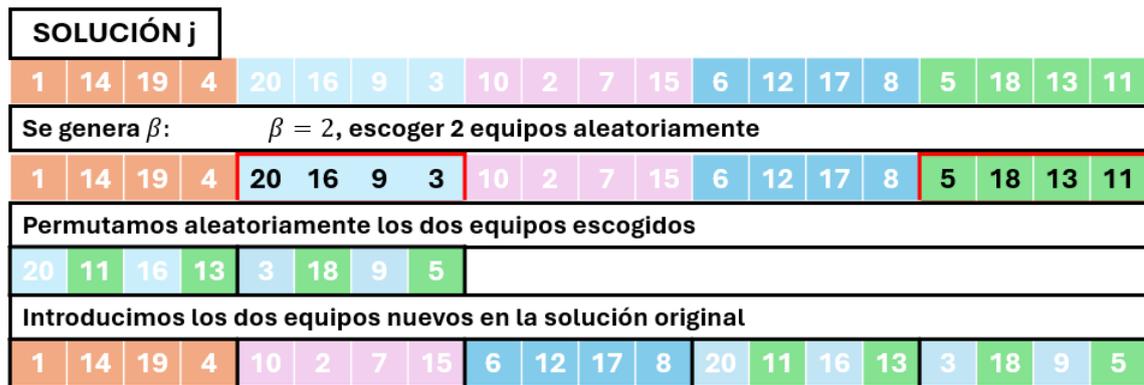


Figura 18: Ejemplo del proceso de mutación personalizado

En la Figura 18, se aprecia que el número aleatorio generado es $\beta = 2$, mediante el cual se procede a elegir dos equipos aleatoriamente. Estos equipos son los conformados por los alumnos [20, 16, 9, 3] por una parte y [5, 18, 13, 11] por otra. Dichos alumnos se guardan en un vector el cual se permuta aleatoriamente para obtener dos equipos diferentes a los originales: [20, 11, 16, 13] y [3, 18, 9, 5]. Estos nuevos equipos reemplazan a los equipos originales para formar un nuevo individuo.

6.5 Software utilizado

El desarrollo del algoritmo y la posterior experimentación se han realizado mediante el lenguaje de programación Python. A grandes rasgos, las dos librerías que se han empleado para llevar a cabo el trabajo han sido la librería *NumPy*³ y *Pymoo*⁴.

NumPy es uno de los paquetes fundamentales para la computación científica en Python. Es una biblioteca de Python que proporciona un objeto de matriz multidimensional, varios objetos derivados (como matrices con máscaras), y una variedad de rutinas para operaciones rápidas en matrices, incluyendo operaciones matemáticas, lógicas, manipulación de forma, ordenamiento, selección, entrada/salida, álgebra lineal básica, operaciones estadísticas básicas, simulación aleatoria y mucho más (Harris, y otros, 2020).

En cuanto a *Pymoo* (abrevación de Python Multiobjective Optimization), es una de las librerías más novedosas en cuanto a programación multiobjetivo. Este marco de trabajo ofrece una inmensa cantidad de algoritmos de optimización tanto para objetivos únicos como multiobjetivo, y muchas más características relacionadas con la optimización multiobjetivo, como visualización y toma de decisiones (Blank & Deb, 2020).

Cabe mencionar, que para la optimización de hiperparámetros, se ha optado por la librería *Optuna*⁵. *Optuna* es un marco de software de optimización automática de hiperparámetros, especialmente diseñado para el aprendizaje automático. El código

³ *NumPy*, <https://numpy.org/>, accedido el 17 de junio de 2024.

⁴ *Pymoo*, <https://pymoo.org/>, accedido el 17 de junio de 2024.

⁵ *Optuna*, <https://optuna.org/>, accedido el 17 de junio de 2024.

escrito con *Optuna* goza de un gran modularidad, y cualquier usuario puede construir dinámicamente los espacios de búsqueda para sus hiperparámetros (Akiba, Sano, Yanase, Ohta, & Koyama, 2019).

7 Experimentación

Este apartado comprende el diseño y resultados de las pruebas realizadas sobre las soluciones implementadas. Cabe destacar que la experimentación se ha llevado a cabo a pequeña escala, ya que el proyecto se ha desarrollado de forma totalmente remota, utilizando un ordenador personal. Por esta razón, no ha sido posible disponer de un equipo de alto rendimiento. En 7.1 se describe el conjunto de datos del que se parte, las instancias de problemas generadas a partir de este y algunas condiciones generales de todos los experimentos. En 7.2 se muestra el proceso y los resultados obtenidos para la optimización general de los hiperparámetros de cruce y mutación. Por último, en el subapartado 7.3 se establecen los objetivos a demostrar en la experimentación, se describen los métodos que se van a utilizar y su razón, como se han diseñado los experimentos y su posterior análisis.

7.1 Creación de instancias

En los proyectos de investigación en optimización es habitual la creación de instancias de problemas para probar cómo los *solvers* o algoritmos de optimización diseñados proceden a su resolución y poder medir el rendimiento de estos. El objetivo en esta sección será describir la manera en que se han creado instancias de problemas que permitan posteriormente evaluar las soluciones aportadas en este trabajo.

Partiendo de la base de que el modelo contiene 3 funciones objetivo relacionados con la personalidad según la prueba Myers-Briggs, la puntuación técnica o calificaciones académicas de los alumnos y la proporción de género del aula, se requiere de un conjunto de datos que represente dichas características de los alumnos.

Por un lado, para los datos relacionados con los rasgos de personalidad según la prueba MBTI, se parte conjunto de datos anonimizado formado por un total de 267 alumnos del grado de Turismo en la Universidad Politécnica de Valencia, datos que han sido recogidos desde los cursos académicos comprendidos entre 2014-2015 hasta 2018-2019 y que contienen los resultados de la aplicación de las MBTI. Originalmente estos datos se utilizaron para los experimentos del artículo de Sánchez-Anguix, Víctor, et al. (Sánchez-Anguix, Alberola, Del Val, Palomares, & Teruel, 2023).

Por otro lado, los datos relacionados con las puntuaciones técnicas de los alumnos y su género han sido simulados con el objetivo de crear 6 tipos de aula diferentes con las que probar los métodos propuestos. Se han intentado simular los datos reales de una clase universitaria convencional. Concretamente, se han generado 2 distribuciones diferentes para la puntuación técnica de la clase y otras dos para la distribución de género, con un tamaño de población $n = 267$. En la siguiente tabla se describen los datos simulados:

Tabla 2: Datos simulados para las puntuaciones técnicas y el género de los alumnos

Datos simulados	Descripción	Estadísticos
R1	Distribución bimodal: 70% datos generados mediante la primera distribución normal y 30% con la segunda distribución normal	$\mu_1 = 5.5, \sigma_1 = 0.5$ $\mu_2 = 8, \sigma_2 = 1$
R2	Distribución normal	$\mu = 7, \sigma = 2$
G1	muestras binarias	% género 1: 0.7
G2	muestras binarias	% género 1: 0.9

Los datos relacionados con las puntuaciones técnicas R comprenden entre los valores 0 y 10. Por esclarecer las dudas que puedan surgir con la primera distribución $R1$, Los datos de $R1$ se han generado mediante la unión de dos distribuciones normales, donde el 70% de los datos proviene de una distribución de media $\mu_1 = 5.5$ y desviación típica $\sigma_1 = 0.5$. La segunda distribución, equivalente al 30% de los individuos, proviene de una distribución normal de media $\mu_2 = 8$ y desviación típica $\sigma_2 = 1$. A continuación, se muestra el histograma del conjunto de datos $R1$ (267 alumnos):

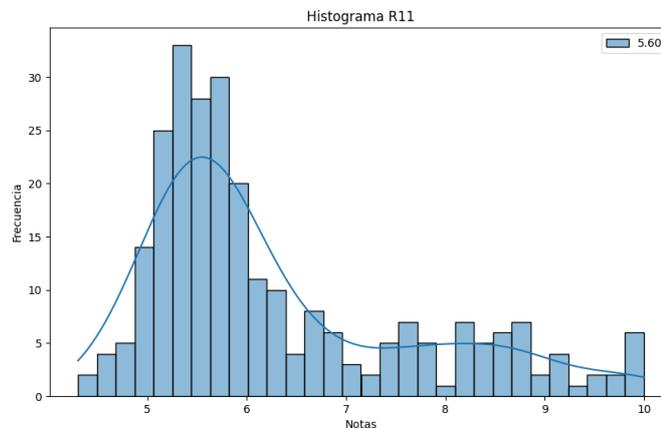


Figura 19: Histograma ilustrativo del conjunto de datos $R1$ [267 alumnos]

Ya expuesto el conjunto de datos con el que se va a trabajar, se pasa a la puesta a punto del modelo propuesto mediante la optimización de dos de sus hiperparámetros.

Una vez definidos los conjuntos de datos a utilizar para cada función objetivo, se exponen en la siguiente tabla las combinaciones realizadas entre sí para generar los 6 tipos de clase:

Tabla 3: Tipos de clase generadas para la experimentación

Tipo de aula	Nº alumnos	R	K	G
1	40	R1	K	G1
2	40	R1	K	G2
3	52	R1	K	G1
4	52	R2	K	G2
5	60	R2	K	G1
6	60	R2	K	G2

Las instancias generadas a partir de los tipos de aulas expuestas en la Tabla 3 se van a especificar en los apartados 7.2 y 7.3.

7.2 Puesta a punto

En primer lugar, se detalla el procedimiento para la obtención de los hiperparámetros principales del algoritmo genético para su mejor rendimiento.

7.2.1 Definición de hiperparámetro

Los hiperparámetros de un modelo de optimización se pueden definir como las variables de la configuración de un algoritmo que incurren directamente en el rendimiento y resultados esperados.

Dar con la combinación de hiperparámetros óptima para un modelo de optimización resulta ser un reto complicado. Además, la eficacia de un modelo de optimización bajo un conjunto de hiperparámetros puede variar drásticamente al alterar la configuración de los hiperparámetros.

Para resolver estos problemas, recurrimos al ajuste u optimización de hiperparámetros. El proceso general es el siguiente (James & Y., 2012):

- Seleccionar los hiperparámetros a sintonizar, en el problema presente, realmente solo interesa optimizar las probabilidades de mutación y cruce, ya que, como es obvio un mayor tamaño de población o generaciones siempre concluirá en mejores resultados.

- Especificar un intervalo de valores aceptables para los hiperparámetros especificados. Por ejemplo, para una tasa de mutación, al ser una probabilidad se especifican los valores que puede tomar dentro del intervalo $\mathbb{R} \in [0,1]$.
- Se entrenan varios modelos correspondientes a cada una de las diferentes configuraciones de hiperparámetros resultantes de los dos pasos anteriores.
- Seleccionar el modelo que mejor funcione del conjunto de modelos entrenados.

Concretamente, en los algoritmos evolutivos, dos de los hiperparámetros más influyentes en términos de rendimiento son la probabilidad de cruce y la probabilidad de mutación. Las buenas prácticas para definir estas tasas se han discutido en la literatura (De Jong, 1975) (Schaffer, Caruana, Eshelman, & Das, 1989), y los parámetros para las tasas de cruce y mutación se han explorado más a fondo (Xarn Retired, Macnish, Vijayan, Turlach, & Gupta, 2004). Los valores sugeridos, por lo general, se sitúan entre 0,6 y 0,8. Para la mutación, los valores típicos no son superiores a 0,1.

Los métodos más comunes en la optimización de hiperparametros son los denominados *Grid Search* o Búsqueda de Rejilla y *Random Search* o Búsqueda Aleatoria (Daves, 2023) Por una parte, En la Búsqueda de Rejilla primero definimos una cuadrícula o rejilla que contiene la lista de hiperparámetros junto con una lista de valores aceptables que queremos que pruebe el proceso de búsqueda. Un ejemplo de este método, en el caso de tener 3 hiperparametros donde cada uno puede tomar 4 valores, seria ejecutar un total de $4 \times 4 \times 4 = 64$ modelos diferentes, probando con todas las combinaciones posibles. Se optaría así por el modelo que mejores resultados diese, estableciendo esa configuración como la predeterminada.

En *Random Search*, en lugar de definir una cuadrícula de valores para los hiperparámetros, especificamos una distribución de la que se podrían muestrear los valores aceptables. La principal diferencia entre la *Random Search* y la *Grid Search* es que, en lugar de probar todas las combinaciones posibles de hiperparámetros, cada combinación de hiperparámetros se muestrea aleatoriamente y los valores de los hiperparámetros proceden de la distribución que especificamos al principio del proceso de búsqueda aleatoria (Daves, 2023).

Como se puede llegar a deducir, estos métodos tradicionales tienen claro inconvenientes (James & Y., 2012).

- La Búsqueda Aleatoria muestrea los hiperparámetros uniformemente al azar, lo que puede llevar a una exploración ineficiente, especialmente en espacios de alta dimensión. La búsqueda de Rejilla, por su parte, muestrea los hiperparámetros en una rejilla, lo que puede hacer que se pierdan regiones importantes del espacio, especialmente si el tamaño de la rejilla es bajo.
- Requieren de un gran número de evaluaciones para encontrar buenas configuraciones de hiperparámetros, especialmente en espacios de alta dimensión. No aprovechan la información de evaluaciones anteriores para guiar la búsqueda, lo que se traduce en una escasa eficiencia de la muestra.

- La búsqueda de Rejilla se vuelve costosa y poco práctica en espacios de altas dimensiones debido al aumento exponencial del número de valores. La búsqueda aleatoria también puede sufrir de ineficiencia computacional en tales espacios porque requiere un gran número de muestras aleatorias para explorar adecuadamente el espacio.
- No hay garantía de que estos métodos encuentren el óptimo global o incluso un buen óptimo local en un número razonable de iteraciones.

Para mejorar e intentar resolver las principales desventajas explicadas, se propone un método más sofisticado y que aplica enfoque inteligente y eficaz para guiar su búsqueda, basado en evaluaciones anteriores, llamado Búsqueda Bayesiana.

7.2.2 Optimización bayesiana de hiperparametros

La optimización bayesiana de hiperparámetros es una técnica para encontrar la mejor configuración para las variables de un modelo. A diferencia de los métodos tradicionales como los mencionados anteriormente, que prueban todas las combinaciones posibles a ciegas, la optimización bayesiana utiliza un enfoque inteligente y eficiente para guiar su búsqueda, informado por evaluaciones anteriores (Daves, 2023).

En este enfoque, se comienza por establecer una creencia inicial a priori que engloba el conjunto de posibles funciones objetivo. Posteriormente, se refina el modelo secuencialmente a medida que encontramos nuevos datos, empleando técnicas bayesianas de actualización posterior. Este modelo bayesiano posterior es un reflejo dinámico de la evolución de los resultados anteriores, conformadas por la información obtenida a partir de los datos observados, en relación con la función objetivo probabilística que se tratan de optimizar.

Matemáticamente, estamos considerando un problema de maximización (o minimización) global de una función objetivo desconocida (Shahriari, Swersky, Wang, Adams, & de Freitas, 2016);

$$x^* = \arg \max f(x); x \in X$$

donde X en este contexto, es un subconjunto compacto de \mathbb{R}^d , esto es, los hiperparametros. Además, se supone que la función de "caja negra" f no tiene una forma cerrada sencilla, sino que puede evaluarse en cualquier punto de búsqueda arbitrario x , esta evaluación produce resultados $y \in \mathbb{R}$ con ruido (estocásticos) los cuales $E \left[\frac{y}{f(x)} \right] = f(x)$. En este contexto, consideramos un algoritmo de búsqueda secuencial que, en la iteración n , selecciona una posición x_{n+1} en el que evaluar f y observar y_{n+1} . Tras N consultas, el algoritmo hace una recomendación final x_N , que representa la mejor estimación del algoritmo del optimizador.

Considérese lo siguiente como una representación de una estimación inicial:

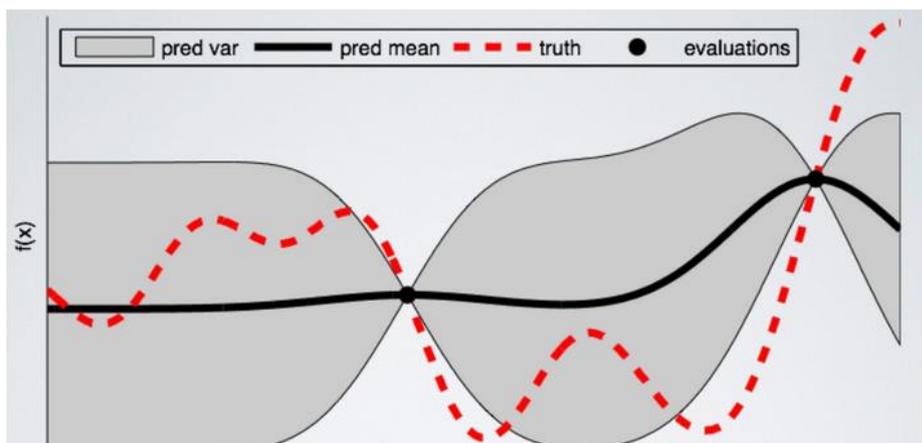


Figura 20: Estimación inicial de una función $f(x)$ (Daves, 2023).

En la Figura 20, La línea negra es la estimación inicial realizada por el modelo aproximado y la línea roja representa los valores reales de la función objetivo. A medida que la búsqueda bayesiana avanza, el modelo consigue imitar los valores reales de la función objetivo de forma mucho más densa:

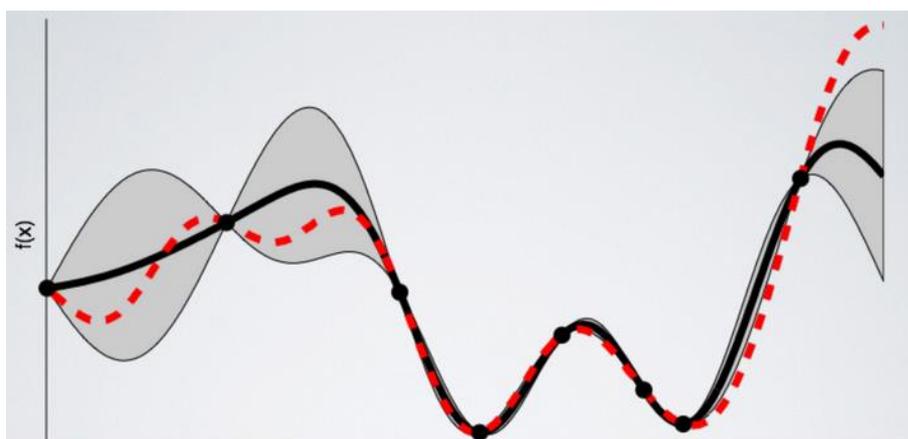


Figura 21: Modelo final para la optimización de los hiperparámetros (Daves, 2023).

Como se puede ver en la Figura 21, mediante esta función probabilística se obtiene la mejor configuración para los hiperparámetros definidos, aplicando una búsqueda más eficiente computacionalmente. Dicha función probabilística, conocida como función subrogada o simplemente como “Muestreador” es la representación probabilística de la función objetivo construida a partir de evaluaciones anteriores (Wikipedia, 2024). Sin entrar en detalles, existen varios tipos de muestreadores, como los procesos gaussianos, la regresión *Random Forest* y Estimadores de *Tree Parzen* (TPE) (Koehrsen, 2018).

7.2.3 Empleo de la optimización bayesiana para la búsqueda de los valores de la tasa de cruce y la tasa de mutación

Una vez comprendido a nivel general el funcionamiento de la Búsqueda Bayesiana se procede a reportar la configuración esencial que se empleara para dicho algoritmo:

- Los hiperparámetros que se optimizarán son la tasa de cruce y la tasa de mutación.
- Dichos parámetros son esencialmente probabilidades por lo que se define el espacio de búsqueda de las tasas de cruce y mutación entre $[0,1] \in \mathbb{R}$.
- El muestreador utilizado es el TPE. A grandes rasgos el TPE es un estimador que utiliza las reglas de Bayes para encontrar espacios de búsqueda prometedores. Para ello hace uso de dos distribuciones condicionales: una para buenos resultados y otra para malos. Selecciona nuevos hiperparámetros maximizando la relación entre estas distribuciones, enfocándose en las regiones prometedoras del espacio de búsqueda (Bergstra, Bardene, Bengio, & Kégl, 2011).

Resulta de suma importancia especificar cual es la función objetivo a maximizar mediante la Búsqueda Bayesiana. En este caso, esta se basará en el hipervolumen promedio obtenido para 6 tipos de aula distintos.

Específicamente, para cada tipo de aula, se emplearán 5 repeticiones del algoritmo NSGA-II (100 generaciones para cada repetición), calculando el hipervolumen promedio obtenido mediante esas 5 repeticiones. Así, se obtiene un promedio de hipervolumen por cada configuración de aula, y finalmente como función a maximizar en cada “trial” de la Búsqueda Bayesiana, se calculará la media de los hipervolumenes promedios obtenidos por cada tipo de aula, obteniendo así una medida de calidad global, de la cual hará uso la Búsqueda Bayesiana para buscar valores en regiones prometedoras en futuras *trials*. A esta función objetivo se le nombra como ‘Hipervolumen Total’ para evitar confusiones en explicaciones posteriores.

Ya definida la configuración del algoritmo para la optimización mediante búsqueda bayesiana, se continua con el planteamiento de los experimentos a realizar para dicha optimización de hiperparametros.

Concretamente, se realizará una Búsqueda Bayesiana en la cual se ejecutarán 200 *trials*. En cada una de ellas, se ejecutará el algoritmo NSGA-II con operadores de cruce y mutación personalizados.

A continuación, se presenta en la siguiente tabla, los 6 tipos de aula utilizadas para la optimización de los hiperparametros:

Tabla 4: Instancias para la optimización de los hiperparámetros

Tipo de aula	Nº alumnos	R	K	G	Tamaño de población	Nº de generaciones
1	40	R11	K1	G11	50	100
2	40	R12	K2	G21	50	100
3	52	R13	K3	G12	50	100
4	52	R21	K1	G22	50	100
5	60	R22	K2	G13	50	100
6	60	R23	K3	G23	50	100

En la Tabla 4 se muestra cómo se han intentado plasmar varias posibles combinaciones de tipos de clase, para obtener unas probabilidades de cruce y mutación que funcionen generalmente bien sea cual sea la tipología del aula. El tamaño de población es de 50 para todas las configuraciones y el criterio de parada se ha fijado en 100 generaciones.

En cuanto a los conjuntos de datos R_{ij} , K_j y G_{ij} , mediante el sufijo j se indica que cada conjunto de datos es una instancia diferente (de tamaño N.º alumnos correspondientes), simulando así varias instancias (aulas) diferentes pero que provienen del mismo conjunto de datos original $R1, R2, K, G1$ o $G2$ (267 alumnos originalmente).

Una vez definido el valor objetivo a optimizar en cada “trial” de la búsqueda bayesiana, se presenta la evolución de los valores del “hipervolumen total” para un total de 200 trials. En la siguiente figura se muestra un gráfico de dispersión donde el eje x representa la tasa de mutación y el eje y la tasa de cruce, mediante un gradiente de color se distinguen los valores obtenidos en cada iteración para el “hipervolumen total”.

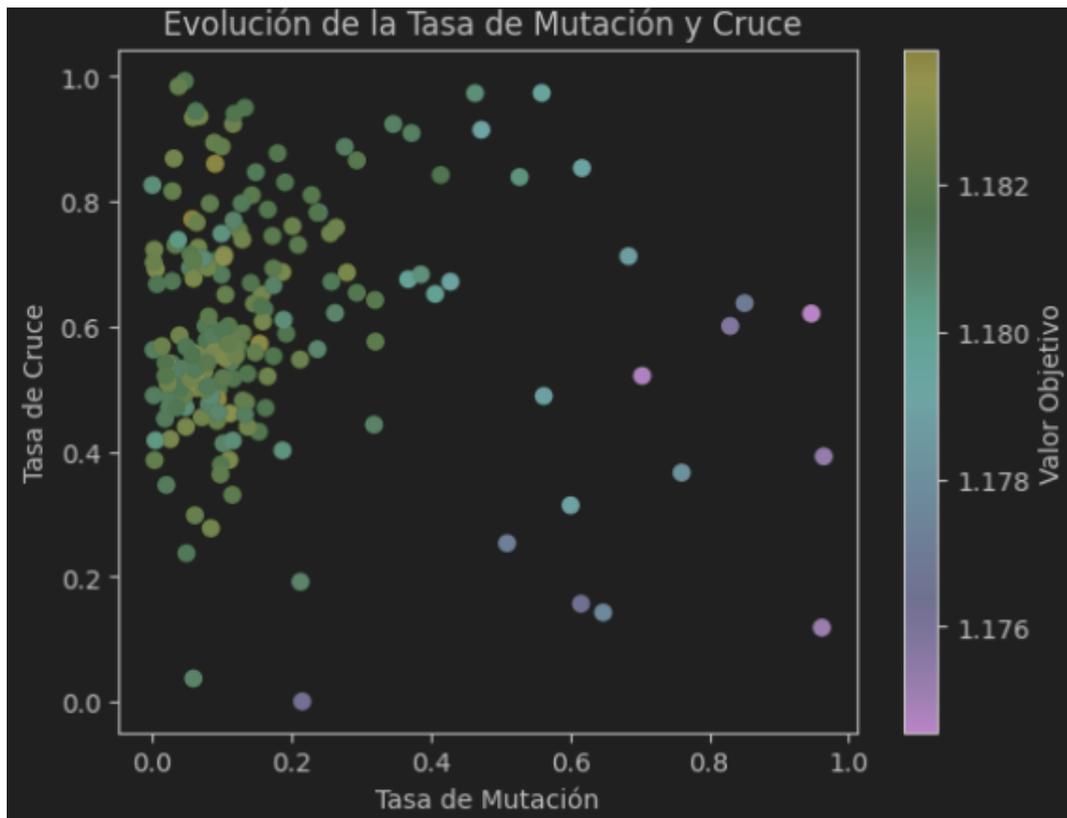


Figura 22: Resultados de la búsqueda bayesiana

Mediante la Figura 22, se puede apreciar que, efectivamente, este algoritmo genético funciona mejor con una probabilidad de cruce alta y una probabilidad de mutación baja, puesto que el hipervolumen total toma valores mayores en estos casos, como es habitual en los algoritmos genéticos.

Por último, mediante el siguiente gráfico se muestra la evolución del hipervolumen total durante el proceso de búsqueda:

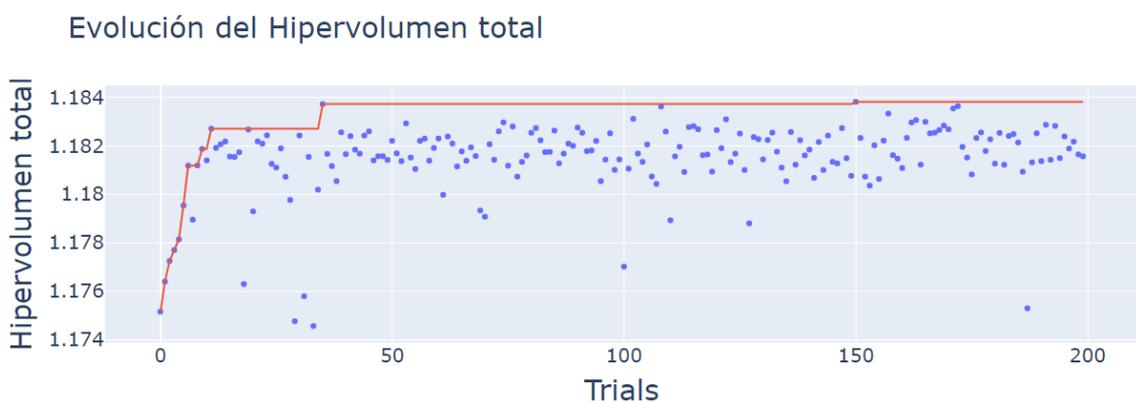


Figura 23: Evolución del "hipervolumen total" durante la búsqueda bayesiana

En la Figura 23 se aprecia que el valor óptimo tras las 200 iteraciones se ha obtenido en la iteración 150, con una tasa de mutación de 5.7% y una tasa de cruce de 77%. Estos parámetros son los que se utilizarán en los posteriores experimentos.

Una vez definidos los valores de los hiperparámetros principales, se finaliza este apartado con los experimentos principales realizados para demostrar la superioridad del Algoritmo NSGA-II frente a otros métodos utilizados con anterioridad en la bibliografía.

7.3 Comparación con otros métodos

Este subapartado se organiza de la siguiente manera: Primero se comentarán los objetivos principales del experimento. Tras ello se describirán los métodos que se van a utilizar en el experimento y la razón por las que se han comparado con el método propuesto en este proyecto. Consecuentemente, se especificarán las instancias que se van a utilizar en el experimento y para finalizar se describirá el proceso de análisis de los resultados.

7.3.1 Objetivo principal del experimento

El objetivo principal de este experimento es demostrar la superioridad en cuanto a rendimiento y resultados del Algoritmo NSGA-II con operadores de cruce y mutación personalizados frente a otros métodos utilizados.

Con dicho fin, tomando en cuenta que el Algoritmo NSGA-II trabaja con una población de soluciones, se hará uso del hipervolumen, por ser el único indicador de calidad conocido que es totalmente sensible a la dominancia de Pareto, es decir, siempre que un conjunto de soluciones domina a otro conjunto, tiene un valor de indicador de hipervolumen más alto que el segundo conjunto (Bader & Zitzler, 2010).

Por lo que, en definitiva, tomando en cuenta que se está trabajando con procesos estocásticos, se emplearán varias ejecuciones de dichos algoritmos obteniendo una población de hipervolumenes por método. Así, mediante los test estadísticos necesarios se demostrará si el algoritmo propuesto en este trabajo obtiene resultados significativamente mejores en cuanto al hipervolumen.

7.3.2 Métodos a comparar

A continuación, se exponen los 4 métodos a comparar experimentación, razonando su inclusión en la experimentación:

NSGA-II CON OPERADORES DE CRUCE Y MUTACIÓN PERSONALIZADOS

Es el algoritmo propuesto con el objetivo de mejorar el proceso de formación de equipos propuesto en este proyecto. A continuación, se especifican los operadores y parámetros que se emplearán:

Parámetro	Valor
Selección	Selección por torneo binario (mediante el operador <i>crowded comparison</i> , ver sección Operador <i>crowded-comparison</i> para el reemplazamiento)
Cruce	Operador de cruce personalizado (tasa de cruce: 77%)
Mutación	Operador de mutación personalizado (tasa de mutación: 5%)
Reemplazamiento	Selección por torneo binario (mediante el operador <i>crowded comparison</i> , ver sección Operador “ <i>crowded-comparison</i> ”)
Tamaño de la población	50 individuos

NSGA-II CON OPERADORES DE CRUCE Y MUTACIÓN CONVENCIONALES

Se empleará el mismo algoritmo, pero con operadores de cruce y mutación más simples. Estos operadores son frecuentemente utilizados en problemas donde se opta por la representación mediante permutación, los cuales se llaman Cruce *Cut and Crossfill* (ver sección 3.3 Algoritmos genéticos) y Mutación Inversa (ver sección 3.3 Algoritmos genéticos). El objetivo de incluir este método es demostrar que los operadores personalizados realmente han influido en la mejora del rendimiento del algoritmo en el problema para el que se han adaptado. Cabe mencionar que las tasas de cruce y mutación se han mantenido iguales puesto que son valores genéricamente normativos.

Parámetro	Valor
Selección	Selección por torneo binario (mediante el operador <i>crowded comparison</i> , ver sección 3.4.3)
Cruce	Operador de cruce <i>Cut and Crossfill</i> (tasa de cruce: 77%)
Mutación	Operador de mutación inversa (tasa de mutación: 5%)
Reemplazamiento	Selección por torneo binario (mediante el operador <i>crowded comparison</i> , ver sección 3.4.3)
Tamaño de la población	50 individuos

HEURÍSTICO MONOBJETIVO DEL CSIC

Con el objeto de demostrar la superioridad técnica del algoritmo NSGA-II respecto a heurísticos monobjetivo presentados en la literatura existente, se procede a exhibir el algoritmo propuesto por (Andrejczuk, Bistaffa, Blum, Rodríguez-Aguilar, & Sierra, 2019).

Primero de todo, se debe recalcar que se ha elegido este algoritmo, llamado *SynTeam*, ya que el problema que presentan en (Andrejczuk, Bistaffa, Blum, Rodríguez-Aguilar, & Sierra, 2019) comparte a grandes rasgos el modo en el que evalúa los equipos. Lo hace mediante lo que llama “equipos sinérgicos”: equipos diversos en cuanto a personalidad y género y cuyos miembros cubran todas las competencias necesarias para completar una tarea.

De modo que, al igual que el presente problema, este algoritmo ha sido pensado para trabajar con 3 funciones objetivo relacionados con las aptitudes técnicas, personalidad y género de los estudiantes. El principal problema del Heurístico propuesto, es que enfoca el problema como monobjetivo mediante la suma ponderada de las 3 características.

Debido a esto, resulta interesante analizar si el algoritmo propuesto en este proyecto es capaz de superar a un Heurístico con enfoque monobjetivo y asentar las afirmaciones realizadas en la revisión bibliográfica sobre que el enfoque multiobjetivo supera al enfoque monobjetivo.

A continuación, se procede a dar una breve explicación del funcionamiento de *SynTeam*:

- Es una heurística basada en la búsqueda local y la exploración de soluciones vecinas.
- Se comienza con una permutación aleatoria de la clase de alumnos que se toma como solución inicial. Principalmente se aplican dos tipos de búsqueda local. La primera de todas denotada como *Búsqueda_local_1*, se basa en escoger aleatoriamente dos equipos de la solución actual y recombinar los alumnos de ambos equipos para encontrar dos nuevos equipos con una aptitud ponderada f_{t_i} mayor a la original, almacenando la nueva solución obtenida como C_m' .
- Cabe destacar que, para comparar el fitness a nivel de equipos en la *Búsqueda_local_1*, al hacer uso de métricas de 3 dimensiones, estas se ponderan equitativamente para lograr un fitness de una sola dimensión.
- Si el algoritmo detecta que se han ejecutado n_l iteraciones no mejoradas (no tienen por qué ser consecutivas) se aplica la segunda búsqueda local.
- En esta nueva búsqueda local, denominada *Búsqueda_local_2*, se realizan intercambios entre estudiantes de distintos equipos, hasta encontrar el primer cambio que resulte en una mejora del fitness, guardando esta solución como C_m' , tras ello se reinicializa el contador c_l .

Debido a que se necesita comparar este algoritmo, que originalmente trabajaba con una única solución, con el algoritmo NSGA-II que trabaja con una población de soluciones, se han realizado dos cambios en para adaptarlo:

- Para cada instancia, se ejecutará el heurístico 50 veces para obtener una población del mismo tamaño que el NSGA-II. Los pesos necesarios para la suma ponderada de la función objetivo se aleatorizarán para cada individuo de la población, con el objetivo de una población de soluciones lo más estocástica posible.
- El criterio de terminación para cada individuo obtenido mediante *Synteam* en este caso será, tiempo de ejecución (t_e) de $\frac{t_e}{n_p}$, siendo n_p el tamaño de población de cada solución. Así se podrá formar una población de soluciones mediante el heurístico, y el tiempo de ejecución total para este será equiparable al del NSGA-II.

RANDOM SEARCH

Por último, Se propone incluir una generación aleatorizada de una población de soluciones iniciales (del mismo tamaño que en el NSGA-II), también conocido como *Random Search*, para demostrar que el algoritmo propuesto es un método que aplican técnicas inteligentes y que converge hacia resultados óptimos.

7.4 Diseño de los experimentos

Una vez definido el conjunto de datos, los objetivos buscados en la experimentación y los métodos que se van a comparar, se continuara con el diseño de los experimentos, con el propósito de aclarar el procedimiento utilizado para la comparativa de rendimientos entre el método propuesto y los demás algoritmos presentados.

Una vez definidas las muestras generadas, mediante la combinación de estas se pretende crear 6 instancias (tipos de aulas) diferentes que se emplearan para la comparación de los diferentes algoritmos presentados en la sección 7.3. Una instancia se entiende como una 4-tupla donde cada elemento indica la siguiente información:

- Número de alumnos en el aula.
- Tipo de distribución de puntuación técnica del aula.
- Tipo de distribución de género del aula.
- Tiempo de ejecución de los algoritmos.

Para cada instancia, siendo los algoritmos propuestos estocásticos, hace falta trabajar con varias soluciones para hacer frente a la aleatoriedad. Cada configuración se ejecutará 50 veces por método, obteniendo así 200 hipervolumenes por instancia (4 métodos).

Mediante la siguiente tabla se presentan las instancias empleadas:

Tabla 5: Instancias empleadas para la experimentación

Tipo de aula	Nº alumnos	R	K	G	Tamaño de población	Nº de generaciones	Tiempo de ejecución
1	40	R1	K	G1	50	100	60 segundos
2	40	R2	K	G2	50	100	60 segundos
3	52	R1	K	G2	50	100	90 segundos
4	52	R2	K	G1	50	100	90 segundos
5	60	R1	K	G1	50	100	120 segundos
6	60	R2	K	G2	50	100	120 segundos

En la Tabla 5, se puede apreciar que el tiempo de ejecución para cada método aumenta junto al tamaño de aula. Esto se debe a que cuanto mayor sea el número de alumnos en el aula, más tiempo necesitan los distintos algoritmos para converger hacia soluciones prometedoras.

En cada una de las instancias, se ha utilizado un conjunto diferente de alumnos, eligiendo aleatoriamente un subconjunto del total de los 267 alumnos según el tamaño de la clase de cada tipo de aula.

Asimismo, como el hipervolumen es sensible a las escalas de las diferentes funciones objetivo, si una de las dimensiones tiene una escala significativamente mayor que las demás, esto hará que esa dimensión tenga un mayor impacto en el cálculo del hipervolumen, lo que puede sesgar los resultados y dar una impresión equivocada de la calidad de las soluciones en el espacio objetivo.

Además, los resultados finales obtenidos en cada ejecución solamente contemplarán los individuos no dominados, puesto que estos son las particiones de clase que nos interesan por tener los mejores resultados en cada ejecución del algoritmo en cuestión.

Debido a esto, en cada repetición, tras obtener el fitness multidimensional de cada uno de los individuos de todos y cada uno los métodos, todas las poblaciones se recopilan en una misma matriz y se les aplica una normalización a norma unitaria en cada dimensión (función objetivo), para que todos las 3 dimensiones se encuentren en el rango [0,1]. Tras esta normalización, se almacenan los fitness normalizados de cada individuo por método, obteniendo una población de soluciones no dominadas normalizada para cada método.

7.5 Análisis de las soluciones

Se procede a analizar las soluciones obtenidas en cada una de las configuraciones mediante un diagrama de violín (Hintze & Nelson, 1998). Los diagramas de violín son similares a los diagramas de caja, salvo que también muestran la densidad de probabilidad de los datos en diferentes valores, normalmente suavizada por un estimador de densidad kernel (Parzen, 1962). Mientras que un diagrama de caja muestra un resumen estadístico como la media/mediana y los rangos intercuartílicos, el diagrama de violín muestra la distribución completa de los datos. El gráfico de violín puede utilizarse en datos multimodales (más de un pico). En este caso, un gráfico de violín muestra la presencia de diferentes picos, su posición y amplitud relativa.

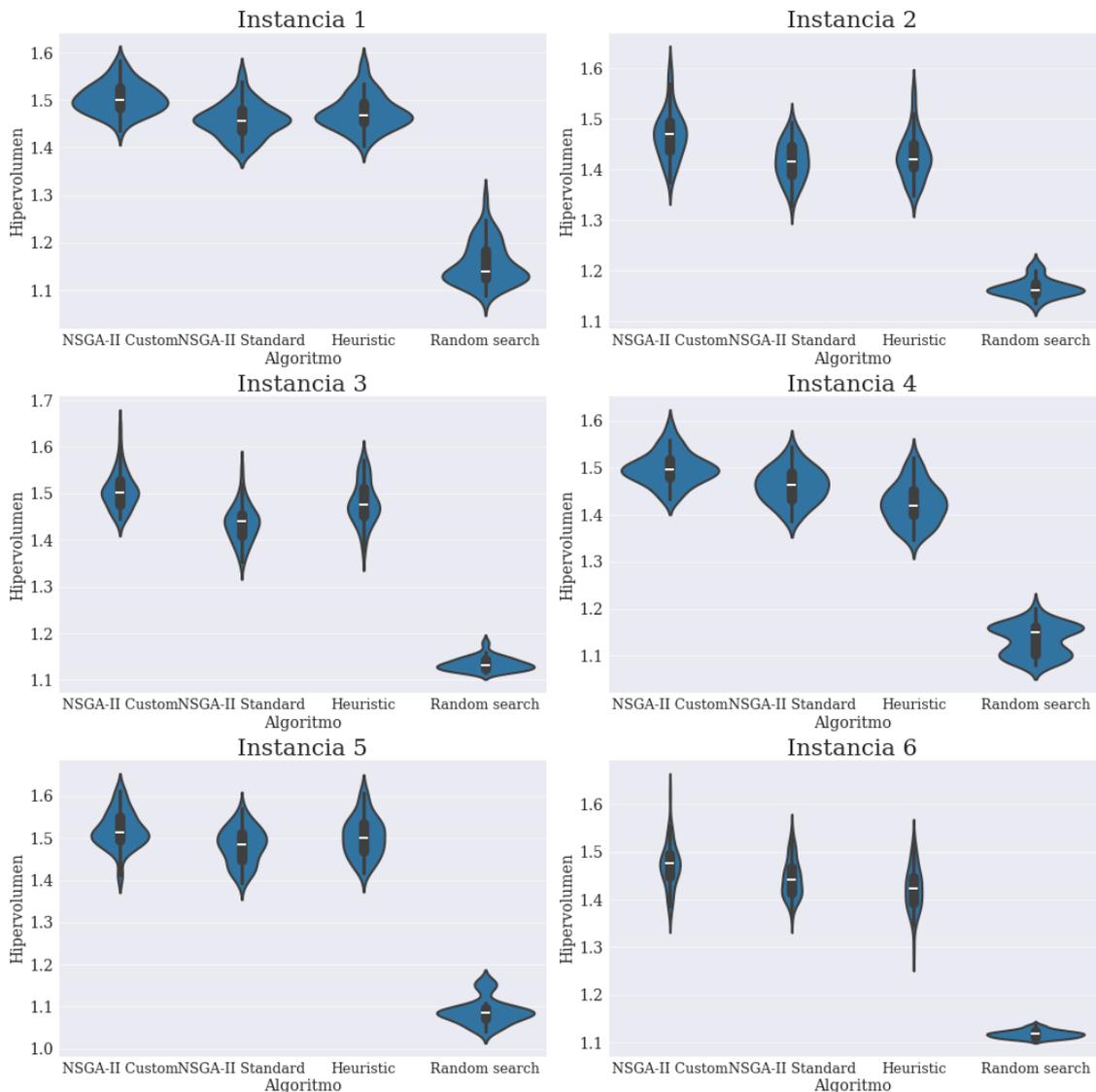


Figura 24: Resultados de los métodos a comparar para cada instancia

Mediante las figuras asociadas a los resultados obtenidos, podemos observar que, en principio, el método propuesto supera con creces las medias de hipervolumen obtenidas en el *Random Search*, demostrando que tanto la heurística simple como el

algoritmo NSGA-II son métodos que aplican técnicas inteligentes y que exhiben una mejora progresiva en los resultados.

En definitiva, el NSGA-II con operadores de cruce y mutación personalizados es el método que mejor media obtiene en valores de hipervolumen para todas las instancias. Aun así, se debe demostrar que dichas diferencias son estadísticamente significativas mediante un ANOVA (Montgomery & St, 2022) y su correspondiente *post-hoc*, en caso de que las poblaciones provengan de una distribución normal. En caso contrario, se deberá aplicar un método no paramétrico de comparación de medias.

Este test ANOVA se aplicaría a nivel de cada instancia. Con el objetivo de saber sí, la aplicación de cualquier factor (algoritmo) tiene efectos estadísticamente significativos sobre la variable respuesta, en este caso, el hipervolumen de la población final obtenida. En caso de no cumplirse la hipótesis nula, los factores tienen un efecto distintivo entre sí para el hipervolumen. Aun así, se debería de comprobar que el algoritmo propuesto es el responsable de esa diferencia significativa estadísticamente, mediante un test *post-hoc*.

Primero de todo se debe comprobar si los datos de cada instancia cumplen con las asunciones tomadas para el test ANOVA (Wikipedia, 2023): Independencia, Normalidad y Esfericidad.

La primera suposición, independencia de las observaciones, no se cumple puesto que estamos analizando los resultados a nivel de instancia, por lo que se parte del mismo conjunto de datos (alumnos) para todos los individuos (hipervolumen de la población final por método) generados dentro de la misma instancia. Debido a esto, se debe dejar de lado el test ANOVA y aplicar un test no paramétrico.

Un test no paramétrico apropiado en este caso sería el test de Kruskal-Wallis (Kruskal & Wallis, 1952). Esta prueba es idéntica a la prueba ANOVA, pero en su versión no paramétrica. La prueba de Kruskal-Wallis no asume normalidad en los datos, en oposición al tradicional ANOVA. Sí asume, bajo la hipótesis nula, que los datos vienen de la misma distribución. Una forma común en que se viola este supuesto es con datos heterocedásticos.

Para comprobar que se cumple la suposición de homocedasticidad, una prueba común es La prueba de Bartlett (Dr. Hossein & Miodrag, 2011) se utiliza para probar la hipótesis nula, que todas las varianzas de los distintos tratamientos de una población son iguales, frente a la hipótesis alternativa de que al menos dos son diferentes. Para cada una de las instancias, la prueba de Bartlett ha dado un p-valor superior a 0.05, afirmando que para todas las instancias se cumple el supuesto de homocedasticidad, por lo que se puede aplicar el test de Kruskall Wallis.

En el caso que se aborda, la prueba de Kruskall-Wallis consistiría en, siendo $X_{m \times n}$ la tabla de datos correspondientes a los hipervolumenes obtenidos en una configuración dada, donde m representa cada resultado obtenido y n sería cada uno de los métodos utilizados, reemplazar la tabla original por otra $r_{m \times n}$, donde los valores de r son el orden de los hipervolumenes de $X_{m \times n}$ en cada método. Con esta nueva tabla, se obtendría un estadístico $H = \frac{SS_e}{SS_t}$, siendo SS_e la varianza intergrupala y SS_t la varianza intra grupala.

Mediante la comparación de este estadístico H la distribución χ^2 con $n - 1$ grados de libertad, con una probabilidad de error tipo I de 0.05, aceptaríamos o no la hipótesis nula (H_0) de que no existen diferencias significativas entre el rendimiento de los distintos métodos propuestos. Es importante recalcar que, mediante el test de Kruskal-Wallis, se comparan rangos (rankings) y no medias.

En este caso, los resultados obtenidos para cada una de las 6 configuraciones devuelven un p-valor menor que $\alpha = 0.05$, por lo que se considera que en todas las configuraciones hay diferencias significativas entre al menos un par de métodos.

Debido a esto, se decide aplicar el test de Dunn (Dunn, 1964), el cual es una prueba post-hoc destinada a encontrar los grupos de datos que difieren después de que una la prueba de Kruskal Wallis haya rechazado la hipótesis nula de que el rendimiento de las comparaciones en los grupos de datos es similar. A continuación, se muestran los p-valor obtenidos en cada configuración:

Tabla 6: Test de Nemenyi: NSGA-II vs 2º algoritmo en términos de hipervolumen

Instancia	Método 1	Método 2	Evidencia	p-valor
Instancia 1	NSGA-II Custom	Heurístico simple	Mayor que	0.01
Instancia 2	NSGA-II Custom	Heurístico simple	Mayor que	0.01
Instancia 3	NSGA-II Custom	Heurístico simple	Mayor que	0.01
Instancia 4	NSGA-II Custom	NSGA-II Estándar	Mayor que	0.01
Instancia 5	NSGA-II Custom	Heurístico simple	Mayor que	0.01
Instancia 6	NSGA-II Custom	NSGA-II Estándar	Mayor que	0.01

Con la tabla 4 demostramos lo que se podía intuir en los diagramas de violín expuestos más arriba. Para todas las instancias, el test de Dunn refleja la superioridad en rendimiento para el método NSGA-II personalizado puesto que el p-valor es menor al error de tipo I ($\alpha = 0.05$) en todos los casos. Definitivamente, el algoritmo NSGA-II con operadores de cruce y mutación personalizados devuelve resultados mejores en términos de hipervolumen en comparación al heurístico simple propuesto y al algoritmo NSGA-II con operadores convencionales.

8 Conclusiones

La principal conclusión obtenida en este trabajo es que se ha logrado el principal objetivo: Adaptar los operadores principales del algoritmo genético multiobjetivo NSGA II, los operadores de cruce y mutación, a la casuística del problema propuesto, obteniendo mejores resultados que los algoritmos que se han propuesto en otros casos de TFP. Demostrando así que, aplicar operadores genéticos inteligentes que hagan uso de los fitness y de la composición de los equipos de una partición de equipos en el aula aporta valor añadido al proceso automático de formación de equipos, tema que no se había profundizado ni tocado en el ámbito de los Problemas de Formación de Equipos.

La experimentación solamente ha incluido configuraciones de aulas con un número de alumnos de 40 a 60, debido a que el tamaño de aula en la mayoría universidades españolas suele rondar estas cifras. Cabe añadir que, el incrementar el tamaño del aula afecta principalmente a la velocidad de ejecución del operador de cruce el cual debe trabajar con más equipos y consecuentemente valorar el fitness de todos ello, cosa que computacionalmente se vuelve cada vez más costoso a la medida que aumenta el conjunto de los equipos.

Cabe destacar que, el autor del TFM no tenía experiencia previa en Python y mucha parte de su aprendizaje ha sido autodidacta, por lo tanto, el nivel de programación de este no es comparable a investigadores que utilizan este lenguaje usualmente, cosa que es muy importante en trabajos que incluyen algoritmos metaheurísticos puesto que la habilidad de programación juega un papel muy importante a la hora de minimizar los tiempos de ejecución de dichos métodos. Con esto se quiere recalcar que, el código se podría llegar a refinar y los resultados podrían haber sido mejores aún, sobre todo depurando y optimizando el código de los operadores de cruce y mutación.

Por último, es importante mencionar que las asunciones sobre las tasas de mutación y cruce, comúnmente establecidas entre el 5-10% y 75-90% respectivamente, se han confirmado mediante la optimización de hiperparámetros utilizando Búsqueda Bayesiana. La librería Optuna, fácil de entender y manejar, ha simplificado significativamente este proceso. Esto demuestra que este método de optimización es altamente útil para la configuración de algoritmos genéticos, una aplicación que no se había explorado previamente en problemas TFP.

9 Bibliografía

- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). *Optuna: A Next-generation Hyperparameter Optimization Framework*.
- Ali Babar, M., & Kitchenham, B. (2007). The Impact of Group Size on Software Architecture Evaluation: A Controlled Experiment. En *First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)* (págs. 420-429).
- Andrejczuk, E., Bistaffa, F., Blum, C., Rodríguez-Aguilar, J., & Sierra, C. (2019). Synergistic team composition: A computational approach to foster diversity in teams. *Knowledge-Based Systems, 182*, 104799.
- Bader, J., & Zitzler, E. (2010). A Hypervolume-Based Optimizer for High-Dimensional Objective Spaces.
- Baker, J. E. (1985). Adaptive Selection Methods for Genetic Algorithms. En *Proceedings of the 1st International Conference on Genetic Algorithms* (págs. 101–111). L. Erlbaum Associates Inc.
- Bekele, R. (2006). Computer-Assisted Learner Group Formation Based on Personality Traits.
- Bergstra, J., Bardene, R., Bengio, Y., & Kégl, B. (2011). *Algorithms for Hyper-Parameter Optimization*. Curran Associates, Inc.
- Blank, J., & Deb, K. (2020). *Pymoo: Multi-Objective Optimization in Python*.
- Briggs Myers, I., McCaulley, M. H., Quenk, N. L., & Hammer, A. L. (1985). *Mbti Manual: A Guide to the Development and Use of the Myers - Briggs Type Indicator*. Consulting Psychologists Press.
- Campion, M. A., Medsker, G. J., & Higgs, C. A. (1993). Relations between work group characteristics and effectiveness: Implications for designing effective work groups. *Personnel Psychology*.
- Candel, G., Sanchez-Anguix, V., Alberola, J., Julián, V., & Botti, V. (2023). An Integer Linear Programming Model for Team Formation in the Classroom with Constraints. Springer Nature Switzerland.
- Carlos A., C. C., Gary B., L., & David A., V. V. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition*. Springer.
- Casotti, A., & Krohling, R. (2023). A multi-objective formulation for the team formation problem using Krippendorff's disagreement and sociometric cohesion with Pareto-solutions obtained via evolutionary algorithms. *Computers & Operations Research, 106444*.
- Chen, S.-H. &.-C. (2011). Operators of the Two-Part Encoding Genetic Algorithm in Solving the Multiple Traveling Salesmen Problem. Proceedings - 2011 Conference on Technologies and Applications of Artificial Intelligence. En *2011 International*

- Conference on Technologies and Applications of Artificial Intelligence* (págs. 331-336).
- Coello Coello, C. A., & Toscano Pulido, G. (2001). A Micro-Genetic Algorithm for Multiobjective Optimization. *Evolutionary Multi-Criterion Optimization* (págs. 126--140). Berlin, Heidelberg: Springer-Verlag.
- Darwin, evolución y selección natural*. (s.f.). Recuperado el 2024 de Mayo de 26, de Khan Academy: <https://es.khanacademy.org/science/ap-biology/natural-selection/natural-selection-ap/a/darwin-evolution-natural-selection>
- Daves, D. (14 de Diciembre de 2023). *What Is Bayesian Hyperparameter Optimization? With Tutorial*. Recuperado el 01 de Junio de 2024, de Weight and Biases: https://wandb.ai/wandb_fc/articles/reports/What-Is-Bayesian-Hyperparameter-Optimization-With-Tutorial---Vmlldzo1NDQyNzcw
- De Jong, K. A. (1975). *Analysis of the behavior of a class of genetic adaptive systems*. USA: University of Michigan.
- Deb, K., & Goldberg, D. E. (1989). An Investigation of Niche and Species Formation in Genetic Function Optimization. *Proceedings of the 3rd International Conference on Genetic Algorithms*, 42–50.
- Deb, K., & Jain, H. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18, 577-601.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.
- Dr. Hossein, A., & Miodrag, L. (2011). Bartlett's Test. *International Encyclopedia of Statistical Science*, 20-23.
- Dunn, O. J. (1964). Multiple comparisons using rank sums. *Technometrics*, 3, 241-252.
- Eiben, A. E., & Smith, J. (2003). *Introduction to Evolutionary Computing*. Computing Reviews.
- Esgario, J., Eguas da Silva, I., & Krohling, R. (2019). *Application of Genetic Algorithms to the Multiple Team Formation Problem*.
- Fonseca, C., & Fleming, P. (1999). Genetic Algorithms for Multiobjective Optimization: Formulation Discussion and Generalization. *the fifth Intl conference on Genetic Algorithms*, 93.
- Gallagher, S., Stepien, W., & Rosenthal, H. (1992). The Effects of Problem-Based Learning On Problem Solving. *Gifted Child Quarterly - GIFTED CHILD QUART*, 36, 195-200.
- Game, P. &. (2020). Bio-inspired Optimization: metaheuristic algorithms for optimization. *ArXiv*, 13, 8775.

- Garshasbi, S., Mohammadi, Y., Graf, S., & Garshasbi, S. (2019). Optimal Learning Group Formation: A Multi-objective Heuristic Search Strategy for Enhancing Inter-group Homogeneity and Intra-group Heterogeneity. *Expert Systems with Applications*, 118, 506-521.
- Gate, R. (s.f.). *Multi-objective optimization approach based on Minimum Population Search algorithm*. Recuperado el 2024 de 06 de 22, de Research Gate.
- Gaviola, M. L., Atanda, A., Howard, M. L., & Yuet, W. C. (2019). A systematic approach to team creation and peer evaluation in a large classroom setting. *Currents in Pharmacy Teaching and Learning*, 11, 1231-1238.
- Gómez-Zará, D. &. (2020). A Taxonomy of Team-Assembly Systems: Understanding How People Use Technologies to Form Teams. *Proceedings of the ACM on Human-Computer Interaction*, 4, 1-36.
- Graf, S. &. (2006). Forming Heterogeneous Groups for Intelligent Collaborative Learning. *Intelligent Tutoring Systems* (págs. 217-226). Taiwan: Springer Berlin Heidelberg.
- Harris, C., Millman, K., Walt, S., Gommers, R., Virtanen, P., Cournapeau, D., . . . Oliphant, T. (2020). Array programming with NumPy. *Nature*, 585, 357-362.
- Hintze, J. L., & Nelson, R. D. (1998). Violin Plots: A Box Plot-Density Trace Synergism. *The American Statistician*, 52(2), 181-184.
- Holland, J. (1992). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. The MIT Press.
- Huxham, M. &. (2000). Assigning Students in Group Work Projects. Can We Do Better than Random? *Innovations in Education and Training International*, 37, 17--22.
- J. -M. Flores-Parra, M. C.-P.-C.-P. (2018). Towards Team Formation Using Belbin Role Types and a Social Networks Analysis Approach. *2018 IEEE Technology and Engineering Management Conference* , 1-6.
- James, B., & Y., B. (2012). Random Search for Hyper-Parameter Optimization. *JMLR.org*, 13, 281-305.
- Johnson, D. W. (2004). *Assessing students in groups : promoting group responsibility and individual accountability*. Corwin: Corwin Press.
- Juan M. Alberola, E. d.-A. (2016). An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowledge-Based Systems*, 101, 1-14.
- Koehrsen, W. (18 de 06 de 2018). *A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning*. (Towards Data Science) Recuperado el 27 de 06 de 2024, de Towards Data Science: <https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f>
- Kruskal, W. H., & Wallis, W. A. (1952). Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*(260), 583--621.

- Laal, M., & Laal, M. (2012). Collaborative learning: What is it? *Procedia - Social and Behavioral Sciences*, 31, 491-495.
- Lamont, G. (2000). Multiobjective Optimization with Messy Genetic Algorithms. En *ACM Symposium on Applied Computing* (págs. 470–476). Como, Italy: Association for Computing Machinery.
- Liu, S., Joy, M., & Griffiths, N. (2013). An Exploratory Study on Group Formation Based on Learning Styles. (págs. 95-99). 2013 IEEE 13th International Conference on Advanced Learning Technologies.
- López-Trujillo, Y., Rosete-Suárez, A., André-Ampuero, M., & Infante-Abreu, A. L. (2020). *Métodos para la formación de múltiples equipos de estudiantes aplicando un enfoque multiobjetivo*. Universidad Tecnológica de La Habana José Antonio Echeverría. Cuba.
- Ma, H., Li, J., Zhu, H., Tang, W., Zhuoxuan, H., & Tang, Y. (2022). Collaborative Optimization of Learning Team Formation Based on Multidimensional Characteristics and Constraints Modeling: A Team Leader-Centered Approach via E-CARGO. *IEEE Transactions on Computational Social Systems*, 11, 184-196.
- Miller, B. I., & Goldberg, D. E. (1995). Genetic Algorithms , Tournament Selection, and the effect of Noise. *Complex Syst.*, 9.
- Miranda, P. &. (2020). A Multi-Objective Optimization Approach for the Group Formation Problem. Expert Systems with Applications. *Expert Systems with Applications*, 113828.
- Montgomery, D., & St, C. (2022). The Analysis of Variance. En *Design and Analysis of Experiments, 9th Edition*. Arizona State University: Wiley.
- Nash, J. F. (1950). The Bargaining Problem. *Econometrica*, 2, 155-162.
- Odo, C., Masthoff, J., & Beacham, N. (2019). Group Formation for Collaborative Learning: A Systematic Literature Review. En *20th International Conference on Artificial Intelligence in Education, AIED 2019* (págs. 206-212). Springer Verlag.
- Osman, I., & Kelly, J. (1996). Meta-Heuristics: An Overview. *Springer US*, 1-21.
- Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The annals of mathematical statistics*, 1065--1076.
- Rudolph, G. (2001). Evolutionary search under partially ordered sets.
- S. Liu, M. J. (2013). "An Exploratory Study on Group Formation Based on Learning Styles. Beijing: IEEE 13th International Conference on Advanced Learning Technologies.
- Sánchez Anguix, V. (s.f.). *Contenido teórico de la asignatura Optimización (Grado en Ciencia de datos, UPV)*. Valencia.

- Sánchez-Anguix, V., Alberola, J. M., Del Val, E., Palomares, A., & Teruel, M. D. (2023). Comparing computational algorithms for team formation in the classroom: a classroom experience. *Applied Intelligence*, 53, 23883-23904.
- Schaffer, J. D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic. *Proceedings of the First Int. Conference on Genetic Algorithms*, Ed. G.J.E Grefenstette, J.J. Lawrence Erlbaum, 93-100.
- Schaffer, J., Caruana, R., Eshelman, L., & Das, R. (1989). A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization. Third international Conference on Genetic Algorithms.
- Sen, S., & Dutta, P. S. (2000). Searching for optimal coalition structures. *Proceedings Fourth International Conference on MultiAgent Systems*, 287-292.
- Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104, 148-175.
- Shi-Jie(Gary), C., & Li, L. (2004). Modeling Team Member Characteristics for the Formation of a Multifunctional. *IEEE Transactions on Engineering Management*, VOL. 51(2), 111-124.
- Srinivas, N., & Deb, K. (1994). Multiobjective Optimization Using Nondominated. *Evolutionary Computation*, 2, 221-248.
- Talal Rahwan, T. P. (2015). Coalition structure generation: A survey. *Artificial Intelligence*, 229, 139-174.
- The Nature of Mathematical Programming*. (s.f.). (INFORMS Computing Society) Recuperado el 21 de 06 de 2024, de Mathematical Programming Glossary: <https://web.archive.org/web/20140305080324/http://glossary.computing.society.informs.org/index.php?page=nature.html>
- Ugarte, N., Aranzabal, A., Arruarte, A., & Larrañaga, M. (2022). Using the Behavioural Tendency of Students in a Team Environment for Team Formation. En *2022 IEEE Frontiers in Education Conference (FIE)* (págs. 1-9).
- Verma, S., Pant, M., & Snasel, V. (2021). A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. *IEEE Access*, 9.
- Wikipedia, C. d. (02 de 05 de 2023). *Análisis de la varianza*. Recuperado el 24 de 06 de 2024, de Wikipedia, la enciclopedia libre: https://es.wikipedia.org/wiki/An%C3%A1lisis_de_la_varianza
- Wikipedia, C. d. (13 de 05 de 2024). *Bayesian optimization*. Recuperado el 19 de 06 de 2024, de Wikipedia: https://en.wikipedia.org/wiki/Bayesian_optimization
- Winston, W., & Goldberg, J. (2004). *Operation research: applications and algorithms*, 4th edition. Belmont, CA.

- Wolpert, D., & Macready, W. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 1, 67-82.
- Xarn Retired, A., Macnish, C., Vijayan, K., Turlach, B., & Gupta, R. (2004). Statistical Exploratory Analysis of Genetic Algorithms. *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, 2, 2288-2295.
- Yannibelli, V., & Amandi, A. (2012). A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Systems with Applications*, 8584-8592.
- Zadeh, L. A. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8, 59-60.
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3, 257-271.
- Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary. *Evolutionary Computation*, 8, 173-195.



**ANEXO I. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA
AGENDA 2030**

**Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster: Relación del trabajo con los
Objetivos de Desarrollo Sostenible de la agenda 2030.**

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				
ODS 2. Hambre cero.				
ODS 3. Salud y bienestar.				
ODS 4. Educación de calidad.				
ODS 5. Igualdad de género.				
ODS 6. Agua limpia y saneamiento.				
ODS 7. Energía asequible y no contaminante.				
ODS 8. Trabajo decente y crecimiento económico.				
ODS 9. Industria, innovación e infraestructuras.				
ODS 10. Reducción de las desigualdades.				
ODS 11. Ciudades y comunidades sostenibles.				
ODS 12. Producción y consumo responsables.				
ODS 13. Acción por el clima.				
ODS 14. Vida submarina.				
ODS 15. Vida de ecosistemas terrestres.				
ODS 16. Paz, justicia e instituciones sólidas.				
ODS 17. Alianzas para lograr objetivos.				

Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto.

***Utilice tantas páginas como sea necesario.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

**Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster: Relación del trabajo con los
Objetivos de Desarrollo Sostenible de la agenda 2030.** (Numere la página)