

01-036 – Review of advances in genetic algorithms for solving the resource-constrained project scheduling problem (RCPSP) – Revisión de los avances en algoritmos genéticos para la resolución del problema de programación de proyectos con recursos limitados (RCPSP)

Ponluisa, Neisser F.¹; Capuz-Rizo, Salvador F¹

(1) Universitat Politècnica de València

 Spanish  Spanish

Resource-constrained project scheduling (RCPSP) is a critical problem in modern project management. The challenge is to allocate resources optimally to meet constraints and ensure project success. A comprehensive review of genetic algorithms as a solution to RCPSP is presented. This paper presents a comprehensive review of Genetic Algorithms (GA) as a solution to the RCPSP, highlighting their effectiveness in generate robust solutions in both real and simulated environments. GA, inspired by the principles of natural evolution, efficiently explore large search spaces through iterative processes of selection, crossover, and mutation. Various chromosome representations, genetic operators, and hybrid strategies that combine GA with other metaheuristics are analyzed. The comparative evaluation is based on studies validated with the standard PSPLIB library and real-case scenarios, demonstrating significant improvements in makespan reduction and resource utilization. This review identifies key research directions, methodological trends, and differences between approaches, offering a solid foundation for future GA-based applications to complex project scheduling problems.

Keywords: *Project scheduling; Limited resources; RCPSP; Metaheuristic techniques; Genetic algorithms*

La programación de proyectos con recursos limitados (RCPSP) representa uno de los desafíos más complejos en la gestión moderna de proyectos. El desafío es asignar los recursos de manera óptima para cumplir con las restricciones y garantizar el éxito del proyecto. Se presenta una revisión exhaustiva de los algoritmos genéticos (GA) como solución para el RCPSP, destacando su eficacia para generar soluciones robustas en escenarios reales y simulados. Los GA, inspirados en los principios de la evolución natural, permiten explorar eficientemente grandes espacios de búsqueda mediante procesos iterativos de selección, cruce y mutación. Se analizan distintas representaciones cromosómicas, operadores genéticos y estrategias híbridas que combinan GA con otras metaheurísticas. La evaluación comparativa se basa en estudios validados con la librería estándar PSPLIB y casos reales, mostrando mejoras significativas en la reducción del *makespan* y en la eficiencia del uso de recursos. Esta revisión permite identificar las principales líneas de investigación, tendencias metodológicas y diferencias entre enfoques, ofreciendo una base sólida para futuras aplicaciones del GA en la resolución de problemas complejos de programación de proyectos.

Palabras claves: *Programación de proyectos; Recursos limitados; RCPSP; Técnicas metaheurísticas; Algoritmos genéticos*



©2025 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

La dirección de proyectos cada vez es más exigente en la programación de proyectos debido al crecimiento competitivo y dinámico de empresas. Por ello, es fundamental realizar una programación en la que se optimicen los recursos disponibles, con el fin de cumplir con los objetivos del proyecto, los cuales incluyen minimizar el makespan (duración total del proyecto), garantizar el alcance y especificaciones de calidad. Esto representa un desafío constante para los directores de proyectos, generando un problema en la asignación de recursos disponibles.

El problema de programación de proyectos con recursos limitados (RCPSP, por sus siglas en inglés de Resource Costrained Project Scheduling Problem) representa un problema fundamental en la dirección de proyectos, esencialmente en entornos donde la asignación de recursos debe ser eficiente por la limitación de recursos y más cuando se necesita minimizar la duración, siendo aspectos críticos para el éxito del proyecto.

El RCPSP está clasificado como NP-hard (Blazewicz, Lenstra, & Rinnooy Kan, 1983, como se cita en Viveros Gutiérrez, 2017), lo que hace inviable su resolución mediante los métodos clásicos o exactos en proyectos reales de gran tamaño (Hartmann, 1998; Kolisch & Hartmann, 2000). Esto ha motivado el desarrollo de enfoques alternativos, entre los que destacan los métodos heurísticos y metaheurísticos, orientados a ofrecer soluciones viables de buena calidad en tiempos razonables.

Entre los enfoques metaheurísticos, el método genético (GA) ha demostrado ser especialmente útil en la programación de proyectos, gracias a su gran capacidad de explorar amplios espacios de búsqueda compleja utilizando los principios de evolución natural como son la selección, cruce y mutación. Desde las contribuciones iniciales de Hartmann (1988), quien propuso una codificación genética basada en permutaciones de sentando las bases de la metodología para futuras investigaciones, además, hizo uso de la librería PSPLIB de Kolisch y Sprecher (1997) para demostrar la eficiencia del algoritmo propuesto.

Los GA han sido objeto de múltiples mejoras estructurales y funcionales, incluyendo variantes híbridas, estrategias de optimización local y extensiones al problema multimodal y al entorno multiproyectos. En contexto, a la amplitud de aplicaciones y variaciones desarrolladas durante las últimas décadas, este trabajo tiene como objetivo presentar una revisión estructurada de la evolución del enfoque genético aplicado al RCPSP. Para ello, se analizan más de treinta contribuciones publicadas entre 1998 y 2024, clasificando las propuestas según sus aportes metodológicos, estructura genética, tipo de problema abordado y desempeño observado en benchmarks como PSPLIB. El análisis permite identificar tendencias en la adaptación de los GA, su integración con otros enfoques metaheurísticos y su capacidad para abordar escenarios reales con restricciones múltiples.

Este documento se estructura de la siguiente manera: la sección 2 ofrece una revisión de la literatura sobre los métodos de programación de proyectos con recursos limitados, con especial atención al algoritmo genético como enfoque metaheurístico. La sección 3 describe la metodología empleada para la clasificación cronológica y presenta una tabla de los algoritmos genéticos aplicados al RCPSP. En la sección 4 se analizan los resultados obtenidos a partir de dicha tabla. La sección 5 expone las conclusiones del estudio. Finalmente, la sección 6 recoge la bibliografía utilizada.

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

La programación de proyectos con recursos limitados ha sido ampliamente estudiada debido a su complejidad y su importancia en la eficiencia operativa de las organizaciones, más aún cuando se busca minimizar el makespan. Para abordar este problema, se han explorado diversos métodos, entre los cuales, los métodos exactos, heurísticos clásicos, metaheurísticos, programación lineal entera (PLE), entre otros. Sin embargo, la efectividad de estos métodos depende significativamente del contexto específico, los objetivos y las restricciones particulares que tiene el proyecto (Kolisch & Hartmann, 2000; Hartmann, 1998; Khalilzadeh, Shadrokh & Kianfar, 2009; Li & Zhang, 2014; Lova et al., 2017).

Entre los métodos metaheurísticos, los algoritmos genéticos han ganado relevancia por su capacidad de buscar y proporcionar soluciones eficientes en tiempo razonables, incluso en proyectos de alta complejidad (Chen, Liu & Luo, 2019; Rostami, Creemers & Leus, 2019).

Esta sección presenta, en primer lugar, una visión general de los enfoques más relevantes empleados en la programación de proyectos con recursos limitados, para luego centrarse específicamente en la evolución, las contribuciones metodológicas y los tipos de aplicación más destacados del algoritmo genético en este campo (Singh & Tiwari, 2018; Tavana et al., 2018; Xu et al., 2024; Zhang et al., 2024).

2.1. Revisión de los métodos para afrontar el RCPSP

El RCPSP ha sido ampliamente estudiado por varios investigadores con el objetivo de encontrar soluciones eficientes y óptimas que puedan aplicarse en proyectos reales. A través de los años, se han desarrollado diferentes estrategias para abordar este problema, los cuales pueden clasificarse, en términos generales, los métodos exactos, heurísticos, metaheurísticos, híbridas, multicriterio y, más recientemente, en métodos hiperheurísticos y de inteligencia artificial. Estos enfoques reflejan la búsqueda constante de soluciones eficientes y óptimas para resolver el RCPSP, ya que sigue representando un desafío significativo en la dirección de proyectos.

Los métodos exactos fueron los primeros métodos en emplearse para solucionar el RCPSP, basándose en modelos matemáticos como la programación lineal entera (PLE), la programación con restricciones y técnicas de enumeración exhaustiva. Aunque estos métodos garantizan soluciones óptimas, su uso queda limitado a instancias de pequeña escala, dado que el tiempo de resolución crece exponencialmente con el tamaño de la complejidad del proyecto (Kolisch & Hartmann, 2000). Aun así, siguen siendo utilizados como referencia para evaluar la calidad de soluciones aproximadas (Viveros, 2017).

Ante las limitaciones de los exactos, se incrementó el interés por resolver instancias más grandes, lo que dio paso al desarrollo de los heurísticos clásicos, que, si bien no garantizan optimalidad, son eficaces y permiten obtener soluciones viables con rapidez y bajo coste computacional. Estos métodos se basan en reglas de prioridad de las actividades según ciertos criterios. Entre los más frecuentes destacan: LFT (Latest Finish Time – Tiempo de finalización más tardío), LST (Latest Start Time – Tiempo de Inicio más tardío), MTS (Minimum Slack Time – Tiempo de holgura mínima), MSLK (Minimum Slack – Holgura mínimo), WRUP (Weighted Resource Utilization and Precedence – Utilización de recursos poderada y precedencia), GRPW (Greatest Rank Positional Weight - Mayor peso posicional de rango o ponderación de actividades). Estos criterios son la base de algoritmos de lista propuesto por Hartmann (1998). Aunque su rendimiento puede ser limitado en proyectos complejos, se utilizan frecuentemente como la base inicial o como parte de métodos más sofisticados.

Los métodos metaheurísticos se desarrollaron como una alternativa robusta y flexible. Entre ellos, los algoritmos genéticos (GA) han sido los más utilizados. Estos métodos simulan el

proceso evolutivo de selección natural, generando soluciones a partir de poblaciones y aplicando operadores de cruce, mutación y selección. Entre otros métodos, se encuentra el recocido simulado (Simulated Annealing, SA), empleado de forma híbrida junto a GA para mejorar la exploración global del espacio de búsqueda (Chen & Shahandashti, 2009). También se ha utilizado la búsqueda tabú (Tabu Search) para gestionar perfiles de recursos flexibles (Tritscher, 2015), así como la optimización por colonia de hormigas (Ant Colony Optimization) y técnicas bioinspiradas como el algoritmo de abejas (Bee Algorithm), aplicado en instancias de la librería PSPLIB (Sadeghi et al. 2011). Otras propuestas incluyen algoritmos meméticos, que integran búsqueda local dentro del proceso evolutivo (Rahman, Chakraborty, & Ryan, 2019), y un enfoque hiperheurístico basado en programación genética para generar operadores adaptativos (Lin, Zhu, & Gao, 2020). Estas técnicas han demostrado ser eficaces en entornos complejos, permitiendo una mayor robustez, flexibilidad y adaptabilidad del cronograma frente a perturbaciones.

Seguidamente, se han propuesto algoritmos genéticos híbridos, combinando GA con otras técnicas para mejorar tanto la calidad como la estabilidad de las soluciones. Chen y Shahandashti (2009) desarrollaron un enfoque híbrido que integra GA con recocido simulado, obteniendo mejoras significativas en el makespan y una mayor capacidad de exploración del espacio de soluciones. En cuanto a la robustez y flexibilidad en los cronogramas, Tritschler (2017) propuso un enfoque metaheurístico que incorpora mecanismos a la programación los perfiles de recursos flexibles. Adicionalmente, Viveros (2017) abordó la necesidad de diseñar cronogramas resilientes ante incertidumbres durante la ejecución y absorber retrasos sin comprometer la continuidad del proyecto.

Por otro lado, Rahman, Chakraborty y Ryan (2019) propusieron un algoritmo memético que combina la evolución genética con intensificación local, logrando un equilibrio más eficiente entre exploración y explotación en entornos complejos.

Finalmente, Lin, Zhu y Gao (2020) introdujeron un enfoque hiperheurístico basado en programación genética con habilidades múltiples. Esta propuesta permite adaptar dinámicamente los operadores evolutivos a las características específicas del problema, mejorando la calidad de la solución.

Los diferentes métodos revisados en la literatura para la programación de proyectos con recursos limitados se resumen en la tabla 1.

Tabla 1: Métodos para resolver RCPSP.

Métodos	Tipos	Características
Exactos	PLE PCE PEM Branch and Bound Árbol de precedencias.	Soluciones óptimas garantizadas. Alta complejidad computacional. Útiles para instancias pequeñas.
Heurísticos	Reglas de prioridad: LFT LST MTS MSLK WRUP GRPW.	Soluciones factibles y rápidas. Bajo coste computacional. Útiles como punto de partida para metaheurísticas.
Metaheurísticos	GA SA ACO Bee Algorithm Tabu Search	Alta flexibilidad. Buen balance entre exploración y explotación. Eficaces para instancias medianas y grandes.

	Algoritmos Meméticos.	
Híbridos	GA + SA, GA + Local Search, GA + VNS.	Combinación de técnicas para mejorar estabilidad y calidad de soluciones. Mejora en makespan y robustez.
Hiperheurísticos	Programación genética adaptativa.	Generación de operadores personalizados. Adaptabilidad sin necesidad de reconfiguración manual.

Fuente: Hartmann (1998); Kolisch & Hartmann (2000); Chen & Shahandashti (2009); Sadeghi et al. (2011); Tritschler (2015), Tritschler (2017); Viveros (2017); Rahman et al. (2019); Lin, Zhu & Gao (2020).

Cada método que se incluye en la Tabla 1 presentan sus propias características para lograr que la programación del proyecto minimice el makespan y optimice los recursos disponibles y que sean asignados a cada una de las actividades del proyecto. El uso de cualquier método dependerá de las particularidades y complejidades del proyecto a vez deben ir alineadas con las estrategias del director de proyectos.

2.2. Revisión del método metaheurístico algoritmo genético (GA)

El (GA), está inspirado en la evolución natural, busca encontrar una programación eficiente para ejecutar las actividades de un proyecto con recursos limitados, considerando las restricciones de precedencia y disponibilidad. A través de un proceso iterativo de selección, cruce y mutación de soluciones candidatas, el GA explora diversas combinaciones posibles sin evaluar todas las opciones, lo que le permite minimizar el makespan y asegurar que cada actividad se realice en tiempo y forma. Esta capacidad lo convierte en una herramienta eficaz para resolver proyectos complejos de manera flexible y optimizada

En 1998, Hartmann presentó un nuevo enfoque de algoritmos genéticos (GA) para resolver el RCPSP. Su enfoque se basaba en una representación por listas de actividades (permutaciones) y operadores genéticos adaptados a las restricciones del problema. Realizo un estudio comparativo para contrastar su propuesta, esto lo hizo con dos enfoques genéticos existentes el primero fue basado en valores de prioridad y el otro en reglas de prioridad, demostrando que su algoritmo basado en permutaciones superaba a ambos, logrando mejores resultados en cuanto a calidad de solución y robustez. Este avance demostró, que su diseño de representación y operadores en los Ga pueden funcionar mejor que los tradicionales. En el 2000, Hartmann y Kolisch ampliaron este trabajo con una evaluación experimental detallada de heurísticas avanzadas, confirmando que los GA cuando se usan con configuraciones adecuadas de representación y técnicas de muestreo inteligente, seguían obteniendo un rendimiento altamente competitivo frente a otros metaheurísticos como la búsqueda tabú y el recocido simulado. Este estudio reforzó la posición de los GA como una de las estrategias más prometedoras para abordar el RCPSP.

En 2007, el algoritmo genético experimentó avances significativos en su aplicación teórica y práctica. Como es el trabajo de Alba y Chicano (2007) que extendieron el uso de algoritmos genéticos al ámbito de la dirección de proyectos de software, agregando restricciones reales, habilidades diferenciadas, salarios y dedicación parcial del personal. Su modelo optimizó simultáneamente coste, calidad y duración del proyecto, destacándose el uso de los GA a entornos prácticos y multicriterio, además desarrollaron una herramienta de simulación automática. En el mismo año, Gutiérrez-Franco y Mejía (2007) propusieron un algoritmo genético para el RCPSP clásico, utilizando recursos renovables y un único modo de ejecución. Su enfoque incluyó una codificación basada en permutaciones, un esquema de generación serial para garantizar la viabilidad de las soluciones, y el desarrollo de una herramienta computacional orientada a usuarios, que permitía definir actividades, restricciones de precedencia y disponibilidad de recursos. Estos trabajos contribuyeron al avance de los

algoritmos genéticos en la resolución de problemas complejos para la programación de proyectos con recursos limitados, tanto en escenarios teóricos como prácticos.

Zhang y Wuliang (2008) propusieron un esquema híbrido que combinaba codificación por prioridad y permutaciones, busco preservar las ventajas de ambos métodos y así asegurar soluciones factibles mediante un esquema de generación serial. Su algoritmo demostró ser competitivo en pruebas con instancias de la PSPLIB, destacando por su simplicidad y eficacia. Seguidamente en el 2009, Chen y Shahandashti desarrollaron un algoritmo híbrido GA y Recocido simulado (SA) para programar múltiples proyectos con restricciones complejas. Este enfoque utilizó GA para prioridades globales y SA como refinador local, aplicado a tres proyectos reales y 3 prácticos superando en rendimiento a GA, SA y otras variantes, marcando un avance en la aplicación de GA a entornos multiproyecto con recursos compartidos.

Klimek en el 2010, reforzó el uso de GA en problemas de gran escala aplicando su algoritmo a 960 instancias de la librería PSPLIB con 30 y 90 actividades. Su estudio se centró en evaluar combinaciones de operadores genéticos y configuraciones de parámetros para maximizar la eficiencia. Concluyendo que los GA con una configuración adecuada, podían ser capaces de obtener soluciones robustas y de alta calidad incluso en instancias complejas. En el mismo año, Cervantes (2010) propuso varias mejoras metodológicas tanto para el RCPSP como para su versión multimodal (MRCPSP), incorporando mecanismos de robustez frente a la incertidumbre, como la inserción de buffers temporales y funciones de evaluación que penalizaban soluciones sensibles a perturbaciones. Sus algoritmos fueron evaluados mediante simulación en entornos variables, evidenciando mejoras en estabilidad sin sacrificar el rendimiento.

Entre 2012 y 2013, se desarrollaron nuevas mejoras en los algoritmos genéticos aplicados al RCPSP. En 2012, Iranagh y Sonmez desarrollaron un GA orientado a la nivelación de recursos en los proyectos de construcción y lo al minimizar el makespan, logrando reducir significativamente las fluctuaciones en la utilización de recursos frente al rendimiento de MS Project, sin comprometer la duración del proyecto. Seguidamente, en 2013, Diana et al. propusieron una mejora al GA clásico de Hartmann basada en una codificación binaria para representar las prioridades en las actividades, logrando mayor flexibilidad en las operaciones genéricas, esto demostró ser más efectiva en instancias difíciles del conjunto 120 actividades de la PSPLIB, mejorando los resultados que el GA original, indicando que los cambios estructurales en la representación cromosómica pueden mejorar su capacidad exploratoria y dar mayor calidad en las soluciones.

El frecuente uso de los algoritmos genéticos en los investigadores hace que se vuelvan a plantear nuevas formas de uso tanto la forma clásica como en variantes híbridas y orientada a contextos reales. Siendo así que, en 2014, Kadam y Kadam propusieron un algoritmo basado en esquema de generación serial (SGS) aplicados a las instancias del PSPLIB, consiguiendo resultados prometedores respecto a los métodos anteriores, así mismo, Bettemir & Sonmez propusieron un algoritmo híbrido que combina el GA y SA (Recocido Simulado / Simulated Annealing), logrando mejores resultados en cuanto a convergencia y calidad frente al GA puro e incluso superando a software Ms Project.

En el 2015, Dunham examinó que tan estables y funcionales son las soluciones obtenidas mediante GA en entornos militares, enfocándose en proyectos donde la estabilidad ante incertidumbre es clave, y demostrando que el GA ofrece alternativas sólidas para la nivelación de recursos. Mientras que, Rajeevan & Nagavinothini aplicaron GA a casos reales y de literatura en proyectos de construcción, usando MATLAB para optimizar la duración de los proyectos, destacando su capacidad de integración en entornos prácticos. Por otro lado, Tritschler desarrolló un nuevo modelo híbrido auto-adaptativo que integra el GA con búsqueda de vecindad variable (VNS) y programación lineal, orientado al RCPSP con perfiles de recursos flexibles, logrando mejoras significativas en calidad de solución y tiempos de

cómputo, dando un paso importante en la evolución práctica de los GA aplicados a problemas reales y complejos.

Con el paso de los años, nuevas variaciones de GA siguen apareciendo, como es el caso de Melo et al. (2017), proponen un algoritmo genético combinado con el esquema de generación serial (SSGS), comparando con el rendimiento del GA frente al algoritmo Cuckoo Search en instancias de PSPLIB, y sus resultados mostraron que el GA seguía siendo competitivo, con soluciones de buena calidad y menor variabilidad. Por su parte, Elshaer et al. propusieron un GA optimizado que introducía ajustes en la representación de cromosomas, estrategias de cruce y mutación, aplicándolo a 100 instancias del conjunto 30 actividades de PSPLIB. Lograron un rendimiento más estable y una convergencia más rápida en comparación con enfoques previos.

Kadri y Boctor en 2018, abordaron una revisión del RCPSP que incluía tiempo de transferencia entre actividades, siendo este aspecto poco estudiado hasta el momento, por lo que, propusieron un GA eficiente que integraba este aspecto en la generación de cronogramas, logrando mejoras significativas frente a los métodos convencionales aplicados al PSPLIB. En el mismo año, Munlin presentó una nueva opción de metaheurística inspirada en PSO (Particle Swarm Optimization / Optimización por Enjambre de Partículas), influenciada por el GA, incorporando la mutación adaptativa y una técnica forward-backward, aunque no es un GA puro, su estructura y operadores genéticos, contribuyendo a un enfoque híbrido en la resolución de RCPSP.

En 2019, Afshar et al. desarrollaron un GA con una nueva búsqueda local, mejorando la resolución del MRCPSP (Multimode Resource constrained Project Scheduling Problem / Problema de Programación de Proyectos con Restricción de Recursos Multimodo), esta combinación mejoró la utilización de los recursos y la calidad del cronograma en un conjunto de instancias de 20 actividades del PSPLIB. Por su parte, Rahman et al. presentaron un MA (Algoritmo Meméticos) basado en GA que combina operadores de cruce mejorados y busca local con reinicios. Este método fue evaluado en 1560 instancias de la librería PSPLIB, demostrando un rendimiento competitivo frente a los métodos existentes, sobre todo se destaca por su simplicidad de implementación y su efectividad en los diferentes tamaños de actividades de proyectos.

Los últimos avances en GA aplicados al RCPSP se centran en enfoques híbridos, evoluciones estructurales y validaciones comparativa. En 2020, Lin et al. desarrollaron un nuevo método para afrontar el RCPSP denominado Hiperheurística basada en la programación genética (GP-HH) enfocada a los recursos con multihabilidad (MS-RCPSP) utilizada para la selección de actividades, operando sobre una estructura de árboles tipo LISP. Al aplicarlo al conjunto de datos iMOPSE, el método demostró una mejor solución frente a heurísticas tradicionales. Los resultados evidenciando su adaptabilidad y eficacia en entornos con mayor complejidad estructural y lógica, posicionando las hiperheurísticas como una herramienta prometedora para la solución de RCPSP complejos. En 2022, Golab et al. realizaron una revisión profunda de los métodos metaheurísticos aplicados al RCPSP, donde concluye que el GA es el método más empleado, tanto en su forma original como en sus combinaciones generando los métodos híbridos, además menciona muchas combinaciones con otros métodos y por eso se destaca que el GA es muy adaptable con otros métodos.

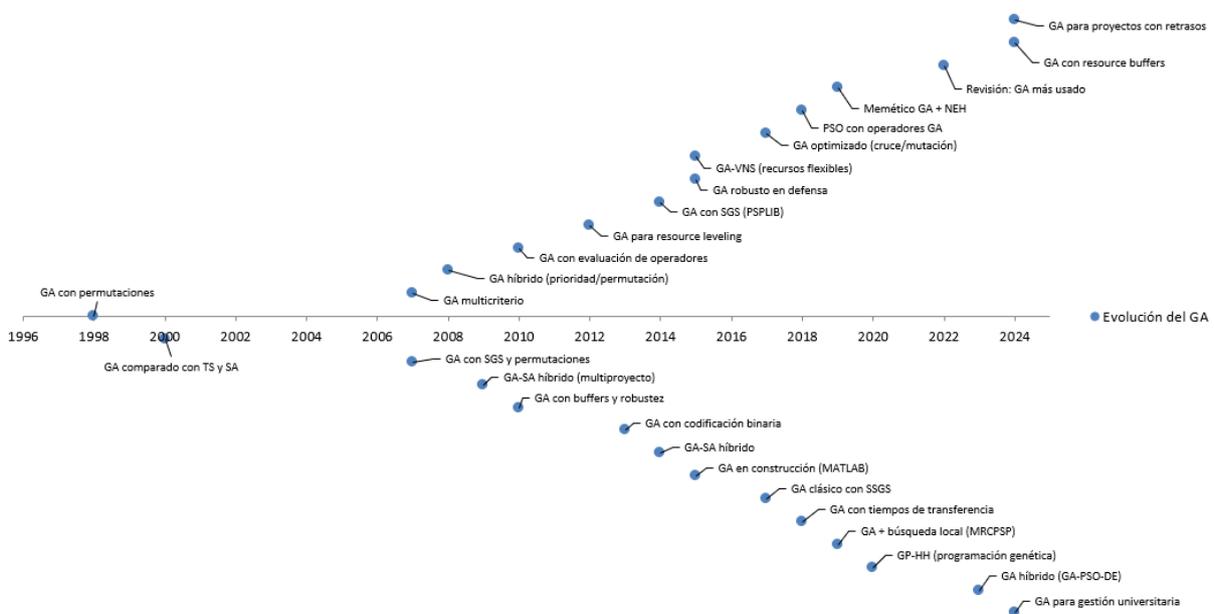
En 2023, Aristotelous y Nearchou, presentaron un nuevo enfoque híbrido innovador con GA, PSO y DE (Differential Evolution / Evolución Diferencial), para resolver el problema de nivelación de recursos. Con este método se hicieron pruebas sobre proyectos reales de construcción, demostrando que la integración de GA en un enfoque híbrido permitió distribuir de manera más equilibrada la carga de trabajo, evitando la sobrecarga de los recursos y manteniendo una carga más uniforme a lo largo del proyecto. Además, logro preservar la duración del proyecto mientras mejoraba significativamente la eficiencia del uso de recursos,

estableciéndose como una buena opción de solución y efectiva para problemas complejos en la gestión de recursos en proyectos reales.

En 2024, Bredel y Vanhoucke propusieron un GA con buffers de recursos para la programación multiproyecto (RCMPSP), incorporando márgenes de seguridad en la programación del uso de recursos sin modificar la estructura de GA, dando como respuesta mejores resultado en el 20% de las instancias evaluadas respecto a otros métodos. Por otro lado, Zhou aplicó GA para optimizar los recursos en proyectos científicos universitarios, considerando las restricciones de tiempo, presupuesto y personal, este modelo fue desarrollado en MATLAB, alcanzado una tasa de convergencia estable de 89,5% y superó a los métodos tradicionales en eficiencia y equilibrio en la carga de trabajo. n 2024, Pérez et al. desarrollaron un algoritmo genético (GA) implementado en Python para abordar la redistribución de recursos en proyectos de construcción con retrasos acumulados. Este enfoque se centró en reprogramar la asignación de personal disponible, tomando en cuenta las horas asignadas a las actividades. El algoritmo utilizó operadores genéticos como selección por torneo, cruce de un punto y mutación uniforme. Los resultados demostraron una mejora significativa en la eficiencia de los recursos y una reducción notable del desfase temporal en los proyectos. Este trabajo confirma que los GA son una herramienta altamente adaptable y efectiva para escenarios dinámicos y complejos, como los que se presentan en la dirección de proyectos reales con restricciones y retrasos.

Finalmente, los algoritmos genéticos han demostrado ser una herramienta eficaz y versátil, capaz de adaptarse a diferentes tipos de proyectos y sus respectivas complejidades. Además, han mostrado su capacidad para optimizar el uso de recursos, reducir desfases temporales y ofrecer soluciones robustas en escenarios dinámicos y complejos. Gracias a su flexibilidad y rendimiento, se han consolidado como una opción confiable para abordar el problema de programación de proyectos con recursos limitados, destacando su relevancia en la dirección de proyectos y su capacidad para enfrentar desafíos cada vez más exigentes (ver figura 1).

Figura 1: Evolución del algoritmo genético en RCPSP (1998 – 2024).



Fuente: Elaboración propia presentados en la literatura.

El gráfico representa la evolución cronológica del uso del algoritmo genético (GA) y sus principales variantes aplicadas al problema de programación de proyectos con recursos limitados (RCPSP) entre 1998 y 2024. Cada punto indica un aporte relevante publicado en ese año, y la etiqueta describe la característica distintiva de dicho enfoque. Los puntos están alternados arriba y abajo para facilitar la lectura, y el eje horizontal muestra la línea de tiempo, permitiendo visualizar de forma clara la progresiva de hibridación y aplicación práctica de los GA en este campo.

3. Metodología

Este estudio presenta una revisión exhaustiva de la literatura enfocada en los métodos basados en algoritmos genéticos, identificando un total de 26 artículos publicados entre 1998 y 2024. Estos trabajos han abordado diferentes formas de resolver el RCPSP mediante variantes del GA, aportando enfoques diversos según el contexto y los objetivos de cada investigación. La recopilación y análisis de esta literatura se ha llevado a cabo mediante una lectura crítica y estructurada, clasificando cada contribución según el tipo de codificación cromosómica utilizada, los operadores genéticos implementados, las técnicas de hibridación aplicadas y los tipos de proyectos abordados en su evaluación. Este proceso ha permitido identificar la evolución en el desarrollo del algoritmo genético, ofreciendo una visión clara del estado actual del arte y una base sólida para futuras investigaciones en el área

Tabla 2: Métodos GA para RCPSP.

Cronología	Tipo de RCPSP	Métodos GA y Combinaciones	Fuente de proyectos
1998	Clásico	GA con codificación por permutación, cruce adaptado, mutación tipo swap, SGS serial.	Librería PSPLIB
2000	Clásico	SGS serial/paralelo, reglas de prioridad, búsqueda local, variantes de GA.	Librería PSPLIB
2007	Múltiples habilidades (multi-skill)	GA con cromosomas binarios ($E \times T$), selección por torneo binario, cruce 2D single point crossover, mutación bit-flip, codificación discreta y fitness multiobjetivo.	Proyectos de software generados artificialmente con múltiples recursos humanos.
2008	Clásico	Crossover híbrido, SGS, mutación adaptativa.	Librería PSPLIB
2009	Multiproyectos	GA con codificación por prioridad, recocido simulado (SA) como refinamiento, selección por torneo.	Múltiples proyectos con restricciones múltiples

2010	Clásico	GA con codificación por Lista de actividades, operadores clásicos, selección, hibridación, decodificación (SGS).	Librería PSPLIB
2011	Clásico	GA + Bee Algorithm, operadores bioinspirados, SGS.	Librería PSPLIB
2012	Nivelación de recursos	GA con codificación por fechas de inicio, mutación, cruce, selección por torneo y función de aptitud multiobjetivo.	Proyectos reales de construcción
2013	Clásico	GA con codificación binaria de prioridades, cruce de un punto, mutación binaria, SGS serial, selección elitista.	Librería PSPLIB
2014	Restricciones energéticas	GA con adaptación a restricciones energéticas, evaluación comparativa.	Librería PSPLIB-Energy
2014	Clásico	GA híbrido con Simulated Annealing (GASA). Codificación por lista de prioridades, cruce exclusivo con control de precedencia, mutación por intercambio y adaptativa, elitismo y selección por ruleta.	Librería PSPLIB
2014	Clásico	GA con codificación mediante listas de actividades, selección por ruleta, cruce de un punto, mutación por intercambio condicionado a restricciones de precedencia, esquema elitista.	Librería PSPLIB
2015	Problema de nivelación de recursos (RLP)	GA con codificación real, cruce aleatorio, mutación por probabilidad, múltiples funciones objetivo.	Proyectos reales de construcción militar.
2015	Clásico	GA con implementación en MATLAB.	Proyectos reales de construcción
2017	Clásico	GA con codificación permutacional, SGS serial, cruce por intercambio de posiciones, mutación con control de duplicados, benchmark contra Cuckoo.	Librería PSPLIB

2017	Clásico	GA clásico con evaluación factorial de operadores, selección RNKS, cruce 2PX, mutación INSM, codificación por lista de actividades (AL), SGS y paralelo.	Librería PSPLIB
2017	Perfiles de recurso flexibles	GA híbrido con VNS (variable neighborhood search), uso de FSGS(Flexible Schedule Generation Scheme), codificación lista de actividades, recursos, retraso de inicio.	Librería PSPLIB
2018	Problema de nivelación de recursos (RLP)	Comparado con HGA y GA, algoritmo SIO inspirado en SONAR.	Librería PSPLIB
2018	Tiempos de transferencia (TT)	GA con codificación modificada, cruce clásico, mutación controlada.	Librería PSPLIB
2019	Clásico	GA con búsqueda local, cruce adaptativo, selección elitista.	Librería PSPLIB
2019	Clásico	GA híbrido (NEH modificado + reglas de prioridad), operador de cruce SBOX, mutación por desplazamiento, búsqueda local VINS, reinicio adaptativo, elitismo.	Librería PSPLIB
2020	Múltiples habilidades (multi-skill)	GA + operadores adaptativos, heurísticas múltiples embebidas.	Librería propia (iMOPSE)
2023	Problema de nivelación de recursos (RLP)	GA cruzado de tipo SBX (simulated binary), mutación polinomial. Se combina con DE (Evolución Diferencial) y PSO (Optimización por enjambre de partículas) en esquemas híbridos (serie y paralelo).	Librería PSPLIB
2024	Multiproyectos	GA con operadores evolutivos, buffers en los recursos, combinación de reglas de prioridad, enfoques secuenciase y vectores RK aleatorios.	MPLIB1 (Van Eynde & Vanhoucke, 2020),
2024	Multiobjetivo	GA con codificación en punto flotante, cruce lineal no uniforme y mutación adaptativos.	Proyectos reales universitarios (datos históricos y encuestas).

2024	Dinámico y reactivo (Redistribución)	GA con codificación en números reales, selección por torneo, cruce de un punto, mutación uniforme.	Actividades simuladas con distribución normal aleatoria.
------	--------------------------------------	--	--

Fuente: (Hartmann, 1998; Kolisch y Hartmann, 2000; Alba y Chicano, 2007; Zhang et al., 2008; Chen y Shahandashti, 2009; Klimek, 2010; Sadeghi et al., 2011; Iranagh y Sonmez, 2012; Diana et al., 2013; Torres et al., 2014; Bettemir y Sonmez, 2014; Kadam y Kadam, 2014; Dunham, 2015; Rajeevan y Nagavinothini, 2015; Melo et al., 2017; Elshaer et al., 2017; Tritschler et al., 2017; Tzanetos et al., 2018; Kadri y Boctor, 2018; Afshar et al., 2019; Rahman et al., 2019; Lin et al., 2020; Aristotelous y Nearchou, 2023; Bredel y Vanhoucke, 2024; Zhou, 2024; Pérez Esquivel et al., 2024).

En la Tabla 2 se presenta un resumen, en orden cronológico de los artículos dedicados a el problema RCPSP mediante el GA y sus diferentes combinaciones. Además, se detallan el tipo de RCPSP y la fuente de datos empleada para su validación, principalmente de librerías. Es importante indicar que la columna cronología es meramente informativa, sirviendo como referencia del año de publicación de cada investigación. Cabe destacar también que esta recopilación no abarca la totalidad de investigaciones existentes sobre el tema, sino una selección representativa.

Esta recopilación constituye a una base de análisis comparativos de los de los diferentes enfoques metodológicos cuyos resultados se desarrolla en la sección siguiente.

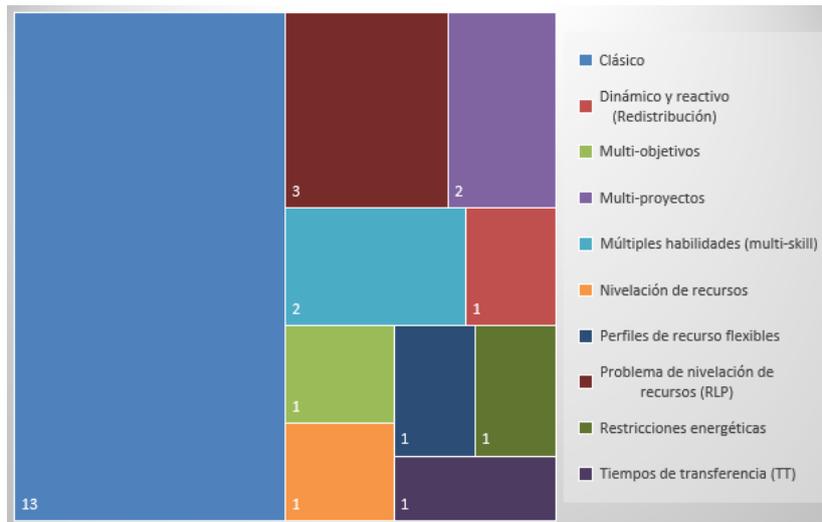
4. Resultados

La tabla 2 anteriormente indicada, permite observa la evolución de la utilización de los métodos GA y sus diferentes combinaciones que ha tenido el método, dándonos una mejor perspectiva a la hora de buscar soluciones para el RCPSP. En esta sección, se analizan los datos recopilados.

4.1. Tipos de RCPSP

En la Figura 2 se observa que el RCPSP clásico es la variante más abordada por los investigadores, con un total de 13 artículos, lo que confirma su papel como punto de partida en el estudio de este problema. En segundo lugar, el problema de nivelación de recursos (RLP) ha ganado relevancia, con 3 estudios dedicados específicamente a esta variante. A continuación, los problemas multiproyectos y multi-skill cuentan con 2 investigaciones cada uno, reflejando un creciente interés en enfoques más integrados y orientados al entorno organizacional. Las variaciones restantes, como el RCPSP con restricciones energéticas, perfiles de recursos flexibles, escenarios dinámicos y reactivos (redistribución), multiobjetivo y tiempos de transferencia, aparecen de forma más puntual, pero apuntan hacia un tratamiento del problema más realista y adaptado a las condiciones del entorno. Esta evolución no solo refleja el interés por optimizar el RCPSP clásico, sino también la necesidad de adaptarlo a contextos reales, ofreciendo soluciones más aplicables y valiosas para los directores de proyectos.

Figura 2: Tipo de RCPSP.

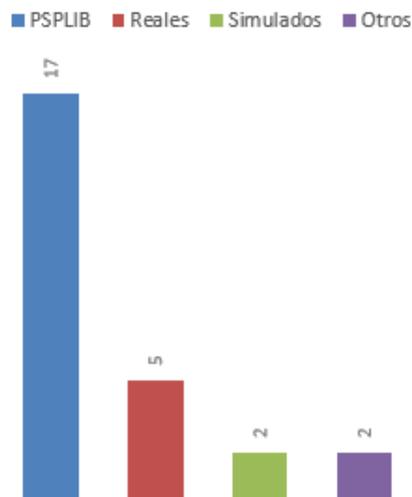


Fuente: Elaboración propia.

4.2. Fuente de proyectos

En la Figura 3 se observa se puede observa que 65% de los artículos ha usado la librería PSPLIB. Esta amplia utilización se debe a su estandarización, disponibilidad pública y compatibilidad con numerosos algoritmos. Sin embargo, también se observa una tendencia creciente hacia el uso de datos reales con el 19% de los artículos. Además, tanto los que se han realizado con simuladores de proyectos y otro tipo de fuentes (iMOPSE y MPLIB1) tienen el 8%.

Figura 3: Fuente de proyectos.



Fuente: Elaboración propia

Esta apertura hacia fuentes más realistas y específicas refleja la intención de los investigadores por acercar sus soluciones al entorno profesional, aumentando la aplicabilidad práctica de los modelos propuestos en proyectos reales.

En conjunto, los resultados presentados permiten identificar patrones claros en la evolución del RCPSP y su abordaje mediante algoritmos genéticos. Se confirma el predominio del

RCPSP clásico como base teórica, pero también una apertura progresiva hacia variantes más complejas y adaptadas al entorno real. De igual forma, se evidencia una preferencia por métodos híbridos y una dependencia considerable como fuente de proyectos la librería PSPLIB, aunque con una tendencia creciente hacia fuentes de datos reales o simuladas. Este panorama muestra una comunidad investigadora que, si bien parte de modelos tradicionales avanza hacia soluciones más flexibles, robustas y aplicables en contextos prácticos.

5. Conclusiones

Los algoritmos genéticos (GA) continúan siendo una alternativa sólida para la resolución del problema de programación de proyectos con recursos limitados (RCPSP), destacándose especialmente su formulación clásica, la cual predomina en la literatura y actúa como base metodológica para abordar nuevas variantes, tales como los problemas multiproyectos, multiobjetivo, la nivelación de recursos, entre otros, que han ido emergiendo de forma dinámica. Asimismo, la librería PSPLIB se mantiene como el principal referente para la validación experimental de métodos GA, especialmente en sus versiones híbridas desarrolladas a lo largo de los años.

Por otra parte, se observa una clara inclinación hacia la hibridación de los algoritmos genéticos con otras técnicas metaheurísticas, con el objetivo de ampliar la exploración del espacio de búsqueda y mejorar la calidad de las soluciones. En conjunto, las tendencias identificadas sugieren que, en el futuro, las soluciones basadas en GA no solo serán más eficientes, sino también más relevantes y aplicables, gracias a enfoques más robustos, adaptativos y alineados con las exigencias actuales, lo que permitirá una toma de decisiones más precisa e informada en la dirección de proyectos.

Finalmente, un avance prometedor en esta línea de investigación es la integración de los algoritmos genéticos con modelos de inteligencia artificial, lo cual representa una innovación clave para dotar a los sistemas de mayor transparencia, confiabilidad y capacidad de adaptación en entornos reales, especialmente aquellos caracterizados por alta variabilidad e incertidumbre.

6. Referencias

- Alba, E., & Chicano, F. (2007). *Management of software projects with GAs*. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO).
- Afshar, M. H., Zandieh, M., & Tavakkoli-Moghaddam, R. (2019). *A genetic algorithm with a new local search method for solving the multimode resource-constrained project scheduling problem*. *Journal of Applied Research on Industrial Engineering*, 6(2), 84–98.
- Aristotelous, C., & Nearchou, A. C. (2023). *Resource leveling optimization by hybrid metaheuristics*. *Engineering Applications of Artificial Intelligence*, 120, 105797.
- Bettemir, Ö., & Sonmez, R. (2014). *Hybrid genetic algorithm with simulated annealing for resource-constrained project scheduling*. *Journal of Construction Engineering and Management*, 140(7), 04014021.
- Bredel, H., & Vanhoucke, M. (2024). *A genetic algorithm with resource buffers for the resource-constrained multi-project scheduling problem*. *Annals of Operations Research*. <https://doi.org/10.1007/s10479-024-05788-z>.

- Cervantes, E. (2010). *Nuevos métodos metaheurísticos para la asignación eficiente, optimizada y robusta de recursos limitados*. Tesis de maestría, Universidad Nacional Autónoma de México.
- Chen, H., & Shahandashti, S. M. (2009). *Hybrid of genetic algorithm and simulated annealing for multiple project scheduling with multiple resource constraints*. *Automation in Construction*, 19(4), 469–476.
- Chen, W., & Shahandashti, S. M. (2007). *A genetic algorithm for the resource-constrained project scheduling problem*. Conference Paper.
- Diana, S., Ganapathy, L., & Pundir, A. K. (2013). *An improved genetic algorithm for resource constrained project scheduling problem*. *International Journal of Computer Applications*, 66(3), 1–6.
- Dunham, D. F. (2015). *Robustness of genetic algorithm solutions in resource-constrained project scheduling*. *Procedia Computer Science*, 51, 15–24.
- Elshaer, R., Shawky, M., Elawady, H., & Nawara, G. (2017). *Solving resource-constrained project scheduling problem using genetic algorithm*. *International Journal of Advanced Computer Science and Applications*, 8(9), 70–75.
- Golab, A., Kowalik, A., & Ziółkowski, M. (2022). *Review of conventional metaheuristic techniques for resource-constrained project scheduling*. *Journal of Project Management*, 7(2), 65–76.
- Hartmann, S. (1998). *A competitive genetic algorithm for resource-constrained project scheduling*. *OR Spectrum*, 20(3), 139–153.
- Iranagh, M. A., & Sonmez, R. (2012). *A genetic algorithm for resource leveling of construction projects*. *Automation in Construction*, 24, 100–108.
- Kadam, S. U., & Kadam, N. S. (2014). *Solving resource-constrained project scheduling problem using genetic algorithm*. *International Journal of Engineering Research and Applications*, 4(10), 12–19.
- Kadri, M., & Boctor, F. F. (2018). *An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case*. *Computers & Industrial Engineering*, 126, 366–380.
- Klimek, M. (2010). *A genetic algorithm for the project scheduling with the constraints of resources*. In Proceedings of the Federated Conference on Computer Science and Information Systems.
- Kolisch, R., & Hartmann, S. (2000). *Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem*. *European Journal of Operational Research*, 127(2), 394–407.
- Lin, Z., Zhu, H., & Gao, L. (2020). *A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem*. *Computers & Industrial Engineering*, 147, 106663.
- Melo, E. B. de, de Melo Junior, G., Calixto, W. P., & Reis, M. R. C. (2017). *An application of genetic algorithm and the serial schedule generation scheme for solving the resource-constrained project scheduling problem*. *Production*, 27, e20162294.
- Pérez Esquivel, J. M., Hernández, D., Cedeño, P., & Loor, J. (2024). *Optimización de la redistribución de recursos en proyectos con retrasos utilizando algoritmos genéticos*. *Revista Iberoamericana de Ingeniería Industrial*, 16(1), 14–22.

- Rahman, A., Chakraborty, R. K., & Ryan, M. J. (2019). *An effective memetic algorithm for resource-constrained project scheduling problem*. *Computers & Operations Research*, 108, 104–117.
- Rajeevan, P., & Nagavinothini, G. (2015). *Time optimization for resource constrained construction projects using genetic algorithm*. *International Journal of Innovative Research in Science, Engineering and Technology*, 4(9), 8522–8529.
- Sadeghi, M., Kalanaki, M. R., Noktehdan, I., Safi Samghabadi, A., & Barzinpour, F. (2011). *Using Bees Algorithm to solve the resource constrained project scheduling problem in PSPLIB*. *Applied Mathematical Sciences*, 5(1), 239–248.
- Torres, F., Barber, F., & Salido, M. A. (2014). *PSPLIB-ENERGY: Una extensión de la librería PSPLIB para la evaluación de problemas con restricciones energéticas*. *Actas del Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*.
- Tritschler, M. (2015). *Metaheuristic approaches for resource-constrained project scheduling with flexible resource profiles*. Tesis doctoral, Technische Universität München.
- Tritschler, M., Naber, A., & Kolisch, R. (2017). *A hybrid metaheuristic for resource-constrained project scheduling with flexible resource profiles*. Technische Universität München.
- Tzanetos, N., Kyriklidis, C., Papamichail, K., Dimoulakis, C., & Dounias, G. (2018). *A nature inspired metaheuristic for optimal leveling of resources in project management*. In *Proceedings of SETN 2018*.
- Tzanetos, N., Kyriklidis, C., Papamichail, K., Dimoulakis, C., & Dounias, G. (2018). *Solving resource-constrained project scheduling problem using metaheuristic algorithm*. Conference Paper.
- Viveros-Gutiérrez, R. (2017). *Robustez y flexibilidad en la programación de proyectos: aplicación de algoritmos genéticos*. Tesis doctoral, Universidad de Santiago de Chile.
- Zhang, H., Xu, Y., & Peng, H. (2008). *A genetic algorithm for solving RCPSP*. In *Proceedings of the International Conference on Computer Science and Software Engineering*.
- Zhou, Y. (2024). *Optimization of university scientific research project management resources based on genetic algorithm*. *Mathematical Problems in Engineering*, 2024, Article ID 5463781.

Comunicación alineada con los Objetivos de Desarrollo Sostenible

