

(01-030) - Review of heuristic methods based on priority rules for constraint resources project scheduling

Ponluisa Marcalla, Neisser Fernando ¹; Capuz-Rizo, Salvador F. ¹

¹ Universitat Politècnica de València

In the field of project management, activity scheduling is crucial for efficient planning and meeting project deadlines. This process is simplified when done under the assumption of unlimited resource availability (a feasible situation by acquiring or hiring resources as needed), however becomes complex when resource limitations exist. This work aims to provide a summary table of heuristic methods based on priority rules and their references, derived from a thorough literature review in the field. Additionally, five methods proven effective in scheduling are selected, then compared through simulating a fictitious project with resource limitations per time unit. Evaluation focuses on deadline compliance and resource allocation efficiency were evaluated, with MS Project scheduling used as a duration reference. Results indicate that three out of the five methods meet deadlines and efficiently optimize resources, however, it is emphasized that these heuristic methods do not guarantee the optimal solution. In summary, this study offers valuable information for selecting the most suitable method to address the RCPSP.

Keywords: Heuristic method; Priority rules; Project scheduling; RCPSP; Project management

Revisión de métodos heurísticos basados en reglas de prioridad para la programación de proyectos con recursos limitados.

En el ámbito de la gestión de proyectos, la programación de actividades es crucial para una planificación eficiente y cumplimiento de plazos establecidos del proyecto. Este proceso se simplifica cuando se realiza bajo la suposición de disponibilidad ilimitada de recursos (una situación factible al poder adquirir o contratar recursos según sea necesario), volviéndose complejo cuando existen limitaciones de recursos. El objetivo del trabajo es proporcionar una tabla resumen de los métodos heurísticos basados en reglas de prioridad y sus referencias, derivada de una revisión exhaustiva de la literatura en el campo. Además, se seleccionan cinco métodos que han demostrado eficacia en la programación, seguidamente se comparan mediante la simulación de un proyecto ficticio con limitaciones de recursos por unidad de tiempo. Se evaluó el cumplimiento de plazos y la eficiencia en la asignación de recursos, como referencia de duración se utiliza la programación realizada en MSProject. Los resultados indican que tres de los cinco métodos cumplen con los plazos establecidos y optimizan eficientemente los recursos, no obstante, se destaca que estos métodos heurísticos no garantizan la solución óptima. En resumen, este estudio ofrece información valiosa para la selección del método más adecuado para abordar el RCPSP.

Palabras clave: Métodos Heurístico; Reglas de prioridad; Programación de Proyectos; RCPSP; Dirección de Proyectos

Correspondencia: Neisser Ponluisa Correo: neisser.ponluisa@outlook.com

Agradecimientos: Mi profundo agradecimiento a Salvador Capuz-Rizo mi tutor y a los profesores del Máster de Dirección y Gestión de Proyectos de la Universitat Politècnica de València.



©2024 by the authors. Licensee AEIPRO, Spain. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introducción

En el campo de la gestión de proyectos, a menudo existe la necesidad de asignar eficazmente los recursos a las diversas actividades del proyecto para cumplir con el cronograma planteado. Si bien este proceso parece relativamente sencillo, cuando se realiza con la hipótesis de disponibilidad de recursos (en el caso que fuese real que se puede adquirir o contratar tantos recursos como sea necesario sería lo ideal), la realidad cambia drásticamente cuando se enfrenta a restricciones de recursos limitados. Estas limitaciones generan un problema que afectan negativamente al desarrollo del proyecto, siendo preciso acudir a métodos que nos permitan resolver de manera eficiente el problema de asignación de los recursos disponibles.

Existen diferentes formas de abordar el problema de programación de proyectos con recursos limitados (Resource Constrained Project Scheduling Problem, RCPSP). En la bibliografía se puede encontrar las diferentes propuestas para abordar este problema, siendo uno de los enfoques más comunes y efectivos los métodos heurísticos. Los métodos heurísticos permiten encontrar soluciones aproximadas en tiempos razonables, los cuales resultan especialmente útiles para proyectos de gran tamaño o complejos. Dentro de los métodos heurísticos se incluyen tanto los métodos metaheurísticos como los métodos basados en reglas de prioridad.

Los métodos heurísticos basados en reglas de prioridad han sido ampliamente investigados y evaluados por numerosos expertos a lo largo de los años (Ortigueira, 1976; Alvarez y Tamarit, 1989a; Brucke et al., 1999; Klein, 2000; Ibrahim y Aydemir, 2014; Azmi et al., 2011; Álvarez-Campana, 2021, etc). Existen múltiples reglas de prioridad que pueden ser utilizadas, estas reglas en su mayoría han sido comprobadas y analizadas mediante simulaciones informáticas utilizando la librería PSPLIB (Project Scheduling Problem Library) desarrollada por Kolish y Sprecher (1997). Sin embargo, la comprensión y aplicación efectiva de estas reglas puede ser desafiante para los directores de proyectos que no tengan experiencia en lenguajes de programación. A menudo, se requieren herramientas informáticas especializadas para implementar estas reglas de prioridad, porque no hay suficiente información que permita aplicarlas de manera práctica a un ejemplo real.

El propósito de este trabajo es recopilar los métodos heurísticos basados en reglas de prioridad, extraídos de una revisión de veinticinco artículos relevantes en la literatura. Se seleccionan cinco reglas de prioridad las reglas más citadas y ampliamente reconocidas para abordar el problema de la asignación de recursos en proyectos. Por otro lado, se busca aplicar estos cinco métodos a un ejemplo de proyecto y determinar cuál optimiza mejor la duración, comparando los resultados obtenidos de los cinco métodos y el resultado obtenido de un software especializado en la programación de proyectos usado para este ejemplo de proyecto. Es importante destacar que la elección del método a utilizar se debe basar en las características y restricciones del proyecto, ya que cada método tiene sus propias ventajas y limitaciones, siendo necesario evaluar cuál se ajusta mejor a las necesidades y objetivos del proyecto.

2. Revisión de la literatura de los métodos para la programación de proyectos con recursos limitados

Los métodos iniciales empleados en la programación de proyectos surgieron en la década de 1950, denominados como PERT y CPM (Fondhal, 1961). Estos métodos se centran en la vinculación de actividades, dependencias y duraciones dentro de un proyecto, facilitando la identificación de las actividades críticas, lo que es fundamental para optimizar la duración total de ejecución del proyecto (Piñeiro, 1995). La representación de la programación del proyecto se realiza comúnmente en tablas o diagramas, destacando al diagrama de Gantt por su capacidad para proporcionar una comprensión visual clara de las actividades, su secuencia

temporal y el orden de ejecución. Durante este proceso de programación, se establecen los tiempos de duración de los hitos y las actividades, asignando fechas de inicio y fin, estableciendo las precedencias, considerando los recursos disponibles y tomando en cuenta las restricciones que componen el proyecto (PMI, 2017; Wilson, 2023).

Los métodos PERT y CPM asumen una disponibilidad ilimitada de recursos, lo que implica que no considera las restricciones relacionadas con la cantidad y disponibilidad de estos recursos. Ante situaciones donde los recursos son limitados o existen restricciones específicas, se recurre a métodos que permitan resolver las limitaciones como los métodos exactos, los métodos heurísticos, entre otros, para abordar este tipo de problemas (Abdolshah, 2014).

Kolish y Hartmann (2006) destaca que, habido una evolución considerable de muchos métodos exactos y heurísticos, impulsado por los constantes requerimientos de la gestión de proyectos, lo que ha llevado al desarrollo de métodos más sofisticados y adaptativos de los ya existentes. Esto implica superar las limitaciones identificadas en los métodos tradicionales, al considerar de manera más precisa las restricciones de recursos, permitiendo así una planificación más realista y eficiente.

2.1. Métodos para RCPSP

La planificación inicial de proyectos comienza con un enfoque general que presupone la disponibilidad de recursos. Sin embargo, es fundamental comprender dos enfoques principales de planificación: la libre, que busca finalizar el proyecto lo antes posible, y la objetivo, que parte de una fecha establecida. La programación desempeña un papel fundamental al asignar recursos y fechas a cada actividad para lograr los objetivos planificados (Simón, Peña y Rincón, 2014).

Aunque la programación de proyectos puede parecer sencilla, enfrenta desafíos y limitaciones que complican su ejecución, lo que supone un reto considerable. Debido a que su objetivo principal radica en la optimización del tiempo, costos, recursos y dependencias, cumpliendo rigurosamente con las restricciones (Gálvez et al., 2016).

Por consiguiente, llevar a cabo una programación con recursos limitados implica centrarse en la gestión, la organización y el control eficiente para alcanzar los objetivos del proyecto en el plazo establecido. Cervantes (2010) subraya la necesidad de enfoques flexibles y adaptables para abordar eficazmente la diversidad de problemas de programación de proyectos, garantizando así su éxito a pesar de las limitaciones de recursos y las configuraciones únicas de la red.

Los constantes cambios en los proyectos requieren soluciones para el RCPSP. Para abordar esto, investigadores han desarrollado algoritmos y metodologías para encontrar soluciones óptimas o aproximadas. Estas contribuciones impulsan el progreso en la programación de proyectos. En esta sección, se resume la diversidad de enfoques en la Tabla 1, que clasifica los tipos de métodos para abordar eficazmente el problema mencionado.

Tabla 1. Métodos para resolver RCPSP

Métodos	Tipos	Características
Exactos	Métodos Numéricos Branch and Bound Árbol de precedencias Divide and Conquer Alternativas de modo y extensión Programación: PLE, PCE y PEM	Resolución matemática exacta. Soluciones Óptimas. Modelos Complejos.
Heurísticos	Reglas de prioridad Metaheurísticos clásicos Metaheurísticos no estándares	Soluciones Factibles. Soluciones no óptimas. Tiempos razonables. Técnicas Sencillas.

Multicriterio	Por metas Por ponderación	Establecimiento de prioridades/jerarquías. Considera los multiobjetivo/multicriterio.
Híbridos	PLE (Integer Linear Programming) y heurística Combinación de métodos heurísticos y metaheurísticos (Algoritmo genético híbrido)	Combinación de diferentes enfoques y técnicas. Explora el conocimiento del problema y encuentra soluciones mejoradas simples, veloces y eficaces.
Redes Petri	TPPN (Timed Place Petri Net)	Herramienta de modelado gráfico. Visualización clara de los procesos de programación. Permite analizar el desempeño del proyecto.
Inteligencia Artificial	Multilayer feed-forward neural network (MLFNN)	Técnica de autoaprendizaje. Eficaces en problemas complejos. Procesamiento de lenguaje natural.

Fuente: Brucker et al. (1999); Kolisch y Hartmann (2006); Valls, Ballestín y Quintanilla (2008); Morillo, Moreno y Díaz (2014a); Morillo, Moreno y Díaz (2014b); Abdolshah (2014); Mejía et al. (2015); Moradi et al. (2019); Golab et al. (2022).

Es importante destacar que cada uno de los métodos mencionados en la Tabla 1 posee sus propias ventajas y desventajas. Sin embargo, todos comparten el mismo objetivo: lograr que la programación del proyecto reduzca el tiempo y optimice los recursos para cumplir con el objetivo establecido. La elección del mejor método depende del criterio del director de proyectos y de las características particulares del proyecto en cuestión, así como de las directrices establecidas por las estrategias empresariales, así como de las herramientas que dispone para analizar la programación.

2.2. Métodos heurísticos para la programación con recursos limitados

La programación de proyectos con recursos limitados (PPRL) constituye un desafío clave en la planificación de proyectos, ya que implica la asignación eficiente de recursos a lo largo del tiempo para cada tarea del proyecto. En este contexto, los métodos heurísticos son herramientas valiosas para la gestión efectiva de proyectos, al permitir encontrar soluciones prácticas y realistas que se ajusten a las restricciones de recursos y plazos del proyecto, abordando así el problema de la programación de proyectos con recursos limitados (RCPSP - Resource-Constrained Project Scheduling Problem).

Los métodos heurísticos han demostrado ser eficaces en la resolución del RCPSP y son ampliamente reconocidos como la mejor opción. Numerosas investigaciones respaldan su éxito al optimizar los tiempos de ejecución y la nivelación de los recursos disponibles, generando resultados de mejor calidad. Estos estudios están documentados en la bibliografía del artículo.

Morillo, Moreno y Díaz (2014a; 2014b) ofrecen una revisión exhaustiva de métodos destinados a alcanzar resultados óptimos del RCPSP, resumiendo los métodos exactos aplicables principalmente en proyectos pequeños o poco complejos, así como los métodos heurísticos más relevantes. Entre estos métodos se incluyen los heurísticos primitivos, la búsqueda local, los métodos codiciosos, los métodos truncados de Branch and Bound, así como los métodos basados en Arcos disyuntivos. Estos métodos cuentan con respaldo empírico en diversas investigaciones (Alvarez y Tamarit, 1989b; Brucker et al., 1999; García y González, 2005; Kolisch y Hartmann, 2006; Valls et al., 2008; Simón et al., 2014; Ghasemi et al., 2015).

Además de métodos heurísticos anteriormente mencionados, se han desarrollado y aplicado otros enfoques para abordar la complejidad del RCPSP. Entre estos se encuentra los métodos metaheurísticos, como la búsqueda tabú, temple simulado, algoritmo genético y optimización por colonia de hormigas, entre otros. Estos métodos se destacan por su versatilidad y capacidad para adaptarse a diferentes tipos de proyectos, además, tienen la ventaja de combinarse con otros enfoques y técnicas, lo que facilita la obtención de mejores resultados

en escenarios complejos para la PPRL (Morillo, Moreno y Díaz, 2014b; Luo, Vanhoucke y Coelho, 2023).

En la búsqueda continua de soluciones innovadoras, Lerch y Trautmann (2019) proponen el algoritmo "Lazy Constraints", que agrega restricciones adicionales solo cuando sea necesario, reevaluando así la programación para encontrar soluciones más óptimas. Ortiz y Díaz (2020) presentan un método que combina programación lineal entera y mixta, formulando un modelo de optimización que maximiza el valor del proyecto considerando restricciones de recursos, asignación de tareas y programación temporal.

Dada la complejidad de los métodos heurísticos y otros enfoques para abordar el RCPSP, es vital reconocer los desafíos que enfrentan los directores de proyectos al implementarlos. Aunque ofrecen soluciones efectivas, su aplicación puede ser exigente debido a la necesidad de conocimientos especializados y habilidades técnicas. Por ende, a pesar de la disponibilidad de enfoques innovadores, los métodos heurísticos convencionales siguen siendo una opción popular y viable. Según Simón, Peña y Rincón (2014), los métodos heurísticos basados en reglas de prioridad son los más utilizados y fáciles de implementar, mientras que Song, Jia y Peng (2022) señalan que destacan por su lógica intuitiva, comprensión y facilidad de cálculo.

2.3. Métodos heurísticos basado en reglas de prioridad para la programación con recursos limitados

Los métodos heurísticos basados en reglas de prioridad se originaron en la década de 1950 con el desarrollo de los métodos PERT y CPM. Estos métodos proporcionan información clave sobre las actividades del proyecto, como fechas de inicio y finalización, holguras, dependencias y duraciones estimadas, siendo esencial para establecer reglas de prioridad que guíen la secuencia de ejecución de las actividades y la programación integral del proyecto.

Alvares y Tamarit (1989b) describen seis reglas heurísticas (MTS, GRPW, LST, LFT, RSM y CUMRED) aplicables en la programación de proyectos con recursos limitados, basadas en el diagrama PERT/CPM. Por otro lado, Kolisch y Hartmann (1999) analizan estos métodos y otros adicionales como MSLK, SPT y WCS, concluyendo que el método LFT ofrece los mejores resultados. La aplicación de estas reglas requiere definir un conjunto de tareas elegibles, un método de selección y una regla de desempate entre tareas similares. El nombre de las siglas mencionadas anteriormente se encuentra en la tabla 2.

Los métodos heurísticos basados en reglas de prioridad son ampliamente empleados por su fácil implementación y eficacia, especialmente en proyectos de gran envergadura. Chand, Singh y Ray (2019) resaltan su importancia en la toma rápida de decisiones en entornos dinámicos ofreciendo una forma eficaz de programar las actividades. Además, en proyectos más pequeños, son preferidos debido a su simplicidad y un enfoque intuitivo, priorizando la practicidad sobre la búsqueda de la optimalidad (Türkakin, Arditi y Manisali, 2021).

Por otro lado, Kolisch y Hartmann (1999) enfatizan que, aunque los métodos heurísticos basados en reglas de prioridad pueden no ofrecer los mejores resultados, siguen siendo esenciales para resolver instancias de problemas grandes en periodos cortos. Además, juegan un papel fundamental en el establecimiento de soluciones iniciales para nuevos procedimientos de métodos.

Estos métodos continúan siendo objeto de interés para los investigadores, quienes buscan encontrar la combinación óptima entre estos métodos, los enfoques heurísticos tradicionales y las metaheurísticas. Por ejemplo, Luo, Vanhoucke y Coelho (2023) han propuesto un diseño automatizado basado en reglas de prioridad utilizando programación genética asistida por modelos sustitutos, reflejando la búsqueda constante de soluciones más eficaces y eficientes en la programación de proyectos con recursos limitados. Aunque los métodos heurísticos no garantizan soluciones óptimas, pueden proporcionar soluciones factibles en un tiempo

razonable y con menor esfuerzo que los métodos exactos, lo que los hace especialmente útiles para proyectos con limitaciones de tiempo y recursos.

En consecuencia, esta investigación identifica la necesidad de actualizar de los métodos heurísticos basados en reglas de prioridad existentes. El propósito principal es brindar a los investigadores un recurso consolidado que abarque la mayoría de las reglas de prioridad existentes en la literatura. Esta necesidad se convierte en el eje central de la investigación que es proporcionar una recopilación exhaustiva y actualizada de las reglas de prioridad utilizadas en los métodos heurísticos para la programación de proyectos con recursos limitados, con el objetivo de facilitar y promover el avance en este campo de estudio.

En consecuencia, la investigación identifica la necesidad crítica de actualizar los métodos heurísticos basados en reglas de prioridad. El propósito principal es brindar a los investigadores un recurso consolidado y actualizado de la mayoría de las reglas de prioridad existentes en la literatura. Esto busca facilitar y promover el avance en la gestión de proyectos con recursos limitados.

3. Metodología

La gestión de proyectos con recursos limitados depende en gran medida de los métodos heurísticos basados en reglas de prioridad, los cuales son cruciales para lograr una planificación eficiente y efectiva. La elección del método adecuado está determinada por una serie de características específicas del proyecto, como los objetivos estratégicos de la organización, la complejidad del proyecto, los recursos disponibles, los riesgos involucrados y la urgencia de la entrega, así como las duraciones y holguras asociadas a las actividades. Con el transcurso del tiempo, se han desarrollado varios métodos heurísticos con diferentes enfoques para establecer el orden de ejecución de las actividades en un proyecto.

En este trabajo, se presenta una recopilación exhaustiva de 97 métodos heurísticos basados en reglas de prioridad, obtenida a través de una revisión meticulosa de 25 artículos científicos relevantes. Aunque esta selección no abarca todos los métodos existentes, proporciona una base sólida para comprender las opciones disponibles y sirve como punto de partida para futuras investigaciones en este campo. La información recopilada se presenta de manera clara y estructurada en una tabla (ver Tabla 2), que incluye la sigla y nombre del método, la base de la regla en la que se aplica la prioridad y las referencias bibliográficas asociadas a cada método. Cada número correspondiente en la columna de referencia de la tabla está vinculado a las referencias completas proporcionadas al final de la tabla.

Tabla 2: Clasificación de métodos heurísticos basados en reglas de prioridad

Nº	SIGLA Y NOMBRE DE LA REGLA	BASE DE LA REGLA	REFERENCIA
1	ACROS - Activity Resource Controls	Recursos	[11]
2	ACS - Average Case Slack Time Rule	Holgura	[8] [11] [20] [25] [20]
3	ACTRES - Activity resources	Recursos	[11]
4	CUMRED - Cumulative Resource Equivalent Duration	Duración	[4] [25] [3]
5	DRC - Dynamic Remaining Capacity	Recursos	[25]
6	DRD - Dynamic Relative Demand	Recursos	[22] [25]
7	DRS - Dynamic Resource Scarcity	Recursos	[25]
8	EDD - Earliest Due Date	Duración	[12] [14] [18]
9	EDF - Earliest Deadline First	Duración	[16]
10	EFT - Early Start Time	Duración	[3] [17] [21] [25] [21]
11	EFTD - Earliest Finishing Time Dynamically	Duración	[11] [23]
12	ERD - Instante De Inicio Más Temprano	Duración	[14] [16]
13	EST - Earliest Start Time	Duración	[6] [11] [17] [19] [22] [24-25]

14	ESTD - Earliest Starting Time Dynamically	Duración	[11]
15	FCFS - First-Come, First-Served	Orden de llegada	[3] [16] [18]
16	GCRWC - Greatest Cumulative Resource Work Content	Trabajo de recursos	[22] [25]
17	GPOP - Greatest Product of Priorities	Mayor prioridad	[3] [22] [25]
18	GPRPW - Greatest Proportional Rank Positional Weight		[3] [22]
19	GRD - Greatest Resource Demand	Recursos	[1-3] [6] [11] [17] [19] [20] [22-25]
20	GRNOP - Greatest Ready Number of Operations in Precedence	Precedencia	[18]
21	GRPW - Greatest Rank Positional Weight	Peso basado en la posición y el rango de la actividad	[3] [6] [8] [13] [17] [19] [20] [22] [24] [25]
22	GRU - Greatest Resource Utilization (HRU1)	Recursos	[1-3] [22-24]
23	GRWC - Greatest Resource Work Content	Recursos	[21]
24	IRSM - Improved Resource Scheduling Method	Recursos	[7] [8] [11] [20] [25] [11]
25	LCFS - Last-Come, First-Served	Orden de llegada	[18]
26	LCR - Least Critical Resource	Recursos	[5]
27	LFS - Least Float per Successor	Holgura	[7] [22] [24] [25] [24]
28	LFT - Late Finish Time	Duración	[1-14] [17] [19] [20] [22-25]
29	LIS - Least Immediate Successors	Sucesoras	[3] [6] [22]
30	LJF - Longest Job First	Trabajo	[16]
31	LNRJ - Least Non-Related Jobs	Trabajo no realizado	[3] [21] [22] [25]
32	LPF - Longest Path Following	Ruta	[3] [7] [15] [22] [25]
33	LPT - Longest Processing Time	Duración	[3] [5] [6] [14] [17-19] [22] [25]
34	LRNOP - Largest Ready Number of Operations in Precedence	Precedencia	[18]
35	LRP - Least Resource Proportion	Recursos	[5]
36	LRPT - Longest Remaining Processing Time	Tiempo de procesamiento	[18]
37	LRPW - Least Rank Positional Weight		[3] [6] [22]
38	LST – Late Start Time (Last Start Time)	Duración	[3] [4] [6] [7] [9-11] [13] [17] [19] [20] [22-25]
39	LTF - Least Total Float	Holgura	[2]
40	LTS - Least Total Successors.	Sucesoras	[3] [5]
41	LWC - Li and Willis Rule, Set C		[25]
42	MAX RWK - Maximum Remaining Work	Trabajo	[5]
43	MAXCAN - Maximum Number Of Subsequent Candidates	Actividades	[5]
44	MAXSP - Maximum Schedule Pressure	Duración	[25]
45	MAXTWK - Maximum Total Work Content	Actividades	[25]
46	MFF - Minimum Free Float		[24]
47	MINSLK - Minimum Job Slack	Duración	[1]
48	MINTWK - Minimum Total Work Content	Recursos	[25]
49	MIS - Most Immediate Successors (MS)	Sucesoras	[3] [5-7] [11] [17] [19-25]
50	MJP - Most Job Possible	Sucesoras	[1] [3]
51	MS/L - Most Successors Per Level	Sucesoras	[3] [22] [25]
52	MSF - Minimum Safety Float	Duración	[24]
53	MSLD - Minimum slack time dynamically	Duración	[11]
54	MSLK - Minimum Slack Time	Holgura	[3] [5-14] [17] [19] [20] [22] [24] [25]
55	MSTD - Most Total Successors dynamically	Sucesoras	[11]
56	MTS - Most Total Successors	Sucesoras	[3] [4] [7-11] [13-15] [19] [20] [22] [23-25]
57	MTSPT - Most Total Successors Processing Time	Tiempo de procesamiento	[22]
58	MTTD - Minimum Time to Due Date	Duración	[16]

59	MWR - Most Work Content Remaining	Actividades	[15]
60	OGRPW - Old Greatest Rank Positional Weight	Peso de posición	[20]
61	RAND - Random	Aleatorio de actividades	[1-3] [7] [11] [12] [22]
62	RED - Resource Equivalent Duration	Recursos	[3] [25]
63	RES - Reschedule Heuristic	Reprogramación	[2]
64	ROT - Resource Over Time	Recursos	[22]
65	RSM - Resource Scheduling Method	Recursos	[1-4] [7] [8] [10] [19] [20] [22] [25]
66	SCUMRD - Smallest Cumulative Resource Demand	Recursos	[22]
67	SFM - Shortest Feasible Mode		[5]
68	SIRO - Service in Random Order	Orden aleatorio de servicio	[14] [18]
69	SJF - Shortest Job First	Duración	[1] [16]
70	SPT - Shortest Processing Time	Duración	[3] [5] [6] [10-13] [17-22] [24] [25]
71	SQNO - Shortest Queue Next Operation	Duración	[14]
72	SRD - Smallest Resource Demand	Recursos	[22] [24]
73	SRPT - Shortest Remaining Processing Time	Tiempo de procesamiento	[14] [18]
74	SST - Shortest Setup Time	Duración	[14]
75	TIMRES		[11] [23]
76	TIMROS		[11] [23]
77	TRD - Total Resource Demand	Recursos	[22] [25]
78	TRS - Total Resource Scarcity	Recursos	[22] [24] [25]
79	WACRU - WACRU (AMEND WRUP)	Recursos	[24] [25]
80	WCLS - Worst Case Latest Starting Time	Duración	[11]
81	WCS - Worst Case Slack	Holgura	[7] [8] [10] [11] [20] [25]
82	WEDD - Weighted Earliest Due Date	Duración	[12]
83	WLFT - Weighted Minimum Latest Finish Time	Duración	[12]
84	WRUP - Weighted Resource Utilization Ratio and Precedence	Relación de utilización de recursos y tiempo.	[9] [11] [17] [19] [22] [24] [25]
85	WSLK - Weighted Minimum Slack	Holgura	[12]
86	WSPT - Weighted Shortest Processing Time	Duración	[12] [14]
87	LNRA - Longest Normalized Remaining Activity	Duración	[17]
88	RD - Resource Demand	Recursos	[17]
89	T-LEVEL - Total Level	Duración	[17]
90	B-LEVEL - Total Level	Actividades	[17]
91	SAD - Sum of All Durations	Duración	[23]
92	HRPW - Highest Resource Productivity Weight	Recursos	[23]
93	EFD - Earliest Finish Date	Duración	[23]
94	STFS - Smallest Total Float Slack	Actividades	[23]
95	STFD - Smallest Total Float Duration	Actividades	[23]
96	HWRU - Highest Weighted Resource Utilization	Actividades	[23]
97	HRU2 - Highest Resource Utilization 2	Recursos	[23]

Nota de referencias: [1- Davis y Patterson, 1975] [2- Richter y Arai 1986] [3- Alvares y Tamarit, 1989a] [4- Alvares y Tamarit, 1989b] [5- Boctor, 1993] [6- Hapke y Slowinski, 1995] [7- Franck y Neumann, 1996] [8- Kolisch, 1996a] [9- Kolisch, 1996b] [10- Kolisch, Hartmann, 1999] [11- Klein, 2000] [12- Kolisch, 2000] [13- Fleszar y Hindi, 2004] [14- Sierra, 2009] [15- Fündeling y Trautmann, 2010] [16- Azmi, Abu, Hanan, Shamsir y Wan, 2011] [17- Gargiulo, Francesco y Quagliarella, 2011] [18- Ibrahim y Aydemir, 2014] [19- Machado, 2014] [20- Chez, Demeulemeester, Bai y Guo, 2018] [21- Adamu, Okagbue y Oguntunde, 2019] [22- Guo, Vanhoucke, Coelho y Luo, 2021] [23- Türkakın, Arditi, Manisali, 2021] [24- Song, Jia y Peng, 2022] [25to - Luo, Vanhoucke, Coelho y Guo, 2022]

En la Tabla 2 se encuentran detallados los 97 métodos identificados. Después de la identificación del grupo de métodos heurísticos basados en reglas de prioridad para la programación de recursos limitados, se procedió a una cuidadosa selección de cinco métodos

para realizar un ejemplo y compararlos. Estos cinco métodos seleccionados basándose en la frecuencia con la que estos métodos fueron mencionados y utilizados en los artículos revisados, lo que sugiere su relevancia y popularidad en el campo.

3.1. Ejemplo de proyecto para aplicación de los métodos heurísticos basado en reglas de prioridad para la programación con recursos limitados

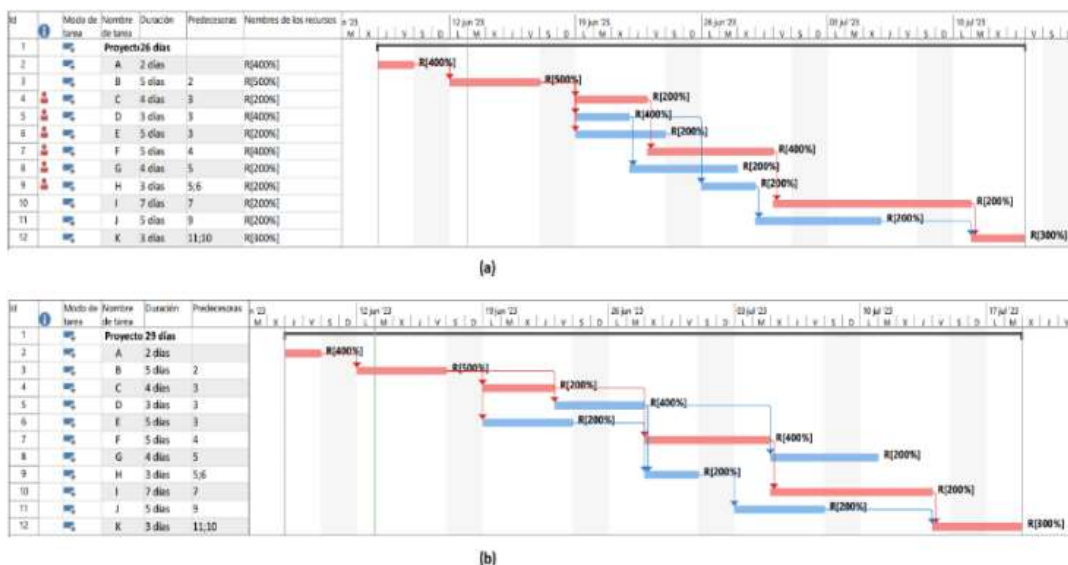
Se ha propuesto un ejemplo de proyecto que consta de 11 actividades para ilustrar la aplicación de los métodos seleccionados. Estas actividades están definidas por su duración, dependencia y la disponibilidad de un único tipo de recurso, como se detalla en la tabla 3. El recurso está sujeto a una limitación de 6 unidades disponibles por unidad de trabajo (UT). El objetivo principal de este caso de estudio es aplicar los métodos previamente seleccionados y comparar la duración total del proyecto con la solución proporcionada por el software MS Project. Se busca abordar específicamente el desafío de programar proyectos con recursos limitados. Para implementar los métodos, es fundamental contar con la información del CPM, ya que proporciona una visión clara de las relaciones entre las actividades y sus duraciones respectivas.

Tabla 3: Ejemplo de proyecto

Actividad	UT	Dependencia	Recursos x UT
A	2		4
B	5	A	5
C	4	B	2
D	3	B	4
E	5	B	2
F	5	C	4
G	4	D	2
H	3	D, E	2
I	7	F	2
J	5	H	2
K	3	I, G, J	3

El ejemplo de proyecto de la Tabla 3 ha sido programado con la ayuda del software MS Project. La representación visual en la Figura 1(a) ofrece una visión clara de la ruta crítica, las duraciones de las actividades, las dependencias y los recursos requeridos. Aunque la duración total del proyecto es inicialmente de 26 días, se observa una sobre asignación de recursos, indicando una limitación en su disponibilidad. En la Figura 1(b), MS Project realiza ajustes automáticos para resolver esta sobreasignación, lo que resulta en un aumento de 3 días en la duración total del proyecto. Este ajuste automático plantea incertidumbres sobre la optimización de la solución.

Figura 1: Programación en MS Project



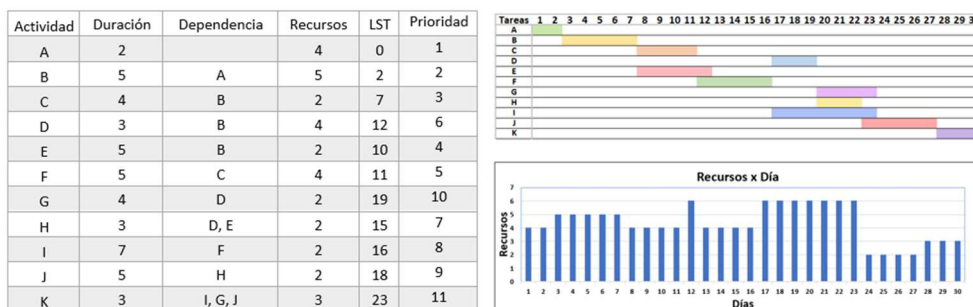
Aquí es donde los métodos heurísticos adquieren relevancia, proporcionando enfoques estratégicos para una planificación más eficiente al considerar la duración de las actividades, sus dependencias y la disponibilidad de recursos, con el fin de optimizar los resultados del proyecto.

Los métodos heurísticos basados en reglas de prioridad seleccionados anteriormente son LST (Late Start Time - Tiempo de Inicio más Tardío), LFT (Late Finish Time - Tiempo de Finalización más Tardío), MST (Most Total Successors - Mayor Cantidad de Sucesoras Totales), MSLK (Minimum Slack Time - Holgura Mínima) y GRPW (Greatest Rank Positional Weight - Mayor Peso Posicional de Rango). A continuación, se aplicarán estos métodos al ejemplo de proyecto presentado en la tabla 3.

3.1.1. Late Start Time – Tiempo de Inicio más Tardío (LST):

Este método prioriza las actividades que pueden comenzar más tarde sin afectar la duración total del proyecto. Se basa en calcular el momento más tardío en el que cada actividad puede iniciar sin retrasar el proyecto en su conjunto. Utilizando la información del Diagrama de Red del Proyecto CPM/PERT. Alvares y Tamarit (1989b) indica que la ecuación para calcular LST de una actividad j es $LST(j) = LFT_j - d_j$, donde LFT_j es el Tiempo de finalización más tardío de la actividad j y d_j es la duración de la actividad j . Interpretar los resultados implica observar si el LST de una actividad es igual a su Tiempo de Inicio más Tardío (EST), lo que significa que la actividad debe comenzar lo antes posible para evitar retrasos. Si el LST es mayor que el EST, indica que la actividad tiene un margen de tiempo disponible antes de volverse crítica para el proyecto, lo que permite gestionar mejor los recursos y evitar posibles demoras.

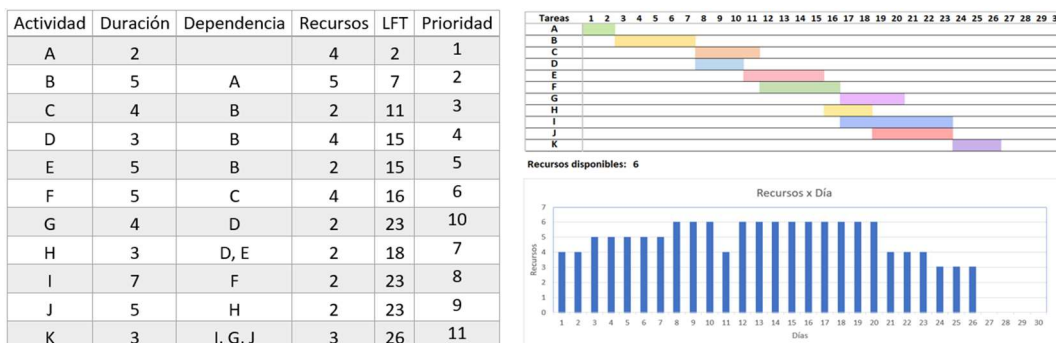
Figura 2: Programación LST



3.1.2. Late Finish Time – Tiempo finalización más Tardío (LFT):

Este método prioriza las actividades según su tiempo más tardío de finalización. Utiliza la información del CPM/PERT para calcular el momento más tardío en el que cada actividad puede finalizar sin afectar la duración total del proyecto. Alvares y Tamarit (1989b) indica que la ecuación para calcular LFT de una actividad j es $LFT(j) = LST(j) + d_j$, donde $LST(j)$ es el Tiempo de Inicio más Tardío de la actividad j y d_j es su duración. Interpretar los resultados del método LFT implica verificar si el LFT de una actividad es igual a su LST significa que la actividad no tiene margen para retrasos y debe completarse tan pronto como sea posible para evitar retrasos. Por otro lado, si el LFT es mayor que el LST, indica que la actividad tiene cierto margen de tiempo disponible antes de convertirse en crítica para el proyecto.

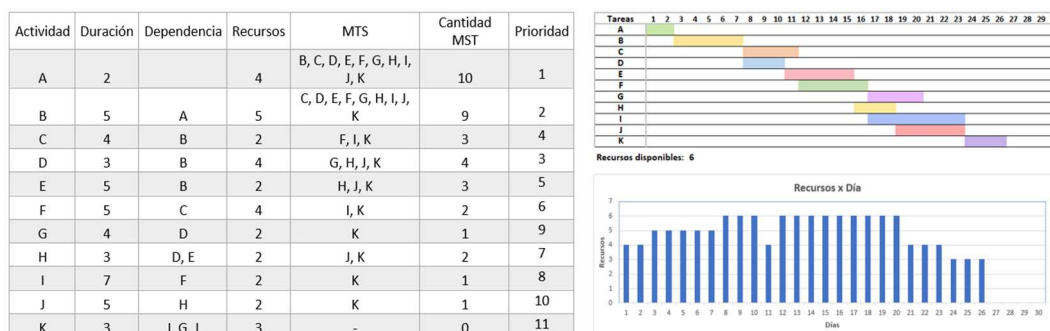
Figura 3: Programación LFT



3.1.3. Most Total Successors - Mayor Cantidad de Sucesoras Totales (MST):

Este método prioriza las actividades según el número total de actividades sucesoras que tiene cada una. Es decir, se enfoca en las actividades que tienen la mayor cantidad de actividades dependientes. La lógica detrás de este enfoque es que las actividades con más sucesoras pueden tener un impacto significativo en el proyecto si se retrasan. Por lo tanto, se les da prioridad para garantizar que se completen a tiempo y eviten posibles retrasos en las actividades posteriores. La aplicación de este método implica identificar las actividades con el mayor número de sucesoras y programarlas en consecuencia para minimizar el riesgo de retrasos en el proyecto.

Figura 4: Programación MST

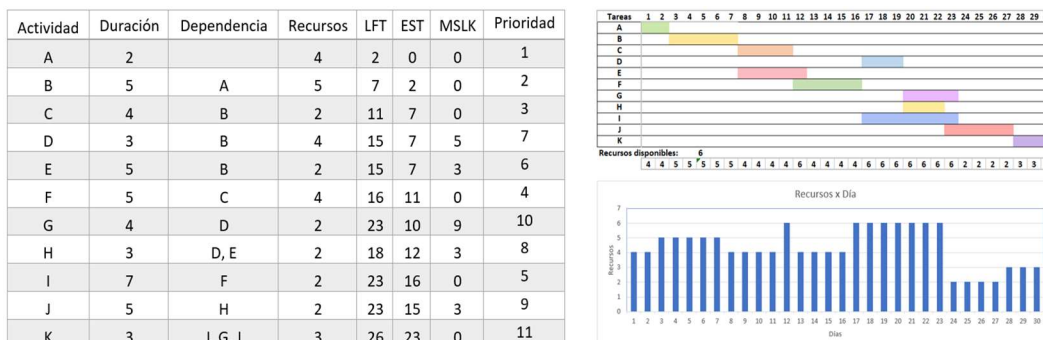


3.1.4. Minimum Slack Time - Holgura Mínima (MSLK)

El método prioriza las actividades del proyecto según el margen de tiempo disponible antes de su fecha límite de finalización. Se centra en identificar las actividades con el menor margen de tiempo disponible para su ejecución, priorizando las actividades que tienen menos flexibilidad en términos de retrasos sin afectar la duración total del proyecto. La fórmula para calcular la MSLK de una actividad j es $MSLK(j) = LFT_j - EST_j - d_j$, donde LFT_j es el Tiempo de

Finalización más Tardío de la actividad j, ESTj es el Tiempo de Inicio más Temprano de la actividad j, y dj es la duración de la actividad j. Este método ayuda a identificar las actividades más críticas que deben completarse sin retrasos para garantizar el cumplimiento de los plazos del proyecto.

Figura 5: Programación MSLK



3.1.5. Greatest Rank Positional Weight - Mayor Peso Posicional de Rango (GRPW)

El método se centra en la asignación de prioridades a las actividades de un proyecto para optimizar la utilización de los recursos disponibles. El método se basa en la asignación de pesos a las actividades, considerando factores como la duración de cada tarea y la suma de las duraciones de sus actividades sucesoras. Alvarez y Tamarit (1989b) menciona la fórmula $GRPW = dj + \sum_{i \in S(j)} di$ donde j es la actividad por programar, dj la duración de la actividad j, di es la duración de la actividad sucesora i, i es la actividad sucesora y S(j) es el conjunto de todas las actividades sucesoras a j. Esto permite identificar las actividades más importantes y asignarles recursos de manera prioritaria para garantizar su finalización oportuna.

Figura 6: Programación GRPW



4. Resultados

En esta investigación, se llevó a cabo un exhaustivo análisis de 25 artículos relevantes en el campo de la programación de proyectos con recursos limitados, centrándose en los métodos heurísticos basados en reglas de prioridad. Estos artículos abordan diversas reglas de prioridad que han sido recopiladas y presentadas en la Tabla 2, donde se identificaron un total de 97 reglas, cada una con sus propias características.

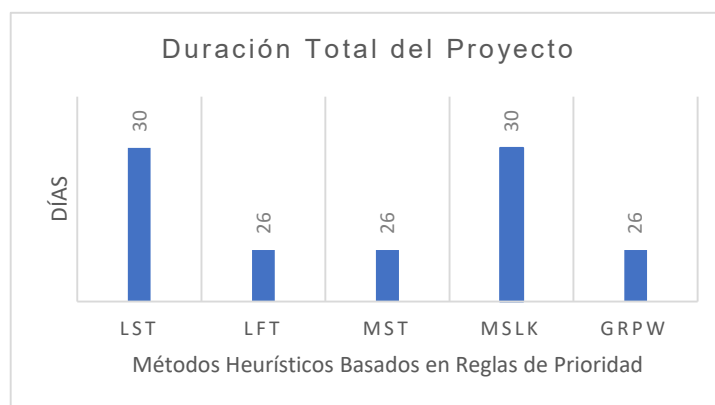
Entre los métodos más citados en la literatura revisada se destacan: LFT con 21, MSKL con 17, MST con 16, LST con 15 y GRPW con 13 referencias. Estos cinco métodos fueron seleccionados para ser aplicados en un ejemplo de proyecto, con el objetivo de evaluar su eficacia en la gestión de proyectos con recursos limitados. En el siguiente párrafo se explicará los resultados encontrados de la comparación realizada en términos de la duración total del

proyecto. Este enfoque permitirá no solo comprender la aplicabilidad y efectividad de estos métodos, sino también identificar cuál de ellos puede ofrecer la mejor solución en situaciones específicas de programación de proyectos con recursos limitados.

Para evaluar la eficacia de los métodos seleccionados, se empleó un ejemplo de proyecto (ver Tabla 3) en el software MS Project. Inicialmente, el proyecto tenía una duración de 26 días y mostraba sobreasignación de recursos (ver Figura 1a). Sin embargo, al resolver automáticamente la sobreasignación, la duración del proyecto aumentó en 3 días, totalizando en 29 días.

Por otro lado, la aplicación de los cinco métodos previamente mencionados reveló una eficiencia en la duración del proyecto. En la figura 7 se puede observar que los métodos LFT, MST y GRPW lograron mantener la duración en 26 días, mientras que los métodos LST y MSLK presentaron una duración de 30 días; ninguno de estos cinco métodos ha presentado sobreasignaciones en sus duraciones. Estos resultados indican que tres de los cinco métodos analizados resultaron ser eficaces y eficientes al mantener la duración del proyecto sin la sobreasignación de recursos que se observó en la solución inicial proporcionada por MS Project.

Figura 7: Programación de los métodos seleccionados



5. Conclusiones

La gestión efectiva de recursos limitados es clave para el éxito de cualquier proyecto. Esto implica la importancia de distribuir los recursos de manera eficiente y ajustar la programación de actividades para abordar las limitaciones. Los métodos heurísticos, especialmente aquellos basados en reglas de prioridad, son altamente recomendados debido a su versatilidad y facilidad de implementación.

La recopilación de 97 métodos heurísticos basados en reglas de prioridad subraya la extensa variedad de métodos disponibles, lo que sugiere la posibilidad de combinar métodos para maximizar la eficiencia y optimizar los resultados del proyecto.

Este trabajo se evaluó cinco métodos heurísticos basados en reglas de prioridad específicos, se encontró que tres de ellos (LFT, MST y GRPW) mantuvieron la duración del proyecto en 26 días, mostrando eficacia en la gestión de recursos. Sin embargo, los métodos LST y MSLK superaron este límite. La elección del método óptimo debe basarse en las características específicas de cada proyecto, pero los métodos heurísticos ofrecen herramientas efectivas para tomar decisiones informadas y estratégicas para cumplir con los objetivos del proyecto.

6. Referencias

- Abdolshah, M. (2014). A Review of Resource-Constrained Project Scheduling Problems (RCPSP) Approaches and Solutions. *International Transaction Journal of Engineering, Management, y Applied Sciences & Technologies*, 5, 253-286. <http://tuengr.com/V05/0253.pdf>.
- Luo, J., Vanhoucke, M., Coelho, J., Guo, W. (2022). Anefficient genetic programming approach to design priority rules for resource-constrained project scheduling problem. *Expert Systems With Applications*, 198. DOI: [10.1016/j.eswa.2022.116753](https://doi.org/10.1016/j.eswa.2022.116753).
- Adamu, P.I., Okagbue, H.I. y Oguntunde, P.E. (2019). A New Priority Rule for Solving Project Scheduling Problems. *Wireless Pers Commun* 106, 681–699. DOI: [10.1007/s11277-019-06185-5](https://doi.org/10.1007/s11277-019-06185-5).
- Álvares-Campana, P. (2021). *Simulador para la resolución de problemas de programación multiproyecto con restricción de recursos*. (Tesis de Maestría, Universidad de Valladolid, Valladolid). <https://uvadoc.uva.es/handle/10324/49313>.
- Alvarez-Valdés, R., y Tamarit, J.M. (1989b). Algoritmos heurísticos deterministas y aleatorios en secuenciación de proyectos con recursos limitados. *Qüestiió*, Vol 13, Nº 173-191. https://dmlle.icmat.es/pdf/QUESTIIO_1989_13_01-02-03_11.pdf.
- Alvarez-Valdés, R., y Tamarit, J.M. (1989a). Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis. *Advances in project scheduling*. 113–134, Amsterdam.
- Azmi, Z., Abu, K., Hanan, A., Shamsir, M., y Wan, W. (2011). Performance Comparison of Priority Rule Scheduling Algorithms Using Different Inter Arrival Time Jobs in Grid Environment. *International Journal of Grid and Distributed Computing*, 4(3), 61-70. Obtenido de http://article.nadiapub.com/IJGDC/vol4_no3/5.pdf.
- Ballestín, F. (2002). *Nuevos métodos de resolución del problema de secuenciación de proyectos con recursos limitados*. (Tesis Doctoral, Universitat de València). <https://roderic.uv.es/handle/10550/14898>.
- Boctor, F. F. (1993). Heuristics for scheduling projects with resource restrictions and several resource duration modes, *International Journal of Production Research*, 31:11, 2547-2558. <http://dx.doi.org/10.1080/00207549308956882>.
- Brucker, P., Drexl, A., Möhring, R., Neumann, K., y Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1), 3-41.
- Cervantes. M. (2010). *Nuevos métodos meta heurísticos para la asignación eficiente, optima y robusta de los recursos limitados*. (Tesis doctoral, Universidad Politècnica de València, 2010). Repositorio Institucional Ruinet. <https://riunet.upv.es/bitstream/handle/10251/7582/tesisUPV3229.pdf?sequence=1>.
- Chand, S., Singh, H., y Ray, T. (2019). Evolving heuristics for the resource constrained project scheduling problem with dynamic resource disruptions. *Swarm and Evolutionary Computation*, 44, 897-912. DOI: [10.1016/j.swevo.2018.09.007](https://doi.org/10.1016/j.swevo.2018.09.007).
- Chen, Z., Demeulemeester, E., Sijun, B., y Yuntao, G. (2018). Efficient priority rules for the stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, 270(3), 957-967. DOI: [10.1016/j.ejor.2018.04.025](https://doi.org/10.1016/j.ejor.2018.04.025).
- Cho, J-H., Kim, Y-D. (1997) A simulated annealing algorithm for resource constrained project scheduling. *Journal of the Operational Research Society*, 48, 736-744.

- Davis, E. y Patterson, J. (1975). A comparison of heuristic and optimun solutions in resource-constrained project scheduling. *Management Science*. Vol. 21, No. 8, USA.
- Fleszar, K., Hindi, K. (2004). Solvin the resource-constrained project scheduling problem by a variable neighbourhood search. *European Journal of Operational Research*, 155, 402-413.
Recuperado de http://yalma.fime.uanl.mx/~roger/work/teaching/class_tso/docs_project/problems/RCPSP/1-s2.0-S0377221702008846-main.pdf.
- Fondhal, J.W. (1961). A Non-Computer Approach to the Critical Path Method for the Construction Industry. Ed. Stanford University, Stanford (California, USA).
- Franck, B., y Neumann, K. (1996.) Priority-rule methods for the resource-constrained project scheduling problem with minimal and maximal time lags: An empirical analysis. *Institut für Wirtschaftstheorie und Operations Research*, Universiät Karlsruhe, Karlsruhe, Germany. <https://publikationen.bibliothek.kit.edu/114796/1090>.
- Fündeling, C. U., y Trautmann, N. (2010). A priority-rule method for project scheduling with work-content constraints. *European Journal of Operational Research*, 203(3), 568-574. DOI: [10.1016/j.ejor.2009.09.019](https://doi.org/10.1016/j.ejor.2009.09.019).
- Gálvez, D., Cisternas, A., Ordieres, J. y Capuz, S. (2016). On project scheduling using DSM. Proceedings from the 20th International Congress on Project Management and Engineering (423-435). Cartagena. http://dspace.aepro.com/xmlui/bitstream/handle/123456789/780/CIDIP2016_01063.pdf?sequence=1&isAllowed=y.
- García-Villalba, M., y González-Velarde, J. (2005). A simulated annealing algorithm for resource-constrained project scheduling. *International Journal of Production Economics*, 93, 217-232.
- Gargiulo, F. y Quagliarella, D. (2011). A Hybrid Genetic Algorithm For The Resource Constrained Project Scheduling Problem (RCPSP). *Evolutionary and Deterministic Methods for Design, Optimization and Control*, 244 - 252. Capua, Italy. 10.13140/2.1.1935.2322.
- Ghasemi, B., Sadeghi, A., y Roghani, M. (2015). The Solution of Multi-Objective Multimode Resource-Constrained Project Scheduling Problem (RCPSP) with Partial Precedence Relations by MultiObjective Bees Algorithm. *Silvae Genetica*, 57(1).
- Golab, A., Gooya, E. S., Al Falou, A., y Cabon, M. (2022). A multilayer feed-forward neural network (MLFNN) for the resource-constrained project scheduling problem (RCPSP). *Decision Science Letters* 11 (2022) 407– 418. DOI: [10.5267/dsl.2022.7.004](https://doi.org/10.5267/dsl.2022.7.004).
- Guo, W., Vanhoucke, M., Coelho, J., Luo, J. (2021). Automatic detection of the best performing priority rule for the resource-constrained project scheduling problem. *Expert Systems With Applications*, 167. DOI: [10.1016/j.eswa.2020.114116](https://doi.org/10.1016/j.eswa.2020.114116).
- Hapke, M., y Slowinski, R. (1996). Fuzzy priority heuristics for project scheduling. *Fuzzy Sets and Systems*, 83(3), 291-299. DOI: [10.1016/0165-0114\(95\)00338-X](https://doi.org/10.1016/0165-0114(95)00338-X).
- Ibrahim, H., Aydemir, E. (2014). A Priority Rule Based Production Scheduling Module on Faborg-Sim Simulation Tool. *Gazi University Journal of Science*, 27(4),1143-1155. https://www.researchgate.net/publication/268686264_A_Priority_Rule_Based_Production_Scheduling_Module_on_Faborg-Sim_Simulation_Tool.
- Klein, R. (2000). Bidirectional planning: improving priority rule-based heuristics for scheduling resource-constrained projects. *European Journal of Operational Research*, 127(3), 619-638. DOI: [10.1016/S0377-2217\(99\)00347-1](https://doi.org/10.1016/S0377-2217(99)00347-1).

- Kolisch, R. (1996a). Efficient priority rules for the resource-constrained project scheduling problem. *Journal of Operations Management*, 14, 179-192.
- Kolisch, R. (1996b). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 90(2), 320-333. DOI: [10.1016/0377-2217\(95\)00357-6](https://doi.org/10.1016/0377-2217(95)00357-6).
- Kolisch, R. (2000). Integrated scheduling, assembly area- and part-assignment for large-scale, make-to-order assemblies. *International Journal of Production Economics*, 64(1-3), 127-141. DOI: [10.1016/S0925-5273\(99\)00052-3](https://doi.org/10.1016/S0925-5273(99)00052-3).
- Kolisch, R., y Hartmann, S. (2006). Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 174, (23-37). DOI: [10.1016/j.ejor.2005.01.065](https://doi.org/10.1016/j.ejor.2005.01.065).
- Kolisch, R., y Sprecher, A. (1997). PSPLIB - A project scheduling problem library: OR Software - ORSEP Operations Research Software Exchange Program. *European Journal of Operational Research*, 96(1), 205-216. DOI: [10.1016/S0377-2217\(96\)00170-1](https://doi.org/10.1016/S0377-2217(96)00170-1).
- Kolisch, R., Hartmann, S. (1999). Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. *Journal of the Operational Research Society*, 50(2), 162-175. DOI: [10.1007/978-1-4615-5533-9_7](https://doi.org/10.1007/978-1-4615-5533-9_7).
- Lerch, D., Trautmann, N. (2019). A Lazy-Constraints Approach to Resource-Constrained Project Scheduling. *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, (144-148). Macao, China. DOI: [10.1109/IEEM44572.2019.8978524](https://doi.org/10.1109/IEEM44572.2019.8978524).
- Luo, J., Vanhoucke, M., y Coelho, J. (2023). Automated design of priority rules for resource-constrained project scheduling problem using surrogate-assisted genetic programming. *Swarm and Evolutionary Computation*, 81, 101339. DOI: [10.1016/j.swevo.2023.101339](https://doi.org/10.1016/j.swevo.2023.101339).
- Machado, L. (2014). *Algoritmo Memético Para Resolver el Problema de Secuenciación con Recursos Limitados Modo Múltiple, MRCPSP*. (Tesis de Maestría, Universidad del Norte, 2014). <https://manglar.uninorte.edu.co/bitstream/handle/10584/9294/120355.pdf?sequence=1&isAllowed=y>.
- Mejía, G., Niño, K., Montoya, C., Sánchez, M. A., Palacios, J., y Amodeo, L. (2016). A Petri Net-based framework for realistic project management and scheduling: An application in animation and videogames. *Computers & Operations Research*, 66, 190-198. DOI: [10.1016/j.cor.2015.08.011](https://doi.org/10.1016/j.cor.2015.08.011).
- Moradi, M., Hafezalkotob, A., Vahidreza Ghezavati, V. (2019). Robust resource-constrained project scheduling problem of the project's subcontractors in a cooperative environment under uncertainty: Social complex construction case study. *Computers & Industrial Engineering*, 133, 19-28. DOI: [10.1016/j.cie.2019.04.046](https://doi.org/10.1016/j.cie.2019.04.046).
- Morillo, D., Moreno, L., y Díaz, J. (2014a). Metodologías analíticas y heurísticas para la solución del problema de programación de tareas con recursos restringidos (RCPPSP): una revisión Parte 1. *Ingeniería y ciencia*, Vol. 10, N°. 19, 247-271.
- Morillo, D., Moreno, L., y Díaz, J. (2014b). Metodologías analíticas y heurísticas para la solución del problema de programación de tareas con recursos restringidos (RCPPSP): una revisión Parte 2. *Ingeniería y ciencia*, Vol. 10, N°. 20, 203-227.
- Ortigueira, M. (1976). Programación de proyectos con recursos limitados. <https://dialnet.unirioja.es/descarga/articulo/2494712>.

- Ortiz, N., Díaz, F., (2020). An optimization model to solve the resource constrained project scheduling problem RCPSP in new product development projects. *DYNA*, 87 (179-188). Colombia: <http://doi.org/10.15446/dyna.v87n212.81269>.
- Piñeiro, S. (1995). PERT y CPM: Programación y control de proyectos. *Cuadernos de Estudios Empresariales*, (5), 271-291. Servicio de Publicaciones UCM. Madrid. <https://dialnet.unirioja.es/servlet/articulo?codigo=164224>.
- PMI (2017). *Guía de los Fundamentos para la dirección de proyectos (PMBOK)*, 6ª edición. Project Management Institute (PMI), Newton Square (Pensilvania, USA)
- Richter, L., Arai, C. (1986). A comparison of heuristics for preemptive resource-constrained project scheduling. *Department of Industrial and Operation Engineering*. University of Michigan. Technical Report 86-35. Recuperado de <https://deepblue.lib.umich.edu/bitstream/handle/2027.42/7264/bam7806.0001.001.pdf?sequence=5>.
- Sierra Sánchez, M. R. (2009). *Mejora de algoritmos de búsqueda heurística mediante poda por dominancia. Aplicación a problemas de scheduling*. (Tesis doctoral). Departamento de Informática. Universidad de Oviedo. Oviedo (España).
- Simón, M., Peña, F., Rincón, P. (2014). Análisis de los métodos heurísticos de resolución del problema de programación de proyectos con recursos limitados (RCPSP: Resource Constrained Project Scheduling Problem). *Revista de Investigación en Educación*, 9, 34-45. https://revistas.uax.es/index.php/tec_des/article/view/591.
- Song, H., Jia, G., y Peng, W. (2022). Bi-objective reactive project scheduling problem under resource uncertainty and its heuristic solution based on priority rules. *IEEE Access*, 10, 3175312. DOI: [10.1109/ACCESS.2022.3175312](https://doi.org/10.1109/ACCESS.2022.3175312).
- Türkakin, O.H.; Arditi, D.; Manisalı, E (2021). Comparison of Heuristic Priority Rules in the Solution of the Resource-Constrained Project Scheduling Problem. *Sustainability*, 13, 9956. DOI: [10.3390/su13179956](https://doi.org/10.3390/su13179956).
- Valls, V., Ballestín, F., Quintanilla, S. (2008). A hybrid genetic algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 185, 495-508. DOI: [10.1016/j.ejor.2006.12.033](https://doi.org/10.1016/j.ejor.2006.12.033).
- Wilson, J. (2003). Gantt charts: A centenary appreciation. *European Journal of Operational Research* 149 (2003) 430 - 437. DOI: [10.1016/S0377-2217\(02\)00769-5](https://doi.org/10.1016/S0377-2217(02)00769-5).
- Yang, X. S. (2010). A new metaheuristic bat-inspired algorithm. *In Nature Inspired Cooperative Strategies for Optimization* (65-74). Springer, Berlin, Heidelberg.
- Zaman, F., Elsayed, S., Saker, R., Essam, D. (2020). Resource Constrained Project Scheduling With Dynamic Disruption Recovery. *IEEE Access*, vol. 8 (144866-144879). DOI: [10.1109/ACCESS.2020.3014940](https://doi.org/10.1109/ACCESS.2020.3014940).

**Comunicación alineada con los
Objetivos de Desarrollo Sostenible**

