



Universidad de Valladolid



ESCUELA DE INGENIERÍAS  
INDUSTRIALES

Máster en Ingeniería Industrial

**MÁSTER EN INGENIERÍA INDUSTRIAL**  
**ESCUELA DE INGENIERÍAS INDUSTRIALES**  
**UNIVERSIDAD DE VALLADOLID**

**TRABAJO FIN DE MÁSTER**

**APLICACIÓN DE ALGORITMOS GENÉTICOS MULTIOBJETIVOS  
AL PROBLEMA MULTIPROYECTO DESCENTRALIZADO:  
SPEA2 - DRCMPSP.**

Autor: D. Javier Dehesa Rodríguez  
Tutor: D. María Elena Pérez Vázquez

Valladolid, abril, 2024



**RESUMEN**

El TFM que se ha desarrollado tiene por objetivo aplicar un algoritmo genético multiobjetivo (SPEA2) al Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos (DRCMPSP). Para ello, ha sido necesario hacer un estudio tanto del algoritmo genético utilizado SPEA2, como de los objetivos más adecuados para la optimización bi-objetivo.

Se utilizará una base de ejemplos ampliamente utilizada en este campo de investigación, y los resultados obtenidos serán analizados en profundidad.

**ABSTRACT**

The developed Master's thesis aims to apply a multi-objective genetic algorithm (SPEA2) to the problem of decentralized resource-constrained multi-project scheduling. To achieve this, it was necessary to study both the genetic algorithm used, SPEA2, and the most suitable objectives for bi-objective optimization. An extensively used benchmark dataset in this research field will be employed, and the obtained results will be thoroughly analyzed.

### AGRADECIMIENTOS

Quiero expresar mi más profundo agradecimiento a todas las personas que han contribuido y de alguna manera han participado en la realización de este trabajo de investigación. En primer lugar, a mi tutora del Trabajo de Fin de Máster, María Elena Pérez Vázquez, por su orientación, paciencia y sabios consejos a lo largo de todo el proceso. Su experiencia en el campo y dedicación han sido fundamentales para el éxito de este trabajo.

Además, extendiendo mi gratitud a María Elena Pérez Vázquez, por proporcionarme acceso a los recursos y herramientas necesarios para llevar a cabo esta investigación.

También quiero agradecer a mi familia por su apoyo incondicional y por comprender los momentos en los que estuve inmerso en mi trabajo de investigación. Su amor y aliento fueron una fuente constante de motivación.

Agradezco a mis amigos y compañeros de clase por sus intercambios de ideas y discusiones que enriquecieron este trabajo.

Por último, agradezco a todas las fuentes bibliográficas, autores y académicos cuyos trabajos y publicaciones han sido fundamentales para el desarrollo de este trabajo.

Sin el apoyo de todas estas personas y recursos, este proyecto no habría sido posible. Estoy profundamente agradecido por todas las contribuciones que hicieron posible este logro.

<b>INTRODUCCION .....</b>	<b>8</b>
I ANTECEDENTES Y MOTIVACIÓN .....	10
II OBJETIVOS.....	12
III CONTENIDO DEL PROYECTO .....	14
<b>1  CONCEPTOS GENERALES EN LA PLANIFICACIÓN DE PROYECTOS.....</b>	<b>15</b>
1.1  CONCEPTOS GENERALES:.....	16
1.1.1 <i>Proyecto:</i> .....	16
1.1.2 <i>Actividades:</i> .....	17
1.1.3 <i>Recursos:</i> .....	18
1.1.4 <i>Función objetivo:</i> .....	19
1.2  MÉTODOS DE RESOLUCIÓN:.....	21
1.2.1 <i>Métodos exactos:</i> .....	21
1.2.2 <i>Métodos heurísticos:</i> .....	24
<b>2  DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO:.....</b>	<b>29</b>
2.1  CARACTERIZACIÓN DE LOS PROBLEMAS .....	29
2.1.1 <i>Dificultad de la resolución:</i> .....	30
2.1.2 <i>Número de Proyectos:</i> .....	31
2.2  DEL PSP AL DRCMPSP: .....	33
2.2.1 <i>Problema de Secuenciación de Proyectos (PSP):</i> .....	33
2.2.2 <i>Problema de Programación de Proyectos con Recursos Limitados (RCPSP):</i> .....	34
2.2.3 <i>Problema de Programación Multiproyecto con Recursos Limitados (RCMPSP):</i> .....	37
2.2.4 <i>Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos</i> <i>(DRCMPSP)</i> .....	38
2.3  FUNCIONES OBJETIVO:.....	42
2.3.1 <i>CLASIFICACIÓN:</i> .....	43
<b>3  SPEA2 (STRENGTH PARETO EVOLUTIONARY ALGORITHM 2): .....</b>	<b>61</b>
3.1  FUNDAMENTOS DEL ALGORITMO SPEA2 .....	62
3.1.1 <i>Descripción general del algoritmo</i> .....	62
3.1.2 <i>Principios clave y características</i> .....	64
3.1.3 <i>Comparación con otros enfoques multiobjetivo</i> .....	65
3.2  PROCEDIMIENTO BÁSICO DE SPEA2: .....	65
3.2.1 <i>Inicialización de la población:</i> .....	66

3.2.2	<i>Evaluación de la aptitud (fitness Assignment) de las soluciones:</i>	68
3.2.3	<i>Cálculo de dominancia y densidad de Pareto:</i>	69
3.2.4	<i>Selección de padres mediante torneo binario:</i>	71
3.2.5	<i>Operadores de cruce y mutación:</i>	72
3.2.6	<i>Generación de la siguiente población:</i>	74
3.3	<b>ACTUALIZACIÓN DEL ARCHIVO EXTERNO:</b>	74
3.3.1	<i>Selección de los mejores individuos:</i>	74
3.3.2	<i>Mantenimiento del archivo externo:</i>	75
3.4	<b>CRITERIOS DE PARADA Y SOLUCIÓN FINAL:</b>	76
3.4.1	<i>Condiciones de terminación o parada:</i>	76
3.4.2	<i>Selección de la mejor solución:</i>	77
<b>4</b>	<b>SIMULACIÓN Y RESULTADOS DEL SPEA2 EN MATLAB</b>	<b>79</b>
4.1	LIBRERÍA MPSPLIB	80
4.2	LANZAMIENTO DE SIMULACIÓN EN MATLAB	82
4.2.1	<i>Fichero AbrirD_Entrada.m</i>	82
4.2.2	<i>Fichero Principal.m</i>	83
4.2.3	<i>Funciones Objetivo</i>	83
4.3	PRESENTACIÓN Y ANALISIS DE LOS RESULTADOS OBTENIDOS	85
4.3.1	<i>Resultados de los lanzamientos</i>	86
<b>5</b>	<b>CONCLUSIONES</b>	<b>93</b>
<b>6</b>	<b>BIBLIOGRAFÍA</b>	<b>95</b>

**FIGURAS:**

FIGURA 1.1 RAMIFICACIÓN Y ACOTAMIENTO. FUENTE: ÁNGEL-MARTÍNEZ ET AL., (2019) ..... 24

FIGURA 1.2 COLONIA DE HORMIGAS. .... 26

FIGURA 2.1 EVOLUCIÓN DEL PSP AL DRCMPSP. REFERENCIA: VILLAFANEZ ET AL. (2014)..... 33

FIGURA 2.2 EJEMPLO GANTT..... 34

FIGURA 2.3 CLASIFICACIÓN RCPSP. REFERENCIA: ALVIN, 2022. .... 37

FIGURA 2.4 FUNCIONES OBJETIVO. REFERENCIA: GÓMEZ ET AL., 2022..... 43

FIGURA 2.5 IDLE RESOURCE (IR). REFERENCIA: LOVA, ET AL., 2000 ..... 54

FIGURA 3.1 DIAGRAMA DE FLUJO SPEA2..... 63

FIGURA 4.1 CODIFICACIÓN DE LOS PROBLEMAS DE LA LIBRERÍA MPSPLIB. FUENTE: (VILLAFANEZ, 2014)..... 81

FIGURA 4.2 EJEMPLO DEL FICHERO DE TERMINACIÓN .XML. .... 81

FIGURA 4.3 ABRIRD\_ENTRADA.M DATOS DE ENTRADA ..... 82

FIGURA 4.4 SELECCIÓN DE LAS FUNCIONES OBJETIVO EN ABRIRD\_ENTRADA.M..... 83

FIGURA 4.5 SPEA2 EN PROBLEMA DE 30 ACTIVIDADES, 10 PROYECTOS, NÚMERO DE REFERENCIA 4 ..... 87

FIGURA 4.6 SPEA2 EN PROBLEMA DE 30 ACTIVIDADES, 20 PROYECTOS, NÚMERO DE REFERENCIA 1 ..... 87

FIGURA 4.7 SPEA2 EN PROBLEMA DE 90 ACTIVIDADES, 10 PROYECTOS, NÚMERO DE REFERENCIA 1 ..... 88

FIGURA 4.8 SPEA2 EN PROBLEMA DE 120 ACTIVIDADES, 10 PROYECTOS, NÚMERO DE REFERENCIA 4 ..... 88





## INTRODUCCION

En el ámbito de la gestión de proyectos, el Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos (DRCMPSP) ha sido objeto de una atención significativa debido a su complejidad y relevancia en diversos contextos empresariales (Sindhya et al., 2007). El DRCMPSP implica la asignación eficiente de recursos limitados a múltiples proyectos interrelacionados, considerando restricciones de tiempo y capacidad de los recursos, con el objetivo de optimizar múltiples objetivos simultáneamente, como minimizar el tiempo total de finalización y reducir los costos asociados.

La planificación de proyectos es esencial para las organizaciones que buscan alcanzar sus objetivos estratégicos y ejecutar eficientemente múltiples proyectos. Sin embargo, en entornos con recursos limitados y descentralizados, la asignación óptima de recursos a los proyectos se vuelve compleja debido a la interdependencia de las tareas, la competencia por los recursos y las restricciones de tiempo (Sindhya et al., 2007). En este sentido, el DRCMPSP plantea desafíos adicionales para los gestores de proyectos, ya que deben considerar la asignación adecuada de recursos para maximizar el rendimiento global de múltiples proyectos.

La optimización multiobjetivo surge como una técnica prometedora para abordar problemas complejos de toma de decisiones en los que existen múltiples objetivos en conflicto. En el caso del DRCMPSP, la optimización multiobjetivo permite encontrar soluciones que equilibren los diferentes objetivos ofreciendo alternativas de asignación de recursos que se ajusten a las necesidades y restricciones del proyecto (Deb et al., 2002).

A lo largo de los últimos años, los algoritmos genéticos multiobjetivo han ido irrumpiendo como herramientas eficaces para abordar este tipo de problemas de optimización con múltiples objetivos. Estos algoritmos facilitan y tienen la capacidad de generar y mejorar soluciones que representan un compromiso entre diferentes objetivos, permitiendo tomar decisiones habiendo explorado y analizado el espacio de soluciones óptimas y no dominadas.

En este Trabajo de Fin de Máster, se propone aplicar el algoritmo genético multiobjetivo SPEA2 (Strength Pareto Evolutionary Algorithm 2) al problema DRCMPSP con el propósito de encontrar

las diferentes soluciones eficientes y equilibradas en términos de los objetivos de optimización establecidos. SPEA2 es un algoritmo genético multiobjetivo que utiliza una función de dominancia basada en la fuerza y la densidad de las soluciones para guiar la búsqueda hacia soluciones no dominadas y bien distribuidas.

El objetivo principal de este TFM es realizar un estudio exhaustivo de los objetivos más adecuados para la optimización bi-objetivo en el contexto del DRCMPSP y evaluar el desempeño del algoritmo genético multiobjetivo SPEA2 en algunas de las funciones objetivo encontradas. Para ello, se utilizará una base de ejemplos ampliamente utilizada en este campo de investigación, y los resultados obtenidos serán analizados para determinar la efectividad y eficiencia del enfoque propuesto.

Mediante este estudio, se espera contribuir al avance de la investigación en la optimización del DRCMPSP, proporcionando conocimientos sobre la aplicabilidad y el rendimiento del algoritmo genético multiobjetivo SPEA2 en este contexto. Además, los resultados obtenidos pueden ser de utilidad para los profesionales involucrados en la gestión de proyectos, brindándoles una herramienta efectiva para la toma de decisiones en entornos con múltiples proyectos y recursos limitados.

## I ANTECEDENTES Y MOTIVACIÓN

El Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos (DRCMPSP) ha sido objeto de estudio en el ámbito de la planificación de proyectos debido a su complejidad, relevancia y utilidad en diversos contextos empresariales. En este problema, se deben asignar de manera eficiente los recursos locales y globales a varios proyectos interrelacionados, teniendo en cuenta restricciones de tiempo y capacidad de los recursos, y optimizando múltiples objetivos simultáneamente, como minimizar el tiempo total de finalización de los proyectos y reducir los costes asociados a ellos.

Un amplio abanico de enfoques se ha propuesto para abordar el DRCMPSP a lo largo de su estudio en el tiempo, pero la optimización multiobjetivo ha demostrado ser una técnica prometedora para lidiar con las dificultades y desafíos inherentes de este problema. Al considerar múltiples objetivos en conflicto, la optimización multiobjetivo permite encontrar soluciones que representen compromisos entre estos objetivos, lo que ayuda a los gestores de proyectos a tomar decisiones más informadas y equilibradas haciendo que estas resulten ser más adecuadas.

En la historia se han aplicado y desarrollado diferentes algoritmos de optimización multiobjetivo para el DRCMPSP, incluyendo algoritmos genéticos, algoritmos basados en colonias de hormigas y algoritmos basados en enjambres de partículas, etc. Estos enfoques han demostrado su eficacia para garantizar la búsqueda de soluciones de una alta calidad para el problema. Sin embargo, es necesario avanzar, seguir explorando y perfeccionando las diversas técnicas existentes para obtener resultados aún óptimos y precisos.

La motivación de este trabajo radica en la necesidad de abordar de manera eficiente y efectiva el problema del DRCMPSP en entornos con recursos limitados y descentralizados, situación habitual en contextos empresariales. La optimización de la asignación de recursos en múltiples proyectos es esencial para garantizar la eficiencia y el éxito de la gestión de proyectos, así como su finalización en tiempo y coste establecido. Al aplicar el algoritmo genético multiobjetivo SPEA2 al DRCMPSP, se busca obtener soluciones de una alta calidad y equilibradas en términos de los objetivos de optimización, como minimizar el tiempo total de finalización y reducir los costos asociados.

La aplicación de un enfoque basado en algoritmos genéticos multiobjetivo tiene varias ventajas. Estos algoritmos son capaces de explorar el espacio de soluciones de manera exhaustiva y generar soluciones no dominadas y bien distribuidas en el frente de Pareto. Además, permiten a los gestores de proyectos considerar diferentes escenarios y evaluar el impacto de las decisiones de asignación de recursos en múltiples objetivos, lo que les brinda una visión más completa para la toma de decisiones.

Al contribuir a la investigación en la optimización del DRCMPSP y evaluar el rendimiento del algoritmo genético multiobjetivo SPEA2 en este contexto, este trabajo busca proporcionar conocimientos valiosos y herramientas prácticas tanto para la comunidad académica como para los profesionales de la gestión de proyectos para situaciones futuras. Los resultados obtenidos podrán mejorar la eficiencia y el rendimiento de la asignación de recursos en proyectos reales, ayudando a las organizaciones a tomar decisiones más informadas y estratégicas en sus actividades de gestión de proyectos.

## II OBJETIVOS

El objetivo principal de este trabajo es aplicar el algoritmo genético multiobjetivo SPEA2 al Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos (DRCMPSP). Esto implica adaptar el algoritmo a las características y restricciones del problema y evaluar su desempeño en términos de la calidad de las soluciones encontradas.

Además del objetivo principal, se plantean objetivos específicos que contribuyen al desarrollo y análisis de la aplicación de SPEA2 al DRCMPSP. Estos objetivos incluyen:

Por un lado, realizar un estudio exhaustivo de los conceptos generales en la planificación de proyectos, así como, estudiar los objetivos más adecuados para la optimización bi-objetivo en el contexto del DRCMPSP. Se analizarán los objetivos relevantes para el problema, como la minimización del tiempo total de finalización y la reducción de los costos asociados, con el fin de definir los criterios de optimización adecuados.

Adaptar el algoritmo genético multiobjetivo SPEA2 al problema del DRCMPSP. Esto implica comprender en detalle el funcionamiento de SPEA2 y realizar las modificaciones necesarias para que se ajuste de manera adecuada a las características del problema. Se considerarán aspectos como las restricciones de recursos, la interdependencia de tareas y la asignación descentralizada de recursos.

Utilizar una base de ejemplos ampliamente utilizada en el campo de investigación del DRCMPSP para evaluar el desempeño del algoritmo genético multiobjetivo adaptado. Se realizarán experimentos utilizando esta base de ejemplos para analizar la calidad de las soluciones encontradas y evaluar la convergencia del algoritmo.

Analizar los resultados en términos de la calidad de las soluciones y la eficiencia del algoritmo genético multiobjetivo SPEA2 aplicado al DRCMPSP. Se identificarán las fortalezas y debilidades del enfoque propuesto y se destacarán las contribuciones y avances realizados en relación con investigaciones previas.

Identificar posibles mejoras o extensiones para futuros trabajos en el campo de la optimización del DRCMPSP con algoritmos genéticos multiobjetivo. Se propondrán

recomendaciones y líneas de investigación adicionales que puedan mejorar aún más la eficiencia y efectividad de la asignación de recursos en proyectos con múltiples objetivos y restricciones.

Estos objetivos permitirán generar conocimientos valiosos en la aplicación de algoritmos genéticos multiobjetivo al problema del DRCMPSP, y contribuirán al avance de la investigación en el campo de la optimización de la asignación de recursos en proyectos con múltiples objetivos y restricciones.

### **III CONTENIDO DEL PROYECTO**

Siguiendo los objetivos anteriores, el documento se organizará como sigue.

En el primer capítulo se presentarán los diferentes conceptos generales relacionados con la planificación de proyectos, de igual manera se explicarán en detalle los diferentes métodos de resolución del problema de programación de actividades.

En el segundo capítulo se presentará la caracterización de estos problemas su evolución a lo largo de los años y se definirán las diferentes funciones objetivos estudiadas para el problema DRCMPSP.

En el tercer capítulo se presentará el algoritmo genético objeto de estudio, el SPEA2, para ello, se describirá en este algoritmo, como funciona y los pasos que se llevan a cabo hasta alcanzar la planificación óptima deseada.

En el cuarto capítulo, se presentarán las simulaciones realizadas de este algoritmo genético SPEA2 en el programa MATLAB tomando como ejemplos los problemas DRCMPSP de la librería MPSPLib. Para ello, se han seleccionado un par de funciones objetivo para llevar a cabo estas simulaciones, el Idle Resource (IR) y el Average Delay Project (APD).

Finalmente, en el quinto capítulo, se presentará la conclusión, tanto del algoritmo genético y de los diferentes resultados obtenidos en la simulación, como del trabajo realizado en general.

### 1 CONCEPTOS GENERALES EN LA PLANIFICACIÓN DE PROYECTOS.

Los diferentes proyectos a los que nos enfrentamos en nuestro día a día están compuestos de una serie de procesos, es decir, una serie de tareas a llevar a cabo. El éxito de los distintos proyectos dependerá del modo de ejecución o desarrollo y de su terminación (fecha más tardía de finalización) del mismo. Entre los procesos encontramos los procesos de iniciación, planificación ejecución, control y cierre, a los cuales se les asigna unos tiempos de inicio y fin. Estos tiempos serán de interés para los responsables de la gestión de los distintos proyectos. Por otro lado, el tiempo de ejecución de cada proyecto es crítico, pues este está directamente vinculado con los costos de todas las actividades individuales. (Pérez et al., 2016).

Uno de los problemas que encontramos en los proyectos es el orden de ejecución del proyecto, es decir, el término denominado Scheduling o Secuenciación. El objetivo de este problema es el de generar un orden de ejecución de las actividades del proyecto, identificando una serie de actividades, como actividades principales que serán las que conforme la ruta o camino crítico. Por otro lado, no solo se debe conocer la ruta crítica, sino que tendremos que ejecutar dichas actividades gestionando unos recursos de una forma eficiente. Como consecuencia, al trabajar con las actividades de un proyecto y con sus respectivos recursos, pasa a tomar el nombre de Problema de Secuenciación de Proyectos con Recursos limitados (RCPSP, Resources Constrained Project Scheduling Problem).

De esta manera, el problema está caracterizado por la estimación y definición de la duración y el consumo de los distintos recursos para cada actividad. En cuanto a los recursos utilizados en la actividad, se les atribuye el nombre de recursos renovables ya que estos pueden volver a utilizarse para otra actividad una vez que se hayan terminado su utilización. Algunos ejemplos de recursos renovables son las materias y la mano de obra. El inconveniente de este tipo de recursos es que se ven restringidos por necesidades de presupuesto.

Por lo tanto, desde un punto de vista más amplio el problema de planificación de recursos se puede considerar como un problema de combinación combinatoria, pues pueden ser codificadas



distintas soluciones factibles a través de la combinación de las distintas actividades que constituyen un proyecto, es decir, mediante el intercambio de la ordenación o posicionamiento de la ejecución de las actividades.

A continuación, explicaremos en detalle, todos los fundamentos conceptos claves para un mayor entendimiento todo lo rodea e interviene con la planificación de un proyecto.

### 1.1 CONCEPTOS GENERALES:

#### 1.1.1 Proyecto:

Un proyecto se define como un conjunto de actividades interrelacionadas y coordinadas que se llevan a cabo de manera planificada. Su objetivo es producir bienes o servicios específicos que satisfagan necesidades o resuelvan problemas, dentro de un presupuesto y un plazo de tiempo establecidos (Coll, 2020). En definitiva, un proyecto implica la planificación y ejecución de acciones para lograr un objetivo determinado. Es esencial incluir en el proyecto una planificación detallada de actividades, así como la asignación de los recursos y medios necesarios para su realización.

Es importante destacar que los proyectos se pueden clasificar de diversas formas, según su financiación (privados, públicos y mixtos), su contenido, su complejidad o su finalidad. Además, cada tipo de proyecto requiere la gestión de elementos específicos que lo conforman. Algunos de los parámetros clave que definen un proyecto son:

- **Alcance:** Define los límites y las metas del proyecto, identificando qué se incluye y qué se excluye del trabajo a realizar.
- **Tiempo:** Implica la planificación y gestión de las actividades en función de los plazos establecidos para su ejecución.
- **Planificación de actividades:** Consiste en la secuenciación y organización de las actividades del proyecto, estableciendo las dependencias y la duración de cada una.

## CONCEPTOS GENERALES EN LA PLANIFICACIÓN DE PROYECTOS

---

- **Coste:** Comprende la estimación, el control y la gestión de los recursos financieros necesarios para la ejecución del proyecto.
- **Calidad:** Implica la definición de estándares de calidad y la implementación de medidas para asegurar que los entregables cumplan con dichos estándares.
- **Gestión de interesados:** Involucra la identificación, el análisis y la comunicación efectiva con todas las partes interesadas en el proyecto.
- **Comunicaciones:** Establece los canales y las estrategias de comunicación necesarios para mantener informados a los interesados y al equipo del proyecto.
- **Gestión del riesgo:** Consiste en la identificación, el análisis, la respuesta y el monitoreo de los riesgos potenciales que puedan afectar al proyecto.
- **Gestión del equipo de trabajo:** Implica la selección, el liderazgo y la coordinación de los miembros del equipo para garantizar un desempeño efectivo y colaborativo.
- **Abastecimiento del material necesario:** Involucra la adquisición y gestión de los recursos materiales necesarios para la realización del proyecto.

En resumen, un proyecto es una secuenciación bien definida de actividades que se planifican y ejecutan con el propósito de alcanzar un objetivo específico. El director del proyecto es responsable de llevarlo a cabo, basándose en parámetros establecidos, como el tiempo, el costo y los recursos, siempre teniendo en cuenta el nivel de calidad requerido (Narváez y Saltos, 2010).

### 1.1.2 Actividades:

En un proyecto, las tareas o actividades son elementos clave que deben ser realizados de manera definida y planificada. Cada actividad tiene asignado un tiempo específico para su ejecución, es decir, se establece una fecha límite para completarla. Algunas actividades pueden ser realizadas de forma individual y con una ejecución específica, mientras que otras pueden ser ejecutadas simultáneamente (Narváez y Saltos, 2010).

Cada tarea puede requerir la definición de objetivos intermedios, que se utilizan para controlar el progreso del proyecto antes de su finalización. Estos objetivos intermedios ayudan a

evaluar el avance y asegurarse de que el proyecto está en camino. Además, cada actividad requiere de la disponibilidad de recursos específicos, como personas, herramientas, equipos e instalaciones. La asignación y gestión adecuada de estos recursos es esencial para garantizar el cumplimiento de las actividades dentro de los plazos establecidos (Narváez y Saltos, 2010).

La identificación y secuenciación de las tareas es un paso fundamental en la planificación de un proyecto. La duración de cada tarea es el tiempo necesario para completarla. Al crear la lista de tareas, también se deben identificar las duraciones estimadas para cada una. La duración más fiable se obtiene a partir de datos históricos recopilados por la empresa. En caso de no contar con datos históricos, la siguiente mejor estimación proviene de la participación de personas con experiencia previa en tareas similares.

En resumen, las actividades o tareas son elementos específicos dentro de un proyecto que deben ser planificados y ejecutados. Cada actividad tiene una duración estimada y puede requerir la definición de objetivos intermedios. La identificación y secuenciación adecuadas de las tareas son fundamentales para la planificación y el control efectivo del proyecto. Además, es esencial asignar los recursos necesarios para llevar a cabo las actividades de manera exitosa.

### **1.1.3 Recursos:**

Los recursos, generalmente, son los elementos productivos necesarios para llevar a cabo una actividad y no se consumen ni se incorporan al producto final de la misma. Al finalizar una tarea, estos recursos quedan disponibles para ser utilizados en otras actividades o proyectos. Algunos ejemplos de recursos son la mano de obra, los equipos y las herramientas (Narváez y Saltos, 2010).

La disponibilidad de un recurso se refiere a la cantidad de dicho recurso que puede ser utilizada en un momento específico. Cada actividad requiere de una cierta cantidad de recursos para su procesamiento, la cual puede ser constante o variable a lo largo del tiempo.

Por todo esto, en cualquier proyecto podemos encontrar y gestionar cualquiera de estos diferentes tipos de recursos:

- **Recursos renovables:** Son aquellos que se renuevan en cada periodo, como las horas de trabajo de los operarios, las máquinas, las herramientas o los equipos de trabajo.
- **Recursos no renovables:** Son aquellos que tienen una cantidad total disponible a lo largo del proyecto y que se consumen a medida que avanza. Algunos ejemplos son el dinero, la materia prima o la energía.
- **Recursos doblemente limitados:** Son aquellos que tienen una limitación tanto en la cantidad total disponible en el proyecto como en cada periodo de tiempo.
- **Recursos parcialmente (no) renovables:** Son aquellos cuya disponibilidad se renueva en intervalos de tiempo específicos.

En resumen, los recursos son los factores productivos necesarios para realizar una actividad, pudiendo ser considerados como renovables o no renovables, y su disponibilidad puede variar en cada periodo. La gestión adecuada de los recursos es esencial para garantizar el éxito del proyecto.

### 1.1.4 **Función objetivo:**

La función objetivo en la planificación de proyectos se refiere al propósito que se desea lograr mediante la programación de las actividades. El tipo de función objetivo puede variar según el contexto en el que se esté llevando a cabo el proyecto. En el contexto del Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos (DRCMPSP), las funciones objetivo juegan un papel crucial en la planificación y optimización de los proyectos involucrados. El DRCMPSP implica la gestión simultánea de varios proyectos, donde cada proyecto tiene sus propias limitaciones de recursos y decisiones de programación independientes.

Las funciones objetivo en el DRCMPSP están diseñadas para buscar soluciones óptimas que equilibren los objetivos y restricciones de los diferentes proyectos. Algunas de las funciones objetivo más comunes en este contexto incluyen:

- **Minimización del tiempo total de finalización de los proyectos:** Esta función objetivo busca reducir la duración total de todos los proyectos en conjunto. El objetivo es completar todos

los proyectos en el menor tiempo posible, teniendo en cuenta las limitaciones de recursos y las dependencias entre las actividades.

- Minimización del retraso total de los proyectos: Esta función objetivo se centra en minimizar el retraso acumulado en la finalización de los proyectos. El objetivo es evitar o reducir los retrasos en la finalización de las actividades y proyectos, considerando la disponibilidad de recursos y las restricciones temporales.
- Minimización del costo total de los proyectos: Esta función objetivo busca minimizar los costos asociados a la ejecución de los proyectos, considerando los recursos limitados y los gastos relacionados. El objetivo es encontrar una asignación de recursos y una programación de actividades que optimice los recursos financieros disponibles.

Según Villafañez et al. (2019), la función objetivo más útil para una empresa genérica sería la minimización de la duración total (TMS), ya que se buscaría reducir la duración en lugar de garantizar que todos los proyectos tengan un retraso similar.

Además de estas funciones objetivo, existen otras que también pueden considerarse, como la minimización del retraso total del proyecto, la minimización de la penalización por retraso del proyecto o programa, la minimización del coste total del proyecto o la maximización de la utilización de recursos, entre muchas otras (Browning y Yassine, 2010). De igual manera pueden considerarse otros criterios específicos según las necesidades y prioridades de cada proyecto y organización. Estos pueden incluir la maximización de la calidad, la optimización del uso de recursos críticos, la minimización de los riesgos o la satisfacción de los stakeholders involucrados.

Es importante destacar que las funciones objetivo pueden variar según el contexto y los objetivos específicos del DRCMPSP. La elección de las funciones objetivo más adecuadas depende de las metas y restricciones particulares de cada proyecto y de la importancia relativa que se le dé a cada objetivo en el proceso de toma de decisiones. Este concepto se desarrollará más en detalle en el capítulo 2.

## 1.2 MÉTODOS DE RESOLUCIÓN:

La resolución de problemas de programación de actividades ha sido objeto de extensos estudios desde la década de 1950, cuando surgieron las primeras necesidades de secuenciar actividades en proyectos para lograr una programación viable. A lo largo de los años, estos métodos han experimentado una evolución y perfeccionamiento significativos, desde sus primeras versiones como GANTT, PERT, CPM y ROY, hasta los métodos más actuales basados en heurísticas y metaheurísticas.

Es importante destacar que hasta el día de hoy no se ha identificado un método que sea superior a los demás en todos los problemas planteados. La eficacia de los métodos de resolución depende en gran medida de las características específicas del problema abordado. Algunos métodos pueden resultar más eficientes en ciertos escenarios, mientras que otros pueden ser más adecuados para diferentes tipos de problemas.

En su publicación, Morillo, et al., (2014) proporcionan una revisión exhaustiva de los métodos de resolución en la programación de actividades. A continuación, se presenta una clasificación y descripción de los principales métodos, atendiendo a las técnicas utilizadas.

### 1.2.1 Métodos exactos:

Los métodos exactos comprenden aquellos procedimientos que utilizan técnicas analíticas o matemáticas con el fin de lograr la convergencia hacia una solución óptima. Sin embargo, su utilidad se limita a condiciones específicas, como la continuidad del problema, un espacio de búsqueda reducido y condiciones de linealidad. La existencia de múltiples métodos exactos para buscar soluciones a estos problemas se debe a que ninguno de ellos es lo suficientemente robusto por sí solo. A continuación, se presentan algunos de estos métodos, destacados por su importancia en la evolución de los métodos de resolución.

### 1.2.1.1 Búsqueda exhaustiva:

La búsqueda exhaustiva es un método de resolución utilizado en problemas de programación de actividades que se caracteriza por su exhaustividad y simplicidad. Este enfoque consiste en generar todas las posibles soluciones factibles dentro del espacio de búsqueda y evaluar cada una de ellas para encontrar la solución óptima.

La principal ventaja de la búsqueda exhaustiva radica en su capacidad para abordar cualquier tipo de problema, ya que no se basa en suposiciones o restricciones específicas. Además, garantiza que no se pasen por alto soluciones potenciales.

Sin embargo, el principal inconveniente de este método es el tiempo de procesamiento requerido, especialmente cuando se enfrenta a un gran número de actividades y proyectos. La explosión combinatoria generada puede hacer que la búsqueda exhaustiva sea inviable en términos de eficiencia.

Es importante tener en cuenta que la utilidad de la búsqueda exhaustiva se limita a los denominados "problemas P", cuyo tiempo de procesamiento crece de manera polinómica. En problemas más complejos, la utilización de este método se vuelve cada vez más impracticable debido al aumento exponencial del espacio de búsqueda. (Álvarez, 2021.)

### 1.2.1.2 Programación lineal:

La programación lineal se basa en la hipótesis de que todas las funciones de un problema son combinaciones lineales de las variables de decisión, donde la función objetivo representa el resultado que se desea maximizar o minimizar. Las restricciones, por otro lado, representan las limitaciones de recursos y las relaciones de precedencia que existen en el problema. Estas restricciones definen el espacio de búsqueda de las soluciones óptimas.

Sin embargo, es importante tener en cuenta que la programación lineal no se considera eficaz para un número grande de actividades. Según Valls, et al., (2005), este método solo se considera eficaz en problemas con menos de 60 actividades. Esto se debe a que el tiempo requerido para la

búsqueda exhaustiva de soluciones se vuelve prohibitivo, incluso con el uso de ordenadores modernos.

La programación lineal ha demostrado ser una herramienta valiosa y versátil en la investigación de operaciones, especialmente después del desarrollo del método símplex por Dantzig en 1947. Se ha utilizado con éxito en diversos problemas, incluido el RCPS. Sin embargo, para problemas más complejos o con un gran número de actividades, se requieren métodos alternativos que sean más eficientes en términos de tiempo de procesamiento.

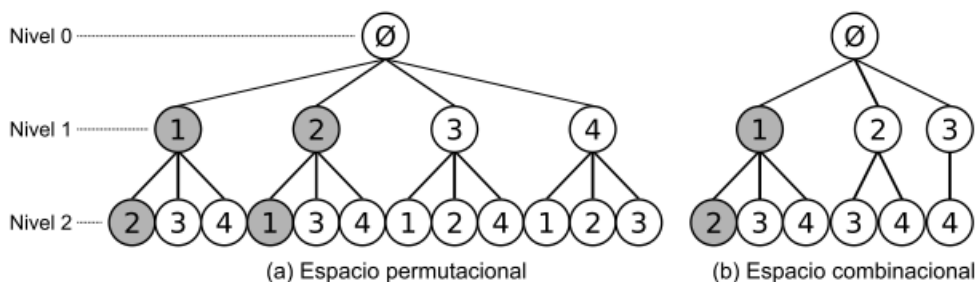
### *1.2.1.3 Ramificación y acotamiento:*

Este método exacto, conocido como "ramificación y acotamiento", se considera igual de robusto que la búsqueda exhaustiva, pero más eficiente en términos de tiempo de ejecución. Su enfoque se basa en dividir el espacio factible de posibles soluciones en subespacios más pequeños, con el objetivo de determinar dónde es más probable que se encuentre la solución óptima. Para lograr esto, se aplica una regla determinada que guía la búsqueda y restringe el espacio de exploración.

La estrategia de ramificación y acotamiento permite reducir el número total de posibles soluciones que deben ser evaluadas. Al dividir el espacio de búsqueda y descartar ramas que no conducen a soluciones óptimas, se acota el conjunto de soluciones consideradas, lo que permite un proceso de búsqueda más eficiente y efectivo.



En la figura 1.1 se muestra un ejemplo ilustrativo de cómo se aplica el método de ramificación y acotamiento, donde se puede observar cómo se realiza la subdivisión del espacio de soluciones y se enfoca la búsqueda en las áreas más prometedoras.



*Figura 1.1 Ramificación y acotamiento. Fuente: Ángel-Martínez et al., (2019)*

### 1.2.2 Métodos heurísticos:

Los métodos heurísticos se caracterizan por encontrar soluciones válidas de buena calidad en un tiempo de procesamiento razonable. Estos métodos surgieron como una alternativa a los métodos exactos, los cuales, aunque logran obtener resultados óptimos, requieren un tiempo considerable debido a la explosión combinatoria generada por proyectos con numerosas actividades. Aunque los métodos heurísticos no pueden garantizar la optimalidad de la solución obtenida, han sido ampliamente utilizados y estudiados debido a su utilidad en problemas donde los métodos exactos no son factibles.

Dado que no es posible determinar si los resultados obtenidos mediante heurísticas son óptimos, es necesario contar con métodos para comparar su desempeño. Para ello, se utilizan “librerías o bibliotecas” de problemas de programación de actividades que permiten subir los resultados de las heurísticas y realizar comparaciones. En este trabajo, se empleará la librería MPSPLib (Homberger, 2007) como referencia para evaluar las heurísticas utilizadas.

Los métodos heurísticos se dividen principalmente en dos categorías: métodos heurísticos primitivos, como técnicas basadas en reglas de prioridad, procedimientos de ramificación y

acotamiento truncados, conceptos de arcos disyuntivos y búsqueda local; y métodos heurísticos basados en metaheurísticas, como el Recocido Simulado o Simulated Annealing, búsqueda tabú, algoritmos genéticos, programación evolutiva y colonia de hormigas. Dado que el método heurístico basado en reglas de prioridad es relevante para este trabajo, se explicará detalladamente, mientras que para más información sobre los métodos heurísticos primitivos se recomienda consultar la publicación de Morillo et al., (2014). También se brindará una breve descripción de los métodos basados en metaheurísticas debido a su utilidad y buenos resultados.

### *1.2.2.1 Métodos heurísticos basados en técnicas metaheurísticas*

Los métodos heurísticos basados en técnicas metaheurísticas ofrecen enfoques innovadores para abordar problemas de programación de actividades. Las diferentes técnicas proporcionan la capacidad de explorar soluciones factibles y emplear estrategias para dirigir la búsqueda hacia regiones más prometedoras del espacio de soluciones.

- **Recocido Simulado (Simulated Annealing):** Este método parte de una solución aleatoria y utiliza un esquema de enfriamiento gradual para explorar soluciones vecinas. A medida que disminuye la temperatura, la búsqueda se vuelve más selectiva y se evita quedar atrapado en óptimos locales. Este enfoque se ha demostrado eficaz en diversos problemas de programación de actividades.
- **Búsqueda Tabú (Tabú Search):** Este método mantiene una lista de soluciones "tabú" que no se pueden volver a explorar, lo que garantiza una mayor diversificación de la búsqueda y ayuda a evitar la convergencia prematura hacia las soluciones.

- **Optimización de la Colonia de Hormigas (Ant Colony Optimization):** Este método se basa en la forma en que las hormigas encuentran el camino más corto hacia el alimento mediante la comunicación a través de feromonas como se representa en la figura 1.2. En el contexto de la programación de actividades, se utiliza para determinar una solución inicial y se asigna un mayor peso a las soluciones que

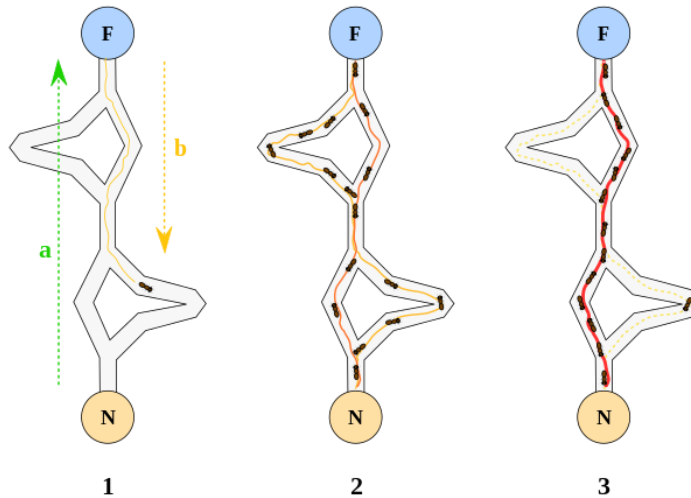


Figura 1.2 Colonia de Hormigas.

Referencia: Álvarez-Campana, P. (2021).

conducen a resultados óptimos.

- **Algoritmos Genéticos (Genetic Algorithms):** Esta técnica se inspira en la evolución biológica y utiliza operadores genéticos, como mutación y recombinación, para generar nuevas soluciones a partir de una población inicial. A través de la selección de los individuos más aptos, se busca mejorar la calidad de las soluciones en cada iteración.

Este método heurístico se basa en la teoría neo-darwiniana de la evolución y utiliza algoritmos genéticos para la optimización. Cada solución se representa como un individuo con un valor de aptitud, que corresponde a su rendimiento en la función objetivo. Las generaciones de soluciones evolucionan mediante la selección y operadores genéticos, lo que permite realizar una búsqueda en paralelo.

La implementación de este método se basa en la representación de las soluciones mediante un código genético en lenguaje computacional. Existen diversas formas comunes de codificar las soluciones: la representación binaria, la representación en valor real, etc. Estas diferencias influyen en la definición de los operadores genéticos. El algoritmo comienza con una población inicial creada aleatoriamente. A partir de esta población, se seleccionan los padres para la siguiente generación. Por lo general, se eligen los individuos con menor valor de aptitud (en problemas de minimización) con alta probabilidad. Sin embargo, también se permite la inclusión de soluciones de mayor valor (soluciones subóptimas) con menor probabilidad para explorar diferentes regiones del espacio factible. (Morillo et al., 2014)

El operador genético más importante en este método es el cruce, mientras que la mutación desempeña un papel secundario al introducir cambios aleatorios para fomentar la exploración en la búsqueda. Se emparejan aleatoriamente los padres y se aplican los operadores genéticos definidos previamente para generar nuevas soluciones (hijos). Luego, se utiliza un criterio de selección para formar la siguiente generación. Este proceso se repite hasta que se cumpla un criterio de parada.

Los algoritmos genéticos son ampliamente utilizados para resolver problemas de programación de proyectos, y existen diversos trabajos que detallan su implementación en este tipo de problemas.

Para este trabajo se ha decidido utilizar este método heurístico, en particular se ha llevado a cabo el uso del algoritmo genético SPEA2 para abordar el problema DRCMPSP por su eficacia en la optimización multiobjetivo, su capacidad para manejar la toma de decisiones descentralizada, la gestión de restricciones de recursos y su robustez en entornos del mundo real. La flexibilidad y adaptabilidad de SPEA2, junto con su capacidad para explorar de manera efectiva el espacio de soluciones, lo convierten en una elección adecuada para este problema complejo de programación multiproyecto

## CONCEPTOS GENERALES EN LA PLANIFICACIÓN DE PROYECTOS

---

En resumen, los métodos heurísticos basados en técnicas metaheurísticas brindan enfoques flexibles y eficientes para abordar problemas de programación de actividades, permitiendo una exploración más amplia del espacio de soluciones y evitando quedar atrapados en soluciones subóptimas. Cada método tiene sus propias características y aplicaciones, y su elección depende del problema específico que se esté abordando

## 2 DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO:

En la actualidad, todos los proyectos están constituidos por una serie de procesos o actividades a realizar, y requieren la asignación adecuada de recursos para alcanzar sus objetivos. El éxito de la ejecución de estos proyectos dependerá en gran medida de la eficiente ejecución y organización de dichas actividades y recursos.

Los problemas de secuenciación de proyectos representan un área crucial en la gestión y planificación eficiente de múltiples proyectos. En este contexto, se busca determinar el orden óptimo de ejecución de las actividades de cada proyecto, teniendo en cuenta las restricciones de recursos y las dependencias entre tareas, con el objetivo de minimizar la duración total de la cartera de proyectos o maximizar otros objetivos relevantes. Estos problemas presentan una complejidad considerable debido a la interacción entre las diferentes actividades y los recursos compartidos, lo que requiere enfoques avanzados de optimización y algoritmos inteligentes para encontrar soluciones viables y efectivas. En este capítulo, se abordará la problemática de secuenciación de proyectos, explorando diversas técnicas y herramientas de optimización para obtener planes de ejecución que mejoren la eficiencia y el rendimiento general del conjunto de proyectos.

### 2.1 CARACTERIZACIÓN DE LOS PROBLEMAS

La programación de proyectos es un proceso fundamental para establecer las fechas de inicio y finalización de las diversas actividades que componen un proyecto. Sin embargo, esta programación enfrenta una serie de desafíos debido a las restricciones asociadas. Una de las principales restricciones es la interrelación de las actividades, donde las dependencias de precedencia deben ser respetadas. En este sentido, las actividades deben cumplir ciertas secuencias, lo que, junto con su duración, determina la programación completa del proyecto. Los métodos tradicionales basados en grafos, como CPM, PERT, ROY y GANTT, han sido utilizados para estimar las programaciones iniciales; no obstante, se ha evidenciado que estos enfoques no siempre reflejan la realidad del entorno de los proyectos. (Villafañez et al., 2018)

Otra restricción relevante es la limitación de recursos, aspecto que inicialmente no fue considerado por los métodos basados en grafos. En la práctica, las actividades utilizan o consumen recursos, los cuales pueden ser escasos debido a su costo, disponibilidad u ocupación (Araújo et al., 2010). Por ende, las programaciones obtenidas a través de los enfoques gráficos a menudo resultan inviables en proyectos reales. Para superar este inconveniente, se requiere desarrollar métodos de resolución que contemplen las restricciones de recursos existentes.

Cabe destacar que pueden surgir situaciones en las que varios proyectos compartan los mismos recursos, lo que puede conducir a una sobreasignación de dichos recursos y hacer inviable la realización de los proyectos según sus estimaciones iniciales. Estos desafíos se consideran como una variante del problema clásico de programación de actividades con restricciones de recursos, pero para múltiples proyectos. La introducción de varios proyectos aumenta la complejidad del problema y disminuye la probabilidad de encontrar soluciones adecuadas mediante métodos de resolución exactos la cual ha sido presentado en la sección 1.3.1.

En particular, cuando dos actividades de diferentes proyectos comparten recursos y no existen recursos suficientes para programar ambas, surge la cuestión de qué proyecto debe recibir preferencia. En la sección 1.2.2, se presentarán dos enfoques para abordar este problema: el enfoque centralizado y el descentralizado.

### **2.1.1 Dificultad de la resolución:**

Cuando enfrentamos la problemática de programar las actividades de múltiples proyectos con recursos limitados, nos encontramos con desafiantes problemas combinatorios que pertenecen a la clase NP-Hard (NP-Difícil) según la teoría computacional de Álvarez-Campana, (2021). Según Los problemas NP-Hard se caracterizan por su complejidad intratable en términos de tiempo de resolución, lo que significa que no existe un algoritmo eficiente que pueda encontrar la solución óptima en un tiempo razonable (Bartusch et., 1988). Por lo tanto, encontrar la planificación óptima de las actividades en este contexto se convierte en una tarea altamente compleja y prácticamente imposible de resolver en un tiempo adecuado.

### 2.1.2 Número de Proyectos:

El problema de programación de actividades con recursos limitados (RCPSP) ha sido tradicionalmente abordado para proyectos individuales. En este contexto, la asignación de recursos se centraba exclusivamente en las actividades específicas del proyecto. Sin embargo, cuando se plantean problemas con dos o más proyectos, la complejidad aumenta significativamente, especialmente si hay recursos globales que pueden ser compartidos entre actividades de diferentes proyectos. En este sentido, el número de proyectos se convierte en un factor relevante para caracterizar estos problemas y determinar el enfoque adecuado para abordarlos.

En los problemas multiproyecto, se han definido dos enfoques distintos según el responsable de la toma de decisiones en la programación de actividades:

- **Enfoque Centralizado:** En este enfoque, todas las actividades de los diferentes proyectos se agrupan en un "macro proyecto". El macro proyecto se forma mediante la combinación de todas las actividades y sus dependencias de los proyectos individuales. Además, se añaden actividades ficticias con duración cero que marcan el inicio y el final del macro proyecto. La responsabilidad de la toma de decisiones recae en un único ente, comúnmente el director del programa de proyectos, quien debe coordinar y optimizar la programación global de todas las actividades.
- **Enfoque Descentralizado:** En este enfoque, cada proyecto conserva su autonomía para programar sus actividades, siempre y cuando se respeten las restricciones de recursos globales y particulares. Los directores de cada proyecto son los responsables de tomar decisiones para la programación de actividades dentro de sus respectivos proyectos. Esta descentralización proporciona una mayor flexibilidad a nivel individual, pero también puede llevar a conflictos potenciales cuando los recursos compartidos deben asignarse entre proyectos (Villafañez et al., 2014).

Por lo tanto, el término "descentralizado" se refiere al enfoque en el cual los proyectos individuales tienen cierta autonomía para tomar decisiones sobre su propia programación. En este contexto, la capacidad de toma de decisiones se distribuye



## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

entre todos los proyectos. Sin embargo, para garantizar la viabilidad general y la coordinación, se requiere una entidad de coordinación. Esta entidad se encarga de organizar, arbitrar y verificar la factibilidad general de los horarios propuestos por cada proyecto, asegurando que se alineen con los objetivos y restricciones globales.

Es importante destacar que, en ambos enfoques, la programación de actividades se realiza con el objetivo de maximizar la eficiencia y el cumplimiento de los objetivos de cada proyecto, teniendo en cuenta las limitaciones de recursos y las interdependencias entre las actividades.

La elección entre un enfoque centralizado o descentralizado dependerá de la naturaleza y complejidad de los proyectos involucrados, así como de las preferencias y capacidades de coordinación de la organización. Ambos enfoques presentan ventajas y desafíos únicos, lo que hace que la elección del enfoque adecuado sea una consideración crucial en la resolución eficiente de los problemas de programación multiproyecto con recursos limitados.

2.2 DEL PSP AL DRCMPSP:

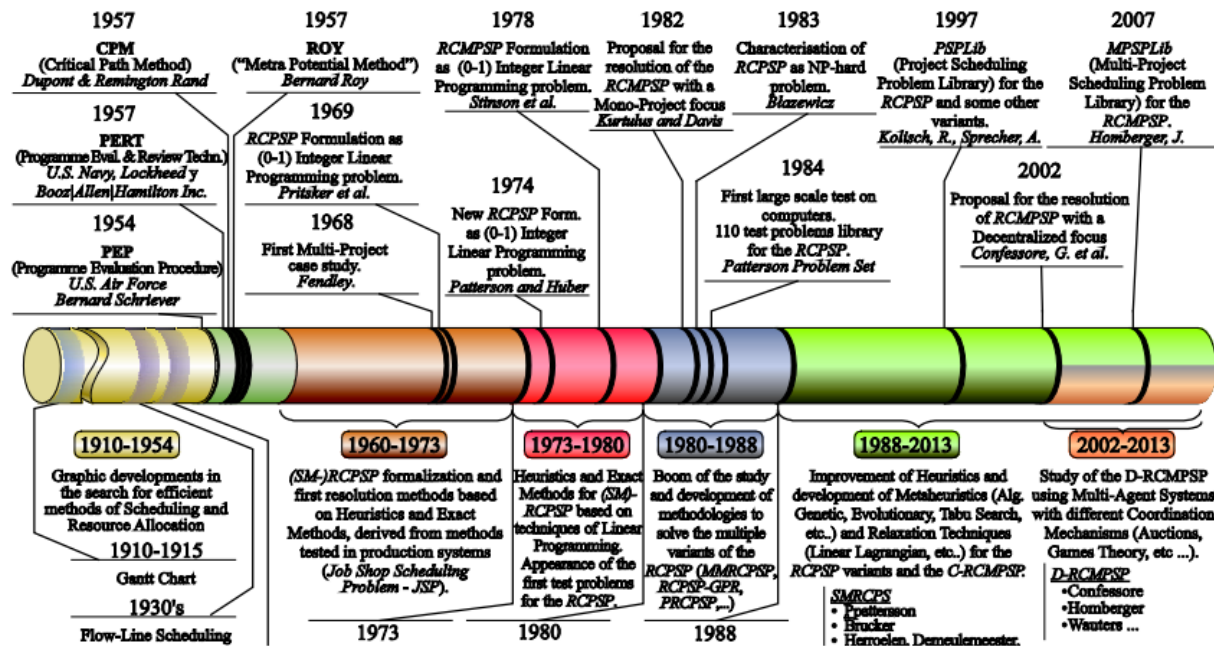


Figura 2.1 Evolución del PSP al DRCMPSP. Referencia: Villafañez et al. (2014)

2.2.1 Problema de Secuenciación de Proyectos (PSP):

El problema de secuenciación o programación de proyectos consiste en la planificación y secuenciación de una serie de tareas a las cuales se asigna un conjunto de recursos para equilibrar el coste total y el plazo de ejecución del proyecto. La incertidumbre siempre está presente en estos problemas debido a la falta de precisión en los tiempos de duración de las actividades del proyecto.

La historia de este tipo de problema se remonta a principios del siglo XX, cuando Gantt comenzó a utilizar el diagrama Gantt. Este gráfico, representado en la figura 2.2, muestra una secuenciación temporal de las tareas, representadas con rectángulos cuya longitud es proporcional a la duración de cada tarea. La duración total del proyecto se define a través del camino más crítico. Posteriormente, a mediados del siglo, surgieron otros métodos para controlar la planificación de proyectos, como las técnicas de evaluación y revisión de programas, así como el método de la trayectoria crítica.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

Sin embargo, el desarrollo de procesos industriales y el aumento de líneas de producción dieron lugar a una nueva necesidad de estudiar la planificación y el control de proyectos, esta vez teniendo en cuenta la existencia de recursos limitados. En este momento, el PSP evoluciona al método RCPSP (Gómez et al., 2022).

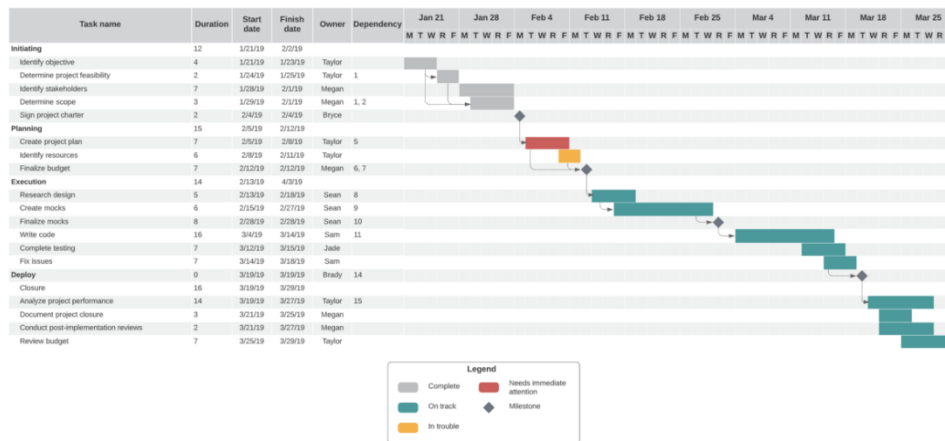


Figura 2.2 ejemplo Gantt.

Por lo tanto, los problemas de secuenciación de proyectos (PSP) están conformados por actividades, recursos, relaciones de precedencia y funciones objetivo. A partir de la combinación de recursos y las diferentes relaciones de precedencia, se obtienen diversas soluciones para este problema. Además, se derivan otras variantes considerando retrasos máximos y mínimos en las tareas (Cervantes, 2010).

### 2.2.2 Problema de Programación de Proyectos con Recursos Limitados (RCPSP):

El Problema de Programación de Proyectos con Recursos Limitados (RCPSP), conocido en inglés como "Resource-Constrained Project Scheduling Problem," es un desafío fundamental en la optimización combinatoria y una problemática ampliamente estudiada en la gestión de proyectos. En el RCPSP, el objetivo es encontrar una planificación óptima para un conjunto de actividades que componen un proyecto, tomando en cuenta las limitaciones de recursos disponibles y las dependencias entre las actividades.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

Un proyecto está compuesto por un conjunto de actividades etiquetadas como "a" ( $a = 1, 2, 3, \dots, A$ ). Las relaciones de precedencia entre estas actividades siguen el tipo final a comienzo (Finish-Start, FS), lo que significa que una actividad sucesora "a" puede comenzar una vez que haya finalizado la actividad predecesora. Cada actividad tiene una duración definida como " $d_a$ " y requiere una serie de recursos disponibles "k" ( $k = 1, 2, 3, \dots, K$ ) para su ejecución. La disponibilidad de cada recurso " $R_k$ " es constante por día, y el consumo de recursos necesario para llevar a cabo cada actividad se representa como " $r_{ak}$ ".

Es importante mencionar que, en cada proyecto, las actividades 1 y A son ficticias y no consumen tiempo ni recursos. En otras palabras, se cumplen las siguientes condiciones:  $d_1 = 0$ ,  $d_A = 0$ ,  $r_{1k} = r_{Ak} = 0$  (Alvin, 2022).

Las actividades A están sujetas a dos tipos de restricciones (Alvin, 2022):

- **Restricción de precedencia:** Esta restricción garantiza que cada actividad "a" ( $a \in A$ ) no puede comenzar hasta que todas sus actividades predecesoras "h" ( $h \in P_a$ ) hayan sido completadas. El conjunto  $P_a$  representa las actividades predecesoras de una actividad "a".
- **Restricción de cantidad de recursos:** La cantidad total de recursos de tipo "K" requeridos para todas las actividades procesadas no puede exceder el valor " $R_k$ " durante ningún período. Se asume que toda la información del proyecto es determinista y conocida de antemano, y que todos los parámetros son positivos y enteros. En consecuencia, el objetivo general de este problema es minimizar la duración del proyecto determinando la fecha más temprana de finalización del proyecto bajo las restricciones mencionadas anteriormente.

Dado que el RCPSPP pertenece a la clase de problemas NP-Hard, no existe un algoritmo eficiente que pueda encontrar la solución óptima en un tiempo polinómico para instancias grandes. Por lo tanto, se emplean diversas técnicas de resolución, como algoritmos heurísticos y metaheurísticas, para obtener soluciones cercanas a la óptima en un tiempo razonable.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

El RCPSP tiene una amplia variedad de aplicaciones en la industria y la gestión de proyectos, incluyendo la planificación de tareas en proyectos de construcción, producción de bienes, desarrollo de software, entre otros. Diseñar una planificación eficiente y realista para proyectos con recursos limitados es esencial para optimizar el uso de los recursos disponibles y cumplir con los plazos establecidos.

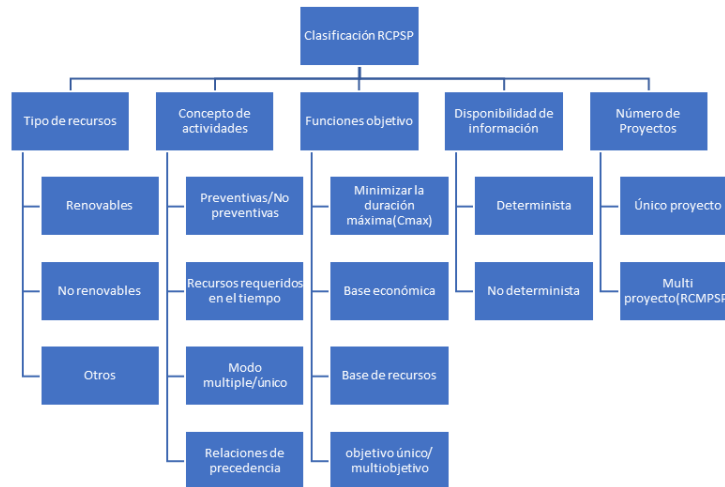
Es relevante mencionar que existen variantes del RCPSP que incorporan restricciones adicionales, como la disponibilidad estacional de recursos, costos de recursos variables, tiempos de actividad inciertos o flexibilidad en la asignación de recursos, lo que incrementa el nivel de complejidad y requiere enfoques de resolución específicos para cada variante.

En definitiva, el Problema de Secuenciación de Proyectos con Recursos Limitados tiene como objetivo minimizar la duración total del proyecto, gestionando de una forma eficiente una serie de recursos limitados, renovables, comunes entre distintas actividades y otros específicos para cada actividad.

Además, Los cinco tipos de clasificación son los siguientes, representados en la figura 2.3 (Alvin, 2022):

- Tipo de recursos: renovables y no renovables.
- Concepto de actividad: preventivo, RCPSPS multimodo, etc;
- Función objetivo: objetivo único o multiobjetivo
- Disponibilidad de información: determinista o no determinista

- Número de proyectos: proyecto único o proyecto múltiple.



*Figura 2.3 Clasificación RCPSP. Referencia: Alvin, 2022.*

### **2.2.3 Problema de Programación Multiproyecto con Recursos Limitados (RCMPSP):**

El (RCMPSP) Resource-Constrained Multi-Project Scheduling Problem, que en español se conoce como el Problema de Programación Multiproyecto con Restricción de Recursos, es una extensión del RCPSP que aborda la planificación de múltiples proyectos simultáneamente, teniendo en cuenta la disponibilidad limitada de recursos compartidos entre ellos.

En el RCMPSP, cada proyecto está compuesto por un conjunto de actividades interrelacionadas, y los proyectos comparten un conjunto común de recursos escasos, como personal, maquinaria o fondos financieros (Pérez et al., 2016). Cada actividad tiene una duración específica y requiere una cantidad de recursos para su ejecución.

Las características clave del RCMPSP son las siguientes:

- **Múltiples proyectos:** El problema implica planificar y programar múltiples proyectos de manera simultánea, teniendo en cuenta las restricciones de recursos compartidos entre ellos.

- **Interdependencia de proyectos:** Las actividades de los diferentes proyectos pueden estar interrelacionadas, lo que significa que la finalización de ciertas actividades de un proyecto puede tener un impacto en el inicio o la duración de actividades en otros proyectos.
- **Recursos compartidos:** Los recursos disponibles son compartidos entre los distintos proyectos, lo que puede dar lugar a conflictos en la asignación de recursos y afectar la programación de las actividades. (Alvin Hodiato, V., 2022).
- **Objetivos múltiples:** El objetivo general del RCMPSP es encontrar una programación que minimice la duración total de los proyectos involucrados, respetando las restricciones de recursos compartidos y las dependencias entre actividades dentro y entre proyectos.

El RCMPSP tiene aplicaciones en diversas áreas, como la gestión de carteras de proyectos, la planificación de programas de inversión, la coordinación de proyectos en grandes organizaciones y la optimización de recursos compartidos entre proyectos con distintas prioridades.

El desafío en la resolución del RCMPSP radica en encontrar una programación que permita equilibrar los recursos entre los proyectos para minimizar la duración total del conjunto de proyectos, garantizando que los recursos se asignen de manera óptima y evitando conflictos y cuellos de botella en la ejecución de las actividades (Villafañez et al., 2014).

### **2.2.4 Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos (DRCMPSP)**

El DRCMPSP es una extensión del problema de programación de proyectos con restricciones de recursos (RCPSP) en el cual se aborda la planificación y coordinación de múltiples proyectos de forma simultánea, teniendo en cuenta la disponibilidad limitada de recursos compartidos entre ellos, utilizando un enfoque descentralizado.

En el DRCMPSP, cada proyecto individual tiene sus propias actividades, recursos y restricciones de precedencia. Los proyectos están interconectados por recursos compartidos, lo que

significa que los recursos utilizados por una actividad de un proyecto pueden afectar la programación de actividades en otros proyectos. La diferencia clave con respecto al RCPSP tradicional es que cada proyecto es gestionado por un director de proyecto independiente y se toman decisiones de planificación de forma descentralizada.

Este enfoque descentralizado implica que cada director de proyecto es responsable de coordinar las actividades de su propio proyecto y tomar decisiones sobre la asignación de recursos, mientras se asegura de respetar las restricciones de recursos compartidos con otros proyectos (Confessore et al., 2007). La interacción entre los distintos proyectos y la competencia por los recursos compartidos pueden dar lugar a problemas de coordinación y asignación subóptima de recursos.

El DRCMPSP es un problema desafiante desde el punto de vista de la optimización, ya que requiere encontrar una solución que permita la finalización temprana de todos los proyectos involucrados, minimizando el tiempo total de finalización y garantizando el uso eficiente de los recursos compartidos. Dado que cada director de proyecto toma decisiones de forma autónoma, la coordinación entre los distintos proyectos es esencial para alcanzar una solución global óptima.

La resolución del DRCMPSP puede requerir el uso de algoritmos genéticos multi-modal, métodos de inteligencia artificial, o técnicas de optimización distribuida que permitan una toma de decisiones colaborativa y una coordinación efectiva entre los directores de proyecto para lograr una programación eficiente y equitativa de todos los proyectos involucrados.

Las características principales del DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem) son las siguientes:

- **Múltiples proyectos:** El DRCMPSP involucra la planificación y coordinación simultánea de múltiples proyectos, cada uno con su propio conjunto de actividades, recursos y restricciones de precedencia.
- **Enfoque descentralizado:** Cada proyecto es gestionado por un director de proyecto independiente, lo que implica que las decisiones de programación se toman de forma autónoma y descentralizada por cada director de proyecto.



## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

- **Recursos compartidos:** Los proyectos comparten un conjunto de recursos escasos, como mano de obra, equipos o materiales, lo que puede dar lugar a conflictos y competencia por los recursos entre los distintos proyectos.
- **Restricciones de precedencia:** Cada actividad dentro de un proyecto tiene relaciones de precedencia que determinan el orden en que deben ser ejecutadas. Además, puede haber dependencias de precedencia entre actividades de diferentes proyectos.
- **Objetivo de minimización:** El objetivo general del DRCMPSP es encontrar una programación óptima para los proyectos involucrados, minimizando el tiempo total de finalización de todos los proyectos y garantizando el uso eficiente de los recursos compartidos.
- **Desafío de coordinación:** La coordinación efectiva entre los directores de proyecto es esencial para evitar conflictos de recursos y lograr una programación equitativa y eficiente para todos los proyectos.
- **Complejidad NP-Hard:** El DRCMPSP pertenece a la clase de problemas NP-Hard, lo que implica que encontrar una solución óptima en tiempo polinómico para instancias grandes es computacionalmente difícil. Por lo tanto, se requieren enfoques heurísticos y metaheurísticos para abordar este problema.

El DRCMPSP tiene aplicaciones en diversos campos, como la gestión de carteras de proyectos en grandes organizaciones, la coordinación de proyectos de construcción o desarrollo de software con recursos compartidos, y la optimización de inversiones en proyectos con restricciones de recursos.

La combinación de múltiples proyectos, recursos compartidos y la descentralización de la toma de decisiones hace del DRCMPSP un problema complejo que requiere un enfoque cuidadoso y métodos de optimización avanzados para alcanzar soluciones eficientes y equitativas.

El mecanismo de coordinación en el DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem) se refiere a cómo los diferentes directores de proyecto colaboran y toman decisiones de manera conjunta para optimizar la programación y asignación de recursos en el contexto de múltiples proyectos.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

Dado que en el DRCMPSP cada proyecto es gestionado de manera autónoma por un director de proyecto independiente, existe la posibilidad de que se produzcan conflictos y competencia por los recursos compartidos entre los distintos proyectos. El mecanismo de coordinación busca resolver estos desafíos y asegurar que se alcancen soluciones eficientes y equitativas para todos los proyectos involucrados.

Existen diferentes enfoques para implementar mecanismos de coordinación en el DRCMPSP, algunos de los cuales incluyen:

- **Comunicación entre directores de proyecto:** Los directores de proyecto pueden establecer canales de comunicación para compartir información relevante sobre el estado de sus proyectos, las necesidades de recursos y las restricciones de programación. Esto permite una mejor comprensión de las interdependencias entre los proyectos y facilita la toma de decisiones informadas.
- **Negociación y colaboración:** Los directores de proyecto pueden negociar y colaborar entre sí para resolver conflictos de recursos y tomar decisiones colectivas sobre la asignación de recursos compartidos. Esto puede implicar acuerdos y compromisos para optimizar el rendimiento global de los proyectos.
- **Sistemas multiagente:** Se pueden utilizar sistemas multiagente, donde cada director de proyecto es un agente independiente que tiene sus propios objetivos y reglas de comportamiento. Los agentes pueden interactuar y coordinarse para lograr un equilibrio en el uso de los recursos compartidos.
- **Mecanismos de mercado:** Se pueden aplicar mecanismos de mercado donde los proyectos compiten por los recursos compartidos mediante ofertas y licitaciones. Estos mecanismos pueden ayudar a asignar recursos de manera eficiente y justa, al tiempo que fomentan la competencia y la colaboración entre los proyectos.

El mecanismo de coordinación seleccionado dependerá de la naturaleza de los proyectos involucrados, la complejidad de las interdependencias y la disponibilidad de información y recursos de comunicación. El objetivo es encontrar una solución global que minimice el tiempo

total de finalización de todos los proyectos y garantice el uso eficiente de los recursos compartidos, evitando conflictos y cuellos de botella en la programación (Villafañez et al. 2014)

La implementación efectiva de un mecanismo de coordinación en el DRCMPSP puede mejorar la colaboración y el rendimiento general de los proyectos involucrados, lo que es esencial para abordar los desafíos de programación en entornos de múltiples proyectos con recursos compartidos.

### 2.3 FUNCIONES OBJETIVO:

En el contexto del DRCMPSP, la planificación y coordinación de múltiples proyectos con recursos compartidos es una tarea compleja y desafiante. Para abordar eficazmente este problema, es fundamental definir funciones objetivo adecuadas que guíen el proceso de optimización y permitan obtener soluciones eficientes y equilibradas.

Las funciones objetivo pueden variar según las metas específicas de los proyectos involucrados y las restricciones de recursos compartidos. Algunos de los objetivos más comunes incluyen la minimización del tiempo total de finalización de los proyectos, la maximización de la utilización de recursos compartidos, la reducción de los retrasos en la ejecución de las actividades y la equitativa distribución de los recursos entre los proyectos.

Durante este apartado, examinaremos cada una de estas funciones objetivo en detalle, describiendo cómo se definen y cómo impactan en la programación de los proyectos. Además, discutiremos la importancia de encontrar un equilibrio adecuado entre los diferentes objetivos, ya que la optimización de uno puede afectar el rendimiento de otro. Estas funciones juegan un papel crucial en la formulación del problema y representan los criterios de optimización que se desean alcanzar durante el proceso de planificación de los proyectos.

Además de presentar las funciones objetivo, exploraremos cómo se integran en el proceso de optimización, especialmente si se utiliza un algoritmo como el SPEA2 (Strength Pareto Evolutionary Algorithm 2) para encontrar soluciones no dominadas en el frente de Pareto.

En resumen, este apartado tiene como objetivo profundizar en las funciones objetivo utilizadas en el DRCMPSP y su impacto en el proceso de optimización. Comprender estas funciones es esencial para abordar con éxito el desafío de coordinación y planificación en entornos de múltiples proyectos con recursos compartidos, permitiendo encontrar soluciones que satisfagan las metas de los proyectos involucrados y promoviendo una programación eficiente y equitativa.

**2.3.1 CLASIFICACIÓN:**

Las funciones objetivo son agrupadas según tres factores principales: recursos, proyectos y tiempo. Esta clasificación es representada en la figura 2.4, donde se propone un esquema que utiliza los colores rojo, verde y naranja para representar las funciones objetivo basadas en recursos, proyectos y tiempo, respectivamente. (Gómez et al., 2022).

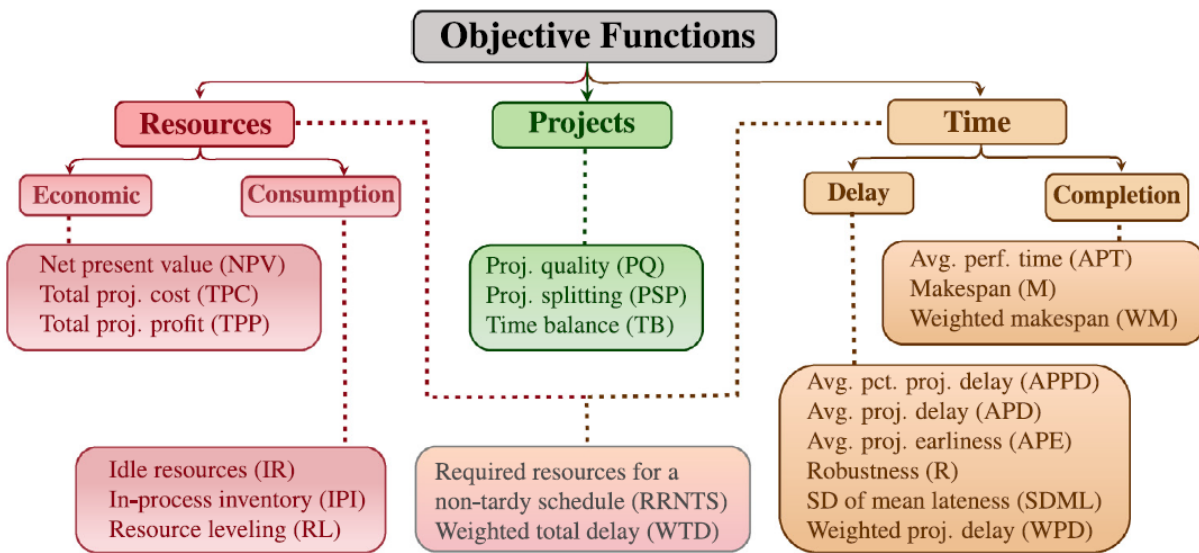


Figura 2.4 Funciones Objetivo. Referencia: Gómez et al., 2022.

La clasificación basada en recursos incluye funciones objetivo relacionadas con la optimización del uso de recursos compartidos entre los proyectos, buscando maximizar su utilización o lograr un equilibrio en su distribución.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

Por otro lado, la clasificación basada en proyectos se refiere a funciones objetivo que buscan optimizar aspectos específicos de cada proyecto, como la minimización del tiempo total de finalización o la reducción de los retrasos en la ejecución de actividades.

Finalmente, la clasificación basada en tiempo se centra en funciones objetivo relacionadas con la eficiencia temporal, buscando minimizar el tiempo total requerido para completar todos los proyectos y actividades.

La combinación de estas funciones objetivo permite abordar de manera integral el problema de planificación y coordinación en el contexto del DRCMPSP, asegurando una asignación óptima y equitativa de recursos, una programación eficiente de actividades y una gestión efectiva del tiempo en la ejecución de los proyectos.

En este trabajo, nos enfocaremos en el desarrollo y análisis de algunas de las funciones objetivo más importantes del tipo basado en recursos y tiempo, en el contexto del DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem). Nuestro objetivo es examinar cómo estas funciones objetivo impactan en la planificación y coordinación de múltiples proyectos que comparten recursos y cómo pueden contribuir a obtener soluciones eficientes y equilibradas para este desafiante problema.

Al profundizar en la optimización de estas funciones, buscamos comprender mejor los aspectos críticos de la asignación óptima de recursos y la gestión eficiente del tiempo en un entorno descentralizado. A través del análisis de estas funciones objetivo, pretendemos proporcionar una perspectiva más clara sobre la toma de decisiones informadas y cómo lograr una programación efectiva de múltiples proyectos, promoviendo una utilización equitativa de los recursos y una ejecución oportuna y coordinada de las actividades.

### *2.3.1.1 Total Profit Project (TTP):*

La función objetivo "Total Profit Project (TTP)" tiene como objetivo principal obtener la máxima ganancia o beneficio total de los proyectos involucrados en el DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem).

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

En el contexto del DRCMPSP, varios proyectos comparten recursos limitados y compiten por su utilización. La función objetivo TTP busca optimizar la programación y coordinación de las actividades de cada proyecto para maximizar los ingresos o beneficios generados por el conjunto de proyectos.

Para ello, se tienen en cuenta factores como los ingresos asociados a cada actividad y la secuencia en que se programan las actividades de los proyectos. La función objetivo TTP busca identificar la mejor combinación de asignación de recursos y secuencia de actividades para obtener el máximo beneficio posible, asegurando que las restricciones de recursos compartidos y las dependencias entre las actividades de los proyectos se cumplan.

Al maximizar el beneficio total, se busca encontrar una solución que equilibre de manera eficiente la utilización de los recursos, la duración de las actividades y la generación de ingresos, lo que puede tener un impacto significativo en la rentabilidad y el éxito global de los proyectos involucrados.

Es importante destacar que la función objetivo TTP puede considerar diferentes factores y métricas económicas específicas para cada proyecto, como el costo de las actividades, los ingresos generados, los gastos operativos, entre otros. La combinación de estos factores en la función objetivo permitirá evaluar y comparar diferentes soluciones y seleccionar aquella que maximice el beneficio total de manera óptima.

Según Shou, et al. (2014) la ecuación 2.1 define la función objetivo para maximizar el beneficio total del proyecto:

$$\max \sum_{p=1}^P \sum_{t=EF_{pj}}^{LF_{pj}} b_{pt} c_{pj} t \quad (\text{Ec. 2.1})$$

$p = \text{índice de proyecto}$

$P = \text{Número total de proyectos}$

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

$j = \text{índice de actividad}$

$t = \text{índice de tiempo}$

$r = \text{índice de recurso}$

$EF_{pj} = \text{Tiempo de finalización más temprana de actividad } (p, j)$

$LF_{pj} = \text{Tiempo de finalización más tardío de actividad } (p, j)$

$b_{pt} = \text{Beneficio del proyecto } p \text{ cuando se completa en el tiempo } t$

Además, mediante la ecuación 2.2, se asegura que todas las actividades seleccionadas están completas.

$$\sum_{t=EF_{pj}}^{LF_{pj}} c_{pjt} = x_p \quad (\text{Ec. 2.2})$$

Estas dos variables se refieren a:

$x_p = \text{selección del proyecto } p$

$c_{pjt} = \text{la programación de las actividades del proyecto}$

$$x_p = \begin{cases} 1 & \text{si el proyecto } p \text{ está incluido} \\ 0 & \text{si el proyecto } p \text{ está excluido} \end{cases}$$

$$c_{pjt} = \begin{cases} 1 & \text{si la actividad } j \text{ del proyecto } p \text{ ha sido completada en el tiempo } t \\ 0 & \text{lo contrario} \end{cases}$$

Para calcular el beneficio del proyecto  $b_{pt}$  (ecuación 2.3), suponemos unos beneficios linealmente decrecientes del tiempo de finalización del proyecto expresados a través de la ecuación 2.4.

$$b_{pt} = B_p \left( 1 - \lambda_1 (t - CPL_p) \right) \quad (\text{Ec. 2.3})$$

$$B_p = \sigma_p \sum_{r=1}^R \sum_{j=1}^J r_{pjr} d_{pj} \quad (\text{Ec. 2.4})$$

$$CPL_p = \text{Duración del camino crítico}$$

$\lambda_1$

= ratio de decrecimiento del beneficio a medida que la complementación del proyecto aumenta

Hay dos niveles del ratio de decrecimiento de beneficio: (2% o 8%).

$$B_p = \text{Beneficio base}$$

$$b_{pt} = \text{beneficio del proyecto}$$

$$\sigma_p = \text{coeficiente de utilización de recursos.}$$

Este coeficiente está sujeto a una distribución uniforme de intervalo [0.5,1.5]

### 2.3.1.2 Total Project Cost (TPC):

La función objetivo "Total Project Cost (TPC)" tiene como objetivo principal minimizar el costo total de los proyectos involucrados en el DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem). La función objetivo TPC busca optimizar la programación y coordinación de las actividades de cada proyecto para reducir los costos asociados a la ejecución de los proyectos.

Para ello, se tienen en cuenta los costos de las actividades, que pueden incluir los costos de recursos utilizados, mano de obra, materiales y otros gastos operativos. La función objetivo TPC busca identificar la mejor combinación de asignación de recursos y secuencia de actividades que minimice el costo total de los proyectos, garantizando que las restricciones de recursos compartidos y las dependencias entre actividades sean satisfechas. La ecuación 2.5 representa la suma total de estos costos.

$$\phi = \text{TTC} + \text{TIC} + \text{TDC} \quad (\text{Ec. 2.5})$$

TTC = Coste total de transferencias de recursos en el multiproyecto



## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

TIC = Coste total de inactividad de recursos en el multiproyecto

TDC = Coste total causado por el retraso de proyectos.

$x_{ijr}$  = Número de unidades de recurso  $r$  transferidas de la actividad  $i$  a  $j$

$x_{ijr}^k$

= Cantidad de recurso  $r$  transferido de la actividad  $i$  a la  $j$  con un tipo de transferencia tipo  $k$ .

$x_{ijr}^{-k}$  = Cantidad de recurso de segundo nivel  $r$  consumido o utilizado durante la

transferencia de primer nivel de recursos de la actividad  $i$  a la  $j$  mediante un

tipo de transferencia  $k$

$$k = \begin{cases} ES: Finish - to - start \\ EE: Finish - to - finish \\ SS: Start - to - start \\ SE: Start - to - finish \end{cases}$$

$$z_{ijr}^k = \begin{cases} 1 & \text{si el recurso } r \text{ es transferido de } i \text{ a } j \text{ mediante el tipo de transferencia } k \\ 0 & \text{en caso contrario} \end{cases}$$

$CT_{ijr}^{fix}$  = coste de transferencia fijo del recurso  $r$  de la actividad  $i$  a la  $j$ .

$CT_{ijr}^{var}$  = coste variable de transferencia de una unidad del recurso  $r$  de  $i$  a  $j$

Según Kruger, et al. (2008) el coste total de transferencia TTC está definido por el coste de todas las transferencias programadas de recursos (ecuación 2.6). El coste variable de una transferencia de la actividad  $i$  al  $j$  depende de la cantidad de unidades transferidas de  $i$  a  $j$  de todos los recursos  $x_{ijr}^k$  excluyendo la cantidad de recursos de segundo nivel utilizados o consumidos  $x_{ijr}^{-k}$ . El coste de transferencia fijo  $CT_{ijr}^{fix}$  aparece una sola vez cuando se realiza una transferencia de recurso  $r$  de  $i$  a  $j$ . Por lo tanto, según Kruger, et al., 2008, el coste fijo de los recursos de segundo nivel no se integra en los costes fijos.

$$TTC = \sum_{i \in J \cup \{So\}} \sum_{j \in J} \sum_{r \in R} (CT_{ijr}^{var} \sum_{k \in S} (x_{ijr}^k - x_{ijr}^{-k}) + CT_{ijr}^{fix} \sum_{k \in S} z_{ijr}^k) \quad (Ec. 2.6)$$

De igual manera según Li, F., & Xu, Z. (2018) el coste de tardanza total (TTC) representado en la ecuación 2.7:

$$TTC = \sum_{p=1}^m tc_p (s_{p(J_p+1)} - ad_p - CPL_p) \quad (Ec. 2.7)$$

$tc_p =$  Coste unitario de retraso

$(s_{p(J_p+1)} - ad_p) =$  El Makespan final del proyecto  $p$

$CPL_p =$  Valor del camino crítico del proyecto  $p$

$m =$  Proyectos a planificar simultáneamente

El coste de oportunidad TIC de los recursos renovables que se encuentran ociosos (ecuación 2.8) se realiza por unidad y por periodo de inactividad. El suministro total de recursos renovables  $r \in R$  viene dada por  $a_r * F_{e0}$ . El recurso es usado de una forma productiva durante la ejecución de la actividad y cuando admite una transferencia de recursos a otro recurso  $s \in \text{sup}(r)$ .

$$TIC = \sum_{r \in R} CI_r \left( a_r * F_{e0} - \sum_{i \in J} u_{ir} d_i - \sum_{i \in J \cup (So)} \sum_{j \in J} \sum_{r_i} \sum_{s \in \text{sup}(r)} \mu_{rs} \Delta_{ijs} x_{ijs}^k \right) \quad (Ec. 2.8)$$

$CI_r$

= Coste de oportunidad de una unidad ociosa de recurso  $r$  por periodo de inactividad.

$a_r =$  Numero de unidades de recurso  $r$  disponible

$F_{e0} =$  tiempo de finalización de la actividad  $i$  donde  $i = e0$ ,

asegurando que todos los recursos son recuperados al final de la actividad.

$u_{ir} =$  Número de unidades de recurso  $r$  requerido en actividad  $i$

$d_i =$  duración de la actividad  $i$

$\mu_{rs}$  = cantidad total de recurso  $r$  requerido para la transferencia de una unidad del recurso  $r$  por cada periodo de transferencia.

$\Delta_{ijs}$  = Tiempo para transferir unidades de recurso  $r$

$x_{ijs}^k$

= Cantidad de recurso  $s$  transferido de la actividad  $i$  a la  $j$  con un tipo de transferencia tipo  $k$ .

Para el monoproyecto, la penalización total por retraso TDC (ecuación 2.9) se compone de la anticipación o retraso y los tipos de penalizaciones dispuestos por periodo de retraso. En el caso de monoproyecto, se calcula la suma de los valores ponderados de puntualidad y retraso.

$$TDC = CD^+ \delta^+ + CD^- \delta^- \quad (\text{Ec. 2.9})$$

Por lo tanto, en el caso de multiproyecto queda definido de mediante la ecuación 2.10:

$$TDC = \sum_{p \in P} (CD_p^+ \delta_p^+ + CD_p^- \delta_p^-) \quad (\text{Ec. 2.10})$$

$CD_p^+, CD_p^-$  = Penalización por retraso y anticipación o puntualidad del proyecto  $p$   
 $\in P$  por periodo de retraso positivo o negativo

No hay índice de penalización en el caso de multiproyecto:  $CD_p^+, CD_p^- \geq 0$ .

$\delta_p^+, \delta_p^-$

= retraso positivo (tardanza), retraso negativo (anticipación o puntualidad) del proyecto  $p$

### 2.3.1.3 Weight project Delay (WPD):

La función objetivo "Weighted Project Delay (WPD)" tiene como objetivo principal minimizar el retraso ponderado de los proyectos involucrados en el DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem).

En el contexto del DRCMPSP, la función objetivo WPD busca optimizar la programación y coordinación de las actividades de cada proyecto para reducir los retrasos en la finalización de las actividades, considerando pesos específicos asignados a cada proyecto.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

Cada proyecto puede tener diferentes niveles de importancia o prioridad, y estos se reflejan en los pesos asignados. La función objetivo WPD busca identificar la mejor combinación de asignación de recursos y secuencia de actividades que minimice el retraso ponderado, asegurando que las restricciones de recursos compartidos y las dependencias entre actividades sean satisfechas (Nabipoor et al., 2018).

Al minimizar el retraso ponderado, se busca encontrar una solución eficiente que priorice los proyectos de mayor importancia y minimice el impacto de los retrasos en su finalización. Esto permite mejorar el rendimiento global y la satisfacción de los interesados al asegurar que los proyectos críticos se completen en tiempo y forma.

Es importante destacar que la función objetivo WPD (ecuación 2.11) combina la eficiencia en el uso de recursos con la gestión de plazos para lograr una planificación equilibrada que considere la importancia relativa de los proyectos involucrados. La elección de los pesos asociados a cada proyecto es fundamental para reflejar las prioridades y los objetivos estratégicos de la organización en el proceso de programación multiproyecto.

$$WPD = \min TTa = \min \sum_{p=1}^P w_p Ta_p \quad (Ec. 2.11)$$

$P =$  Número total de proyectos

$TTa =$  Retraso total ponderado de los proyectos

$w_p =$  El peso (grado de prioridad) del proyecto  $p$

$Ta_p =$  tardanza o retraso del proyecto  $p$

El retraso o tardanza de cada proyecto se define mediante la ecuación 2.12 como la diferencia entre la duración (Makespan) del proyecto y su fecha de vencimiento:

$$Ta_p \geq S_{i+1,p}^h - DD_p \quad (Ec. 2.12)$$

$S_{i,p}^h =$  El tiempo de comienzo de la actividad  $i$  del proyecto  $p$  bajo el escenario  $h$

$DD_p =$  Fecha de vencimiento o límite del proyecto  $p$

### 2.3.1.4 Resource Leveling / Nivelación de recursos (RL):

Resource Leveling, también conocido como Nivelación de Recursos, es una técnica utilizada en la gestión de proyectos para equilibrar la carga de trabajo de los recursos disponibles en el tiempo, evitando picos de sobreutilización y subutilización de los recursos.

En el contexto del DRCMPSP, el objetivo es optimizar la asignación y programación de las actividades de los proyectos de manera que se evite la asignación excesiva de recursos en ciertos períodos y se garantice una utilización más uniforme a lo largo del tiempo.

La Nivelación de Recursos implica reorganizar las actividades de los proyectos, ajustar las fechas de inicio y finalización, o incluso cambiar la secuencia de actividades para evitar conflictos de recursos. Esto se realiza sin modificar la secuencia lógica de las actividades o alterar el calendario total de finalización del proyecto.

Al implementar la Nivelación de Recursos, se busca lograr una distribución más equitativa y eficiente de los recursos, lo que puede conducir a una mejor utilización de los recursos disponibles, una reducción de los costos asociados con el uso de recursos o tiempos extras y una mejora en la planificación y ejecución de los proyectos.

Es importante destacar que la Nivelación de Recursos es un proceso iterativo y dinámico, ya que los cambios en la programación de un proyecto pueden tener un impacto en otros proyectos y en la carga de recursos en general. La implementación exitosa de la Nivelación de Recursos requiere una planificación cuidadosa y la consideración de las restricciones de recursos, las prioridades de los proyectos y los objetivos estratégicos de la organización.

Según Mao et al., (2009) la nivelación de recursos, definida mediante la ecuación 2.13, trata de minimizar la variación del uso de recursos a lo largo del tiempo. Este objetivo suele alcanzarse minimizando la suma del coste de utilización al cuadrado.

$$RL = \min \left( c_r^u \sum_{t=0}^{\infty} u_{t,r}^2(S) \right) \quad (Ec. 2.13)$$

$c_r^u > 0 =$  Coste de utilización del recurso tipo  $r$  por intervalo de tiempo discreto

$u_{t,r}$  = Utilización del recurso  $r$  durante el intervalo de tiempo discreto etiquetado por  $t$  según la programación  $S$

La utilización del recurso tipo  $r$  en  $t$  se define como representamos en la ecuación 2.14:

$$u_{t,r}(S) = \sum_{s_{p,j} \in S | s_{p,j} \leq t \leq s_{p,j} + p_{p,j}} r_{p,j}^r \quad (\text{Ec. 2.14})$$

$p_{p,j}$  = duración de procesamiento no preferente de la actividad  $j$  del proyecto  $p$

$r_{p,j}^r$  = cantidad del recurso de tipo  $r$  para la actividad  $j$  del proyecto  $p$

$R$  = Tipos recursos disponibles

$s_{p,j}$  = Tiempo de comienzo de la actividad  $j$  del proyecto  $p$

Según Hauder, et al. (2020), no son considerados los costes puesto que no hace diferenciación entre los distintos tipos de recursos ya que estos son igualmente importantes. Por lo tanto, el objetivo es minimizar los diferentes picos de utilización de recursos y así que haya un balance o equilibrio de utilización de recursos.

$$RL = \min (\max_{r \in R^*} \left( \frac{u_r}{C_r} \right)) \quad (\text{Ec. 2.15})$$

$R^*$  = Selección de recursos pertenecientes a  $R$

$u_r$  = máxima utilización de recurso (pico de utilización) del recurso  $r \in R^*$

Según Lova et al. (2000), podemos analizar la nivelación de recursos  $RL$  mediante la ecuación 2.16 como la media del coeficiente de variación ( $E(\delta_r)$ ) ratio de comparación entre la desviación estándar y la media. Cuanto menor es la media del Coeficiente de variación mejor es la nivelación de recursos.

$$E(\delta_r) = \frac{\sum_{r=1}^R \frac{\sqrt{\text{var}(x_r)}}{E(x_r)}}{R} \quad (\text{Ec. 2.16})$$

### 2.3.1.5 *Idle Resource (IR):*

El concepto de "Idle Resource (IR)" se refiere a las unidades de un recurso que están disponibles y no están siendo utilizadas en ninguna actividad del proyecto o multiproyecto en un momento dado. En otras palabras, son las unidades de recursos que están sin asignar y pueden ser empleadas para realizar tareas adicionales.

En un contexto de disponibilidad constante de recursos, y como se representa en la figura 2.5, minimizar los recursos ociosos (IR) implica minimizar la duración total del multiproyecto. Si el tiempo de ejecución del multiproyecto se reduce, habrá menos oportunidades para que las unidades de recursos estén sin asignar, lo que resulta en una mejor utilización de los recursos disponibles.

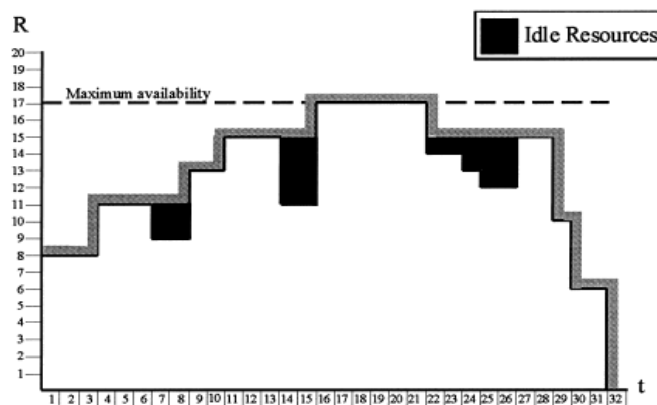


Fig. 3. Resource histogram that illustrates idle resources.

*Figura 2.5 Idle Resource (IR). Referencia: Lova, et al., 2000*

Para proporcionar otra definición de recursos ociosos, se considera que una unidad de recurso asignada al multiproyecto permanecerá asignada desde el primer instante en que se utilice hasta que no sea necesaria para el multiproyecto. Bajo esta definición, una unidad de recurso estará ociosa cuando, estando asignada al multiproyecto, las actividades del multiproyecto no la requieran. En esta situación, se asume que la disponibilidad de cada recurso es la máxima necesidad

disponible, lo que indica que una unidad de recurso se considera disponible y ociosa en el momento en que no se necesita para ninguna de las actividades programadas del multiproyecto. (Lova,1997)

El objetivo de gestionar y reducir los recursos ociosos (IR) es mejorar la eficiencia en la asignación y uso de recursos, lo que puede conducir a una mejor utilización de los recursos disponibles y a una optimización del rendimiento y la duración del multiproyecto.

Según Amirian et al. (2017), para calcular el valor de recursos inutilizados nos ayudamos de la ecuación 2.17:

$$IR = \min \sum_{r=1}^R \left( \left( \sum_{t=1}^T R_{rt} \right) - \sum_{p=1}^P Y_p \cdot \sum_{j=1}^{J_p} d_j \cdot r_{pjr} \right) \quad (Ec. 2.17)$$

*p = índice del proyecto p = 1, ..., P*

*h, j = índice de actividad h, j = 1, ..., J<sub>i</sub>*

*r = Tipo de recurso r = 1, ..., R*

*t = índice de tiempo t = 1, ..., T*

*R<sub>rt</sub> = máxima cantidad de recurso de tipo r disponible en el periodo t*

$$Y_p = \begin{cases} \text{Proyecto } p \text{ es seleccionado} \rightarrow Y_p = 1 \\ \text{Proyecto } p \text{ no es seleccionado} \rightarrow Y_p = 0 \end{cases}$$

*d<sub>j</sub> = Duración de la actividad j del proyecto p*

*r<sub>pjr</sub> = Recurso del tipo r requerido en la actividad j del proyecto p*

### 2.3.1.6 Makespan o Maximum Completion Time (TMS):

El "Makespan" o "Maximum Completion Time (TMS)" es una función objetivo comúnmente utilizada en problemas de programación de proyectos y programación multiproyecto, incluido el DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem).



## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

En el contexto del DRCMPSP, el objetivo principal de la función Makespan es minimizar la duración total del proyecto o del conjunto de proyectos involucrados. Representa el tiempo más largo necesario para completar todas las actividades y proyectos considerando las restricciones de recursos y las dependencias entre actividades.

Al minimizar el Makespan, se logra una planificación más eficiente y optimizada, lo que puede resultar en una reducción de los costos de ejecución de los proyectos y una mejora en la satisfacción de los interesados al entregar los proyectos dentro de los plazos acordados.

Es importante destacar que el Makespan puede estar influenciado por diversas variables, como la capacidad de los recursos, las dependencias entre actividades, los retrasos en la programación y los cambios en los requerimientos del proyecto. Por lo tanto, la búsqueda de soluciones óptimas o aproximadas para el Makespan en problemas de programación multiproyecto puede requerir el uso de algoritmos heurísticos o metaheurísticos para encontrar soluciones eficientes y de alta calidad en un tiempo razonable.

Esta función tiene como objetivo el maximizar el mayor de los tiempos de finalización del último de los trabajos. La minimización de esta función objetivo consiste en tomar la fecha más tardía de entre todos los proyectos menos la fecha de inicio más temprana. En definitiva, según Hao et al. (2018), el TMS, representado en la ecuación 2.18, sería el número de unidades temporales transcurridas entre el inicio del primer proyecto hasta la finalización del último proyecto dentro de una cartera.

$$C_j = \text{maximo}T - \text{minimo}T \quad (\text{Ec. 2.18})$$

*maximoT = Maximo de la duración final del proyecto*

*minimoT = Tiempo minimo de llegada del proyecto*

$$C_{max} = \text{max}C_j$$

### 2.3.1.7 Average Project delay (APD):

El "Average Project Delay (APD)" es una función objetivo utilizada en problemas de programación multiproyecto, que busca minimizar el retraso promedio de todos los proyectos involucrados.

En el contexto del DRCMPSP (Decentralized Resource-Constrained Multi-Project Scheduling Problem), donde múltiples proyectos comparten recursos limitados, el APD tiene como objetivo encontrar una secuencia de actividades y asignación de recursos que permita minimizar el tiempo promedio de retraso de todos los proyectos, considerando las restricciones de recursos y las dependencias entre actividades.

El APD considera tanto los proyectos críticos que tienen un impacto significativo en el tiempo total del multiproyecto como los proyectos no críticos. El objetivo es minimizar la probabilidad de retraso en cada proyecto y mejorar el rendimiento global del conjunto de proyectos.

Al minimizar el APD, se busca lograr una planificación más eficiente y equilibrada que reduzca el tiempo de finalización promedio de todos los proyectos involucrados. Esto puede conducir a una mejor utilización de los recursos disponibles, una reducción de los costos asociados con los retrasos en los proyectos y una mayor satisfacción de los interesados al cumplir con los plazos acordados para la entrega de los proyectos.

Es importante tener en cuenta que el APD (ecuación 2.19) puede estar sujeto a diversas restricciones y factores, como la disponibilidad de recursos, la duración de las actividades, las restricciones de precedencia y los cambios en los requerimientos del proyecto. Por lo tanto, encontrar soluciones óptimas o aproximadas para el APD en problemas de programación multiproyecto puede requerir el uso de algoritmos de optimización y métodos heurísticos para obtener resultados eficientes y de alta calidad en un tiempo razonable. (Zheng et al., 2014).

$$APD = \frac{\sum_{p=1}^P (MS_p - CPD_p)}{P} \quad (Ec. 2.19)$$

$$MS_p = \text{Valor del Makespan}$$

$CPD_p = \text{Duración del camino crítico}$

$P = \text{Número total de proyectos}$

Según Pérez et al. (2016), la minimización del porcentaje medio del retraso consiste en minimizar la suma del porcentaje medio de retrasos de todos los proyectos como se define en la ecuación 2.20.

$$APD = \frac{\sum_{p=1}^P (D_p - T_{op} - \text{Valor}CP_p)}{P} \quad (\text{Ec. 2.20})$$

$D_p = \text{duración del proyecto } p$

$T_{op} = \text{Tiempo de llegada del proyecto } p$

$\text{Valor } CP_p = \text{Tiempo total del camino crítico}$

$P = \text{número de proyectos}$

### 2.3.1.8 Tiempo de ejecución (DPD):

El "Tiempo de Ejecución (DPD)", definido en la ecuación 2.22, se refiere al tiempo total necesario para completar la ejecución de un proyecto o un conjunto de proyectos en el contexto del problema de programación de proyectos o programación multiproyecto.

En el problema de programación multiproyecto, el DPD (ecuación 2.21) tiene como objetivo minimizar el tiempo necesario para finalizar todas las actividades y proyectos involucrados, considerando las restricciones de recursos, las dependencias entre actividades y otras limitaciones (Pérez et al., 2016).

El DPD es una medida crucial para evaluar la eficiencia y el rendimiento de la planificación de proyectos. Un menor DPD implica una programación más eficiente y una finalización más rápida de los proyectos, lo que puede resultar en una reducción de los costos y una mayor satisfacción de los interesados.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

Sin embargo, es importante encontrar un equilibrio entre minimizar el DPD y cumplir con las restricciones y objetivos específicos de cada proyecto. En algunos casos, minimizar en exceso el DPD puede generar conflictos de recursos o incumplimiento de fechas límite, lo que afecta negativamente la calidad y eficacia general del proceso de planificación.

Por lo tanto, encontrar soluciones óptimas o aproximadas para el DPD en problemas de programación multiproyecto puede requerir el uso de algoritmos de optimización, técnicas de programación y métodos heurísticos para encontrar la mejor combinación de secuenciación de actividades y asignación de recursos que optimice el tiempo de ejecución general de los proyectos.

$D_p = \text{duración del proyecto } p$

$T_{op} = \text{Tiempo de llegada del proyecto } p$

$\text{Valor } CP_p = \text{Tiempo total del camino crítico}$

$P = \text{número de proyectos}$

$$DPD = \frac{\sum_{p=1}^P (D_p - T_{op} - \text{Valor } CP_p - APD)^2}{P} \quad (\text{Ec. 2.21})$$

$$DPD = \sqrt{\frac{DPD}{P - 1}} \quad (\text{Ec. 2.22})$$

La elección de las funciones Idle Resource (Recursos Ociosos) y Average Project Delay (Retardo Promedio del Proyecto) para el estudio del DRCMPSP con el algoritmo SPEA2 en este trabajo se sustenta en lo siguiente:

### **Idle Resource (IR):**

#### Fortalezas:

- Destaca la eficiencia en la asignación de recursos.
- Permite evaluar la capacidad de aprovechamiento de los recursos disponibles.

## DESCRIPCIÓN DEL PROBLEMA Y LAS FUNCIONES OBJETIVO

---

- Proporciona una medida cuantitativa directa de cómo se gestionan y utilizan los recursos en el conjunto de proyectos, lo que facilita la comparación y análisis.

### Debilidades:

- Puede no capturar completamente la complejidad del problema en entornos dinámicos.

### **Average Project Delay (APD):**

#### Fortalezas:

- Mide el tiempo promedio de retraso en la finalización de proyectos.
- Refleja la capacidad del sistema para cumplir con los plazos establecidos.

#### Debilidades:

- Enfoque promedio puede ocultar variaciones significativas entre proyectos.

En definitiva, estas funciones han sido seleccionadas por su capacidad para proporcionar una evaluación integral del rendimiento del sistema, abordando tanto la eficiencia en la utilización de recursos como la puntualidad en la entrega de proyectos. La combinación de IR y APD ofrece una perspectiva equilibrada de los resultados, permitiendo una comprensión más completa de los desafíos y oportunidades en el contexto del DRCMPSP.

---

### 3 SPEA2 (STRENGTH PARETO EVOLUTIONARY ALGORITHM 2):

Los Algoritmos Genéticos (AG) son técnicas de optimización inspiradas en la evolución biológica que se utilizan para resolver una amplia gama de problemas de optimización. SPEA2, que significa "Strength Pareto Evolutionary Algorithm 2", es una variante avanzada de los AG diseñada para abordar problemas de optimización multiobjetivo.

Este algoritmo es especialmente efectivo cuando se trata de encontrar soluciones que representen un equilibrio óptimo entre múltiples objetivos, lo que lo convierte en una herramienta valiosa en campos como la ingeniería, la planificación y la toma de decisiones.

SPEA2 se destaca por su capacidad para encontrar conjuntos de soluciones no dominadas, conocidas como "frentes de Pareto", que representan soluciones óptimas en términos de múltiples objetivos, sin que ninguna de ellas sea dominada por otra. A través de una combinación de selección, cruce y mutación de soluciones, SPEA2 evoluciona poblaciones de soluciones en busca de estas soluciones óptimas.

Este algoritmo es altamente adaptable y puede aplicarse a una amplia variedad de problemas de optimización multiobjetivo (Zitzler et al., 2003). A medida que evoluciona la población, SPEA2 mantiene un frente de Pareto aproximado que representa las soluciones no dominadas encontradas hasta el momento. A lo largo de las generaciones, busca mejorar continuamente este frente de Pareto hasta que converja hacia un conjunto de soluciones que represente el equilibrio deseado entre los objetivos del problema.

En resumen, SPEA2 es una herramienta poderosa para la resolución de problemas de optimización multiobjetivo y es ampliamente utilizada en la investigación y la aplicación práctica. Su capacidad para encontrar soluciones no dominadas lo convierte en una elección popular cuando se trata de encontrar soluciones óptimas en problemas que involucran múltiples objetivos conflictivos.

### 3.1 FUNDAMENTOS DEL ALGORITMO SPEA2

El algoritmo SPEA2 (Strength Pareto Evolutionary Algorithm 2) es un método avanzado de optimización diseñado para resolver problemas de optimización multiobjetivo. Para comprender adecuadamente los fundamentos de SPEA2, vamos a explorar los siguientes aspectos clave:

#### 3.1.1 Descripción general del algoritmo

SPEA2 se basa en una estructura típica de algoritmo genético, lo que significa que opera en poblaciones de soluciones a lo largo de múltiples generaciones. Comienza con la creación de una población inicial de soluciones, donde cada solución representa una posible asignación de valores para los objetivos del problema.

Durante cada generación, SPEA2 aplica operadores genéticos para evolucionar la población actual. Estos operadores incluyen:

- **Selección:** Determina qué soluciones se reproducirán para formar la siguiente generación. La selección se basa en la calidad relativa de las soluciones y se enfoca en preservar la diversidad en la población.
- **Cruce:** Combina las características de dos soluciones para crear nuevas soluciones. El cruce busca combinar las fortalezas de los padres para generar descendientes de alta calidad.
- **Mutación:** Introduce pequeñas modificaciones aleatorias en las soluciones existentes. La mutación agrega diversidad a la población y ayuda a explorar nuevas áreas del espacio de soluciones.

El funcionamiento y las distintas etapas del algoritmo SPEA2 se representan gráficamente mediante el diagrama de flujos presentado a continuación en la figura 3.1:

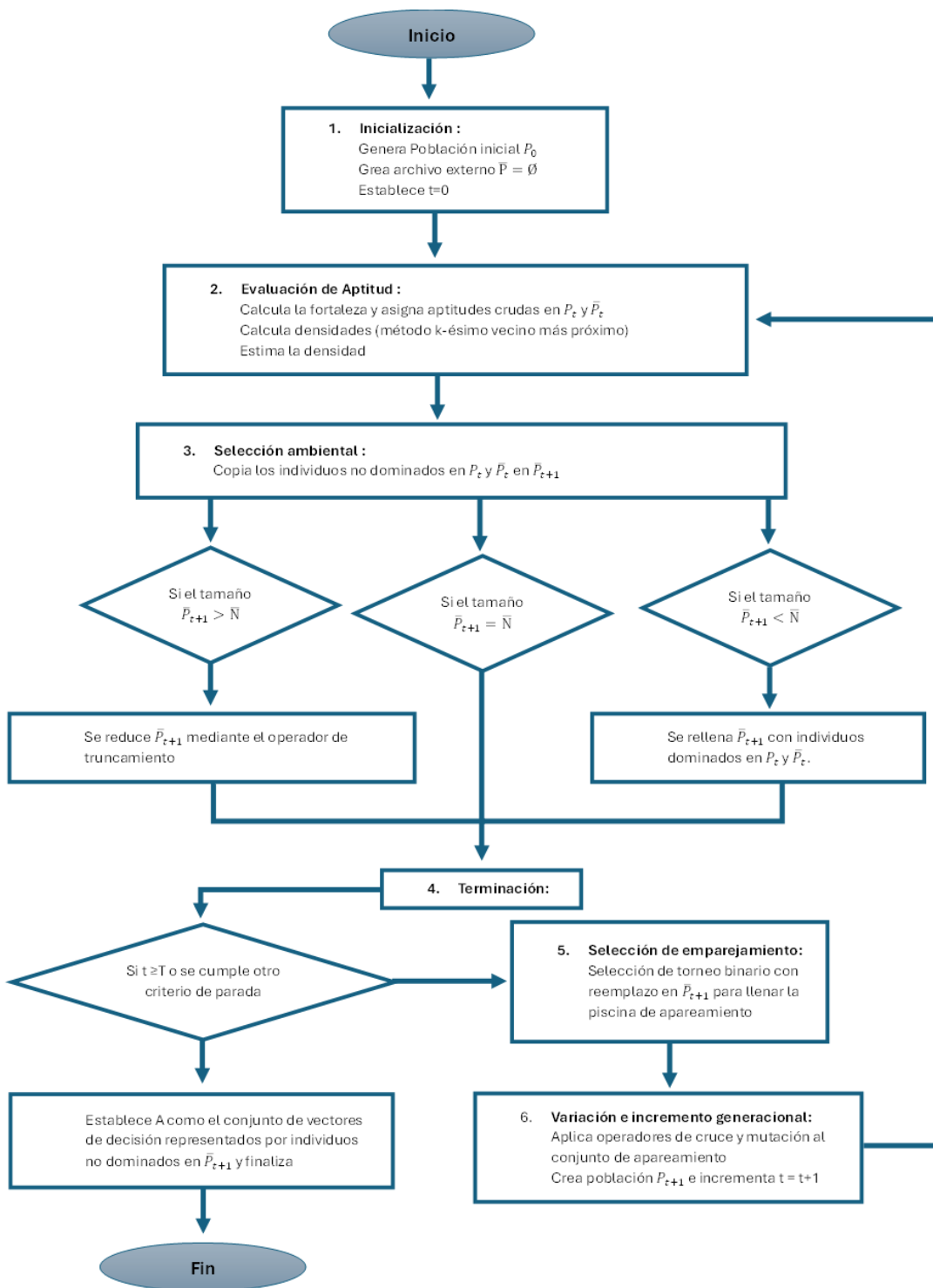


Figura 3.1 Diagrama de flujo SPEA2



### 3.1.2 Principios clave y características

Según Zitzler et al., (2001), SPEA2 se destaca por varios principios y características fundamentales:

- **Evaluación de Dominancia:** SPEA2 utiliza la evaluación de dominancia para determinar si una solución es mejor que otra en términos de múltiples objetivos. Una solución se considera dominante sobre otra si es al menos tan buena en todos los objetivos y estrictamente mejor en al menos uno de ellos.
- **Archivo Externo:** El algoritmo incorpora un archivo externo que actúa como un repositorio de soluciones no dominadas. El archivo externo garantiza que las soluciones de alta calidad se conserven y se utilicen para guiar la evolución de la población. Este archivo se actualiza continuamente durante la ejecución del algoritmo y garantiza que las mejores soluciones se conserven y guíen la evolución de la población.
- **Equilibrio entre Exploración y Explotación:** SPEA2 busca un equilibrio entre explorar nuevas regiones del espacio de soluciones y explotar las regiones conocidas del frente de Pareto. Esto se logra mediante un proceso de selección que tiene en cuenta la densidad de las soluciones en el archivo externo y la calidad de las soluciones en la población actual. El objetivo es mantener una población diversa y avanzar hacia el frente de Pareto óptimo.
- **Convergencia:** A medida que avanza a través de las generaciones, SPEA2 trabaja para mejorar gradualmente el frente de Pareto. La convergencia se logra a medida que el archivo externo contiene soluciones de alta calidad y la población evoluciona para reflejar estas soluciones.

En resumen, la descripción general del algoritmo SPEA2 se basa en la evolución de una población de soluciones mediante operadores genéticos, la evaluación de dominancia, el mantenimiento de un archivo externo y el equilibrio entre exploración y explotación. Estos

---

elementos permiten a SPEA2 encontrar soluciones no dominadas en problemas de optimización multiobjetivo de manera efectiva y eficiente

### 3.1.3 Comparación con otros enfoques multiobjetivo

SPEA2 ha demostrado su eficacia en la resolución de problemas de optimización multiobjetivo en comparación con otros algoritmos (Zitzler et al., 1999). A menudo supera a sus predecesores en términos de calidad de las soluciones encontradas y eficiencia computacional. Esta superioridad se debe a su enfoque en el mantenimiento de un archivo externo de soluciones no dominadas y su equilibrio entre exploración y explotación.

En resumen, los fundamentos del algoritmo SPEA2 se basan en su estructura de algoritmo genético, la evaluación de dominancia, el uso de un archivo externo y su capacidad para lograr un equilibrio óptimo entre exploración y explotación. Estos fundamentos le permiten sobresalir en la resolución de problemas de optimización multiobjetivo, convirtiéndolo en una herramienta valiosa en la búsqueda de soluciones óptimas en problemas complejos

## 3.2 PROCEDIMIENTO BÁSICO DE SPEA2:

En el siguiente apartado, exploraremos en detalle el procedimiento básico del algoritmo SPEA2 (Strength Pareto Evolutionary Algorithm 2). Este algoritmo es ampliamente reconocido en la comunidad de optimización multiobjetivo debido a su capacidad para encontrar conjuntos de soluciones de alta calidad en problemas con múltiples objetivos y restricciones. Vamos a desglosar sus componentes clave según lo explican Zitzler et al., (2001) y entender cómo contribuyen a la búsqueda eficaz de soluciones no dominadas en el espacio de búsqueda multiobjetivo.

SPEA2 se controla mediante tres parámetros clave de entrada:

- **Tamaño de la Población (N):** Este parámetro define cuántas soluciones candidatas se mantendrán y evolucionarán en cada generación del algoritmo. Un valor adecuado de N depende de la complejidad del problema y la capacidad computacional disponible. Se define un tamaño inicial de población, que puede variar según el

problema y las restricciones de recursos disponibles. Una población más grande puede aumentar la diversidad, pero también puede requerir más tiempo de procesamiento.

- **Tamaño del Archivo Externo ( $\bar{N}$ ):** Indica el límite máximo de soluciones no dominadas que se almacenarán en un archivo externo a lo largo de la ejecución del algoritmo. La elección de  $N$  afecta la diversidad y calidad de las soluciones encontradas.
- **Número Máximo de Generaciones (T):** Este parámetro establece un límite en la cantidad de generaciones que el algoritmo realizará antes de detenerse. Ayuda a controlar la duración de la ejecución y se ajusta según la complejidad del problema.

Por otro lado, la salida (output) del algoritmo genético SPEA2 es de suma importancia, ya que es la culminación de un proceso de búsqueda que busca encontrar un conjunto de soluciones óptimas en problemas multiobjetivo. A medida que el algoritmo evoluciona y explora el espacio de soluciones, acumula un conjunto de soluciones no dominadas, lo que significa que cada una de ellas representa un equilibrio entre los múltiples objetivos del problema.

- **Conjunto de Soluciones No Dominadas (A):** La salida deseada del algoritmo es un conjunto de soluciones que son óptimas en el sentido de que no pueden mejorarse en ningún objetivo sin empeorar al menos uno de los otros objetivos. Estas soluciones representan un equilibrio entre los objetivos múltiples del problema y proporcionan opciones de alta calidad para la toma de decisiones.

### 3.2.1 **Inicialización de la población:**

La etapa inicialización es fundamental en cualquier algoritmo genético, incluido SPEA2. Aquí se crea una población inicial de soluciones candidatas a partir de las cuales el algoritmo evolucionará en busca de soluciones óptimas. En este primer paso, se genera una población inicial  $P_0$  que consiste en un conjunto de soluciones candidatas. Además, se crea un conjunto vacío llamado archivo externo (external set)  $\bar{P}$ , que se utiliza para almacenar las soluciones no dominadas

---

a lo largo de la ejecución del algoritmo. La variable  $t$  se inicializa en cero para llevar un seguimiento de las generaciones. A continuación, se detalla el proceso de inicialización en SPEA2:

1. Creación de Individuos Iniciales: Se genera un conjunto de soluciones candidatas al azar o mediante un método específico según las características del problema. La diversidad en la población inicial es importante para explorar ampliamente el espacio de soluciones y evitar la convergencia temprana hacia un conjunto limitado de soluciones no dominadas.
2. Evaluación de Aptitud Inicial: Cada solución generada en el paso anterior se evalúa mediante las funciones objetivo definidas para el problema. Estas funciones proporcionan una medida de la calidad de la solución en relación con los objetivos del problema. La forma en que se generan las soluciones depende de la representación de las variables y la naturaleza del problema. Por ejemplo, en problemas de programación, las soluciones podrían representar programaciones de tareas y recursos.
3. Creación del Archivo Externo: Se crea un archivo externo (external archive) inicialmente vacío para almacenar soluciones no dominadas. Este archivo es esencial para llevar un registro de las soluciones de alta calidad a lo largo de la ejecución del algoritmo.
4. Inicialización de Parámetros: Se inicializan otros parámetros importantes, como el contador de generaciones ( $t$ ) a cero, que se utilizará para controlar el progreso del algoritmo.

La etapa de inicialización sienta las bases para la búsqueda subsiguiente en SPEA2. La población inicial, junto con el archivo externo, proporciona el punto de partida para la exploración y evolución de soluciones no dominadas a lo largo de las generaciones. Es importante destacar que la calidad de la población inicial puede influir en la eficacia global del algoritmo. Por lo tanto, se deben utilizar estrategias de inicialización adecuadas para abordar eficazmente el problema multiobjetivo en cuestión.

### 3.2.2 Evaluación de la aptitud (fitness Assignment) de las soluciones:

En SPEA2, la evaluación de aptitud es un paso crucial que determina la calidad de cada individuo en la población y el archivo externo. A continuación, se describe cómo se lleva a cabo este proceso en SPEA2 de una manera diferente pero equivalente:

**Cálculo de la Fortaleza (Strength):** Con el fin de evitar que varios individuos dominados por los mismos miembros del archivo tengan la misma aptitud, SPEA2 considera tanto las soluciones que un individuo domina como las que lo dominan. La fortaleza  $S(i)$  de cada individuo  $i$  en el archivo  $\bar{N}$  y la población  $P_t$  se calcula para representar cuántas soluciones están siendo dominadas por él.

**Cálculo de la “Raw Fitness”:** La “Raw fitness”  $R(i)$  de un individuo  $i$  se deriva de la fuerza de los individuos que lo dominan tanto en el archivo como en la población. Esto se calcula utilizando la siguiente fórmula:  $R(i) = \sum S(j)$ , representando el sumatorio de todas las fortalezas de los individuos que lo dominan. Vale la pena destacar que  $R(i)$  se minimiza; es decir,  $R(i) = 0$  indica que un individuo no está dominado, mientras que un valor elevado de  $R(i)$  sugiere que  $i$  es dominado por muchos individuos (que a su vez dominan a muchos otros).

**Estimación de Densidad (Density):** La “Raw fitness” proporciona un mecanismo de nicho basado en la dominancia de Pareto. Sin embargo, esta “Raw fitness” podría no ser efectiva cuando la mayoría de los individuos no se dominan mutuamente. Por lo tanto, SPEA2 incorpora información adicional mediante la densidad para diferenciar entre individuos con valores de “Raw fitness” iguales. Se utiliza una técnica de estimación de densidad basada en el método del  $k$ -ésimo vecino más cercano (Silverman 1986). Se calculan y almacenan las distancias de cada individuo  $i$  a todos los demás individuos en el archivo y la población. Luego, estas distancias se ordenan y se determina la distancia al  $k$ -ésimo vecino más cercano, denominada  $\sigma_i^{(k)}$  donde  $k = \sqrt{P_t + \bar{N}}$ , es decir, el coeficiente  $k$  es la raíz cuadrada de la suma del tamaño de la población y del archivo, en otras palabras, se establece  $k$  como la raíz cuadrada del tamaño de la muestra. Finalmente, la densidad  $D(i)$  de un individuo  $i$  se define mediante la ecuación 2.23 como:

$$D(i) = \frac{1}{\sigma_i^{(k)} + 2} \quad (\text{Ec. 3.1})$$

Añadimos ese (+2) al denominador debido a que queremos que nuestra solución de densidad esté entre  $0 < D(i) < 1$ .

Este valor de densidad se suma a la “Raw fitness”  $R(i)$  para calcular la “Raw fitness” final  $F(i)$  del individuo. La evaluación de aptitud en SPEA2 desempeña un papel esencial al evaluar la calidad relativa de cada individuo. La inclusión de información sobre la densidad permite diferenciar soluciones con “Raw fitness” idénticas, lo que mejora la capacidad del algoritmo para explorar y explotar eficazmente el espacio de búsqueda.

### **3.2.3 Cálculo de dominancia y densidad de Pareto:**

Se refiere al proceso de determinar qué soluciones son dominadas por otras y cómo se asigna la densidad a cada solución. En este paso, se realiza una comparación exhaustiva de todas las soluciones dentro de la población y el archivo (un conjunto externo de soluciones no dominadas) para determinar las relaciones de dominancia entre ellas.

La dominancia se basa en la relación Pareto, que establece que una solución A domina a otra solución B si A es al menos igual de buena que B en todos los objetivos y mejor en al menos un objetivo.

Para cada solución en la población y el archivo, se calcula su fuerza (strength) en función de cuántas soluciones dominadas y cuántas soluciones dominan. Esto se hace para evitar que dos soluciones dominadas tengan el mismo valor de “fitness”.

Luego, se calcula la densidad de Pareto para cada solución. La densidad mide como de cerca están otras soluciones de la solución dada en el espacio objetivo. Esto se logra calculando la distancia de la solución a sus vecinos más cercanos. Cuanto más densamente rodeada esté una solución, mayor será su densidad.

La combinación de la fuerza y la densidad permite asignar un valor de aptitud (fitness) a cada solución. Este valor se utiliza para determinar cuáles soluciones tienen más probabilidades de ser seleccionadas como padres en la etapa de reproducción del algoritmo, lo que influye en la evolución de la población hacia soluciones no dominadas y bien distribuidas en el espacio objetivo.

En resumen, el cálculo de dominancia y densidad de Pareto es esencial en SPEA2 para identificar y seleccionar las soluciones más prometedoras y contribuir a la mejora de la población de soluciones no dominadas.

#### Cálculo de Dominancia (Strength):

1. Para cada solución individual (individuo) en la población actual y el archivo de soluciones no dominadas (external archive), se calcula cuántos individuos dominan a ese individuo y cuántos individuos son dominados por él.
2. Un individuo A se considera dominado por otro individuo B si B es al menos igual de bueno que A en todos los objetivos y mejor que A en al menos un objetivo. Este cálculo se realiza para todos los individuos en la población y el archivo.
3. La fuerza (Strength) de un individuo se define como el número de individuos que lo dominan. Cuanto mayor sea la fuerza de un individuo, menos dominado está en comparación con otros individuos.

#### Cálculo de Densidad de Pareto:

1. Para cada individuo en la población y el archivo, se calcula su densidad de Pareto, que mide cuán cerca están otros individuos de él en el espacio objetivo.
2. Para calcular la densidad, se determina la distancia entre el individuo actual y sus vecinos más cercanos en el espacio objetivo. Esta distancia se puede calcular utilizando diversas métricas, como la distancia euclidiana o la distancia de Hamming, según el tipo de problema y los objetivos utilizados.
3. La densidad se calcula como la inversa de la distancia promedio a los k individuos más cercanos, donde k es un parámetro. La fórmula general definida anteriormente con la ecuación 2.23.

4. Usualmente, se establece  $k$  como la raíz cuadrada del tamaño de la población más el tamaño del archivo externo.

El cálculo de la fuerza y la densidad se utiliza para asignar valores de aptitud (fitness) a cada individuo en la población y el archivo. Estos valores de aptitud se utilizan en la selección de padres y en la evolución de la población en el algoritmo SPEA2 para promover soluciones no dominadas y bien distribuidas en el espacio objetivo.

#### **3.2.4 Selección de padres mediante torneo binario:**

La selección de padres mediante el torneo binario es una técnica utilizada en algoritmos genéticos, incluido el algoritmo SPEA2, para elegir qué individuos serán los progenitores de la siguiente generación. Este proceso se llama "torneo binario" porque implica emparejar a dos individuos al azar y seleccionar al mejor de los dos como padre o madre. Aquí se describe cómo funciona la selección de padres mediante torneo binario:

- 1- **Población de Torneo:** Primero, se selecciona un subconjunto aleatorio de individuos de la población actual. Este subconjunto se conoce como la "población de torneo". Por lo general, se eligen dos individuos al azar de la población de torneo para competir en cada torneo binario.
- 2- **Competencia:** Luego, los dos individuos seleccionados de la población de torneo compiten entre sí para determinar cuál de ellos será seleccionado como padre o madre. La competencia implica comparar sus valores de "fitness".
- 3- **Selección del Mejor:** El individuo con el valor de aptitud más alto entre los dos competidores se elige como padre o madre y se coloca en una "piscina de apareamiento" o "mating pool". Este individuo se considera más apto y, por lo tanto, más propenso a transmitir sus características a la descendencia.
- 4- **Repetición:** Este proceso de selección de padres mediante torneo binario se repite hasta que se haya llenado la mating pool con la cantidad requerida de padres para la siguiente generación.



El torneo binario es beneficioso porque permite una exploración más diversa del espacio de soluciones. Los individuos menos aptos aún tienen la oportunidad de ser seleccionados como padres en algunos torneos, lo que evita que la población se estanque en soluciones locales óptimas y promueve la diversidad genética en la población.

En el contexto de SPEA2, esta técnica de selección de padres se utiliza para formar la mating pool a partir de la población de individuos en cada generación. Los padres en la mating pool se utilizarán luego en los operadores de cruce y mutación para generar la siguiente generación de individuos.

### **3.2.5 Operadores de cruce y mutación:**

En este paso, se lleva a cabo la creación de una nueva población a partir de los individuos seleccionados como padres en el paso anterior (Selección de padres mediante torneo binario). Esto se logra mediante dos operadores genéticos esenciales: el operador de cruce “crossover” y el operador de mutación.

- **Operador de Cruce (Crossover):** Se usa para la exploración. El operador de cruce implica la combinación de dos o más soluciones padres seleccionadas para crear una o más soluciones descendientes. La forma en que se realiza el cruce puede variar según la implementación y la naturaleza del problema. Comúnmente se utilizan técnicas como el cruce en un punto, el cruce en dos puntos, el cruce uniforme, entre otros. Este proceso permite que las soluciones compartan información genética útil, generando descendientes que heredan características prometedoras de sus padres. En este trabajo utilizaremos el cruce Binario (SBX, Simulated Binary Crossover) el cual describe la simulación de la distribución binomial y permite controlar la probabilidad de intercambio y la extensión de la distribución. Se ha decidido utilizar este tipo de cruce puesto que permite una buena exploración en problemas continuos y es ajustable mediante parámetros.

- **Operador de Mutación:** Es utilizado para la explotación. El operador de mutación introduce pequeñas modificaciones aleatorias en las soluciones descendientes. Esto puede incluir cambios en uno o más genes de una solución. La mutación es importante para mantener la diversidad genética en la población y explorar nuevas regiones del espacio de búsqueda. La probabilidad de mutación y la magnitud de las mutaciones suelen ser parámetros ajustables.

Hay diferentes tipos de métodos de mutación como la simple la uniforme o no uniforme, por inserción, por intercambio, etc. En nuestro trabajo hemos decidido aplicar la mutación uniforme. Mediante esta mutación agregamos una variación aleatoria en los genes de los individuos.

Esta mutación se realiza ajustando los genes de los individuos (Hijos). Se generan valores aleatorios (delta) y se suman o restan a los genes dependiendo de otro conjunto de valores aleatorios y la probabilidad de mutación. La mutación está limitada para asegurarse de que los genes resultantes estén dentro del rango  $[0, 1]$ .

La combinación de los operadores de cruce y de mutación permite generar una nueva población de soluciones, llamada "población hija". Estas soluciones hijas se someten a evaluación de "fitness" en el siguiente ciclo del algoritmo. Es importante destacar que la elección adecuada de los operadores de cruce y mutación, así como sus parámetros, puede influir significativamente en el rendimiento y la eficacia del algoritmo SPEA2.

El proceso de operadores de cruce y mutación se repite hasta que se ha generado la población completa para la siguiente generación. Luego, se avanza al último paso del procedimiento básico de SPEA2: la generación de la siguiente población (paso 6 - Apartado 3.2.6).

Este proceso de combinación y mutación genética, junto con la selección de padres basada en torneos y el cálculo de dominancia y densidad de Pareto, constituye el núcleo del algoritmo SPEA2 para la búsqueda y optimización de soluciones multiobjetivo.

---

### **3.2.6 Generación de la siguiente población:**

En este último paso del procedimiento básico de SPEA2, se forma la población de la siguiente generación. Esto se logra mediante la combinación de la población hija, que se obtuvo mediante los operadores de cruce y mutación, con la población actual (Zitzler, et al., 2001). Sin embargo, dado que SPEA2 tiene como objetivo mantener una población de tamaño constante, es necesario realizar un proceso de selección para elegir cuáles individuos se mantendrán en la población y cuáles serán reemplazados.

El proceso de selección implica comparar los individuos de ambas poblaciones (actual y hija) utilizando el concepto de dominancia de Pareto y su densidad, que se calculó previamente. Se eligen los individuos no dominados de ambas poblaciones y se añaden a la población de la siguiente generación hasta que se alcance el tamaño deseado.

Si la población actual y la población hija en conjunto tienen más individuos no dominados de los que se permite en la población de la siguiente generación, se utiliza el operador de truncamiento para seleccionar los individuos más prometedores. Este operador mantiene a los individuos más sobresalientes según su aptitud en la población y descarta los demás.

El proceso se repite durante un número fijo de generaciones ( $T$ ), o hasta que se cumpla algún otro criterio de parada predefinido. Al final del algoritmo, la población resultante (normalmente, población de la última generación) representa el conjunto de soluciones no dominadas, es decir, el conjunto de Pareto, que proporciona soluciones óptimas en términos de los múltiples objetivos del problema.

## **3.3 ACTUALIZACIÓN DEL ARCHIVO EXTERNO:**

### **3.3.1 Selección de los mejores individuos:**

En el proceso de actualización del archivo externo de SPEA2, se da prioridad a la preservación de soluciones de alta calidad, es decir, aquellas que están más cerca del frente de

---

Pareto verdadero. Esto se logra seleccionando los mejores individuos tanto de la población actual como de la población hija generada en la etapa anterior.

Primero, se combinan ambas poblaciones en un solo conjunto. Luego, se calcula el valor de “fitness” de cada individuo en este conjunto utilizando las métricas de dominancia y densidad de Pareto, similares a las empleadas en las etapas anteriores del algoritmo. Esta puntuación conjunta se utiliza para determinar la calidad relativa de los individuos en comparación con otros (Zitzler et al, 2001).

A continuación, se realiza una selección basada en clasificación, donde los individuos se ordenan según sus puntuaciones de aptitud de menor a mayor. Seleccionar los mejores individuos significa tomar los primeros N, donde N es el tamaño deseado del archivo externo. Estos individuos se añaden al archivo externo y constituyen el conjunto de soluciones no dominadas más prometedoras encontradas hasta el momento.

Esta selección de los mejores individuos garantiza que el archivo externo se mantenga enriquecido con soluciones de alta calidad a lo largo de las generaciones, lo que contribuye a que el algoritmo converja hacia el frente de Pareto óptimo.

### **3.3.2 Mantenimiento del archivo externo:**

En el algoritmo SPEA2, el archivo externo juega un papel fundamental en la búsqueda y conservación de soluciones no dominadas. Para mantener su calidad y diversidad, se implementa un mecanismo de eliminación periódica y selección de las soluciones más prometedoras. El proceso se desarrolla de la siguiente manera según explican Zitzler et al., (2001):

- **Eliminación de Soluciones Redundantes:** Periodicamente, se realiza una limpieza en el archivo externo para eliminar soluciones que sean redundantes o estén muy cerca unas de otras en el espacio de objetivos. Esto se hace para evitar la congestión de soluciones en regiones específicas del frente de Pareto.

- **Selección de las Soluciones más Prometedoras:** Luego de la eliminación, se seleccionan las soluciones restantes en función de sus métricas de dominancia y densidad de Pareto. Este proceso es similar a la selección de los mejores individuos en la población principal, pero se enfoca en el archivo externo.
- **Actualización del Archivo Externo:** Finalmente, el archivo externo se actualiza con las soluciones seleccionadas. Estas soluciones representan un conjunto diverso y de alta calidad de soluciones no dominadas encontradas durante la ejecución del algoritmo.

La periodicidad de estos pasos de mantenimiento puede variar según la configuración del algoritmo o la naturaleza del problema, pero su función principal es asegurar que el archivo externo mantenga una representación efectiva del frente de Pareto a lo largo del tiempo.

### **3.4 CRITERIOS DE PARADA Y SOLUCIÓN FINAL:**

#### **3.4.1 Condiciones de terminación o parada:**

El algoritmo SPEA2 necesita criterios de parada claros para determinar cuándo debe finalizar su ejecución. Estos criterios de terminación garantizan que el algoritmo no siga ejecutándose innecesariamente una vez que se ha alcanzado una solución satisfactoria o cuando se ha agotado el número máximo de generaciones permitido. A continuación, se describen algunos de los criterios de terminación comunes utilizados en SPEA2:

- **Número Máximo de Generaciones (T):** Una condición de terminación común es especificar un número máximo de generaciones (T). Cuando el algoritmo ha generado T generaciones de población sin mejoras significativas en el frente de Pareto, se detiene.
- **Convergencia del Archivo Externo:** El algoritmo también puede terminar si el archivo externo no muestra mejoras durante un número determinado de generaciones. Esto indica que el algoritmo ha convergido y no se esperan mejoras adicionales en las soluciones encontradas.

- **Criterio de Calidad de las Soluciones:** Se puede establecer un umbral de calidad para las soluciones encontradas. Si el algoritmo encuentra al menos una solución que cumple con este umbral, puede detenerse, ya que se ha alcanzado un nivel de calidad deseado.
- **Límite de Evaluaciones de Aptitud:** En algunos casos, es importante limitar el número total de evaluaciones de aptitud realizadas. Si se alcanza este límite sin encontrar soluciones satisfactorias, el algoritmo puede finalizar.
- **Estabilidad del Frente de Pareto:** Se puede monitorear la estabilidad del frente de Pareto. Si durante varias generaciones el frente de Pareto no cambia significativamente, el algoritmo puede considerarse estable y terminar.

Estos criterios de terminación son fundamentales para controlar la ejecución del algoritmo y garantizar que no se ejecute indefinidamente. La elección de un criterio de terminación depende de la naturaleza del problema y los objetivos del usuario.

### 3.4.2 Selección de la mejor solución:

Una vez que el algoritmo SPEA2 ha alcanzado su criterio de parada y se ha detenido, es necesario seleccionar la mejor solución del conjunto de soluciones no dominadas en el archivo externo de Pareto. Este paso es crucial para proporcionar una solución óptima o bien equilibrada al usuario, dependiendo de las preferencias y necesidades específicas del problema.

La elección de la mejor solución generalmente se basa en los objetivos específicos del problema (Lesinski, 2016). Si el objetivo es encontrar una única solución óptima, se puede seleccionar la solución que tenga el valor mínimo en una de las funciones objetivo (por ejemplo, minimizar el costo total). Si se busca un conjunto diverso de soluciones, se pueden aplicar métodos de análisis multicriterio para seleccionar soluciones que cubran diferentes compromisos entre los objetivos.

Además, la selección de la mejor solución puede depender de las preferencias del usuario. Algunos algoritmos SPEA2 permiten a los usuarios explorar el conjunto de soluciones no dominadas y seleccionar la que mejor se ajuste a sus necesidades.

La elección de la mejor solución debe estar respaldada por un análisis adicional, como la revisión de las funciones objetivo, el rendimiento en condiciones diversas o el impacto de las soluciones en el problema real. (Zitzler et al., 1999).

### 4 SIMULACIÓN Y RESULTADOS DEL SPEA2 EN MATLAB

En el marco de este estudio, el Capítulo 4 se dedica a explorar la aplicación y los resultados del algoritmo genético Strength Pareto Evolutionary Algorithm 2 (SPEA2) en la resolución del Problema de Programación Multiproyecto Descentralizado con Restricción de Recursos (DRCMPSP).

La programación multiproyecto es una disciplina esencial en la gestión de proyectos que involucra la asignación de recursos limitados a múltiples proyectos para optimizar objetivos relacionados con el tiempo y los costos. Resolver eficazmente el DRCMPSP es un desafío crucial en este contexto.

El Capítulo 4 se enfoca en la simulación y evaluación de los resultados obtenidos al aplicar el algoritmo SPEA2 a este problema específico. Para llevar a cabo esta tarea, hemos aprovechado la potencia y versatilidad del entorno de programación MATLAB tomando como referencia los problemas de la librería MPSPLib.

A lo largo de esta sección, exploraremos en detalle la metodología empleada, los parámetros configurados y, lo que es más importante, los resultados obtenidos en esta investigación, tomando como ejemplo dos funciones objetivo presentadas anteriormente en el capítulo 2.

Al hacerlo, contribuimos al avance en la resolución de problemas complejos de programación multiproyecto y proporcionamos información valiosa para la toma de decisiones en la gestión de proyectos en diversos sectores industriales.

En última instancia, este capítulo será fundamental para comprender la efectividad de SPEA2 en la optimización de proyectos multiproyecto con recursos limitados y descentralizados, y, por supuesto, los resultados clave que hemos obtenido en la aplicación de SPEA2 al DRCMPSP.



### 4.1 LIBRERÍA MPSPLib

Las diferentes simulaciones realizadas para la presente de este trabajo se han llevado a cabo tomando como base los problemas presentes en la librería MPSPLib, el cual abarca un conjunto de 140 problemas de DRCMPSP, cada uno representando un portafolio de proyectos.

La cantidad de proyectos en estos problemas varía entre 2, 5, 10 y 20. Además, la cantidad de actividades en cada proyecto varía entre 30, 90 y 120 actividades. La combinación de distintos números de proyectos y actividades da origen a los mencionados 140 problemas, los cuales presentan ciertas características comunes:

- Todos los problemas disponen de cuatro recursos (R1, R2, R3 y R4), que pueden ser de carácter global, es decir, compartidos por todos los proyectos, o particulares, restringidos a un proyecto específico.
- En todos los problemas, al menos un recurso es de naturaleza global, ya que, si todos fueran recursos particulares, el problema se descompondría en varios problemas independientes de RCPS.
- Cada recurso es de tipo renovable, y su disponibilidad máxima se mantiene constante a lo largo del período considerado en los problemas.

Cada problema recibe una denominación única que refleja sus características. En primer lugar, se utiliza el prefijo "mp" para indicar que se trata de un problema de múltiples proyectos (multi-project). En segundo lugar, se representa el número de actividades en cada proyecto mediante los caracteres "j30," "j90," o "j120." En tercer lugar, se utiliza la letra "a" seguido del número de proyectos en el problema, como "a2," "a5," "a10," o "a20." En cuarto lugar, se encuentra el par de caracteres "nr" seguidos de un número ordinal para distinguir problemas con las mismas características anteriores. Por último, en quinto lugar, puede incluirse "AgentCopp" para indicar que el problema incluye cuatro recursos, todos los cuales son globales (Villafañez, 2014).

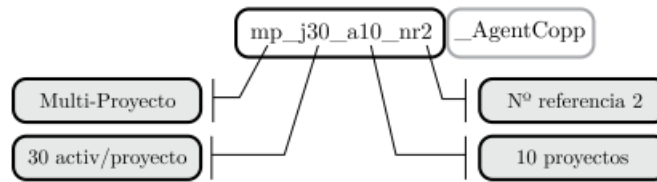


Figura 4.1 Codificación de los problemas de la librería MPSPLIB. Fuente: (Villafañez, 2014).

Cada problema se describe mediante un archivo con extensión .xml. En estos archivos, se detallan las particularidades y especificaciones del problema, incluyendo el nombre del problema (Figura 4.2-I), los nombres de los proyectos involucrados (Figura 4.2-II), las fechas de inicio de cada proyecto (Figura 4.2-III) y las restricciones o límites de los recursos globales (Figura 4.2-IV).

```

▼ <mp-list>
  ▼ <mp>
    I <name>mp_j30_a2_nr1</name>
      ▼ <project-list>
        ▼ <project>
          II <filename>KolischInstanzen/j30/j3015_5.sm</filename>
          III <start>0</start>
        </project>
        ▼ <project>
          <filename>KolischInstanzen/j30/j3015_5.sm</filename>
          <start>0</start>
        </project>
      </project-list>
      ▼ <resources>
        IV <resource>32</resource>
          <resource>62</resource>
          <resource>0</resource>
          <resource>0</resource>
        </resources>
      </mp>
    </mp-list>
  
```

Figura 4.2 ejemplo del fichero de terminación .xml.

### 4.2 LANZAMIENTO DE SIMULACIÓN EN MATLAB.

En este apartado se explicará en detalle cómo utilizar el programa SPEA2 en MATLAB centrándonos en los principales ficheros en los que se introducirán los “inputs” o datos de entrada, las diferentes funciones objetivo programadas y utilizadas y la visualización e interpretación de los resultados.

#### 4.2.1 Fichero AbrirD Entrada.m

En este bloque fichero se carga el número de proyectos y de actividades, para ello se deberán introducir los siguientes datos de entrada:

- Número de ejecuciones: realizaremos 20 en nuestro caso para realizar suficientes ejecuciones como para obtener un resultado aceptable.
- Número de Individuos: 200 para los problemas multiobjetivo, 100 para los de un único objetivo.
- Número de evaluaciones: Este valor será de 100.000 para los problemas pequeños y de 300.000 para los problemas grandes.
- Selección: Según el Torneo Binario por lo que introduciremos el valor 2.

```
NumEjecuciones=2; %luego haremos 20
NumIndividuos=200; %200 siempre para dos objetivos
NumEvaluaciones=100000; %100000 par problemas pequeños 300000 para grandes
Sel=2; %Torneo binario
```

*Figura 4.3 AbrirD\_Entrada.m Datos de entrada*

Además, deberemos indicar las dos funciones objetivo que queremos estudiar, las cuales han sido programadas en el programa y numeradas. Se indicará cada uno de estos dos números en las variables de entrada FuncionOnjetivo1 y FuncionObjetivo2.

```
%Selección Funciones Objetivo:  
%1-TMS - Makespan  
%2-APD  
%3-DPD  
%4-IR1 - Iddle Resources Método1 (recursos Locales y globales)  
%5-NR - Nivelación de Recursos (recursos Locales y globales)  
%6-IR2 - Iddle Resources Método 2 (recursos Locales y globales)  
%7-IR1G - Iddle Resources Método 1 (solo globales)  
%8-NRG - Nivelación de Recursos (solo globales)  
%9-IR2G - Iddle Resources Método 2 (solo globales)  
FuncionObjetivo1=2;  
FuncionObjetivo2=4;
```

*Figura 4.4 Selección de las Funciones Objetivo en AbrirD\_Entrada.m*

### 4.2.2 Fichero Principal.m

Este fichero permitirá abrir el archivo con todos los datos de entrada, y será sobre el cual ejecutaremos e iniciaremos el programa una vez introducido los inputs en el fichero AbrirD\_Entrada.

El fichero .xml descargado de la librería MPSPLib deberá ser guardado junto con el programa con el nombre de D\_Entrada.xml, el cual tendrá el aspecto presentado en la ilustración 4.2. De este fichero se extraerán todos los datos necesarios para la ejecución del programa y la obtención de los resultados.

### 4.2.3 Funciones Objetivo

Las funciones objetivo utilizadas en el SPEA2 dentro de este programa desarrollado en MATLAB deberán ser programadas en los ficheros Evaluación.m y GuardarD\_Salida\_TODO.m.

En este caso nos centraremos tan solo en un par de funciones objetivo las cuales ejecutaremos en distintos tipos de problemas y sobre las que obtendremos los diferentes resultados pertinentes los cuales serán comentados y analizados en detalle.

Las funciones Objetivo utilizadas para este trabajo son:

- APD: Average Project Delay
- IR: Iddle Resources

Para la programación de la función objetivo del APD Average Project Delay nos hemos basado en la ecuación 2.20 presentada en el capítulo 2. La forma de presentar APD utilizada es la siguiente:

$$APD = \frac{\sum_{p=1}^P (D_p - T_{op} - ValorCP_p)}{P}$$

$$D_p = \text{duración del proyecto } p$$

$$T_{op} = \text{Tiempo de llegada del proyecto } p$$

$$Valor CP_p = \text{Tiempo total del camino crítico}$$

$$P = \text{número de proyectos}$$

Por otro lado, si nos referimos a la función objetivo Iddle Resource nos hemos ayudado de la ecuación 2.17 también presentada anteriormente en el capítulo 2, la cual es la siguiente:

$$IR = \min \sum_{r=1}^R \left( \left( \sum_{t=1}^T R_{rt} \right) - \sum_{p=1}^P Y_p \cdot \sum_{j=1}^{J_p} d_j \cdot r_{pjr} \right)$$

$$p = \text{índice del proyecto } p = 1, \dots, P$$

$$h, j = \text{índice de actividad } h, j = 1, \dots, J_i$$

$$r = \text{Tipo de recurso } r = 1, \dots, R$$

$$t = \text{índice de tiempo } t = 1, \dots, T$$

$$R_{rt} = \text{máxima cantidad de recurso de tipo } r \text{ disponible en el periodo } t$$

$$Y_p = \begin{cases} \text{Proyecto } p \text{ es selecionado} \rightarrow Y_p = 1 \\ \text{Proyecto } p \text{ no es selecionado} \rightarrow Y_p = 0 \end{cases}$$

$d_j =$  Duración de la actividad  $j$  del proyecto  $p$

$r_{pjr} =$  Recurso del tipo  $r$  requerido en la actividad  $j$  del proyecto  $p$

Con la combinación y estudio de estas dos funciones objetivo buscamos analizar la planificación óptima de tal forma que el retraso medio del proyecto y la suma de los recursos ociosos medidos en unidades de tiempo sean mínimas quedando representadas en la curva de Pareto.

### 4.3 PRESENTACIÓN Y ANALISIS DE LOS RESULTADOS OBTENIDOS

En esta sección, se presentará un análisis detallado de los resultados obtenidos en el marco de este estudio. Tras la implementación del algoritmo genético SPEA2 en la resolución del Decentralized Resource-Constrained Multi-Project Scheduling Problem (DRCMPSP) utilizando MATLAB y la biblioteca MPSPLib, se han generado conjuntos de soluciones que representan diferentes compromisos entre los objetivos de optimización.

El objetivo de este análisis es proporcionar una comprensión profunda de cómo las soluciones se comportan en función de las funciones objetivo APD (Average Project Delay) e IR (Idle Resource).

En primer lugar, se presentarán el frente de Pareto resultante que visualiza la relación entre APD e IR. Estas curvas destacarán los compromisos inherentes entre la minimización del retraso promedio de proyectos y la maximización de la utilización de recursos disponibles.

A continuación, se explorarán casos específicos de soluciones en la curva de Pareto para ilustrar los resultados y las implicaciones prácticas. Se prestará atención a las soluciones que demuestran una eficiencia óptima y cómo estas soluciones pueden influir en la toma de decisiones en la gestión de proyectos descentralizada.

Este análisis proporcionará una visión integral de las relaciones entre APD e IR y permitirá a los responsables en la toma de decisiones de los proyectos comprender mejor cómo equilibrar estos objetivos en un entorno de gestión de proyectos descentralizado con restricciones de recursos.

Además, se examinarán posibles áreas de mejora y consideraciones para futuras investigaciones en este campo. A través de este análisis de resultados, se busca arrojar luz sobre los desafíos y oportunidades que surgen al abordar problemas complejos de programación multiproyecto en situaciones descentralizadas, lo que en última instancia contribuye al avance de la toma de decisiones eficiente en el ámbito de la gestión de proyectos.

### **4.3.1 Resultados de los lanzamientos**

En el contexto de la planificación de proyectos multiproyecto con restricción de recursos, es crucial evaluar múltiples objetivos simultáneamente para tomar decisiones informadas y equilibradas. En este trabajo, se aplicó el algoritmo genético SPEA2 a la resolución del problema DRCMPSP utilizando las funciones objetivo APD (Average Project Delay) e IR (Idle Resource).

La generación de un frente de Pareto permitió explorar el trade-off entre estos dos objetivos y proporcionar una visión clara de las soluciones óptimas disponibles. En los siguientes gráficos estudiamos diferentes simulaciones para una serie de problemas en los cuales se representan todas las soluciones encontradas durante las ejecuciones permitiendo analizar como el algoritmo mejora con cada generación obteniendo una solución más óptima mediante la descripción del frente de Pareto.

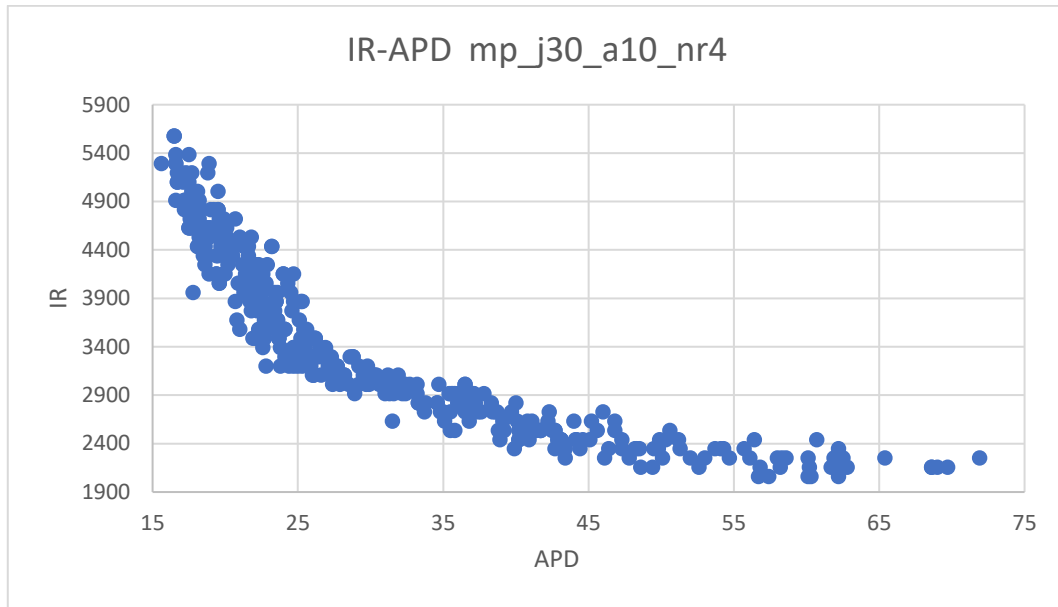


Figura 4.5 SPEA2 en problema de 30 actividades, 10 proyectos, número de referencia 4

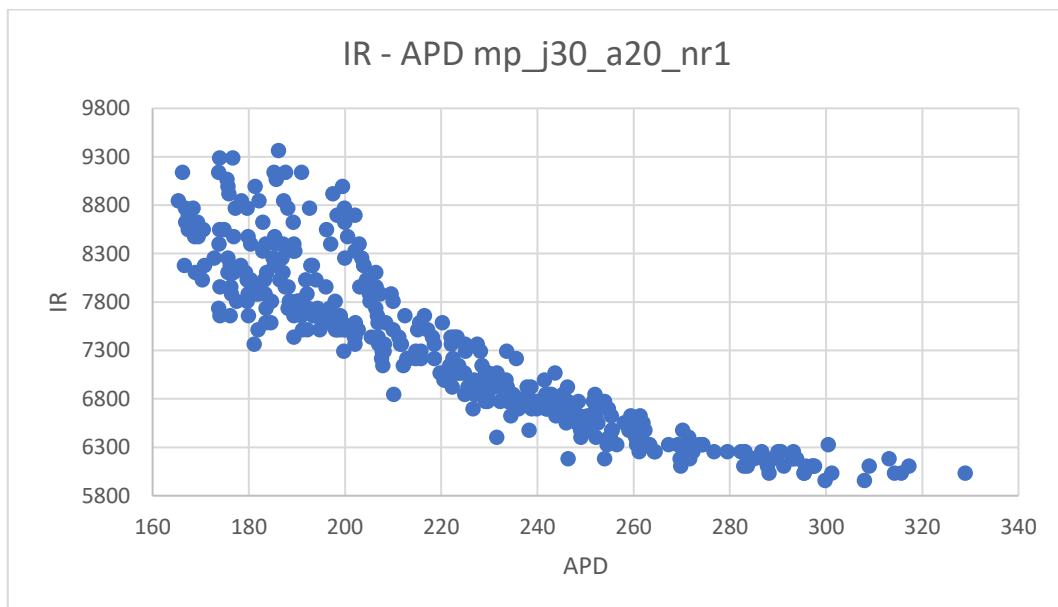
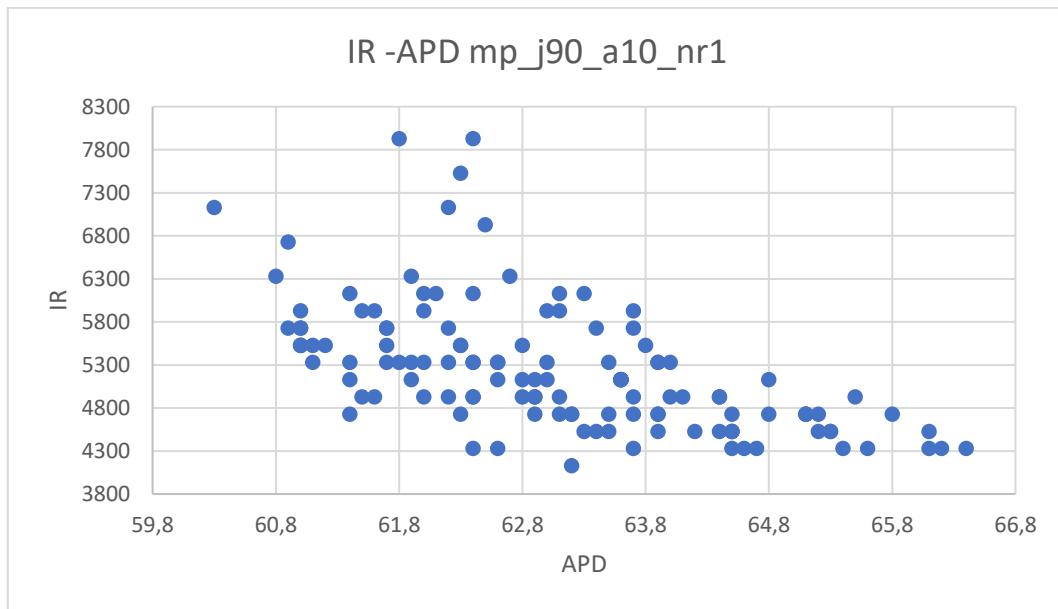
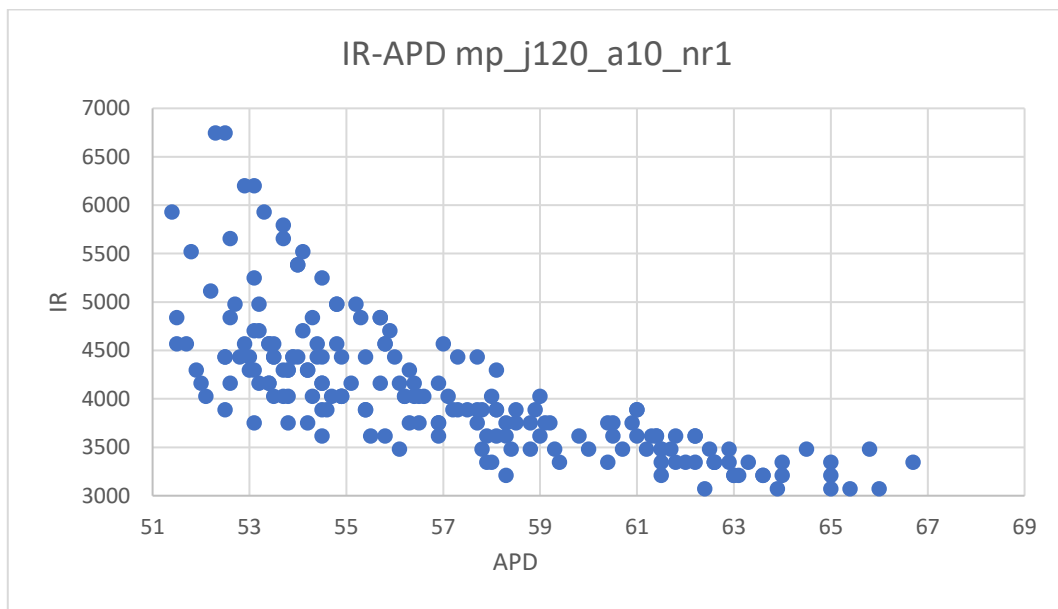


Figura 4.6 SPEA2 en problema de 30 actividades, 20 proyectos, número de referencia 1





*Figura 4.7 SPEA2 en problema de 90 actividades, 10 proyectos, número de referencia 1*



*Figura 4.8 SPEA2 en problema de 120 actividades, 10 proyectos, número de referencia 4*

Los resultados de las simulaciones revelaron un frente de Pareto bien definido que muestra la relación inversa entre el IR y el APD. A medida que se buscaba reducir el IR (indicando un uso

más eficiente de los recursos), el APD tendía a aumentar y viceversa. Esta observación confirma la naturaleza de problema multiobjetivo, donde la optimización de un objetivo puede afectar negativamente al otro.

1. **Trade-off entre el Idle Resource y el Average Project Delay:** El frente de Pareto muestra claramente el trade-off entre las dos funciones objetivo. A medida que se intenta mejorar un objetivo, como el IR, el otro, en este caso, el APD, tiende a empeorar. Esto significa que no existe una solución única que sea la mejor en ambos objetivos. Los tomadores de decisiones deben elegir un punto en el frente de Pareto que equilibre sus preferencias entre el uso eficiente de los recursos (IR) y la minimización del retraso promedio (APD).
2. **Diversidad de soluciones:** Un frente de Pareto bien distribuido puede indicar que existe una variedad de soluciones óptimas disponibles, lo que brinda a los tomadores de decisiones flexibilidad para elegir según sus preferencias y restricciones específicas.
3. **Definición de prioridades:** La elección de un punto en el frente de Pareto debe basarse en las prioridades y necesidades del problema y la organización. Si la reducción del tiempo de finalización de proyectos es de alta prioridad, se puede elegir una solución en la parte inferior izquierda del frente de Pareto. Si se prefiere conservar los recursos o minimizar el uso de recursos ociosos, se podría optar por una solución en la parte superior derecha.
4. **Sensibilidad:** El análisis de la curva de Pareto permite evaluar la sensibilidad de las soluciones a los cambios en los objetivos. Pequeñas modificaciones en las preferencias pueden resultar en cambios significativos en la solución óptima seleccionada.
5. **Toma de decisiones:** El frente de Pareto proporciona a los responsables de proyectos información sólida para tomar decisiones. Pueden evaluar diferentes escenarios y seleccionar la solución que mejor se adapte a las necesidades y restricciones de su proyecto o empresa.

Un hallazgo interesante es la diversidad de soluciones óptimas disponibles en el frente de Pareto. Este aspecto proporciona a los responsables en la toma de decisiones la flexibilidad para elegir la solución que mejor se adapte a sus preferencias y restricciones específicas. La elección del punto en el frente de Pareto dependerá de las prioridades y necesidades del proyecto o la organización. Si se valora más la eficiencia en el uso de recursos, se puede seleccionar una solución en la parte superior derecha del frente de Pareto. Por otro lado, si la minimización de las demoras es la principal prioridad, se podría optar por una solución en la parte inferior izquierda.

La sensibilidad de las soluciones a las modificaciones en las preferencias también se destacó durante el análisis. Como se ha explicado anteriormente hemos podido observar que pequeños cambios en los objetivos pueden llevar a cambios sustanciales en la solución óptima seleccionada. Esto subraya la importancia de definir claramente las prioridades y considerar diferentes escenarios al tomar decisiones.

El uso del algoritmo SPEA2 en el contexto del problema DRCMPSP con los objetivos IR (Idle Resource) y APD (Average Project Delay) presenta varias fortalezas y debilidades en función de los resultados obtenidos en este TFM.

### **Fortalezas:**

1. Optimización Multiobjetivo: SPEA2 es una técnica probada en la optimización multiobjetivo, lo que permite abordar simultáneamente múltiples objetivos en el problema DRCMPSP, como la minimización de recursos ociosos (IR) y la minimización de retrasos promedio (APD).
2. Frente de Pareto: La generación de un frente de Pareto proporciona una variedad de soluciones no dominadas que permiten a los tomadores de decisiones elegir el equilibrio deseado entre los objetivos. Esto es fundamental en situaciones donde los objetivos pueden estar en conflicto.
3. Robustez: SPEA2 es una técnica robusta que tiende a generar resultados consistentes en diferentes conjuntos de datos y condiciones, lo que garantiza una solución estable en problemas del mundo real.

### Debilidades:

1. Complejidad: SPEA2 es un algoritmo relativamente complejo en comparación con otros enfoques más simples. Requiere una configuración adecuada y una mayor cantidad de recursos computacionales.
2. Interpretación de Resultados: El análisis del frente de Pareto puede ser desafiante y requiere experiencia para interpretar las soluciones. Las relaciones existente entre los objetivos puede que no se presente de una forma evidente a simple vista.
3. Tiempo de Cálculo: La búsqueda en el espacio de soluciones multiobjetivo puede requerir más tiempo de cálculo en comparación con enfoques de optimización monoobjetivo. Esto puede ser una limitación en problemas con restricciones de tiempo estrictas.
4. Configuración de Parámetros: SPEA2 implica una serie de parámetros que deben ajustarse cuidadosamente. La elección inadecuada de parámetros puede afectar significativamente los resultados.

En resumen, el uso de SPEA2 en el problema DRCMPSP con objetivos IR y APD ofrece ventajas significativas al abordar la naturaleza multiobjetivo y de toma de decisiones descentralizada del problema. Sin embargo, la complejidad y la interpretación de resultados requieren un enfoque cuidadoso y la consideración de las necesidades específicas del caso.

En conclusión, el análisis del frente de Pareto en el contexto del problema DRCMPSP ha proporcionado una visión valiosa de las compensaciones y las soluciones óptimas disponibles. Este enfoque permite a los responsables de los proyectos tomar decisiones informadas basadas en sus prioridades y restricciones específicas. Además, destaca la naturaleza multiobjetivo y la sensibilidad de las soluciones a las preferencias.



## 5 CONCLUSIONES

La resolución del Problema de Programación de Proyectos Multiproyecto con Restricción de Recursos (DRCMPSP) es fundamental en la gestión de proyectos en diversas industrias. En este trabajo, hemos aplicado el algoritmo genético SPEA2 en MATLAB para abordar este desafiante problema con un enfoque multiobjetivo centrado en las funciones APD (Average Project Delay) e IR (Idle Resource). A través de nuestras investigaciones y análisis, hemos llegado a varias conclusiones importantes:

Los resultados de nuestras simulaciones demuestran la eficacia del algoritmo SPEA2 en la búsqueda de soluciones óptimas en un entorno de múltiples objetivos. El SPEA2 es capaz de encontrar soluciones que equilibran las demoras en los proyectos y la utilización de los recursos, lo que es esencial en situaciones donde múltiples proyectos comparten recursos limitados.

La obtención de un frente de Pareto nos proporciona un conjunto de soluciones eficientes que representan el compromiso entre APD e IR. Este frente de Pareto permite a los responsables de proyectos seleccionar soluciones óptimas según sus prioridades y restricciones específicas.

Nuestros resultados resaltan la importancia de considerar las preferencias del tomador de decisiones. Diferentes stakeholders pueden tener preferencias distintas, lo que influirá en la elección de soluciones. La sensibilidad a las preferencias destaca la necesidad de una toma de decisiones colaborativa y la flexibilidad en la elección de soluciones.

La metodología y las herramientas desarrolladas en este trabajo tienen aplicaciones en una amplia gama de industrias, incluyendo la construcción, la ingeniería, la manufactura y más. La capacidad de equilibrar las demoras en los proyectos y la utilización de los recursos es esencial para el éxito de proyectos complejos.

La planificación cuidadosa y la definición clara de preferencias son cruciales en la gestión de proyectos multiproyecto. La toma de decisiones informadas y la evaluación de diversos escenarios son elementos clave en la búsqueda de soluciones eficientes.

En resumen, este trabajo proporciona una valiosa contribución a la resolución del DRCMPSP utilizando el algoritmo SPEA2 en un contexto multiobjetivo. Las soluciones encontradas son sensibles a las preferencias, lo que subraya la necesidad de una toma de decisiones estratégica y colaborativa. Este enfoque tiene aplicaciones en diversas industrias y puede contribuir a la eficiencia y la rentabilidad en la gestión de proyectos multiproyecto con restricción de recursos.

Para futuras investigaciones, se pueden explorar enfoques adicionales para abordar el DRCMPSP, como algoritmos de optimización multiobjetivo alternativos y técnicas de gestión de proyectos más avanzadas. Además, se pueden considerar casos de estudio específicos en diversas industrias para evaluar la aplicabilidad y la eficacia de este enfoque en situaciones del mundo real. La mejora continua de los algoritmos y la adaptación a desafíos emergentes en la gestión de proyectos serán áreas clave para futuras investigaciones.

## 6 BIBLIOGRAFÍA

Álvarez-Campana, P. (2021). Simulador para la resolución de problemas de programación multiproyecto con restricción de recursos. Tesis de Máster en Dirección de Proyectos. Departamento de Organización de Empresas y C.I.M., Universidad de Valladolid, Valladolid, España.

Alvin, V.(2022). Multi-mode resource constrained multi project scheduling problem optimization with symbiotic organism's search. *Dimensi Utama Teknik Sipil*, Vol.9. Taiwan.

Amirian, H., Sahraeian, R. (2017). Solving a grey project selection scheduling using a simulated shuffled frog leaping algorithm. *Computers & Industrial Engineering*, 107, 141-149.

Araúzo, J. A., Pajares, J. y Lopez-Paredes, A. (2010) 'Simulating the dynamic scheduling of project portfolios', *Simulation Modelling Practice and Theory*, 18(10), pp. 1428–1441. doi: 10.1016/j.simpat.2010.04.008.

Bartusch, M., Möhring, R. H. y Radermacher, F. J. (1988) 'Quantitative Models , Data Structuring and Information Processing', *Annals of Operations Research*, 16, pp. 201–240.

Browning, T. R. y Yassine, A. A. (2010). Resource-constrained multi-project scheduling: Priority rule performance revisited. *International Journal of Production Economics*, 212–228.

Cervantes, M(2010). Nuevos Métodos Meta Heurísticos para la Asignación Eficiente, Optimizada y Robusta de Recursos Limitados. Universidad Politécnica de Valencia, Valencia España.

Coll, F. (2020). Proyecto. Economipedia.com.



Confessore, G., Giordani, S. y Rismondo, S. (2007). A market-based multi-agent system model for decentralized multi-project scheduling. *Annals of Operations Research*, 150, pp 115-135. DOI: 10.1007/s10479-006-0158-9.

Deb, K., Pratap, A., Agarwal, S., y Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

Del Angel-Martínez, E., Fraire-Huacuja, H., Soto, C., Rangel, N., Cruz-Reyes, L., y Gómez-Santillán, C. (2019). Un nuevo algoritmo de ramificación y acotamiento para el problema de la bisección de vértices. *Tecnológico Nacional de México / Instituto Tecnológico de Ciudad Madero, Tamaulipas, México; Universidad de Cádiz, Escuela Superior de Ingeniería, Cádiz, España*

Gómez, M; Lalla-Ruiz, E; Fernández, A; Castro, C y Vob, S. (2022). Resource-constrained multi-project scheduling problem: A survey. *European Journal of Operational Research*, ELSEVIER.

Hao, X., Sun, L., Gen, M., y Sun, L. (2018). Solving Robust Resource Constrained Scheduling Problem by Multi-objective Optimization Method based on Hybridization of EDA and GA. *Procedia Manufacturing*, 17, 1187-1194. DOI: 10.1016/j.promfg.2018.10.001.

Hauder, V. A., Beham, A., Raggl, S., Parragh, S. N., y Affenzeller, M. (2020). Resource-constrained multi-project scheduling with activity and time flexibility. *Computers & Industrial Engineering*. DOI: 10.1016/j.cie.2020.106904.

Homberger, J. (2007) 'A multi-agent system for the decentralized resource-constrained multi-project scheduling problem', *International Transactions in Operational Research*, 14(6), pp. 565–589. doi: 10.1111/j.1475-3995.2007.00614.x.

Krüger, D., y Scholl, A. (2008). Managing and modelling general resource transfers in multiproject scheduling. *OR Spectrum*, 32, 369-394. <https://doi.org/10.1007/s00291-008-0144-5>.

Lesinski, G. (2016). Application of the Strength Pareto Evolutionary Algorithm (SPEA2) Approach to Rail Repair Investments. Department of Systems Engineering, United States Military Academy.

Li, F., y Xu, Z. (2018). A multi-agent system for distributed multiproject scheduling with two-stage decomposition. School of Economics and Management, Beihang University, Beijing, China.

Lova, A., Maroto, C., y Tormos, P. (2000). A multicriteria heuristic method to improve resource allocation in multiproject scheduling. *European Journal of Operational Research*, 127, 408-424.

Mao, X., Salden, A. H., y Roos, N. (2009). Stable multi-project scheduling of airport ground handling services by heterogeneous agents. Conference Paper. DOI: 10.1145/1558013.1558087.

Morillo, D., Moreno, L. y Díaz, J. (2014) ‘Metodologías Analíticas y Heurísticas para la Solución del Problema de Programación de Tareas con Recursos Restringidos (RCPS): una revisión Parte 1’, *Ingeniería y Ciencia*, 10(19), pp. 247–271. doi: 10.17230/ingciencia.10.19.12.

Morillo, D., Moreno, L. y Díaz, J. (2014) ‘Metodologías Analíticas y Heurísticas para la Solución del Problema de Programación de Tareas con Recursos Restringidos (RCPS): una revisión. Parte 2’, *Ingeniería y Ciencia*, 10(20), pp. 203–227. doi: 10.17230/ingciencia.10.20.12.

Nabipoor, E., Aghaie, A., y Naja, A. A. (2018). Robust optimization for the resource-constrained multi-project scheduling problem with uncertain activity durations. *Scientia Iranica Transactions E: Industrial Engineering*, 27(1), 361-376.

Narváez, G., y Saltos, R. (2010). Implementación de un Algoritmo Genético para resolver el problema de Programación de Proyectos con Recursos Limitados. Guayaquil.

Pérez, E., Posada, M., y Lorenzana, A. (2016). Taking advantage of solving the resource constrained multi-project scheduling problems using multi-modal genetic algorithms. *Soft Computing*, 20, 1879-1896. DOI: 10.1007/s00500-015-1610-z.

Shou, Y., Xiang, W., Li, Y., y Yao, W. (2014). A Multiagent Evolutionary Algorithm for the Resource-Constrained Project Portfolio Selection and Scheduling Problem. *Mathematical Problems in Engineering*, Volume 2014, Article ID 302684, 9 pages. Hindawi Publishing Corporation. Doi: 10.1155/2014/302684.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London: Chapman and Hall.

Sindhya, K., Deb, K., y Mohan, M. (2007). Multi-objective optimization problems with resource constraints: A review. *Evolutionary Multi-Criterion Optimization*, 435-449.

Valls, V., Ballestín, F. y Quintanilla, S. (2005) 'Justification and RCPSP: A technique that pays', *European Journal of Operational Research*, 165(2), pp. 375–386. doi: 10.1016/j.ejor.2004.04.008.

Villafañez, F., López-Paredes, A., y Pajares, J., (2014) 'From the RCPSP to the DRCMPSP: Methodological foundations', in *Proceedings of the 2014 International Conference on Artificial Intelligence, ICAI 2014 - WORLDCOMP 2014*, pp. 594–598.

Villafañez, F. (2014) *Programación multi-proyecto con restricciones de financiación (F-DRCMPSP)*.

Villafañez, F., Poza, D., Pajares, J., López-Paredes, A., y Del Olmo, R (2018) 'A unified nomenclature for project scheduling problems (RCPSP and RCMPSP)', *Dirección y Organización*, 64, pp. 56–60.

Villafañez, F., Poza, D., Pajares, J., López-Paredes, A., y Del Olmo, R., (2019). A generic heuristic for multi-project scheduling problems with global and local resource constraints (RCMPSP). *Soft Computing*, 23(10), 3465–3479. doi: 10.1007/s00500-017-3003-y.

Zheng, Z., Guo, Z., Zhu, Y., y Zhang, X. (2014). A critical chains-based distributed multi-project scheduling approach. *Neurocomputing*, 143, 282-293.

Zitzler, E., Laumanns, M., y Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimization and Control* (pp. 95-100).

Zitzler, E., Laumanns, M., y Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. TIK Report 103. ETH Library. DOI: 10.3929/ethz-a-004284029.

Zitzler, E., y Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257-271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., y Da Fonseca, V. G. (2003). Performance assessment of multi-objective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2), 117-132.