

Desarrollo de un Algoritmo Genético para Resolver el Problema de Programación de Proyectos con Recursos Restringidos (RCPSP) con Duración de Actividades Aleatorias Soportado en un Proceso de Simulación de Monte Carlo

Vanesa Carolina Calderón Barreto y Sergio Aldemar Madrid Guiza

Trabajo de grado para optar el título de Ingeniería Industrial

Director

Néstor Raúl Ortiz Pimiento

PhD. En Ingeniería

Universidad Industrial de Santander

Facultad de Ingenierías Físico - Mecánicas

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2021

### **Dedicatoria**

A Dios por darnos la salud y el entendimiento, a nuestras madres Sandra y Martha por darnos el apoyo, el cariño y su sabiduría para enfrentar el día a día, a nuestros padres Carlos y Aldemar por ser un ejemplo de esfuerzo y dedicación ante las situaciones más difíciles, a nuestros hermanos por darnos su compañía y alegría para disfrutar la vida.

A nuestros amigos Karina, Camila, Mónica, Karen, David y todos aquellos que nos han apoyado y siempre nos comparten la mejor de sus sonrisas.

Al doctor Néstor Ortiz por brindarnos su acompañamiento y guía en el desarrollo de nuestro trabajo de grado, a nuestra alma mater UIS por abrirnos sus puertas y permitir desarrollarnos para la vida personal y profesional.

A la vida y a la madre tierra por entregarnos las mejores experiencias y conocimientos para perseguir nuestros mayores anhelos.

**Tabla de Contenido**

Introducción .....	14
1. Generalidades del proyecto .....	16
1.1. Planteamiento del Problema .....	16
1.2. Objetivos.....	18
1.2.1. Objetivo General.....	18
1.2.2. Objetivos Específicos .....	18
2. Revisión de Literatura .....	19
2.1. Análisis Bibliométrico Scopus .....	19
2.1.1. Documentos por año .....	19
2.1.2. Autores.....	20
2.1.3. Territorio.....	21
2.1.4. Áreas de investigación .....	22
2.1.5. Keywords .....	23
2.1.6. Citación de autores .....	25
2.1.7. Acoplamiento bibliográfico .....	26
2.1.8. Citación por revistas .....	27
2.2. Análisis Bibliométrico Web Of Science.....	29
2.2.1. Documentos por año .....	29
2.2.2. Autores.....	30
2.2.3. Territorio.....	31
2.2.4. Áreas de Investigación.....	32
2.2.5. Keywords web of science .....	33

2.2.6.	Citación de Autores Web of Science .....	35
2.2.7.	Acoplamiento bibliográfico .....	36
2.2.8.	Citación por revistas. ....	37
2.3.	Análisis Preliminar de la Literatura.....	38
3.	Marco Teórico .....	44
3.1.	Proyecto .....	44
3.2.	Representación de un proyecto.....	46
3.2.1.	Diagrama de barras. ....	46
3.2.2.	Diagramas de red. ....	46
3.3.	Problemas de programación de proyectos .....	49
3.4.	Simulación de Monte Carlo.....	50
3.5.	Distribución Beta.....	52
3.6.	Esquemas Generadores De Secuencia.....	53
3.6.1.	SGS en serie.....	53
3.6.2.	SGS en paralelo .....	54
3.7.	Reglas De Prioridad Basadas En Heurísticas .....	55
3.8.	Metaheurísticas de Solución.....	56
3.8.1.	Búsqueda tabú.....	57
3.8.2.	Templado simulado. ....	57
3.8.3.	Algoritmo Genético .....	58
3.8.3.1.	Codificación de las Soluciones.....	61
3.8.3.2.	Función de evaluación .....	62
3.8.3.3.	Selección.....	62
3.8.3.4.	Cruce.....	62

3.8.3.5. Mutación.....	63
4. Metodología .....	64
5. Modelo Matemático .....	66
6. Algoritmo Genético Propuesto.....	69
6.1. Generación de la Población Inicial. ....	72
6.1.1. Individuo.....	72
6.1.1.1. Regla LFT (late finish time) .....	75
6.1.1.2. Distribución Beta $\beta$ .....	76
6.1.2. Población Inicial .....	76
6.2. SGS Paralelo.....	77
6.3. Guardar el Mejor Cromosoma.....	77
6.4. Selección de Padres .....	77
6.5. Cruce.....	78
6.6. Mutación.....	79
7. Validación Del Algoritmo .....	82
7.1. Análisis de varianza para instancias j30.....	83
7.2. Análisis de varianza para instancias j60.....	86
8. Resultados .....	89
8.1. Aplicación de Riesgo.....	90
8.1.1. Valor Esperado .....	92
8.1.2. Medidas de Robustez.....	92
8.1.2.1. Robustez de la calidad .....	93
8.1.2.2. Robustez de la solución .....	93
8.2. Análisis Según Indicadores .....	94

8.3.	Análisis Según Tiempo de Ejecución.....	99
9.	Conclusiones .....	100
10.	Recomendaciones.....	102
	Referencias Bibliográficas .....	104

**Lista de Figuras**

	Pág.
Figura 1. Publicación por año del RCPSP en Scopus .....	20
Figura 2. Autores de mayor relevancia en Scopus .....	21
Figura 3. Países líderes en investigación del RCPSP de Scopus .....	22
Figura 4. Áreas de investigación del RCPSP en Scopus.....	23
Figura 5. Red bibliométrica de palabras clave en Scopus.....	24
Figura 6. Red de citación de autores Scopus.....	25
Figura 7. Revistas más citadas en línea de tiempo .....	28
Figura 8. Publicaciones por año en Web of Science .....	30
Figura 9. Autores de mayor relevancia en web of science.....	31
Figura 10. Países líderes en investigación del RCPSP de Web of Science.....	32
Figura 11. Áreas de investigación de RCPSP en Web of Science .....	33
Figura 12. Red bibliométrica de palabras clave en Web Of Science .....	34
Figura 13. Red de citación de autores Web of Science.....	35
Figura 14. Procedimiento algoritmo genético .....	60
Figura 15. Ejemplo de cruce .....	63
Figura 16. Fases metodológicas de la investigación .....	64
Figura 17. Diagrama del Algoritmo Genético.....	70
Figura 18. Creación del Primer Individuo.....	73
Figura 19. Representación del individuo.....	75
Figura 20. Representación del cruce .....	79
Figura 21. Representación de la Mutación.....	81

Figura 22. Gráfico de Pareto instancias j30 .....	84
Figura 23. Gráfica de efectos principales instancias j30 .....	85
Figura 24. Gráfico de Pareto Instancias j60 .....	87
Figura 25. Gráfica de efectos principales instancias j60 .....	88

**Lista de Tablas**

	Pág.
Tabla 1. Cumplimiento de objetivos del proyecto .....	15
Tabla 2. Palabras clave con Scopus .....	24
Tabla 3. Citaciones por autores .....	26
Tabla 4. Acoplamiento bibliográfico entre autores .....	27
Tabla 5. Citación por revistas en Scopus .....	27
Tabla 6. Palabras clave con Web of Science.....	34
Tabla 7. Citación por autores de Web of Science .....	36
Tabla 8. Acoplamiento bibliográfico entre autores.....	37
Tabla 9. Citación por Revistas en Web of Science .....	37
Tabla 10. Ejemplo de Instancia de Proyecto .....	71
Tabla 11. Experimentos para el análisis factorial 2k para instancias j30.....	82
Tabla 12. Experimentos para el análisis factorial 2k para instancias j60.....	83
Tabla 13. Análisis de Varianza Instancias j30 .....	83
Tabla 14. Análisis de Varianza Instancias j60 .....	86
Tabla 15. Riesgos Instancias j30.....	91
Tabla 16. Riesgos Instancias j60.....	91
Tabla 17. Valor Esperado Instancias j30.....	94
Tabla 18. Valor Esperado Instancias j60.....	95
Tabla 19. Resultados Robustez de la Calidad Instancias j30 .....	95
Tabla 20. Resultados Robustez de la Calidad Instancia j60.....	96
Tabla 21. Resultados Robustez de la Solución Instancias j30 .....	97

Tabla 22. Resultados Robustez de la Solución Instancias j60 .....	98
Tabla 23. Tiempo de procesamiento computacional para instancias j30 y j60.....	99

### **Lista de Apéndices**

(Ver apéndices adjuntos y pueden ser consultados en la base de datos de la Biblioteca UIS)

Apéndice A. Código algoritmo genético

Apéndice B. Diseño de experimentos instancias j30

Apéndice C. Diseño de experimentos instancias j60

Apéndice D. Resultados Algoritmo Genético

Apéndice E. Resultados Método Exacto

Apéndice F. Código SGS paralelo

Apéndice G. Resultados simulación de Montecarlo para las líneas base del AG

Apéndice H. Resultados simulación de Montecarlo para las líneas base del ME

Apéndice I. Artículo de carácter publicable

## Resumen

**Título:** Desarrollo de un Algoritmo Genético para Resolver el Problema de Programación de Proyectos con Recursos Restringidos (RCPSP) con Duración de Actividades Aleatorias Soportado en un Proceso de Simulación de Monte Carlo\*

**Autores:** Vanesa Carolina Calderón Barreto y Sergio Aldemar Madrid Güiza\*\*

**Palabras Clave:** Problema de Programación con Recursos Restringidos (RCPSP) Duración, Incertidumbre, Actividad, Estocástico, Makespan.

### Descripción:

En la presente investigación se abordó el Problema de Programación de Proyectos con Recursos Restringidos (RCPSP) no determinístico, el cual tiene como objetivo minimizar el tiempo de ejecución de un proyecto teniendo en consideración las relaciones de precedencia de actividades y la cantidad de recursos a consumir para llevar a cabo la ejecución de tareas. No obstante, el problema se encuentra sujeto a la interrupción de actividades por parte la materialización de ciertos riesgos de tipo externo o interno que afectan directamente la duración total del proyecto.

En la primera fase de la presente investigación se usó una estrategia proactiva para dar solución al problema a través de un algoritmo genético, el cual, utiliza los datos generados de una simulación de Monte Carlo para realizar un proceso que genera una población inicial como punto de partida a un proceso evolutivo y a través de las generaciones conseguir el individuo más apto el cual indica una lista de actividades ordenadas de acuerdo con sus tiempos de inicio. En una segunda fase, se integra una estrategia reactiva mediante el uso de un sistema generador de secuencias (SGS) en paralelo, el cual permitió realizar una simulación de Montecarlo con 10000 escenarios de solución con el fin de evaluar la capacidad de la línea base generada en la estrategia proactiva y compararla con la línea base obtenida a partir de un método exacto, por medio de la comparación del valor esperado de la solución y los indicadores de robustez de la solución y de la calidad.

---

\* Trabajo de Grado

\*\* Facultad de Ingenierías Físico - Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: Néstor Raúl Ortiz Pimiento, Doctorado en Ingeniería.

### Abstract

**Title:** Development of a Genetic Algorithm to Solve the Resource Constrained Project Scheduling Problem (RCPSP) with Random Activity Durations Supported in a Monte Carlo Simulation Process\*

**Authors:** Vanesa Carolina Calderón Barreto and Sergio Aldemar Madrid Güiza\*\*

**Keywords:** Resource constrained project scheduling problem (RCPSP), Duration, Uncertainty, Activity, Stochastic, Makespan.

### Description:

The present research addressed the non-deterministic Resource Constrained Project Scheduling Problem (RCPSP), which aims to minimize the execution time of a project taking into consideration the precedence relationships of activities and the amount of resources to be consumed to carry out the execution of tasks. However, the problem is subject to the interruption of activities due to the materialization of certain external or internal risks that directly affect the total duration of the project.

In the first phase of this research, a proactive strategy was used to solve the problem through a genetic algorithm, which uses the data generated from a Monte Carlo simulation to perform a process that generates an initial population for an evolutionary process and through the generations to get the fittest individual which indicates a list of activities ordered according to their start times. In a second phase, a reactive strategy is integrated through the use of a sequence generator system (SGS) in parallel, which allowed to perform a Monte Carlo simulation with 10000 solution scenarios in order to evaluate the capacity of the baseline generated in the proactive strategy and compare it with the baseline obtained from an exact method, by comparing the expected value of the solution, the robustness quality and robustness of the solution.

---

\* Bachelor Thesis

\*\* Physical – Mechanical Engineering Faculty. School of Industrial and Business Studies. Director: Néstor Raúl Ortiz Pimiento. PhD in Engineering

## **Introducción**

En el desarrollo de nuevos proyectos es necesario establecer una programación que se pueda ejecutar de manera eficiente, sin embargo, para el gerente de un proyecto es inevitable que se puedan presentar condiciones que afectan la duración de las actividades, esto debido a que en el entorno se presentan eventos inciertos.

En los últimos años se ha abordado el problema de programación de proyectos con recursos restringidos RCPSP con el objetivo de minimizar el tiempo de ejecución a través de métodos de solución metaheurísticos, los diferentes investigadores lo hacen con el fin de aportar metodologías que puedan ser usadas por los gerentes de proyectos con el fin de generar una óptima programación de proyectos que incluya las duraciones de las actividades y la capacidad limitada de recursos que se emplean.

En la presente investigación se desarrollará un algoritmo genético con actividades de duración aleatoria, mediante el cual se obtendrá como solución una línea base lo suficientemente robusta para soportar cualquier tipo de interrupción en la ejecución del proyecto en el menor tiempo posible, además se comparará con 10.000 escenarios simulados a través de la técnica Monte Carlo.

Se espera que la herramienta programada sea aplicada a proyectos analizados bajo riesgo, esto con el fin de facilitar a los gestores de proyectos el proceso de toma de decisiones, sin embargo, es imprescindible que cuenten con técnicas para identificar los riesgos que pueden afectar un proyecto.

La investigación se organiza de la siguiente forma: en el capítulo 2 se encuentran las generalidades y objetivos planteados para el proyecto, luego, en el capítulo 3 se realiza la revisión de literatura con su respectivo análisis bibliométrico, en el que se estudia todo el material científico publicado referente al problema de programación de proyectos con recursos restringidos (RCPSP) y duraciones aleatorias, seguidamente en el capítulo 4 se presenta el marco teórico que brinda las herramientas conceptuales para el entendimiento de la investigación, luego en el capítulo 5 se describe la metodología del proyecto por medio de 5 fases, posteriormente en el capítulo 6 se describe el modelo matemático adoptado para darle solución al RCPSP, en el capítulo 7 se explica el algoritmo genético propuesto que incluyen los procesos de generación de la población inicial, cruce y mutación. En el capítulo 8 se realizó un diseño de experimentos para validar el mejor comportamiento del algoritmo, en el capítulo 9 se realiza una simulación de Montecarlo para hallar los indicadores de robustez de la línea base obtenida por el algoritmo genético y posteriormente compararlos con los indicadores obtenidos de la línea base del método exacto. Por último, en los capítulos finales se encuentran las conclusiones del proyecto y las recomendaciones para futuras investigaciones. En la tabla 1 se puede apreciar el cumplimiento de los objetivos.

**Tabla 1**

*Cumplimiento de objetivos del proyecto*

<b>Objetivo</b>	<b>Cumplimiento</b>
1. Realizar una revisión literaria del problema de programación de proyectos con recursos restringidos no determinístico y duración aleatoria de actividades.	Capítulo 2
2. Realizar el proceso de simulación de Monte Carlo, con el fin de generar datos aleatorios de duración de actividades.	Capítulo 6

*Continuación tabla 1 (Cumplimiento de objetivos del proyecto)*

Objetivo	Cumplimiento
3. Desarrollar un algoritmo genético que permita solucionar el problema de programación de proyectos con recursos restringidos.	Capítulo 6
4. Evaluar y comparar, a partir de otro proceso de simulación de Montecarlo asociado a la fase de programación reactiva, el algoritmo genético con el modelo basado en programación lineal entera, propuesto en el artículo Ortiz, N. R., y Diaz, J. F. (2019) Desviación media relativa como medida de robustez en el problema de programación del proyecto estocástico <i>Revista Facultad de Ingeniería, 28(52)</i> , mediante indicadores de robustez.	Capítulo 8
5. Elaborar un artículo de carácter publicable en base a los resultados obtenidos en la investigación.	Apéndice I

## 1. Generalidades del proyecto

### 1.1. Planteamiento del Problema

La gestión de proyectos es una disciplina que ha impactado múltiples campos de ejecución de las pequeñas y grandes industrias del mundo, las cuales compiten entre sí para alcanzar sus metas. Teniendo en cuenta que el entorno en que se desenvuelven los proyectos es fluctuante y difícil de predecir, las actividades y sus duraciones pueden interrumpirse al materializarse un riesgo (sea o no identificado con anterioridad) ocasionando incumplimientos en los tiempos de entrega. Por tanto, es importante el uso de estrategias y herramientas que permitan que se lleve a cabo la gestión y ejecución de un proyecto de manera correcta y eficaz.

Desde la investigación de operaciones se proponen modelos para los problemas de programación de proyectos, los cuales pueden ser resueltos desde enfoques determinísticos y no determinísticos a partir del uso de diferentes técnicas matemáticas.

Siendo así, inmerso en la gestión de proyectos se encuentra el Problema de programación de proyectos con recursos restringidos (RCPSP) no determinístico, el cual consiste en hallar soluciones altamente robustas y que al mismo tiempo provean el mínimo tiempo posible de ejecución de un proyecto (Makespan).

Dentro de la planificación del proyecto se tienen en cuenta tanto las actividades a ejecutar con sus tiempos de inicio y finalización, como los recursos necesarios para realizar dichas actividades. El problema surge cuando al momento de ejecutar el proyecto en alguna de sus fases, se materializa un riesgo donde se supera la cantidad disponible de recursos planificados, por lo cual se ocasiona un retraso en la duración de las actividades puesto que hay que reajustar la planificación y por consiguiente también puede llegar a extenderse la duración total del proyecto.

Este problema es considerado como NP-HARD debido a la alta complejidad que se presenta al momento de programar las múltiples actividades de un proyecto, ya que dependiendo de su tamaño las posibilidades de solución son más extensas. Por tanto, en esta investigación se pretende contribuir a la solución y análisis del RCPSP no determinístico, mediante un algoritmo genético basado en un proceso inicial de simulación de datos sobre la duración de las actividades (producto del riesgo materializado) y junto a esto, una técnica de simulación de Montecarlo que permita hallar indicadores de robustez de la calidad y de la solución para así, comparar con otros métodos de solución propuestos y determinar cuan aplicable puede ser la metodología aquí expuesta.

## 1.2. Objetivos

### 1.2.1. *Objetivo General*

Desarrollar un algoritmo genético que genere una línea base robusta como solución al problema de programación de proyectos con recursos restringidos (RCPSP) no determinístico y duración aleatoria de actividades, haciendo uso de un proceso de simulación de Monte Carlo, el cual permitirá obtener diferentes escenarios para la duración de las actividades de un proyecto.

### 1.2.2. *Objetivos Específicos*

- Realizar una revisión literaria del problema de programación de proyectos con recursos restringidos no determinístico y duración aleatoria de actividades.
- Realizar el proceso de simulación de Monte Carlo, con el fin de generar datos aleatorios de duración de actividades.
- Desarrollar un algoritmo genético que permita solucionar el problema de programación de proyectos con recursos restringidos.
- Evaluar y comparar, a partir de otro proceso de simulación de Montecarlo asociado a la fase de programación reactiva, el algoritmo genético con el modelo basado en programación lineal entera, propuesto en el artículo de Ortiz & Díaz (2019) “Desviación media relativa como medida de robustez en el problema de programación del proyecto estocástico”, *Revista Facultad de Ingeniería*, 28(52), mediante indicadores de robustez.
- Elaborar un artículo de carácter publicable en base a los resultados obtenidos en la investigación.

## 2. Revisión de Literatura

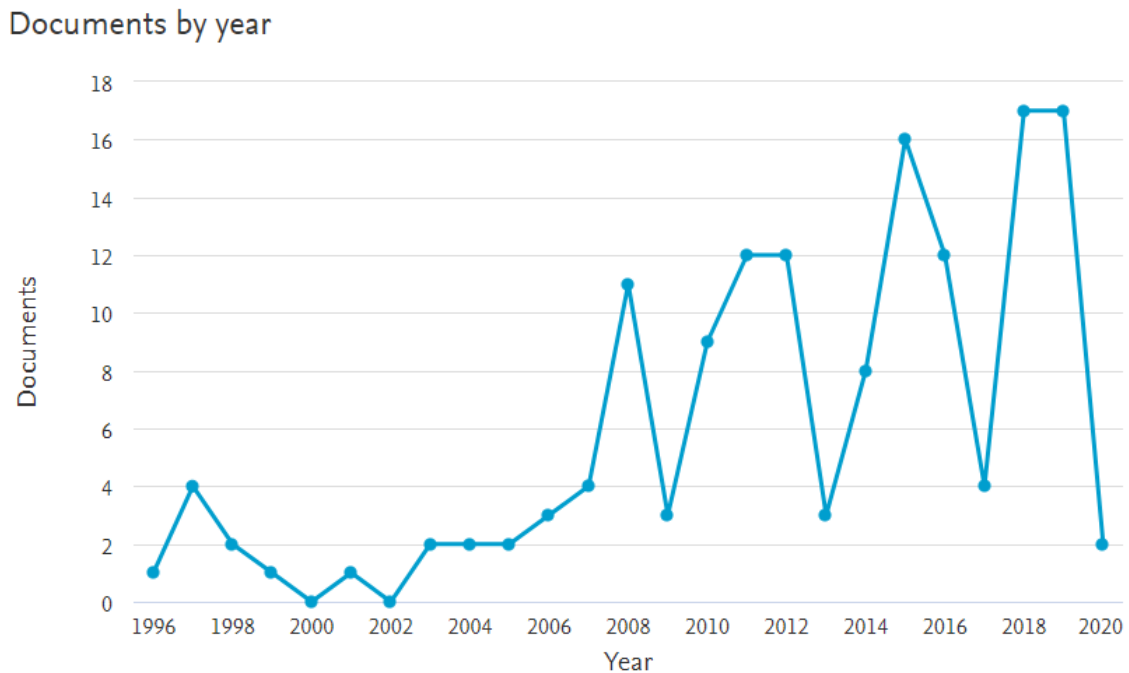
### 2.1. Análisis Bibliométrico Scopus

El siguiente análisis bibliométrico parte de la necesidad de obtener una visión más amplia acerca de la programación de proyectos con recursos restringidos, que permita analizar la actividad de la investigación científica desde el año 1996 hasta el presente.

A través de la base de datos Scopus se ingresó la ecuación de búsqueda: (*"critical chain"* AND *rcpsp*) OR (*"resource constrained project scheduling problem"* OR *rcpsp*) AND (*duration*) AND (*uncertain* OR *uncertainty* OR *stochastic* OR *random*) de la cual se obtuvo un total 148 artículos en los cuales se analizarán sus autores, año de publicación, países en donde fueron publicados y áreas del conocimiento.

#### 2.1.1. Documentos por año

La figura 1 muestra que en los primeros años los aportes a la literatura fueron entre uno y cuatro artículos hasta el 2007, posteriormente aumentó la actividad de investigación desde el año 2008 en adelante, donde se registraron entre 11 y 18 publicaciones en diferentes periodos; tanto así que los años 2015, 2018 y 2019 es donde hubo mayor producción literaria lo cual muestra que el problema está siendo estudiado con gran auge en las diferentes áreas de investigación.

**Figura 1***Publicación por año del RCPSP en Scopus*

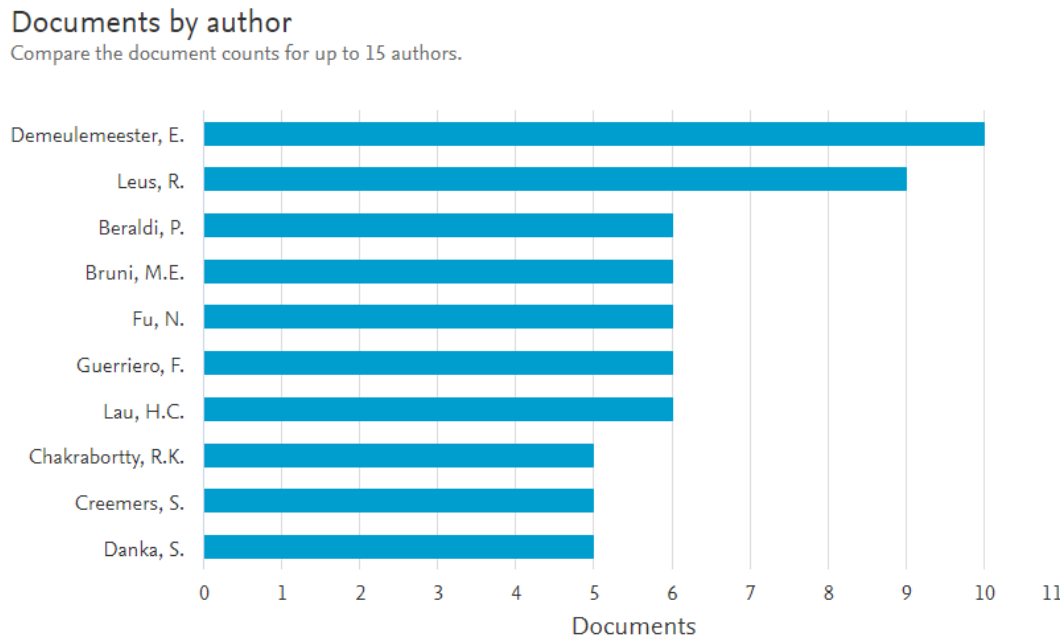
*Nota.* Tomado de la base de datos Scopus, 2021, Elsevier.

**2.1.2. Autores**

La figura 2 muestra los 10 autores que más aportes han hecho al RCPSP no determinístico, siendo Demeulemesster, E. el autor que lidera con 10 artículos publicados desde el 2006 hasta el 2019; Leus, R. quien cuenta con nueve publicaciones, seguidamente Beraldi, P. Bruni, M.E. Fu, N. Guerriero, F. y Lau, H.C. con seis artículos cada uno y por último Chakrabortty, R.K. Creemers, S. y Danka, S. con cinco documentos publicados desde el 2011 en adelante.

**Figura 2**

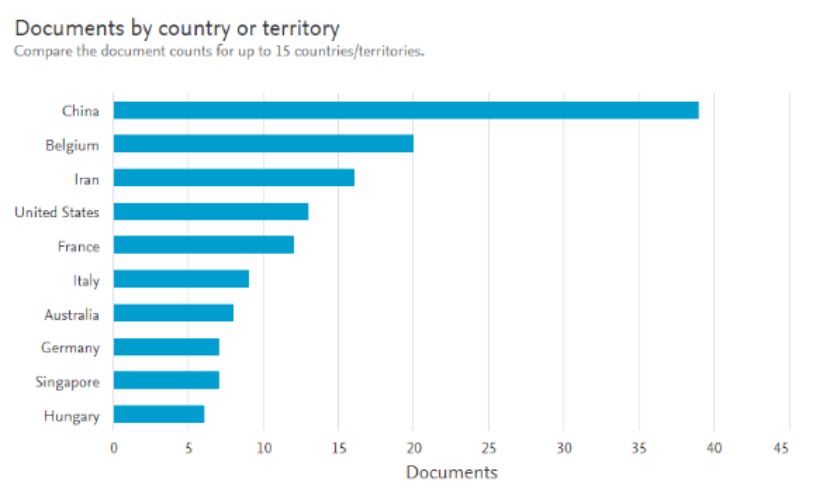
*Autores de mayor relevancia en Scopus*



*Nota.* Tomado de *la base de datos Scopus, 2021, Elsevier.*

**2.1.3. Territorio**

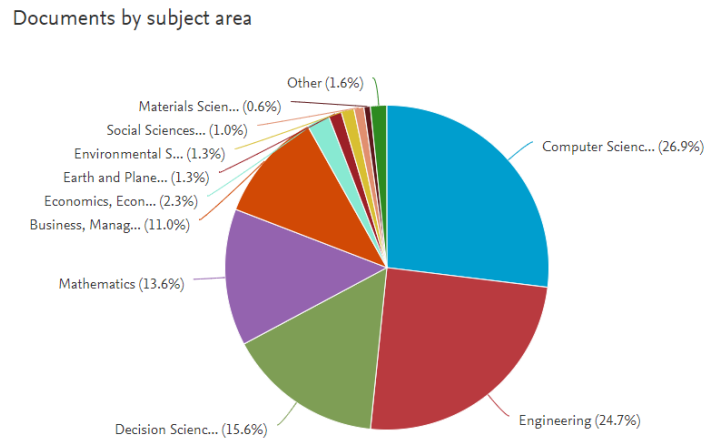
Tal como muestra la figura 3 China es el país con mayor producción científica concerniente al RCPSP no determinístico, con un total de 39 artículos, Bélgica e Irán por su parte oscilan entre 15 y 20 publicaciones, Estados Unidos y Francia no superan los 13, y por último Italia, Australia, Alemania, Singapur y Hungría han publicado entre seis y nueve artículos.

**Figura 3***Países líderes en investigación del RCPSP de Scopus*

*Nota.* Tomado de la base de datos Scopus, 2021, Elsevier.

**2.1.4. Áreas de investigación**

La figura 4 muestra el porcentaje de publicaciones realizadas sobre el RCPSP desde 1996 hasta el presente en 13 diferentes áreas del conocimiento, las más relevantes son la ingeniería y la ciencia de la computación en 51,6%, seguidamente se encuentran la ciencia de la decisión, las matemáticas y la gestión empresarial en 40,2% y un 8,2% que se distribuye en las seis áreas restantes.

**Figura 4***Áreas de investigación del RCPSP en Scopus*

*Nota.* Tomado de la base de datos Scopus, 2021, Elsevier.

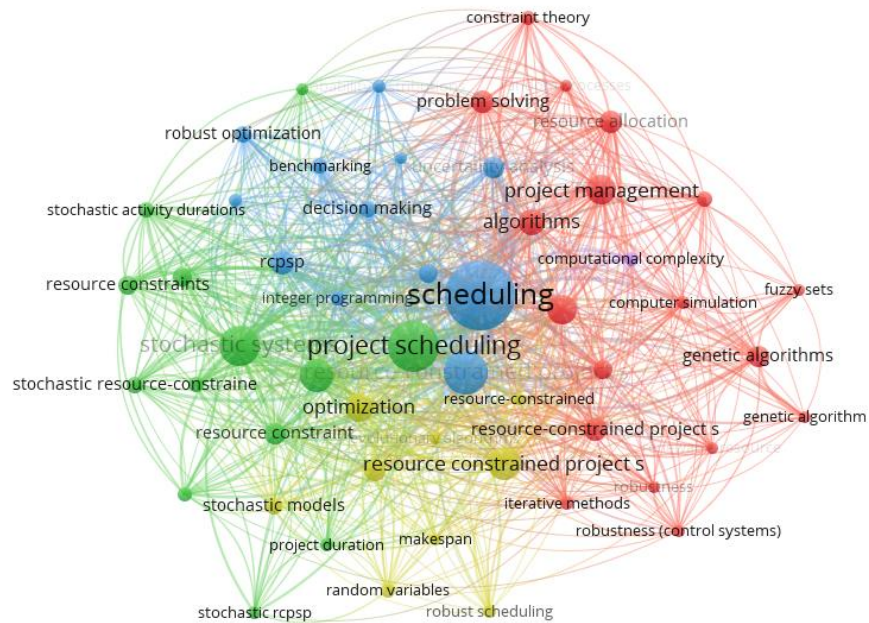
Además de las gráficas expuestas anteriormente, se usó el software de libre programación VOSviewer que sirve como herramienta para visualizar datos bibliométricos y estudiar que tan fuerte es la relación de ciertos factores dentro de los artículos según la distancia de estos.

**2.1.5. Keywords**

La figura 5 muestra el mapa de las palabras clave de scopus por medio de círculos que determinan la frecuencia de aparición de dichas palabras según su tamaño (entre más grande sea el círculo mayor aparición tiene la palabra). Se obtuvo 51 palabras distribuidas en cuatro clusters cada uno agrupado por temáticas. Para el caso del conjunto azul la palabra con mayor relevancia es “scheduling”, para el conjunto verde “project scheduling” y “stochastic system”, en el conjunto amarillo fue “resource constrained project scheduling”, por último, para el conjunto rojo fue “heuristic methods”.

**Figura 5**

*Red bibliométrica de palabras clave en Scopus*



*Nota.* Tomado del software *VOSviewer*, por N. J. Van Eck y L. Waltman, 2021, Centro de Estudios de Ciencia y Tecnología (CWTS).

En La tabla 2 se define la relación entre las principales palabras que aparecen en Scopus con el RCPSP no determinístico, se organizan de mayor a menor debido a su ocurrencia en los resultados obtenidos. El total link strength representa la fuerza total de enlace de las palabras clave entre los artículos.

**Tabla 2**

*Palabras clave con Scopus*

Palabras clave	Ocurrencia	Total link Strength
Scheduling	110	611
Project scheduling	64	335

*Continuación tabla 2 (palabras clave con Scopus)*

Palabras clave	Ocurrencia	Total link Strength
Resource-constrained Project scheduling problem	50	292
Stochastic systems	44	272
Activity duration	31	212
Resource constrained Project scheduling	32	192
Heuristic methods	26	170
Optimization	24	158
Project management	27	151
Algorithms	22	145

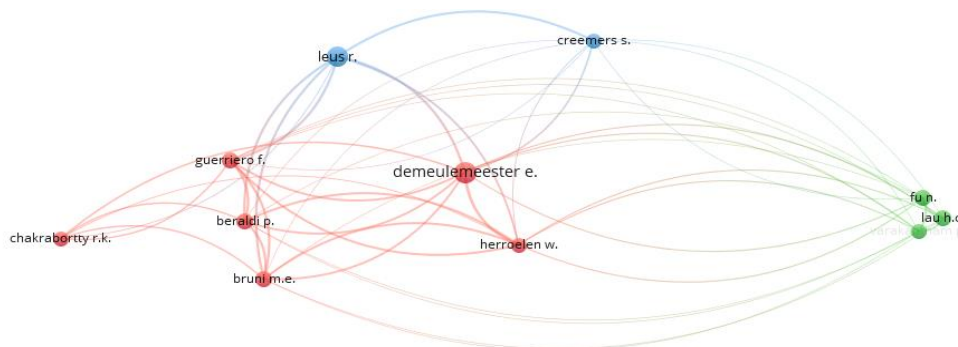
*Nota.* Tomado de la base de datos Scopus, 2021, Elsevier.

### 2.1.6. Citación de autores

La figura 6 muestra la relación entre las citaciones de un investigador con el otro mediante el número de veces que se citan entre ellos. Siendo Demeulemeester e. Leus r. y Herroelen los autores más nombrados entre la comunidad.

### Figura 6

*Red de citación de autores Scopus*



*Nota.* Tomado del software VOSviewer, por N. J. Van Eck y L. Waltman, 2021, Centro de Estudios de Ciencia y Tecnología (CWTS).

La tabla 3 muestra la cantidad de documentos y el número de citas por autor. Demeulemeester es el autor que más ha realizado aportes en el tema, también es el más citado, seguido de Leus R. y Herroelen W. con más de 300 citas de sus diferentes artículos.

**Tabla 3**

*Citaciones por autores*

<b>Autores</b>	<b>Documentos</b>	<b>Citaciones</b>	<b>Total link Strength</b>
Demeulemeester E.	10	414	77
Leus R.	9	373	71
Herroelen W.	5	404	68
Beraldi P.	6	119	64
Bruni M.	6	119	64
Guerriero F.	6	119	64
Fu N.	6	78	26
Lau H. C.	6	72	26
Varakantham P.	5	72	26
Chakraborty R. K.	5	39	25

*Nota.* Tomado de la base de datos Scopus, 2021, Elsevier.

### **2.1.7. Acoplamiento bibliográfico**

En cuanto al acoplamiento bibliográfico se encuentra la tabla 4 que tiene como orden de referencia el total link strength en los que dos o más investigadores han trabajado en conjunto sobre el tema y validan la usabilidad del documento mediante las citas. Se observa que los investigadores con mayor relación entre ellos son Beraldi P., Bruni M. E. y Guerriero F. con 119 citas cada uno.

**Tabla 4***Acoplamiento bibliográfico entre autores*

<b>Autores</b>	<b>Documentos</b>	<b>Citaciones</b>	<b>Total link Strength</b>
Beraldi P.	6	119	12
Bruni M. E.	6	119	12
Guerriero F.	6	119	12
Fu N.	6	78	10
Lau H. C.	6	72	9
Varakantham P.	5	72	9
Herroelen W.	5	404	6
Demeulemeester E.	10	414	5
Leus R.	9	373	5
Creemer S.	5	58	2

*Nota.* Tomado de la base de datos Scopus, 2021, Elsevier.

### 2.1.8. Citación por revistas

La tabla 5 muestra las revistas ordenadas de mayor a menor número de citaciones y documentos publicados, aquellas que han presentado mayor actividad en el tema de investigación son Journal of scheduling y European journal of operational research con más de 10 documentos y numerosas citaciones.

**Tabla 5***Citación por revistas en Scopus*

<b>Revista</b>	<b>Documentos</b>	<b>Citaciones</b>	<b>Total link Strength</b>
European journal of operational research.	11	1430	27
Journal of scheduling.	10	408	38

*Continuación table 5 (citación por revistas en Scopus)*

Revista	Documentos	Citaciones	Total link Strength
Computers and industrial engineering	6	158	13
Ieee international conference on industrial engineering and engieneering management.	4	10	5
Procedia computer science	4	28	5
Lecture notes in computer science	4	11	2

*Nota.* Tomado de la base de datos Scopus, 2021, Elsevier.

La línea de tiempo representada en la figura 7 muestra las revistas más importantes respecto a citaciones en un periodo de tiempo que abarca desde el 2012 al 2014.

## Figura 7

*Revistas más citadas en línea de tiempo*



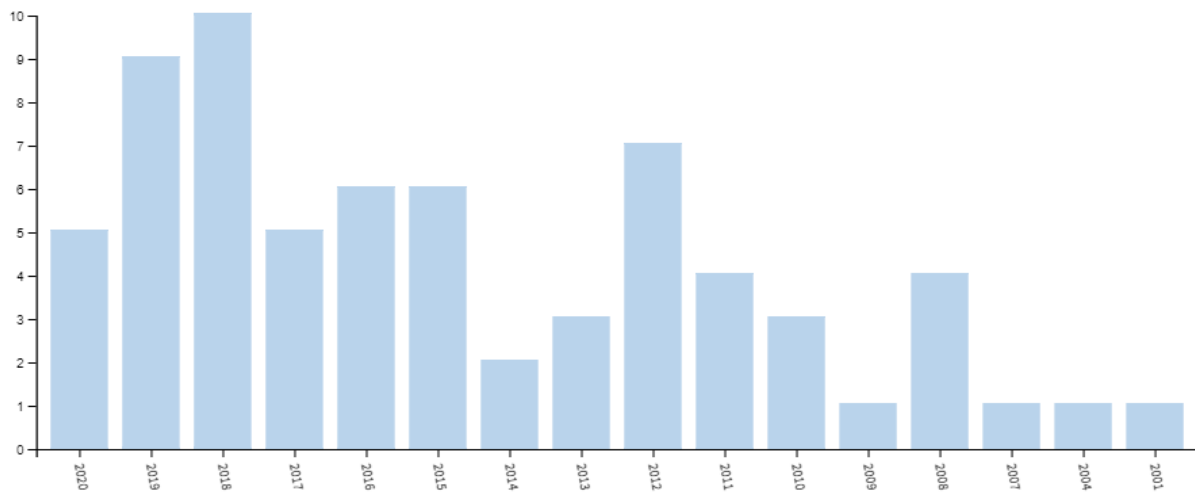
*Nota.* Tomado del software VOSviewer, por N. J. Van Eck y L. Waltman, 2021, Centro de Estudios de Ciencia y Tecnología (CWTS).

## **2.2. Análisis Bibliométrico Web Of Science**

Para obtener una mayor comprensión de la actividad de investigación sobre el RCPSP, también se abordó la base de datos web of science donde, por medio de la ecuación de búsqueda  $TS = ((\text{"critical chain"} \text{ AND rcpsp}) \text{ OR } (\text{"resource constrained project scheduling problem"} \text{ OR rcpsp}) \text{ AND } (\text{duration}) \text{ AND } (\text{uncertain OR uncertainty OR stochastic OR random}))$ , se obtuvo un total de 68 artículos publicados entre los años 2001 y 2020; a partir esto se realizó el siguiente análisis bibliométrico.

### **2.2.1. Documentos por año**

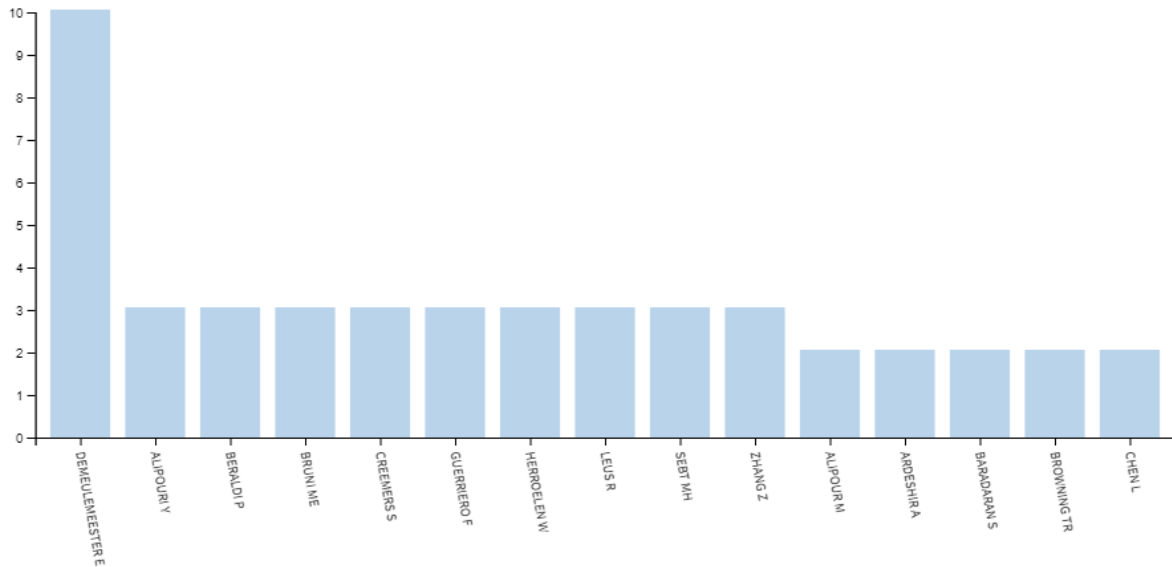
La figura 8 representa la cantidad de artículos publicados por año en web of science en las 2 últimas décadas, se puede observar que la actividad de investigación del RCPSP no determinístico ha sido muy diferente en cada año, ya que en los primeros años la cantidad máxima de artículos encontrados fue de 4 en el 2008 y 2011 en comparación con los años 2001 a 2007 con una cantidad de 1 artículo cada uno, también es notorio que el interés de investigar en el tema ha aumentado ya que desde el año 2012 al 2020 la cantidad de artículos publicados fue entre 2 y 10 por año.

**Figura 8***Publicaciones por año en Web of Science*

*Nota.* Tomado de la base de datos *Web of Science*, 2021, Clarivate Analytics.

**2.2.2. Autores**

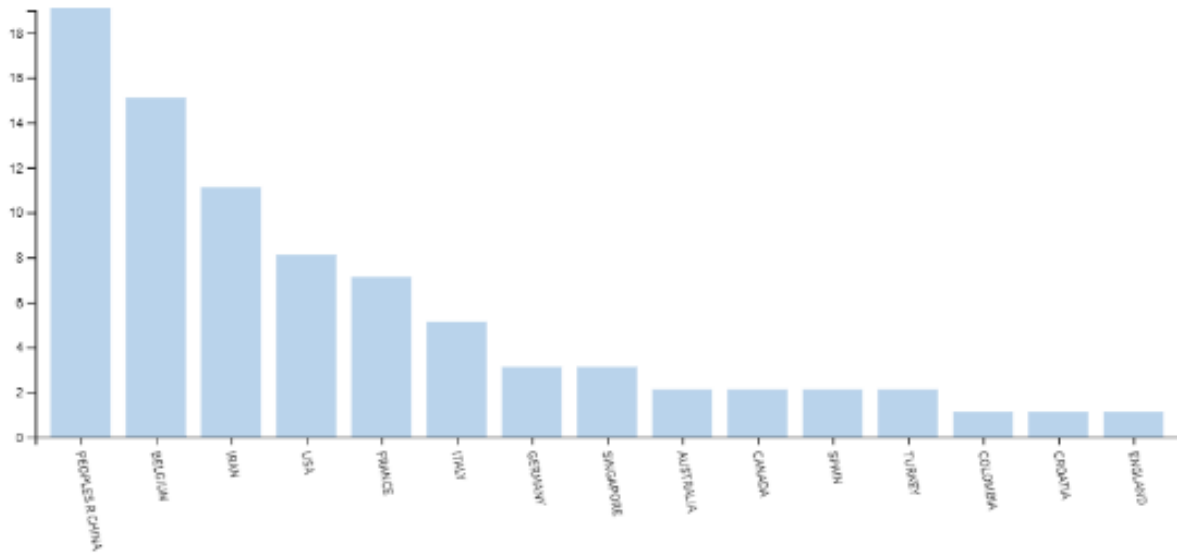
La figura 9 representa los autores más relevantes encontrados en web of science y la cantidad de artículos que han publicado a lo largo de los últimos 20 años, al igual que se encontró en Scopus el autor que más aportes ha realizado al RCPSP no determinístico es Demeulemeester E. con un total de 10 artículos, por otra parte también se observa que han publicado como mínimo 3 artículos varios autores como Alipouri Y., Beraldi P., Bruni M. E., Creemers S., Guerrero F., Herroelen W., Leus R., Sebt M. H., y Zhang Z.; y otros de los autores más representativos en el RCPSP no determinístico que han publicado como mínimo 2 artículos son Alipour M., Ardeshir A., Baradaran S., Browning T. R., y Chen L.

**Figura 9***Autores de mayor relevancia en web of science*

*Nota.* Tomado de la base de datos *Web of Science*, 2021, Clarivate Analytics.

### 2.2.3. Territorio

La figura 10 representa los principales países del mundo que han realizado aportes al RCPSP no determinístico, el país líder en investigación es china con un total de 19 artículos publicados, seguido de este se encuentran Bélgica con 15 artículos e Irán con 11; otros países como Estados Unidos, Francia, Italia, Alemania y Singapur han publicado entre 3 y 8 artículos. Otra gran cantidad de países han realizados aportes de 1 y 2 artículos dentro de estos se encuentra Australia, Canadá, España, Turquía, Croacia, Inglaterra y Colombia.

**Figura 10***Países líderes en investigación del RCPSP de Web of Science*

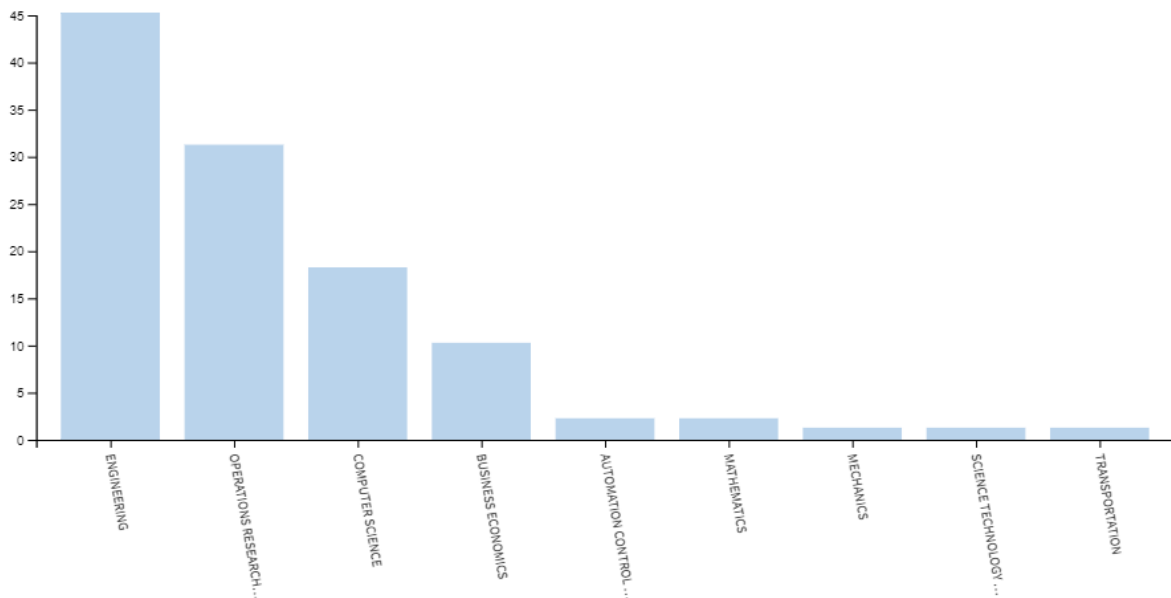
*Nota.* Tomado de la base de datos Web of Science, 2021, Clarivate Analytics

#### **2.2.4. Áreas de Investigación**

La figura 11 muestra las áreas de investigación que han realizado aportes cada año en materia del RCPSP no determinístico, cabe aclarar que la cantidad de artículos que muestra cada área puede ser compartida con otra área que haya trabajado en conjunto por lo cual se muestra una suma mayor a los 68 artículos encontrados. El área líder de investigación es la ingeniería con un total de 45 y artículos, seguido de esta se encuentra las ciencias de gestión de la investigación de operaciones con 31 artículos, la ciencia de la computación con 18, y negocios económicos, sistemas de control de automatización, matemáticas, mecánica, ciencia de la tecnología y transporte han publicado entre 1 y 10 artículos.

**Figura 11**

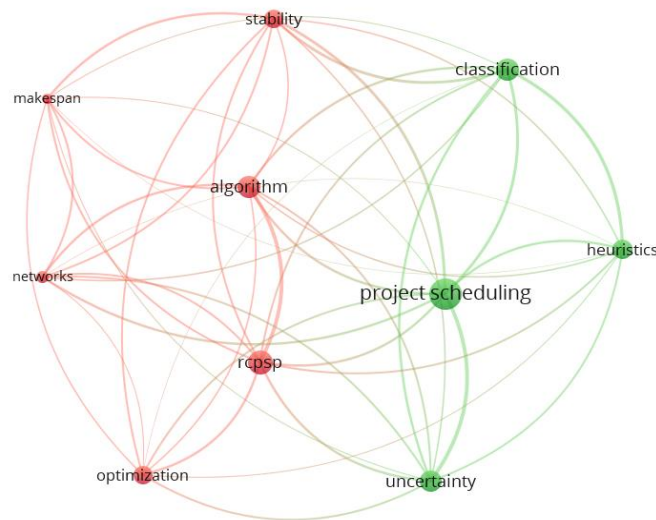
*Áreas de investigación de RCPSP en Web of Science*



*Nota.* Tomado de la base de datos *Web of Science*, 2021, Clarivate Analytics

### 2.2.5. *Keywords web of science*

La figura 12 muestra el mapa de las palabras clave que se encontraron en Web of Science, se obtuvo 10 palabras claves como las principales y de mayor relevancia dentro de los artículos las cuales se identificaron en 2 clusters, el conjunto verde con la palabra de mayor relevancia project scheduling y el rojo con la palabra rcpsp.

**Figura 12***Red bibliométrica de palabras clave en Web Of Science*

*Nota.* Tomado del software *VOSviewer*, por N. J. Van Eck y L. Waltman, 2021, Centro de Estudios de Ciencia y Tecnología (CWTS).

La tabla 6 expone la relación entre las principales palabras que aparecen en Web of Science con el RCPSP no determinístico, tienen una ocurrencia de 7 a 23 y un total link strength que oscila entre 27 a 46, la cual representa la fuerza total de enlace entre las palabras.

**Tabla 6.***Palabras clave con Web of Science*

<b>Palabras clave</b>	<b>Ocurrencia</b>	<b>Total link Strength</b>
Project Scheduling	23	46
Rcpsp	17	43
Algorithm	16	40
Classification	16	39
Uncertainty	15	39
Stability	13	38

Continuación tabla 6 (Palabras clave con Web of Science)

Palabras clave	Ocurrencia	Total link Strength
Networks	8	32
Heuristics	14	30
Optimization	13	28
Makespan	7	27

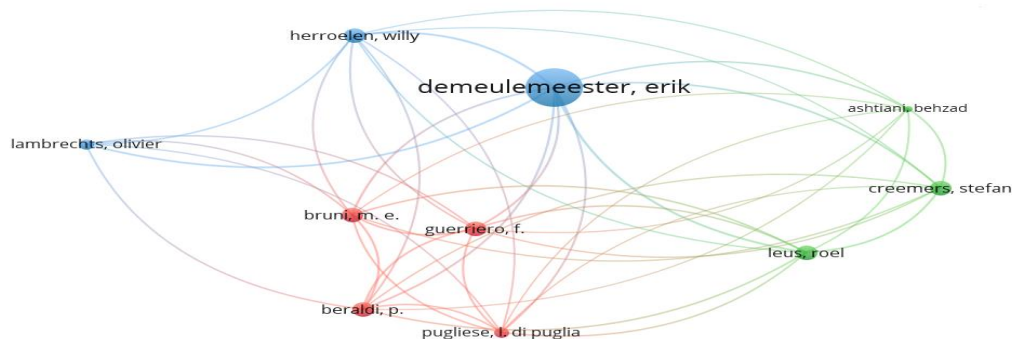
Nota. Tomado de la base de datos Web of Science, 2021, Clarivate Analytics

### 2.2.6. Citación de Autores Web of Science

La figura 13 ilustra la relación entre las citaciones de un autor con el otro en la base de datos web of science, Demeulemeester Erik es el autor más citado entre la comunidad de investigadores. Los otros presentan frecuencias en citaciones similares.

### Figura 13

Red de citación de autores Web of Science



Nota. Tomado del software VOSviewer, por N. J. Van Eck y L. Waltman, 2021, Centro de Estudios de Ciencia y Tecnología (CWTS).

La tabla 7 muestra el número de veces que cada autor ha sido citado y en cuántos y cuáles documentos, de igual forma su total link strength. Demeulemeester E., al igual que en el análisis

realizado para Scopus, lidera con mayor número de documentos y citas, seguido por Beraldi P., Bruni, M. E. y Guerriero F., cada uno con 86 citas y una producción de 3 documentos.

**Tabla 7**

*Citación por autores de Web of Science*

<b>Autores</b>	<b>Documentos</b>	<b>Citaciones</b>	<b>Total link Strength</b>
Demeulemeester E.	10	297	60
Beraldi P.	3	86	35
Bruni, M. E.	3	86	35
Guerriero F.	3	86	35
Leus Roel	3	120	34
Herroelen W.	3	205	34
Pulgieste D.	2	45	25
Creemers S.	3	51	23
Lambrechts O.	2	151	22
Ashtiani B.	1	48	14

*Nota.* Tomado de la base de datos *Web of Science*, 2021, Clarivate Analytics

### **2.2.7. Acoplamiento bibliográfico**

La tabla 8 representa el acoplamiento bibliográfico entre los diferentes investigadores que trabajan usualmente en conjunto sobre temas de investigación, la tabla tiene como orden de referencia el total link strength y se confirma que los autores Beraldi P., Bruni, M. E. y Guerriero F., son los que más colaboraciones han realizado en investigar el RCPS no determinístico.

**Tabla 8***Acoplamiento bibliográfico entre autores*

<b>Autores</b>	<b>Documentos</b>	<b>Citaciones</b>	<b>Total link Strength</b>
Beraldi P.	3	86	8
Bruni M. E.	3	86	8
Guerrero F.	3	86	8
Pugliese D.	2	45	6
Baradaran S.	2	28	4
Ghomi S.	2	28	4
Hashemi S.	2	28	4
Demeulemeester E.	10	297	3
Herroelen W.	3	205	3
Leus R.	3	120	0

*Nota.* Tomado de la base de datos *Web of Science*, 2021, Clarivate Analytics

### 2.2.8. Citación por revistas.

La tabla 9 muestra las revistas en las cuales se registra mayor aparición de artículos y citas en referencia al tema de investigación que se presenta, *Journal of scheduling* y *European journal of operational resource* tienen la mayor actividad con 364 y 173 citas respectivamente.

**Tabla 9***Citación por Revistas en Web of Science*

<b>Revista</b>	<b>Documentos</b>	<b>Citaciones</b>	<b>Total link Strength</b>
<i>Journal of scheduling</i>	10	364	30
<i>European journal of operational resource</i>	5	137	18

*Continuación table 9 (Citación por Revistas en Web of Science)*

Revista	Documentos	Citaciones	Total link Strength
Computers & industrial engineering	5	138	12
Journal of industrial and production engineering	3	16	5
International journal of production research	3	19	3

*Nota.* Tomado de la base de datos *Web of Science*, 2021, Clarivate Analytics

### 2.3. Análisis Preliminar de la Literatura

El RCPSP no determinístico ha sido estudiado desde mediados del siglo XX por diferentes investigadores que han aportado a la literatura, se han propuesto diferentes modelos de optimización y su metodología de solución. Para tener un acercamiento al problema de investigación se recomienda la lectura del artículo escrito por Ortiz & Diaz (2018) en el cual realizaron una revisión extensa de la literatura entre 1996 y 2018, donde se encontró que la principal técnica de solución es el algoritmo genético.

Con el fin de complementar el análisis sobre los principales aportes en torno al problema, se presentan a continuación, 21 artículos publicados entre 2019 y 2020: Davari & Demeulemeester (2019a) estudiaron el RCPSP con duración estocástica, mediante la formulación de un problema de programación proactivo-reactivo integrado con una función de costos combinada que incluye un costo de la programación de la línea base, así como los costos de una serie de reacciones. Para ello introdujeron soluciones al problema como políticas proactivas reactivas (PR) y discutieron que las políticas proactivas-reactivas son más poderosas e informativas que sus contrapartes tradicionales (es decir, una combinación de un programa de referencia y una política reactiva),

brindan una mejor estabilidad y solidez, y son más flexibles cuando se agregan restricciones adicionales al problema. Dentro del estudio se proponen cuatro modelos basados en programación dinámica aplicados en cuatro diferentes políticas proactiva-reactiva donde los cálculos arrojan que tres de los modelos mencionados brindan una mejor solución en comparación con el método convencional proactivo de criticidad de tiempo y reactivo con un sistema generador de secuencias en paralelo robusto.

En el mismo año, Davari & Demeulemeester (2019b) presentaron dos clases de reacciones para una política proactiva-reactiva (PR): ya que esta es una solución para el problema proactivo y reactivo de programación de proyectos con recursos restringidos. Reacciones basadas en selección y reacciones basadas en búfer, a partir de esto obtuvo como resultado que ambas reacciones, especialmente las que se basan en amortiguación, contribuyen en mayor medida a la construcción de una política óptima PR.

Chakraborty & Ryan (2019) proponen un enfoque tolerante a la incertidumbre (UTA) para el problema de programación de proyectos con restricción de recursos estocásticos (SRCPSP) con el objetivo de minimizar el resultado combinado de la duración esperada y costos de inestabilidad. Se propuso una heurística basada en búsqueda de vecindad variable (VNSH), que incorpora diferentes reglas de prioridad para solucionar el problema mediante un extenso experimento basado en simulación, dicho experimento reveló que el VNSH (LST-la última hora de inicio) y VNSH (CPRU- el camino crítico y regla de utilización de recursos) superaron a los demás.

Por otra parte, Chakraborty et al. (2020) abordan el problema RCPSP en entornos dinámicos con el fin de identificar la mejor regla de prioridad eficiente de seis que fueron propuestas. Mediante el uso de una heurística de búsqueda local basada en vecindario variable (VNSLSH) se simuló las diferentes PR y se concluyó que ninguno de estos es dominante sobre otros en el entorno dinámico de programación de proyectos.

Ahmadpour & Ghezavati (2019) presentaron un nuevo modelo para el problema de programación de proyectos con recursos limitados (RCPSP) considerando un calendario de trabajo para los miembros del proyecto, a partir de un factor de habilidad basado en el concepto de eficiencia. Con dicho modelo se busca disminuir el riesgo de desviación de la programación al asignar miembros con factores de habilidad más altos a actividades críticas; adicional a esto se obtuvo una disminución en la mano de obra requerida y una reducción en la duración del proyecto.

De otra parte, Alipouri et al. (2019) abordaron el Problema de programación de proyectos con restricciones de recursos estocásticos difusos (FS-RCPSP) a través de una política hiperheurística denominada Evolución diferencial auto adaptativa para la programación (SADESP) que tiene dos módulos clave: (1) un módulo (policyEvolver) que desarrolla la política de programación y (2) un procedimiento de programación dinámica (Schedule) que toma decisiones de programación utilizando una política de programación.

Posteriormente Alipouri et al. (2020) presentaron un modelo matemático basado en una red de flujo de recursos con variables aleatorias difusas. Luego, este modelo se transforma en un modelo de programación lineal entera mixta con variables nítidas, por lo cual, realizaron los siguientes aportes al problema: (1) se propone un modelo de programación lineal entera mixta (MILP) para resolver RCPSP cuando la aleatoriedad y la falta de claridad coexisten en las estimaciones de duraciones de actividades; (2) las incertidumbres están representadas por un nuevo enfoque en teoría de probabilidad difusa y variables aleatorias difusas.

Brčić et al. (2019) proponen funciones de costo basadas en umbrales que incluyan penalizaciones por incumplimiento, de esta manera se obtiene un modelo más flexible y realista. Dichas funciones dan como resultado la posibilidad de cambios proactivos que a su vez cumplan financieramente con lo presupuestado. Para lo cual se presentan dos enfoques metaheurísticos, despliegue flexible y descenso iterativo en línea basado en la simulación, estos logran mejoras en

cuanto al costo-rendimiento a cambio de una mayor carga computacional y mayor complejidad del espacio de la solución.

Wang et al. (2019) construyeron un modelo de optimización bi-objetivo para mostrar explícitamente la compensación entre la robustez de una programación y los costos de transferencia de recursos. El modelo emplea una representación alternativa de soluciones orientada a los recursos que no solo permite obtener el número de unidades de cada tipo de recursos que se mueven a través de la red del proyecto, sino que también especifica qué unidades de recursos se transfieren entre cada par de actividades. Se aplicaron dos metaheurísticas para solucionar el problema, NSGA-II que es un algoritmo genético evolutivo multiobjetivo basado en la población de datos y el PSA un algoritmo de recocido simulado de Pareto basado en la búsqueda local de datos. Ambos algoritmos fueron puestos a prueba con un caso de la vida real y comparados con un método de restricción E-Constraint donde demostraron ser beneficiosos y aplicables para el problema.

Ma et al. (2019) llevaron a cabo un experimento computacional basado en simulación reactiva usando el algoritmo de búsqueda tabú propuesto por Lambrechts et al. (2008b) para abordar el RCPSP proactivo. Los autores propusieron un marco general de medidas de robustez sustitutas basadas en holgura, mediante la introducción de tres parámetros para distinguir diferentes cálculos alternativos de cada medida y además teniendo en cuenta dos entornos de incertidumbre: disponibilidad de recursos estocásticos y duración de actividad estocástica. Los resultados fueron analizados de acuerdo con el impacto presentado por los parámetros sobre el desempeño de las medidas de robustez propuestas y se encontró que fueron eficaces comparadas con las medidas de referencia.

Chakraborty et al. (2019) desarrollaron un marco de evaluación de riesgos de programación (SRAF) para modelar las incertidumbres en la duración y los recursos y observar su impacto en los objetivos del proyecto. Para demostrar la aplicabilidad introdujo un heurístico de

desarrollo propio, titulado heurística de búsqueda local mejorada basada en movimientos EMBLSH, el cual puede resolver conjuntos de datos de referencia RCPSP unimodal y multimodal.

Kuhlemann et al. (2019) propusieron un marco para la toma de decisiones en condición de incertidumbre, el cual fue aplicado a tres problemas de optimización, entre ellos, el RCPSP.

Moradi et al. (2019) modelaron el problema teniendo en cuenta la disponibilidad de los recursos y la duración de las actividades simultáneamente, adicionalmente evaluaron una coalición formada entre contratistas de proyectos de construcción con el fin de reducir el makespan, aumentar la disponibilidad de los recursos y maximizar sus ganancias. Gracias a un modelo de optimización robusta (RO) para resolver el RCPSP y mediante un método de solución que contempla múltiples objetivos, en el que se modela la incertidumbre en un conjunto de escenarios, se demostró que las ganancias son mayores para los contratistas cuando se trabaja en coalición y no de manera independiente.

Balouka & Cohen (2019) sugieren un enfoque de optimización robusto para el RCPSP multimodal con duraciones de actividad inciertas. El modelo matemático se compone de: (1) un problema maestro que determina los modos seleccionados y la asignación de recursos para minimizar la duración nominal del proyecto, y (2) un subproblema que encuentra, para la solución maestra, la duración del peor de los casos dentro de un conjunto de incertidumbre poliédrica. Finalmente, el problema se resuelve mediante un enfoque de descomposición de Benders con cortes especializados.

Creemers (2019) planteó que el tamaño y la complejidad de una red de proyecto aumenta simultáneamente con el beneficio de la división de actividades, sin embargo, la misma puede ser afectada por la variabilidad de la duración de las actividades. Por esta razón, estudió el problema estocástico preventivo de programación de proyectos con recursos restringidos (PSRCPSP) y

presentó un modelo de solución exacto usando distribuciones hipo exponenciales para modelar la duración de las actividades y definir una nueva cadena de Markov de tiempo continuo (CTMC).

Tackenberg & Duckwitz (2019) presentaron una extensión del problema de programación de proyectos con restricción de recursos (RCPSP) aplicado el caso de mantenimiento frecuente de infraestructuras. Además, aplicaron un algoritmo evolutivo multicriterio con una nueva representación del problema que es capaz de revisar las asignaciones de tareas técnicas mientras que la duración de la tarea puede ser estocástica. El objetivo principal fue desarrollar planes de turnos que maximizan la tasa de utilización del personal técnico.

Subulan (2019) abordó el RCPSP no determinístico bajo la teoría de conjuntos difusos mediante la incorporación de parámetros y variables de decisión con valores de intervalo. Las operaciones aritméticas de intervalo se realizan utilizando información complementaria obtenida del director del proyecto. Así, el enfoque propuesto también puede tener en cuenta la actitud de los gestores de proyectos hacia el riesgo y produce soluciones más aceptables y libres de riesgo.

Faria et al. (2020) propusieron que la capacidad de los recursos renovables para realizar un trabajo pueda aumentarse para que puedan realizar trabajo adicional en una unidad de tiempo o puede disminuirse con la consiguiente reducción del trabajo realizado, lo cual combinado con la holgura que tienen algunas actividades en un cronograma concreto, puede utilizarse para absorber las desviaciones que puedan producirse durante la ejecución de un proyecto

Kucusayacigil & Ulusoy (2020) estudiaron el problema de programación de proyectos bi-objetivo, multi-proyecto y multi-modo con restricciones de recursos adoptando pares de objetivos y combinaciones de medidas de desempeño y como método de solución se trabaja con un algoritmo genético híbrido de ordenación no dominante. Para mejorar el algoritmo aplicaron un proceso de pase de atrás hacia adelante (BFP) y con esto generaron nuevas poblaciones y así mejoraron el posprocesamiento. Para implementar el BFP probaron diferentes alternativas reportando resultados

para diferentes combinaciones de funciones objetivo, adicional a esto implementaron un procedimiento de inyección para aumentar la diversidad en la población ya que genera un numero alto de soluciones no dominadas adquiriendo así mayor diversidad.

Khalilzadeh et al. (2020) propusieron un modelo difuso de programación de proyectos con recursos limitados multimodo y para dar solución a este, desarrollaron un algoritmo heurístico hibrido basado en un algoritmo genético en el que se utiliza el método de simulación 99 para tratar los parámetros inciertos.

Por último, se encontró una gran aplicación del MRCPSP. Sun et al. (2020) presentaron un simulador generalizado para investigar el impacto de diferentes tipos de interdependencia en la recuperación de la funcionalidad de las infraestructuras. Consideraron que hay múltiples modos posibles de ejecutar una tarea de restauración enmarcando el proceso de restauración de sistemas interconectados como un problema de programación de proyectos multimodal con recursos limitados (MRCPSP). Además, tuvieron en cuenta tres conjuntos de incertidumbres: la duración de la restauración y la demanda de recursos para ejecutar una tarea, así como la dependencia funcional entre sistemas. También implementaron tres tipos de interdependencias: interdependencia de recursos compartidos de recursos, la dependencia de la precedencia de la restauración y la dependencia de la funcionalidad.

### **3. Marco Teórico**

#### **3.1. Proyecto**

Un proyecto consiste en un conjunto de actividades o trabajos con las siguientes características:

- Cada actividad tiene una duración o tiempo requerido para realizarla o ejecutarla.
- Existe un orden en la ejecución de las actividades, que se describe así: una actividad A precede a una actividad B, (o A es predecesor de B, o B es sucesor de A) si A debe completarse antes de que B comience.
- Hay actividades que no tienen predecesoras, o actividades iniciales, y también hay actividades sin sucesoras, o actividades finales. (Kong, 2013, p.216).

En el área de administración de la producción y operaciones se dice que “un proyecto como una serie de trabajos relacionados que, por lo habitual, se dirigen hacia un producto mayor y cuyo desempeño requiere de un periodo considerable” (Chase et al., 2009, p.59).

- **Actividad:** “Es la realización secuencial e integrada de diversas acciones necesarias para alcanzar objetivos generales y específicos de un proyecto” (Chase et al., 2009, p.77).
- **Recursos:** Generalmente, las actividades de los proyectos necesitan una cantidad de recursos para llevarse a cabo, los recursos pueden ser personas, energía, dinero, máquinas, materiales, entre muchos otros. De manera general, los recursos pueden ser definidos por tres elementos: unidades, tipos y categorías. Las unidades son la cantidad base del recurso a utilizar, los tipos de recursos hacen referencia a los conjuntos de unidades de recursos que tienen capacidad de realizar una misma función y las categorías son características que comparten un conjunto de restricciones (Morillo, 2017, p.5).
- **Recursos renovables:** “Son aquellos cuyo total de unidades se encuentra disponible en cada unidad de tiempo” (Morillo, 2017, p.5).

- **Recursos no renovables:** Son aquellos que tienen una cantidad total de recursos disponibles para todo el proyecto, o un intervalo de este. A medida que las actividades consumen dicho recurso, este se va agotando: Así, las unidades de recurso asignadas a una actividad no podrán estar disponibles para su uso en otra actividad durante todo el proyecto (Morillo, 2017, p.6).
- **Recursos doblemente restringidos:** “Estos están limitados de dos maneras simultáneas, en un intervalo de tiempo (como los no renovables) y por unidad de tiempo (como los renovables)” (Morillo, 2017, p.6).

## 3.2. Representación de un proyecto

### 3.2.1. Diagrama de barras.

La forma más intuitiva de representación gráfica de la planificación son los diagramas de barras. son muy utilizados en proyectos porque proporcionan una representación fácil de entender por personas de procedencia muy diferente, pero aportando simultáneamente datos precisos. La forma básica de un diagrama de barras consiste en una lista de actividades en el lado izquierdo cada una de las cuales tiene asociada una barra horizontal de longitud proporcional a su duración (González et al., 2014, p.385).

### 3.2.2. Diagramas de red.

“Las redes proporcionan una forma natural de desplegar de manera gráfica el flujo de actividades de un proyecto a gran escala o de un proyecto de investigación y desarrollo (Hillier, 2010, p.368). Una red se compone de:

- Un nodo (es un símbolo que) representa la ocurrencia del evento de terminación de conjunto de actividades y el inmediato inicio de las actividades siguientes.
- Un arco es una actividad que sale del nodo donde se inicia su ejecución y se conecta al nodo en que concluye y el valor del arco es la duración de la actividad (Kong, 2013, p.217).

Las redes tienen tipos de actividades y de holguras dentro de un proyecto y se definen de la siguiente manera:

- Actividades restrictivas: En el tiempo de preparación se pueden realizar una serie de actividades restrictivas, que no son actividades internas a la ejecución del proyecto, pero que pueden afectar el inicio de algunas actividades. Suelen ser del tipo: petición de autorización, permisos municipales, condiciones estacionales, consecución de financiación, etc. (Velasco, 2015, p.29).
- Actividades ficticias: En una red puede ocurrir que, entre un mismo suceso inicial y final, aparezcan paralelamente varias actividades. Para cumplir el “principio de designación unívoca” se crean “actividades ficticias” y así evitar la posibilidad de que dos actividades diferentes tengan el mismo nodo inicial y final. Se procura dejar la conexión directa a la actividad de mayor duración (Velasco, 2015, p.30).
- Holgura de nodo: Es el margen de tiempo del cual se dispone para conseguir el suceso (i). si su valor es nulo, se dice que el nodo es restringido o crítico.
- Holgura total: Es el margen de tiempo que queda empezando la actividad lo antes posible y acabándola lo más tarde permisible (Velasco, 2015, p.44).

- Holgura libre: Es el margen de tiempo que queda empezando la actividad lo antes posible y acabándola también lo antes posible.
- Holgura independiente: Es el margen disponible cuando la actividad anterior ha terminado lo más tarde permisible y se quiere acabar lo más pronto posible.
- Holgura programada: Tiene por objeto distribuir la holgura de una sub ruta no crítica según algún criterio de razones subjetivas, como: incertidumbre de una duración, proximidad de una actividad al camino crítico, desconfianza en los plazos de entrega de algún proveedor, etc. (Velasco, 2015, p.44).

Dentro de los diagramas de redes también se incluyen técnicas para la administración de proyectos (Torres, 2014), las cuales se describen de la siguiente manera:

- PERT: son siglas de Program (o Project) Evaluation and Review Technique, “mediante esta técnica se trata de lograr que un proyecto determinado se ajuste a los tiempos establecido en su programa, eliminando los atrasos que en el desarrollo de este se puedan presentar” (Taibo, 2002, p.57). “La red PERT es un diagrama arborescente, con ramificaciones como las de los árboles, que señala los diferentes caminos que son necesarios recorrer para finalizar el proyecto y también indica la secuencia en que se deben realizar las tareas” (Taibo, 2002, p.58)
- CPM: El método de ruta crítica (critical path method) consiste en definir una cadena de actividades críticas que une los eventos inicial y final del programa de red, esto se define mediante el cálculo de los tiempos más tempranos y tardíos de cada actividad, y aquellas actividades en las que coincida dichos tiempos se definen como actividades críticas ya que no puede ser retardada sin afectar la duración total del proyecto (García, 2020).

- PDM: el método de diagrama de precedencias (precedence diagram method) es una herramienta que permite la programación de las actividades de un proyecto utilizando cajas que se denominan nodos en las que se anotan las que se llevarán a cabo y que se conectan entre sí por flechas y que muestran las dependencias que preceden a esas actividades (García, 2020).
- GERT: Acrónimo de Graphical Evaluation and Review Technique, es un procedimiento para el estudio de redes estocásticas de nodos o vértices del tipo lógico OR exclusivas, OR inclusivas y AND y ramas de multiparámetro (García, 2020).

### 3.3. Problemas de programación de proyectos

Algunos de los problemas de programación de proyectos son:

- Hallar el tiempo mínimo para completar todas las actividades del proyecto siguiendo el orden establecido y además encontrar cuales actividades son críticas en el proyecto, esto es, aquellas cuyo retraso aumenta la duración mínima del proyecto, y cuales pueden retrasarse sin afectar este tiempo.
- Determinar el costo mínimo de ejecución del proyecto, dados los costos de las actividades, para terminarlo en un tiempo especificado (Kong, 2013, p. 216).

Dentro del contexto general de los recursos limitados, existen dos problemas claramente diferenciados: la nivelación y la asignación de recursos.

- **Nivelación de recursos:** El problema de nivelar recursos surge cuando el consumo de ellos que se hace a lo largo de proyecto no es uniforme y consiste en hacer una redistribución de las tareas sin que afecte la duración del proyecto (González et al., 2014, p.403).

- **Asignación de recursos:** La asignación de recursos es un problema parecido al de nivelación pero con algunas particularidades. El problema surge cuando, según la planificación del proyecto, en alguna etapa del proyecto el nivel de utilización de recursos teóricos supera a los disponibles realmente, entonces hay que ajustar la planificación de las tareas de forma que en cada periodo de tiempo no se utilicen más recursos que los existentes, retrasando la realización de ciertas tareas. Puede ocurrir incluso que el retraso de estas tareas origine una mayor duración del proyecto (González et al., 2014, p.407).

### 3.4. Simulación de Monte Carlo

Simulación se entiende como el desarrollo de un modelo lógico-matemático de un sistema, de tal forma que se obtenga una copia del sistema a través del tiempo. La simulación involucra la generación de una historia artificial del sistema y la observación de esta con el uso de la manipulación experimental; además, ayuda a deducir y entender las características operacionales de tal sistema. (Azofeifa, 2008)

Dentro de los modelos de simulación se encuentran los elementos aleatorios o probabilísticos; estos contienen ejemplos de sistemas de colas, de inventario y modelos de análisis de riesgos, se les conoce como Simulación Monte Carlo (Azofeifa, 2008)

Un precursor de la simulación hasta la actualidad es el experimento De Montecarlo, que es un esquema de modelado que estima parámetros estocásticos o determinísticos en base en un muestreo aleatorio. La meta final de la simulación es estimar algunas medidas de desempeño deseables que describa el comportamiento del sistema simulado (Taha, 2017, p.647)

La simulación Monte Carlo es básicamente un muestreo experimental cuyo propósito es estimar las distribuciones de las variables de salida que depende de variables probabilísticas

de entrada. Los investigadores acuñaron este término por su similitud al muestreo aleatorio en los juegos de ruleta en los casinos de Monte Carlo.

Por otra parte, puesto que hoy en día los modelos de simulación pueden crearse y ejecutarse en una PC. El nivel de conocimientos de computación y matemática requeridos para diseñar y correr un simulador útil se ha reducido considerablemente. La capacidad de los modelos de simulación es una herramienta poderosa (Azofeifa, 2008, p.99).

- Se gana “experiencia” en forma barata simulando en el computador sin correr riesgos reales.
- Se obtienen resultados de manera rápida.
- Con los modelos de simulación es posible analizar sistemas muy complejos, donde los modelos analíticos no pueden llegar (Azofeifa, 2008, p.100).

“En la práctica, las pruebas aleatorias se sustituyen por resultados de ciertos cálculos realizados con números aleatorios” (Franco, 2013).

“En la simulación de Monte Carlo se agrupan una serie de procedimientos que analizan distribuciones de variables aleatorias usando simulación de números aleatorios. El método es aplicable a cualquier tipo de problema, ya sea estocástico o determinístico” (Federico, 2007, p.10).

Permite tener en cuenta para el análisis un elevado número de escenarios aleatorios, por lo que, se puede decir que hace posible llevar la técnica del análisis de escenarios al infinito ampliando la perspectiva de los escenarios posibles. De esta forma, se pueden realizar análisis que se ajusten en mayor medida a la variabilidad real de las variables consideradas. La aplicación de esta técnica se basa en la identificación de las variables que se consideran más significativas, así como las

relaciones existentes entre ellas (aunque esto puede resultar realmente complejo), para explicar la realidad a estudiar mediante la sustitución del universo real, por un universo teórico utilizando números aleatorios (Pérez, 2015, p.19).

### 3.5. Distribución Beta

Esta distribución de probabilidad es muy utilizada para representar variables de duración cuyos valores se encuentran restringidos a intervalos de longitud finita para obtener cantidades con límites de tolerancia sin la necesidad de plantear una hipótesis de normalidad.

Se dice que una variable aleatoria de tipo continuo sigue una distribución beta de parámetros  $p > 0$  y  $q > 0$  si su función de densidad está dada por la función (Román, 2017, p.21).

$$f(x) = \begin{cases} \frac{1}{\beta(p, q)} x^{p-1} (1-x)^{q-1} & 0 < x < 1 \\ 0 & \text{si no es así} \end{cases} \quad (1)$$

Román (2017), afirma:

Donde para los distintos valores de  $p$  y  $q$ , la forma de la función de densidad toma formas muy diferentes

- Si  $p = q$  la función de densidad es simétrica siendo el eje de simetría la recta  $x = 1/2$ .
- Si  $p = q = 1$ ,  $\beta(p, q) \equiv U(0, 1)$ .
- Si  $p < q$  es asimétrica a la derecha.
- Si  $p > q$  es asimétrica a la izquierda.
- Si  $p < 1$  y  $q \geq 1$  es decreciente y cóncava.

- Si  $q < 1$  y  $p \geq 1$  es creciente y cóncava.
- Si  $p > 1$  y  $q > 1$  tiene un sólo máximo.
- Si  $p < 1$  y  $q < 1$  tiene un sólo mínimo (p.23).

### 3.6. Esquemas Generadores De Secuencia

Un esquema generador de secuencia (Sequence generator schemes, SGS) es uno de los procedimientos más importantes en las metaheurísticas, porque permiten construir una solución factible a partir de un esquema de representación de soluciones que define una prioridad entre las actividades de un proyecto. Es decir, son algoritmos iterativos que permiten decodificar una solución expresada (Morrillo, 2017, p.37).

Hay dos SGS diferentes disponibles. El llamado SGS en serie realiza un incremento de actividad y el llamado SGS paralelo realiza un incremento de tiempo (Kolisch & Hartmann, 1996).

#### 3.6.1. *SGS en serie*

Se define como un algoritmo constructivo que genera una secuencia en  $n$  iteraciones, es decir en un número de pasos igual al número de actividades ( $n - 2$  si se incluyen las ficticias, la inicial y la final). El método toma una a una, cada actividad dependiendo de la representación, por ejemplo, de una lista de actividades, y la programa lo más pronto posible, teniendo en cuenta las restricciones de precedencia y de consumo de recursos de la solución parcial. El proceso se repite hasta que no quedan actividades por programar (Morrillo, 2017, p.37).

El procedimiento se describe de la siguiente manera por Kolisch (1995) existen 2 conjuntos de actividades: el conjunto programado  $S$  que son las actividades que están programadas y pertenecen a un programa parcial, el conjunto de decisiones  $D$  contiene las actividades que faltan por programar con cada predecesor. En cada iteración se programa una actividad de acuerdo con una regla de prioridad y se programa con su primer recurso y precedencia; posteriormente se elimina la actividad programada del conjunto de decisiones y se coloca en el conjunto programado.

### 3.6.2. *SGS en paralelo*

Este algoritmo genera soluciones en máximo  $n$  iteraciones. Cada iteración está asociada a un punto de decisión en un tiempo  $t$ , denominado  $PD(t)$ . Estos puntos están definidos como el instante  $t$  en el cual se liberan o están disponibles los recursos.

Adicionalmente, en cada  $PD(t)$  se crea un conjunto de actividades elegibles ( $AE(t)$ ) para ese instante  $t$ . Estas actividades deben ser factibles en precedencias. Así, en cada  $PD(t)$  se programa una actividad que se encuentre dentro del conjunto  $AE(t)$ , la cual será elegida basado en una representación, por ejemplo, en el orden de prioridad de una lista de actividades. Este proceso se repite hasta que no se puedan programar más actividades en ese instante y se define un nuevo punto de decisión  $PD(t)$ . El algoritmo finaliza cuando el conjunto  $AE(t)$  está vacío (Morrillo, 2017, p.38).

Se ha demostrado que el SGS paralelo construye programas sin demora. Un programa sin demora es un programa en el que, incluso si se permite la preferencia de actividad, ninguna de las actividades puede iniciarse antes sin retrasar alguna otra actividad. El conjunto de horarios sin demora es un subconjunto del conjunto de horarios activos. Por tanto, tiene en promedio, una cardinalidad menor (Kolisch & Hartmann, 1996).

La diferencia principal entre un sistema generador de secuencia y el otro es que el SGS en serie produce soluciones del tipo activas (AS) donde ninguna actividad puede iniciar más temprano sin retrasar el proyecto o violando la restricción de recursos; mientras que el SGS en paralelo produce soluciones del tipo no retrasadas (NDS) donde ningún recurso se mantiene inactivo cuando se puede utilizar para programar una actividad. A pesar de que las soluciones óptimas siempre pueden ser alcanzadas por el esquema en serie, el esquema en paralelo, por lo general, produce soluciones más compactas (de menor makespan). Por este motivo, en la literatura se usan los dos esquemas generadores (Morrillo, 2017, p.41).

### **3.7. Reglas De Prioridad Basadas En Heurísticas**

Las reglas de prioridad (Priority rules) son heurísticas que determinan el orden de prioridad de las actividades que son expresadas en los esquemas de representación o bien son usadas directamente por los SGS para construir una solución.

Existen numerosas reglas de prioridad propuestas en la literatura. Desde la más simple, asignando la prioridad aleatoriamente, hasta las más complejas que son calculadas de manera dinámica. Siguiendo la clasificación hecha por Klein (2000), estas podrían ser catalogadas en cuatro grupos (Morrillo, 2017, p.42).

Según el tipo de información empleada para calcular  $v(j)$ , se pueden distinguir las reglas basadas en la red, el tiempo y los recursos, así como las reglas de límite superior e inferior. Las reglas de límite inferior calculan para cada actividad un límite inferior del valor de la función objetivo, las reglas de límite superior calculan para cada actividad un límite superior del valor de la función objetivo.

La cantidad de información empleada se puede distinguir en reglas locales o globales. Las reglas locales emplean solo información de la actividad en consideración, como el tiempo de procesamiento, mientras que las reglas globales hacen uso de una gama más amplia de información. La Tabla ofrece una descripción general de las reglas de prioridad conocidas GRPW (mayor peso posicional de rango), LFT (última hora de finalización), LST (última hora de inicio), MSLK (holgura mínima), MTS (la mayoría de los sucesores totales), RSM (método de programación de recursos), SPT (tiempo de procesamiento más corto) y WCS (holgura en el peor de los casos) (Kolisch & Hartmann, 1996).

No existe una regla de prioridad que supere a las demás en todos los casos. Esto se debe a que establecer un orden específico en el cual las actividades deben ser programadas, es equivalente a resolver el problema. Sin embargo, es de gran importancia estudiar las reglas que mejores resultados obtienen para su uso en los métodos metaheurísticos (Morrillo, 2017, p.43).

### **3.8. Metaheurísticas de Solución**

Entre los metaheurísticos se destacan aquellos que imitan el comportamiento de sistemas naturales, tanto biológicos como físicos, como puede ser la evolución natural de las especies, la termodinámica, el trabajo cooperativo en una colonia de hormigas o el comportamiento de las neuronas en el cerebro. De la gran variedad de metaheurísticos existentes, se puede destacar, por sus excelentes resultados y la diversidad de problemas a los que están siendo aplicados, a los algoritmos genéticos, búsqueda tabú y templado simulado (Maroto et al., 2013, p. 269).

Hillier & Lieberman (2010) afirman:

la metaheurística es un tipo general de método de solución que organiza la interacción entre los procedimientos de mejora local y las estrategias de más alto nivel para crear un proceso que sea capaz de escapar de un óptimo local y realizar una búsqueda vigorosa de una región factible (p.566).

### **3.8.1. *Búsqueda tabú***

Definida por primera vez por Fred Glover y se conoce como un procedimiento metaheurístico que guía a un procedimiento de búsqueda local para explorar el espacio de soluciones más allá de los óptimos locales. Utiliza procedimientos diseñados para cruzar los límites de la factibilidad (para así evitar quedar atrapados en óptimos locales), que normalmente eran tratados como barreras infranqueables (Maroto, 2013, p.291).

En un procedimiento de búsqueda tabú cada solución  $S$  se asocia a un conjunto de soluciones vecino  $V(S)$ , y una solución  $S'$  que pertenece a  $V(S)$  es alcanzada por una operación  $m$  a la que se le llama movimiento. La idea es construir en cada iteración para la solución  $S$  el conjunto  $V(S)$  y elegir dentro de este el mejor individuo como una nueva solución. Dado que se puede elegir una solución peor que la anterior existe la posibilidad de visitar una solución más de una vez, por lo tanto, se utiliza la memoria para evitar volver visitar soluciones recientes (Maroto, 2013, p.292).

### **3.8.2. *Templado simulado.***

Es una técnica secuencial donde en cada iteración se obtiene una solución a partir de una solución anterior y el proceso se repite hasta que se satisface una condición de finalización.

Se puede denotar igual que en una búsqueda Tabú como  $V(S)$  como el conjunto de soluciones que puede ser obtenido a partir de la solución  $S$ , el procedimiento de iteración parte de una solución inicial  $S$  que se elige normalmente de forma aleatoria, a una solución cualquiera  $S'$  que pertenece  $V(S)$ . Esta solución  $S'$  es aceptada como nueva solución actual si es mejor que  $S$ . Sin embargo, si  $S'$  es peor que  $S$ , también existe cierta probabilidad de ser aceptada como solución actual. Dicha probabilidad depende tanto el valor objetivo de dicha solución como de un parámetro  $t$ , el cual es denominado por la analogía que tiene con un proceso físico de fundición de un sólido a altas temperaturas y después es dejado enfriar lentamente hasta volver a su estado sólido.

La elección del parámetro  $t$  influye de manera determinante en el comportamiento del algoritmo. Se debe fijar su valor inicial y la forma en la que va decreciendo dicho valor a lo largo del proceso, lo que en términos físicos se denomina programa de enfriamiento. Si el enfriamiento es demasiado rápido, la técnica tiende a comportarse como un simple mecanismo de búsqueda local, pudiendo quedar atrapado en un óptimo local o en una solución de baja calidad. Por otro lado, si el enfriamiento es demasiado lento, el tiempo de ejecución del algoritmo puede convertirse en prohibitivo (Maroto, 2013, p.295).

### **3.8.3. Algoritmo Genético**

Los algoritmos genéticos están muy influidos por otra forma de un fenómeno natural. En este caso, la analogía es con la teoría biológica de la evolución formulada por Charles Darwin a mediados del siglo 19. Cada especie de plantas y animales muestra grandes variaciones individuales. Darwin observó que aquellos individuos con variaciones que significan una ventaja de supervivencia a través de una mejoría en la adaptación al entorno

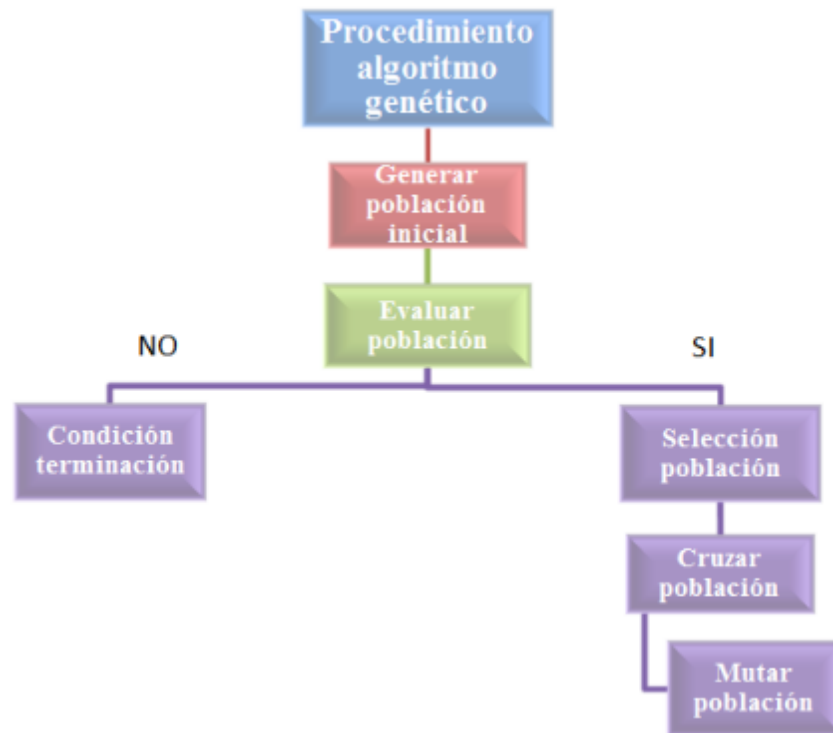
tienen una posibilidad mayor de sobrevivir en la siguiente generación. Este fenómeno se conoce desde entonces como en la supervivencia del más apto.

En algunas especies que se reproducen por vía sexual, cada descendiente hereda algunos de los cromosomas de cada uno de los padres, donde los genes de los cromosomas determinan las características individuales del hijo. Un hijo que hereda las mejores características de los padres tiene una mayor posibilidad de sobrevivir en su adultez y se convierte en un padre que pasa algunas de estas características a la siguiente generación. La población tiende a mejorar de manera lenta a través del tiempo por medio de este proceso (Páez y Moreno 2017, p.31).

Los algoritmos genéticos siguen paso a paso el procedimiento que ha seguido en la naturaleza. Trabajan sobre una población de individuos, cada uno de los cuales representa una posible solución al problema al que se aplican. A cada individuo se le asigna un valor de adaptación que representa la calidad de dicha solución. Los individuos de la población se cruzan entre sí para producir nuevas soluciones, de forma que los individuos con mejor valor de adecuación tienen más posibilidades de ser seleccionados para el cruce. Los individuos menos cualificados, es decir, las soluciones con un peor valor de adecuación tienen menos posibilidades, aunque también tienen algunas, de cruzarse con otras soluciones y hacer que sus características pasen a la siguiente generación (Maroto et al., 2013, p. 270).

**Figura 14**

*Procedimiento algoritmo genético*



*Nota.* Tomado de *Investigación operativa en administración y dirección de empresas* (p.272), por C. Maroto et al., 2013. Universidad Politécnica de Valencia.

El primer paso es generar la población inicial de soluciones, una vez que se tenga dicha población, que puede haberse obtenido bien de forma aleatoria, o bien a través de algún heurístico, cada uno de los individuos de dicha población es evaluado, es decir, le es asignado un valor de adecuación

Una vez ha sido generada la población inicial y ésta ha sido evaluada, se repiten los siguientes pasos y mientras no se cumpla la condición de finalización. Primero se realiza la selección de la población. En este proceso cada uno de los individuos de la población se copia un número determinado de veces, de forma que los mejores individuos tendrán en

general mayor número de copias que los individuos menos cualificados. De esta forma se obtiene una nueva población que reemplaza al anterior.

A continuación, se realiza el proceso de cruce. los individuos de la población se emparejan de forma aleatoria y sobre cada una de las parejas se aplica el mecanismo de cruce con una determinada probabilidad. Si el cruce se llevó a cabo por la pareja (padres), se crea dos nuevas soluciones (hijos) que remplazan a las anteriores.

Finalmente, algunos individuos de la población pueden sufrir mutaciones, es decir, algunas soluciones se pueden alterar parcialmente, lo que permite introducir en la población características nuevas o que se han perdido con la evolución (Maroto et al., 2013, p. 271).

En el proceso de implantar el algoritmo, es imperativo saber la respuesta a ciertas preguntas, por ejemplo, ¿Cuál será el tamaño de la población?, ¿Cómo deben seleccionarse los padres?, ¿Cuáles deberían ser las características que los padres deriven a sus hijos?, ¿Cómo deben insertarse las mutaciones en los hijos?, ¿Cuántas iteraciones deben usarse? Las respuestas dependerán de la estructura del problema.

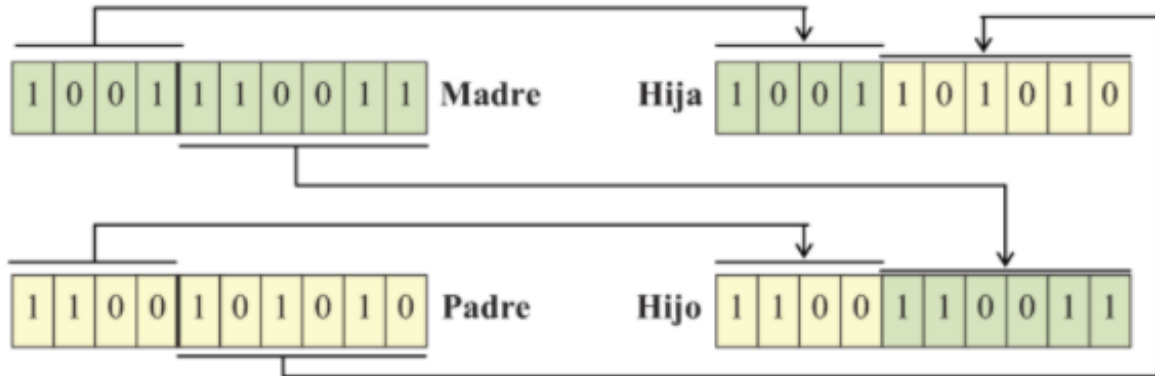
**3.8.3.1. Codificación de las Soluciones.** La posible solución a un problema se puede representar como un conjunto de parámetros. Estos parámetros, conocidos como genes, se puede poner uno a continuación del otro formando una cadena de valores, a la que se hace referencia como cromosomas. En términos genéticos el conjunto de parámetros representados por un cromosoma se denomina genotipo. El genotipo contiene la información necesaria para construir un organismo, denominado fenotipo (Maroto et al., 2013, p. 272).

**3.8.3.2. Función de evaluación.** Es aquella que asigna a cada uno de los individuos de la población el valor de adecuación, que indica la idoneidad de ese individuo con respecto al resto de individuos que forman parte de la población.

**3.8.3.3. Selección.** Serán los individuos con un mejor valor de adecuación aquellos que tendrán más posibilidades de permanecer en la población para poder así cruzarse con otros individuos y pasar su material genético a las siguientes generaciones. Los individuos con peor valor de adecuación tendrán más probabilidades de desaparecer.

Para simular este proceso natural, cada individuo se copia un determinado número de veces para formar una población intermedia que sustituye a la población anterior y que tiene el mismo tamaño que está. Es decir, los mejores individuos recibirán mayor número de copias que los peor dotados, que pueden incluso no recibir copia alguna y por tanto desaparecer.

**3.8.3.4. Cruce.** Se emparejan de forma aleatoria todas las soluciones de la población. Cada una de estas parejas se reproduce con una determinada probabilidad de cruce que es fija para toda la población. Si la pareja no se reproduce, cada uno de los individuos queda inalterado en la población. Por el contrario, si la pareja lleva a cabo el cruce, se crean dos nuevas soluciones que heredan una combinación de características de sus progenitores y que reemplazan a estos. De esta forma, tras el proceso de cruce, el tamaño de la población queda inalterado.

**Figura 15***Ejemplo de cruce***Figura 9.2. Ejemplo Cruce 1-punto**

*Nota.* Tomado de *Investigación operativa en administración y dirección de empresas* (p.272), por C. Maroto et al., 2013. Universidad Politécnica de Valencia.

**3.8.3.5. Mutación.** Es una especie de búsqueda a ciegas que pretende asegurar que en el espacio de soluciones no existe ningún punto con una probabilidad nula de ser exterminado.

La mutación afecta a cada uno de los genes de cada uno de los individuos de la población con una probabilidad determinada, la probabilidad de mutación suele tomar valores bastante pequeños mucho menores que los de la probabilidad de cruce. Es decir, todos y cada uno de los genes tienen igual probabilidad de ser afectados por la mutación. Mutar un gen consiste en alterar su valor (Maroto et al., 2013, p. 281).

Por medio de una simulación se puede investigar una hipótesis o conjunto de hipótesis utilizando modelos, se usa para conducir experimentos en una computadora digital, de esta manera

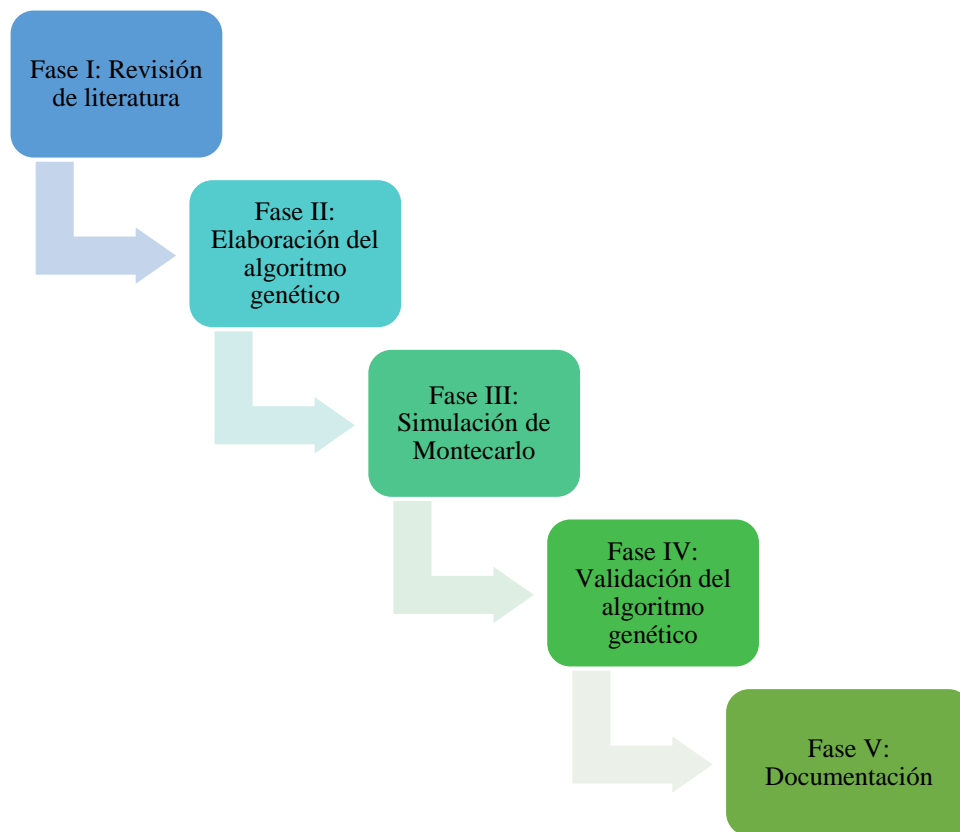
se evitan hacer pruebas en el sistema real evitando los costos que acarrearía un experimento en un entorno real.

#### 4. Metodología

El desarrollo del presente trabajo de investigación para solucionar el problema de programación de proyectos con recursos restringidos se hará a partir de las siguientes fases:

##### Figura 16

*Fases metodológicas de la investigación*



**Fase 1:** Revisión de la literatura

- Definir las palabras clave para la elaboración de la ecuación de búsqueda a través de las bases de datos disponibles en la biblioteca de la universidad
- Crear una ecuación de búsqueda con las palabras clave encontradas en las bases de datos Scopus y web of science (Esta revisión de literatura se completará en la entrega de libro final para el año 2020), para obtener los artículos relacionados con el RCPSP no determinístico.
- Realizar un análisis bibliométrico en cuanto a la actividad de las publicaciones de los artículos arrojados por la ecuación de búsqueda
- Revisión de la literatura donde se identifiquen los modelos propuestos y métodos de solución para el problema planteado

**Fase 2:** Elaboración del algoritmo genético

- Comprender y utilizar el lenguaje de programación Python para codificar el algoritmo genético
- Estudiar los sistemas generadores de secuencia en serie y en paralelo para programar las actividades del proyecto.
- Generar datos aleatorios para la duración de las actividades de un proyecto a partir de una simulación de Montecarlo.
- Desarrollar un algoritmo genético en Python basado en el algoritmo propuesto en la figura 9 del artículo de Ballestin del año 2007.

**Fase 3:** Simulación de Montecarlo

- Realizar la simulación de Monte Carlo para obtener 10.000 escenarios de ejecución para cada una de las instancias del proyecto que serán evaluadas en el algoritmo genético.

**Fase 4:** Validación del algoritmo genético

- Probar el algoritmo elaborado con diferentes instancias de redes proyectos tomadas de las librerías SPLIB
- Comparar la línea-base arrojada por la solución del AG con los 10.000 escenarios simulados con la técnica de Monte Carlo mediante indicadores de robustez de la solución y de la calidad.
- Comparar los resultados obtenidos del AG con el artículo de Ortiz & Diaz (2019) en el cual propusieron un modelo basado en programación lineal entera mediante indicadores de robustez de la solución y de la calidad.

**Fase 5:** Documentación

- Elaboración del documento del proyecto de grado con sus respectivas conclusiones
- Elaboración de un artículo de carácter publicable que contenga la metodología, resultados obtenidos y conclusiones.

## 5. Modelo Matemático

Como fue descrito anteriormente, el RCPSP no determinístico tiene como objetivo optimizar la ejecución de actividades de un proyecto en el mejor tiempo posible, sin sobrepasar los

límites de los recursos a utilizar en cada actividad y tomando en consideración las precedencias, con el fin de que se genere una línea base robusta que soporte cualquier tipo de disrupción que se presente al materializarse un riesgo durante la ejecución de una actividad.

Siendo así, el problema se encuentra sujeto a la condición de que cada actividad no sea ejecutada si se necesita una actividad precedente para realizarse, es decir, que se encuentra estrictamente necesaria la realización completa de una actividad para poder ejecutarse la siguiente, por ejemplo, cuando se realiza la construcción de una carretera y antes de realizar la excavación se necesita un estudio previo de suelos. Otra condición importante, es que no se sobrepasen los límites de cada recurso a utilizar entre varias actividades debido a que se pueden generar retrasos por el mal uso de estos, por ejemplo, en una empresa de tecnología en la que el recurso humano es tan necesario como limitado para el desarrollo de softwares especializados.

En la presente investigación se adopta el modelo propuesto por Ortiz (2019), donde resuelve el RCPSP no determinístico con duración de actividades aleatorias e inserción de tareas; Para efectos del modelo las actividades se identifican mediante un subíndice ( $i$  o  $j$ ) con valores entre 1 y  $n$ , y se toman la primera y última actividad como ficticias de duración cero con el fin de pautar el inicio y la finalización del proyecto ( $n+2$ ), el tiempo se representa con otro subíndice ( $t$ ) y toma valores entre 1 y  $T$ . La cantidad máxima de recurso para cada periodo  $t$  se define como  $W_{kt}$ . En cada período, la actividad  $i$  consume  $r_{ik}$  unidades del recurso renovable  $k$ . Se puede expresar como un programa lineal donde se definen las restricciones de recursos, y se establece el tiempo de inicio y duración de cada actividad.

### **Parámetros de entrada**

$D_i$ : Duración estimada de cada actividad

$P_{ij}$ : Parámetro binario que denota la precedencia entre las actividades. Tomará el valor de 1 si la actividad  $i$  precede a la actividad  $j$  y tomará como valor el cero en el caso contrario.

### Variables de decisión

$S_i$ : Tiempo de inicio planeado para cada actividad  $i$

$A_{it}$ : Variable binaria la cual indica la ejecución de la actividad  $i$  en un periodo de tiempo  $t$ .

Tomará el valor de 1 si se encuentra activa en el periodo  $t$  y tendrá valor cero en el caso contrario.

### Modelo

$$\text{Minimizar} \quad S_n \quad (2)$$

Sujeto a las siguientes restricciones:

$$S_j \geq P_{ij} * (S_i + d_i) \quad \forall (i, j) \in E \quad (3)$$

$$S_i + d_i - 1 \geq t * A_{it} \quad \forall i, \forall t \quad (4)$$

$$S_i \leq t + (1 - A_{it}) * M \quad \forall i, \forall t \quad (5)$$

$$\sum_{t=1}^T A_{it} = d_i \quad \forall i \quad (6)$$

$$\sum_{i=1}^n (r_{ik} * A_{it}) \leq R_{W_{kt}} \quad \forall k, \forall t \quad (7)$$

$$S_i \geq 0 \quad \forall i \quad (8)$$

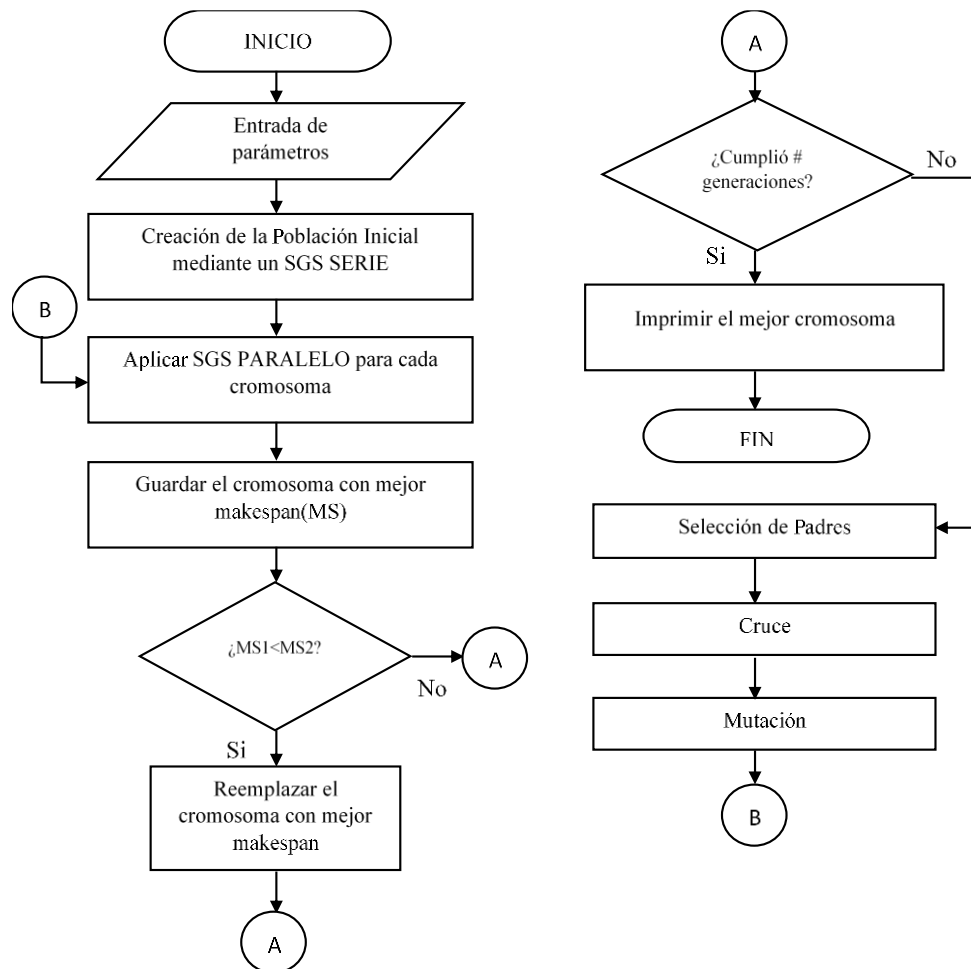
$$A_{it} \in \{0,1\} \quad \forall i, \forall t \quad (9)$$

$S_n$  indica el tiempo de inicio de la actividad ficticia  $n$ , la ecuación 3 indica que las actividades pueden iniciar solamente cuando las actividades que la preceden hayan terminado, las ecuaciones 4, 5 y 6 indican que  $A_{it}$  debe ser igual a 1 cuando la actividad  $i$  se encuentre activa en el momento  $t$ , de lo contrario  $A_{it}$  toma el valor de cero de la siguiente manera: la ecuación 4 indica que  $A_{it}$  es uno si y solo si la actividad  $i$  no ha finalizado en el momento  $t$ , la ecuación 5 permite que  $A_{it}$  sea 1 solamente cuando la actividad  $i$  haya iniciado después del momento  $t$ ,  $M$  representa un número muy grande, la ecuación 6 asegura que  $A_{it}$  sea uno para el intervalo  $\{S_i, S_i+d_i\}$

Adicionalmente, la ecuación 7 permite la programación simultanea de varias actividades tomando en consideración que no se superen los recursos disponibles. Por ultimo las ecuaciones 8 y 9 aseguran la no negatividad y el tipo de variable necesario.

## 6. Algoritmo Genético Propuesto

Con el fin de dar solución al modelo planteado se codifica un algoritmo genético en el lenguaje de programación Python en su versión 3,8, soportado en un sistema generador de secuencias serie y un sistema generador de secuencias en paralelo, por medio del cual se obtendrá una lista de actividades la cual se usará como orden prioridad para la programación de actividades en la simulación de Montecarlo. El presente algoritmo fue desarrollado a partir de un prototipo propuesto por Ballestin (2007). En la figura 17 se puede contemplar el diagrama de flujo del algoritmo y además en el apéndice A se encuentra la codificación realizada en Python.

**Figura 17***Diagrama del Algoritmo Genético*

Se establecieron los siguientes parámetros de entrada:

- **Tamaño de la población:** Numero de cromosomas a usar.
- **Número de generaciones:** Cantidad de poblaciones en diferentes periodos de tiempo.
- **Instancias:** Conjunto de datos agrupados donde se encontrará número de actividades, recursos usados por cada actividad, precedencias por cada actividad, duraciones de actividades y disponibilidad de recursos.

En la tabla 10 se muestra una instancia de tamaño j30, la cual ilustra las 30 actividades de un proyecto, su duración, actividades sucesoras y los 4 tipos de recurso R a consumir por actividad; para efecto de la descripción del algoritmo genético será ajustada dicha instancia para explicar cada una de las fases del proceso evolutivo.

**Tabla 10***Ejemplo de Instancia de Proyecto*

<b>Actividad</b>	<b>Actividades Sucesoras</b>	<b>Duración</b>	<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>R4</b>
			<b>14</b>	<b>10</b>	<b>11</b>	<b>14</b>
<b>1</b>	2 3 4	0	0	0	0	0
<b>2</b>	10	2	0	0	2	0
<b>3</b>	5 9 15	10	0	8	0	0
<b>4</b>	7 13	7	0	0	7	0
<b>5</b>	6 17	1	8	0	0	0
<b>6</b>	13 16 18	3	0	0	0	8
<b>7</b>	8 11 19	6	0	0	5	0
<b>8</b>	27	2	0	9	0	0
<b>9</b>	14 28	10	0	0	0	2
<b>10</b>	12 22	5	1	0	0	0
<b>11</b>	21 26	9	2	0	0	0
<b>12</b>	24	4	0	0	6	0
<b>13</b>	25	4	0	5	0	0
<b>14</b>	25	2	0	0	10	0
<b>15</b>	21	4	0	0	0	10
<b>16</b>	20 26	10	10	0	0	0
<b>17</b>	21 26 29	10	0	0	7	0
<b>18</b>	23	7	5	0	0	0
<b>19</b>	25	10	0	0	0	7
<b>20</b>	31	7	0	0	3	0
<b>21</b>	31	2	0	7	0	0
<b>22</b>	27 29	1	10	0	0	0
<b>23</b>	31	2	8	0	0	0
<b>24</b>	29	2	0	0	0	8

*Continuación tabla 10. Ejemplo de instancia de proyecto)*

<b>Actividad</b>	<b>Actividades</b>	<b>Duración</b>	<b>R1</b>	<b>R2</b>	<b>R3</b>	<b>R4</b>
	<b>Sucesoras</b>		<b>14</b>	<b>10</b>	<b>11</b>	<b>14</b>
<b>25</b>	30	2	0	0	0	7
<b>26</b>	27	9	7	0	0	0
<b>27</b>	30	1	0	0	0	1
<b>28</b>	30	4	8	0	0	0
<b>29</b>	32	7	0	3	0	0
<b>30</b>	32	8	0	0	8	0
<b>31</b>	32	9	0	6	0	0
<b>32</b>		0	0	0	0	0

## **6.1. Generación de la Población Inicial.**

Es conformada por un conjunto de posibles soluciones aleatorias pertenecientes al espacio de búsqueda y se genera a partir del cromosoma de un primer individuo. Es importante tener en cuenta que, aunque hay distintos métodos y se pueden obtener distintas soluciones, no siempre son factibles o pueden tener una aptitud muy baja, es decir la solución al problema no es la mejor.

### **6.1.1. Individuo**

El individuo que representa la solución del problema al RCPSP no determinístico será una lista de actividades (LA) ordenada de acuerdo con los tiempos de inicio. En esta investigación se lleva a cabo inmerso en la codificación del algoritmo un proceso de simulación de Montecarlo para hallar duraciones aleatorias y como resultado de esto crear el primer individuo; dicha simulación hace cumplimiento al objetivo número 2 del proyecto y a través de la figura 18 se puede observar la línea de código que se describe de la siguiente manera. Además, se puede encontrar la programación completa en el apéndice A.

**Figura 18***Creación del Primer Individuo***Algoritmo:** *Creación del Primer Individuo***Entrada:** *Instancia de Actividades***Salida:** *Lista de Actividades Ordenada de acuerdo con los Tiempo de Inicio*

```

def set_single_chromosome(self) -> Chromosome:
    """Creates a single chromosome, this process runs 1000 times to get a good average"""
    ## Initial list, range goes from 0 to 999 discounting this initial
    ## asignment
    chromosome = Chromosome(genes=self.get_single_sgs_serie())
    ## Division by 1001 due to the extra sample that is being taken
    ## (initial list).
    for act in chromosome.genes:
        act.start/=1001

    initialTime = time.time()
    for i in range(0, 5):
        print(f'Progress: {(i+1)/20}%')
        res :List[Chromosome] = self.multi_threading(threads=2)
        print(f'Number of lists: {len(res)}')
        for ch in res:
            for j in range(0, len(ch.genes)):
                chromosome.genes[j].start += ch.genes[j].start / 101

    print()
    print(f'Time elapsed for creating one Chromosome: \n {time.time()-initialTime}s\n')
    return chromosome

```

Se genera un vector con duraciones aleatorias para cada actividad con la característica de variar de acuerdo con una distribución de probabilidad beta, aplicando un esquema generador de secuencias (SGS) en serie que utiliza una regla de prioridad para programar cada actividad de la instancia; el proceso se repite 1.000 veces con el fin de generar nuevos tiempos de inicio que posteriormente se promediaran (tiempos de inicio y duraciones por cada actividad). Seguidamente los valores promedios para cada actividad se ordenan de menor a mayor para obtener la lista de actividades que constituiría el primer individuo (cromosoma) del algoritmo genético. En la figura 19 se encuentra un ejemplo gráfico de las duraciones aleatorias halladas por la distribución beta y el primer individuo del algoritmo genético de una instancia de proyecto j30.

**Figura 19***Representación del individuo*

DURACIONES ALEATORIAS		INDIVIDUO	
Actividad	Duración Beta	Actividad	Tiempos de Inicio
1	0.000	1	0.00
2	3.050	2	0.00
3	19.039	3	0.00
4	5.245	4	0.00
5	5.167	10	3.05
6	11.529	7	5.25
7	5.360	12	8.22
8	0.984	22	8.22
9	1.361	24	13.58
10	11.006	11	16.77
11	14.338	19	16.77
12	1.334	5	19.04
13	2.868	8	19.04
14	10.981	9	19.04
15	8.623	17	20.37
16	8.296	29	29.00
17	2.765	28	30.02
18	2.777	14	30.02
19	2.377	6	31.11
20	4.937	15	31.11
21	1.408	27	66.31
22	3.160	13	33.49
23	5.884	18	33.49
24	3.864	21	39.37
25	3.651	25	39.37
26	10.372	16	33.49
27	3.671	23	43.86
28	6.473	20	43.86
29	22.449	26	43.86
30	15.358	31	50.33
31	9.582	30	67.72
32	0.000	32	77.30

**6.1.1.1. Regla LFT (late finish time).** Al momento de programar la primera actividad se activan aquellas que son elegibles, como paso siguiente se escoge aleatoriamente una de ellas, se debe tener en cuenta que tiene mayor probabilidad de ser elegida aquella con mayor valor se calcula una probabilidad de selección  $P_j$  para cada actividad elegible  $j$  ( $P_j$ ) candidata a secuenciar en ese momento.

$$p_j = \frac{LFT_j}{\sum LFT_i} \quad (10)$$

Para ejemplificarlo, se toman 3 actividades, suponiendo que la LFT para las actividades 2, 5, y 9, son 14, 26 y 32 respectivamente, las  $P_j$  se hallarían de la siguiente manera:

$$p_2 = \frac{14}{14+26+32} \quad p_5 = \frac{26}{14+26+32} \quad p_9 = \frac{32}{14+26+32} \quad (11)$$

Las probabilidades son 0,194 0,361 y 0,444 se genera un número aleatorio y por sorteo se escoge la actividad a secuenciar, en este caso la actividad la actividad  $p_9$  tiene mayor probabilidad de ser seleccionada, sin embargo, no siempre será el caso.

**6.1.1.2. Distribución Beta  $\beta$ .** La distribución beta utilizada para hallar las duraciones aleatorias de las actividades en el algoritmo genético está determinada por parámetros  $p=2$  y  $q=5$ , lo cual al cumplirse que  $p < q$  la distribución es asimétrica a la derecha, además se cumplen que las duraciones estarán comprendidas entre un rango de duración de la actividad  $a = (11/20)$  di y  $b = (23/8)$  di.

### **6.1.2. Población Inicial**

Inicialmente se debe fijar el parámetro de entrada para el algoritmo que es la Cantidad de individuos  $n_{Pob}$  por población, después de haber creado el primer individuo, el segundo se obtiene programando las actividades de la instancia de proyecto en un SGS en serie que también toma como referente para la programación de actividades, la regla de prioridad LFT y además la

asignación de la duración por actividad que tiene en cuenta la duración promedio hallada para el primer individuo, después de completarse la programación de cada una de las actividades se obtiene una segunda lista de actividades, es decir, un segundo individuo.

El tercer individuo se obtiene aplicando nuevamente un SGS en serie, así como se realizó para el segundo individuo (con las mismas duraciones promedio). De igual forma, se obtendrán los siguientes individuos y al final se completará una población inicial nPob de individuos que corresponde a unas listas de actividades LA.

## **6.2. SGS Paralelo**

Cuando se completa la población inicial, a cada una de las Listas de Actividades (LA) se le aplica un SGS paralelo, el cual, programa las actividades basado en el orden de prioridad que representa el individuo y asigna la duración de las actividades de una manera determinista mas no de manera aleatoria, como lo hace en SGS serie. Al completarse la programación de todas las actividades, se obtendrá un makespan de cada individuo.

## **6.3. Guardar el Mejor Cromosoma**

De la primera población obtenida y programada por el SGS paralelo, se debe registrar y guardar en la memoria del algoritmo genético, el individuo que tenga el mejor makespan ya que para efectos de evolución del algoritmo, se espera que haya mejores soluciones a las de la primera población.

## **6.4. Selección de Padres**

Se lleva a cabo por medio del método de selección por torneo de tipo probabilístico, la selección se realiza en base a comparaciones entre los individuos, se selecciona aleatoriamente un

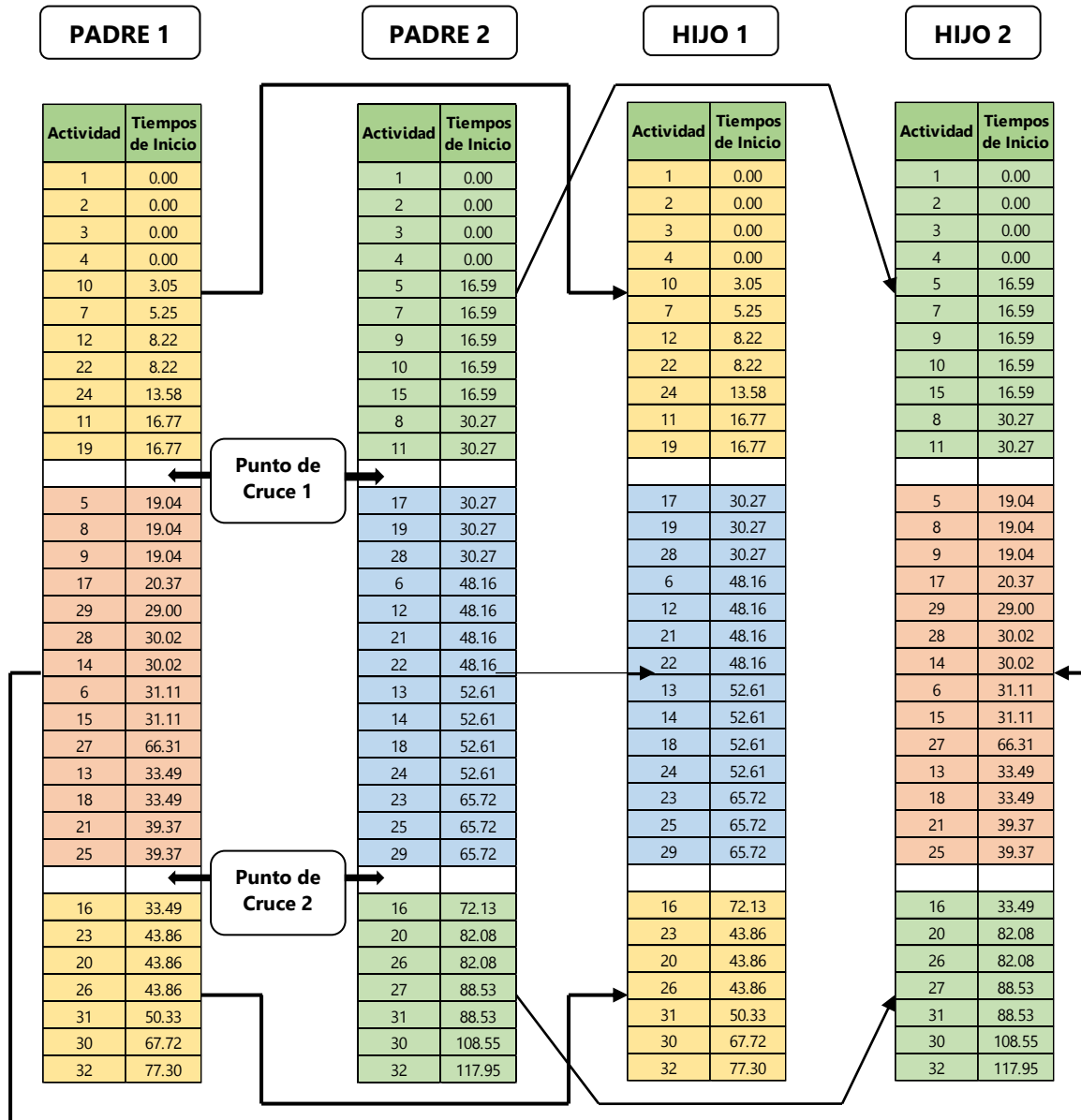
numero  $p$  de individuos a los cuales se les asigna un valor entre  $0.4 < p < 1$ , si es mayor que el parámetro  $p$ , fijado para el proceso evolutivo, se escoge el individuo más apto, y si es menor, se escoge al menos apto.

### **6.5. Cruce**

De los individuos ganadores del proceso de selección anterior, se realiza una operación de cruce de dos puntos, donde se intercambia el orden de las actividades en dos partes del cromosoma, en la figura 20 se aprecian dos cromosomas elegidos por el torneo probabilístico intercambiando el orden de sus listas de actividades para dar lugar a dos nuevos cromosomas, los cuales deben cumplir las restricciones de precedencia de la instancia del proyecto y además que no se repitan los genes (actividades) dentro del individuo, de lo contrario el cromosoma no se tendrá en cuenta para las futuras generaciones del algoritmo.

Figura 20

Representación del cruce



### 6.6. Mutación

Con el objetivo de diversificar la población, cada cromosoma sufre un proceso de mutación.

Las actividades que conforman dicho cromosoma tienen una probabilidad de cambiar de posición

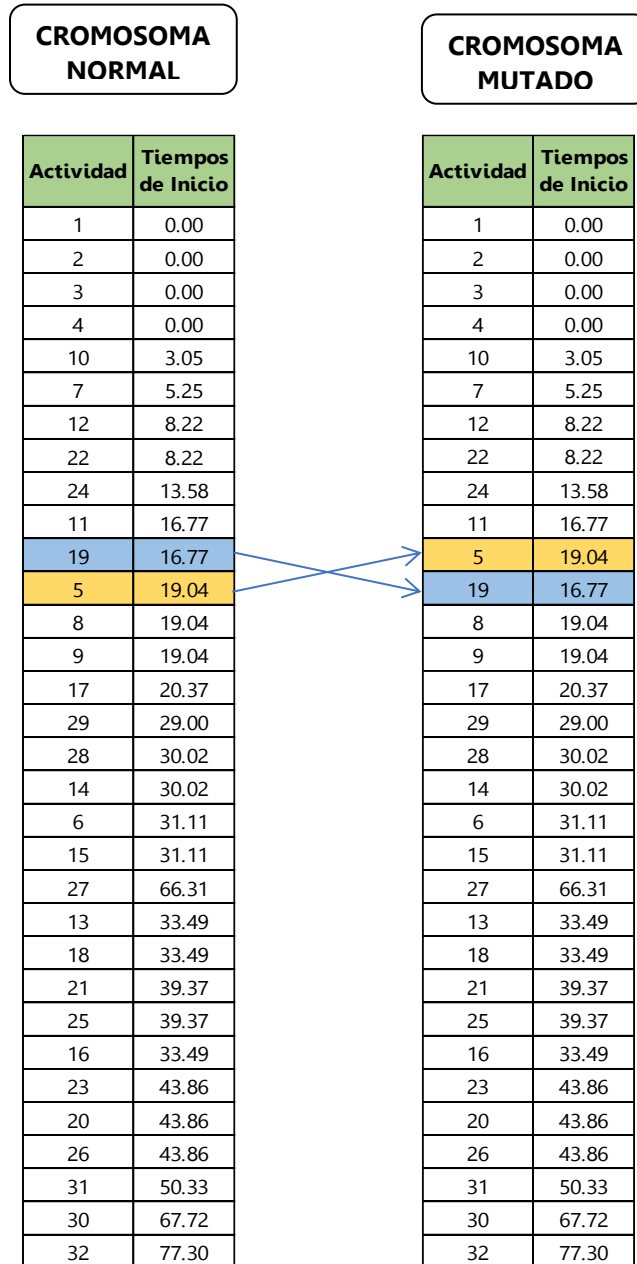
del 10%, si se cumple la probabilidad el individuo muta, ninguno de sus genes se debe repetir y además, debe cumplir con las restricciones de precedencia de la instancia del proyecto, si no cumple con dichas restricciones el organismo es invalido y no es tenido en cuenta para futuras generaciones. La figura 21 representa la mutación para una lista de actividades de instancia j30.

Los cromosomas validos se someten nuevamente a un SGS en paralelo por medio el cual se obtienen los makespan de las listas de actividades de la segunda generación. Seguidamente se escoge el individuo con mejor makespan, se compara con el guardado en memoria de la primera generación, y de ser mejor se reemplaza.

Finalmente se crea la tercera generación del algoritmo genético, a partir de la segunda, por medio de las operaciones de cruce y mutación, y se repite el procedimiento hasta que se completen el número de generaciones ingresadas como parámetro de entrada. Se obtiene el mejor makespan con su correspondiente línea base.

**Figura 21**

*Representación de la Mutación*



## 7. Validación Del Algoritmo

Para hacer el análisis del comportamiento del algoritmo genético en términos de la duración final de proyecto (makespan) en el RCPSP no determinista, se realizó un diseño de experimentos  $2^k$  en el software estadístico minitab 2019. Se usaron 2 instancias de proyectos tomada de la librería PSPLIB, una de 30 actividades con 4 tipos de recursos y otra de 60 actividades con 4 tipos de recurso igualmente. Los resultados obtenidos se encuentran en el apéndice B y C.

El diseño experimental es  $2^2$ , con 5 réplicas; los factores son A(población) y B(generaciones). El nivel bajo de A es de 20 individuos y el nivel alto es de 100 individuos, para B el nivel bajo es de 10 generaciones y el alto es de 50 generaciones. En este capítulo se analizará qué factor (A, B o AB) incide mayormente en la generación del makespan. Para esto se tiene en cuenta cinco réplicas como lo muestra la tabla 11 para las instancias J30 y la tabla 12 para instancias j60.

**Tabla 11**

*Experimentos para el análisis factorial  $2^k$  para instancias j30*

<b>A:Población</b>	<b>B:Generación</b>	<b>MS 1</b>	<b>MS 2</b>	<b>MS 3</b>	<b>MS 4</b>	<b>MS 5</b>
<b>20</b>	<b>10</b>	87,13	86,09	80,99	85,33	79,58
<b>100</b>	<b>50</b>	86,92	88,65	89,01	84,12	88,41
<b>20</b>	<b>50</b>	75,18	76,54	80,58	75,23	78,53
<b>100</b>	<b>10</b>	89,21	92,86	88,82	87,08	100

**Tabla 12***Experimentos para el análisis factorial 2k para instancias j60*

A:Población	B:Generación	MS 1	MS 2	MS 3	MS 4	MS 5
100	50	135,9	137,16	138,47	135,59	137,57
20	50	124,25	124,33	126,28	127,08	127,48
100	10	134,51	138,71	137,54	137,68	137,33
20	10	130,93	124,3	125,21	127,3	128,73

**7.1. Análisis de varianza para instancias j30****Tabla 13***Análisis de Varianza Instancias j30*

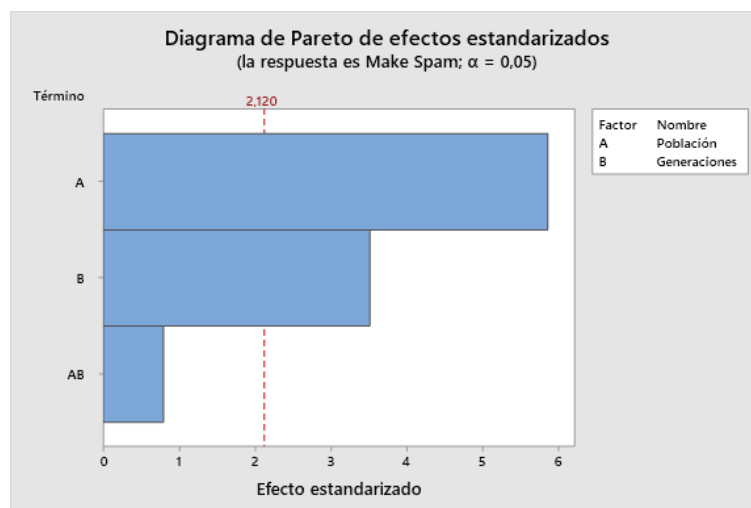
Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Modelo	3	556,911	185,637	15,79	0,000
Lineal	2	549,469	274,734	23,36	0,000
Número tabú	1	404,101	404,101	34,37	0,000
Número de iteraciones	1	145,368	145,368	12,36	0,003
Interacciones de 2 términos	1	7,442	7,442	0,63	0,438
Población*Generaciones	1	7,442	7,442	0,63	0,438
Error	16	188,144	11,759		
Total	19	745,055			

Los resultados obtenidos en la tabla 13 para el análisis de varianza permiten evidenciar que el factor población es significativo en la obtención del makespan, porque su valor  $p=0,00$  es menor al valor de significancia utilizado que corresponde a  $0,05$  puesto que el nivel de confianza del experimento fue de  $95\%$  y además el valor  $F=34,37 > F_{\text{fisher}}(1,12)=4,94$  confirma el efecto de la población en la variable respuesta; por otra parte el factor generación también muestra ser significativo ya que el valor  $p=0,003$  es menor al valor  $p=0,5$  de referencia, también al hacer la

comparación del valor  $F=12,36 > F_{\text{fisher}}(1,12)=4,94$  se confirma que el factor generación tiene efecto sobre la duración total del proyecto. La interacción de la población y las generaciones tiene un valor  $p=0,438$  mayor a  $0,05$  por lo que se afirma que no hay significancia. Por último, al comparar el valor  $f=0,63 < F_{\text{fisher}}(1,12) = 4,494$  se comprueba que la interacción de los 2 factores no tiene efecto sobre el makespan.

## Figura 22

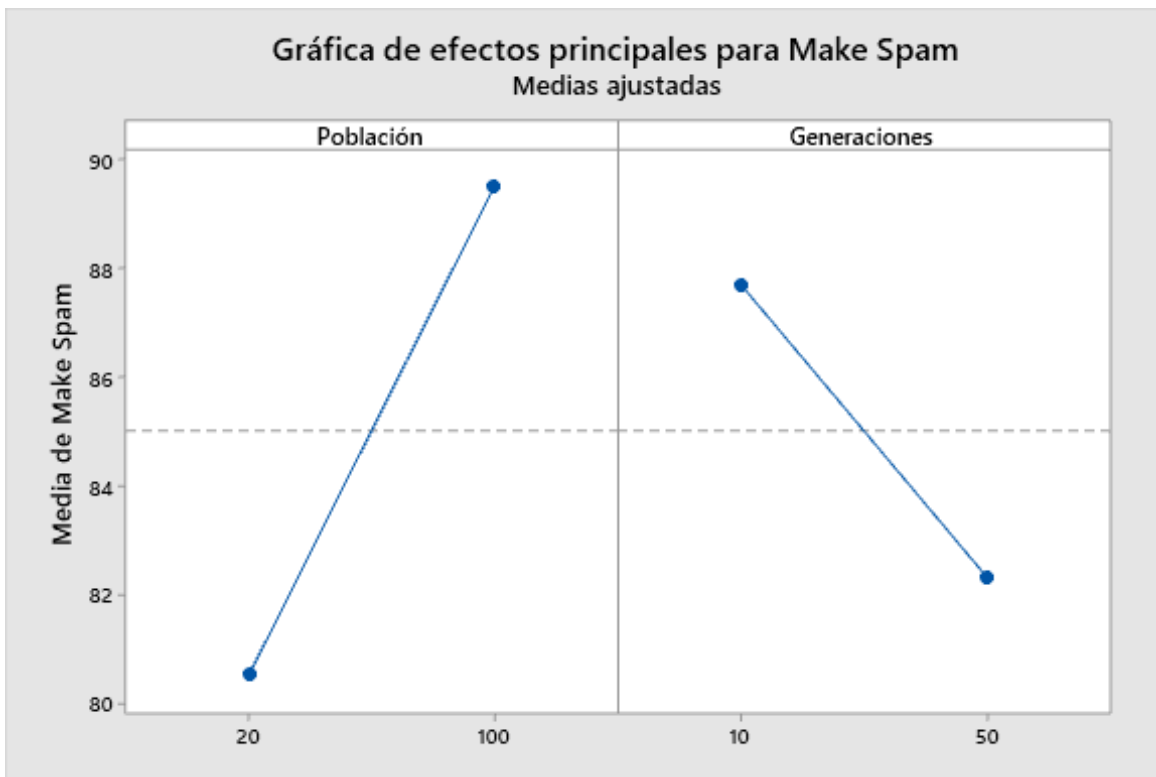
*Gráfico de Pareto instancias j30*



En el diagrama de Pareto en la figura 22 se observa que los factores A: población y B: Generaciones tienen incidencia en el makespan, siendo más fuerte el factor población y seguido el factor generaciones; también se evidencia que la interacción población y generaciones no tiene efecto en el makespan.

**Figura 23**

*Gráfica de efectos principales instancias j30*



Según la figura 23: gráfica de efectos principales para makespan, se encuentra que el efecto de factor población sobre el makespan presenta un efecto positivo en su nivel bajo=20 ya que el experimento arroja una duración final promedio de 80,518 y para el nivel alto=100 se muestra mayor duración del makespan promedio de 89,508 por lo cual se considera negativo para la obtención del makespan. Por otra parte, para el factor generaciones se encontró que para el factor bajo=10 arroja un makespan promedio de 87,809, dado esto se considera negativo en comparación con el nivel alto=50 ya que arroja uno menor al promedio de 82,317.

## 7.2. Análisis de varianza para instancias j60

**Tabla 14**

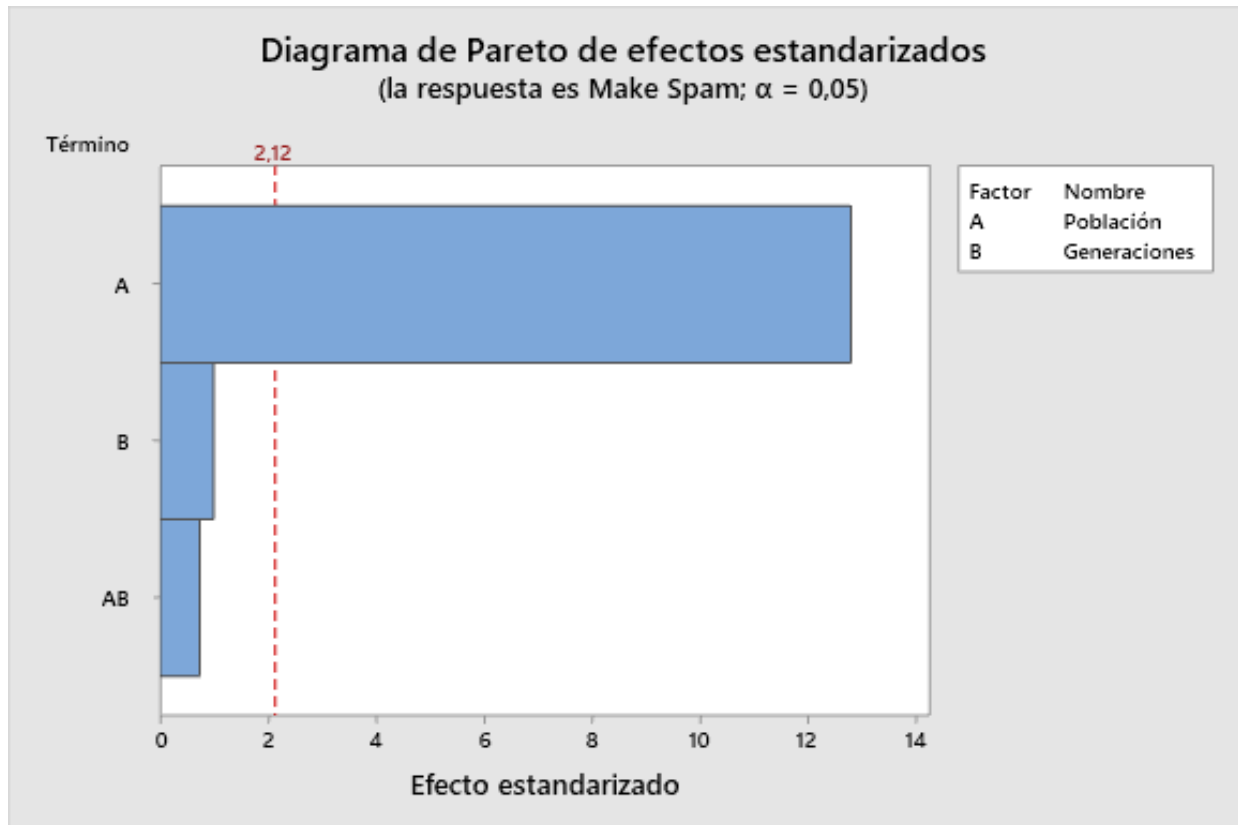
*Análisis de Varianza Instancias j60*

<b>Fuente</b>	<b>GL</b>	<b>SC Ajust.</b>	<b>MC Ajust.</b>	<b>Valor F</b>	<b>Valor p</b>
Modelo	3	551,831	183,944	55,15	0,000
Lineal	2	550,049	275,025	82,45	0,000
Número tabú	1	546,744	546,744	163,92	0,000
Número de iteraciones	1	3,305	3,305	0,99	0,334
Interacciones de 2 términos	1	1,782	1,782	0,53	0,475
Población*Generaciones	1	1,782	1,782	0,53	0,475
Error	16	53,368	3,335		
Total	19	605,199			

En la tabla 14 se muestra los resultados obtenidos para el análisis de varianza, permiten evidenciar que el factor población es significativo en la obtención del makespan porque su valor  $p=0,00$  es menor al valor de significancia utilizado que corresponde a  $0,05$  puesto que el nivel de confianza del experimento fue de  $95\%$  y además el valor  $F=163,92 > F_{fisher}(1,12)=4,94$  confirma el efecto de la población en la variable respuesta; por otra parte el factor generación se encontró no significativo para este tipo de instancia ya que el valor  $p=0,334$  es mayor al valor  $p=0,5$  de referencia, también al hacer la comparación del valor  $F=0,99 < F_{fisher}(1,12)=4,94$  se confirma que el factor generación no tiene efecto sobre el makespan. La interacción de la población y las generaciones tiene un valor  $p=0,475$  mayor a  $0,05$  por lo que se afirma que no hay significancia y al comparar el valor  $f=0,53 < F_{fisher}(1,12)=4,494$  se comprueba que la interacción de los 2 factores no tiene efecto sobre el makespan.

**Figura 24**

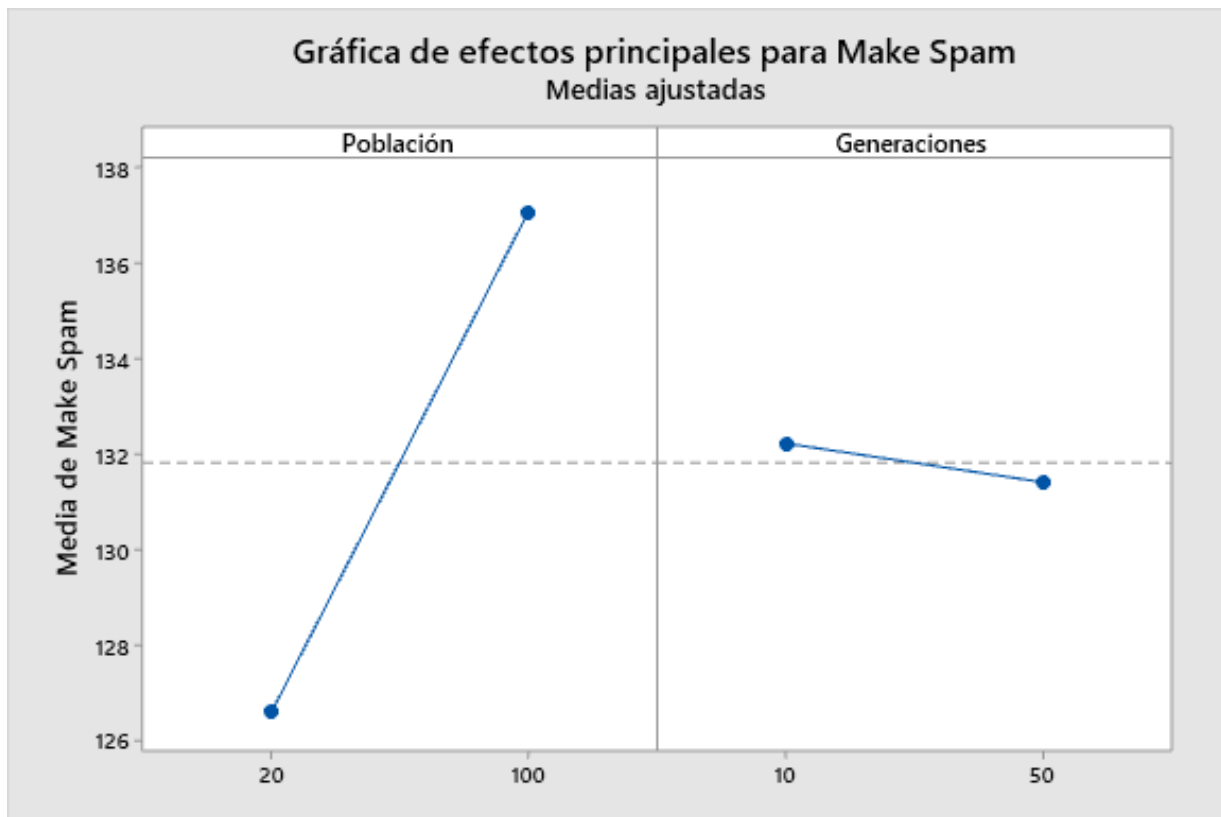
*Gráfico de Pareto Instancias j60*



En el diagrama de Pareto de la figura 24, se observa que el factor A: población tiene incidencia en el makespan y el factor B: Generaciones, y la interacción AB población y número de generaciones no tienen incidencia en el makespan.

**Figura 25**

*Gráfica de efectos principales instancias j60*



Según la figura 25: gráfica de efectos principales para makespan, se encuentra que el efecto de factor población sobre el makespan presenta un efecto positivo en su nivel bajo=20 ya que el experimento arroja un makespan promedio de 126,589 y para el nivel alto=100 se muestra mayor duración del makespan promedio de 137,046, por lo cual, se considera negativo. Por otra parte, para el factor generaciones que no se encontró significativo para la variable respuesta, se encontró que para el factor bajo=10 arroja un makespan promedio de 132,224, para el cual no se encuentra mayor diferencia con el nivel alto=50 que arroja un makespan promedio de 131,411.

De acuerdo con el análisis del diseño de experimentos para las instancias j30 y j60 se concluye que el factor población es altamente significativo para ambas puesto que presenta mayor incidencia en el makespan con un nivel bajo=20 individuos. Respecto al factor generación se encontró significativo en instancias j30 y no significativo en instancias j60, pero aun así mostró arrojar un menor makespan en su nivel alto=50 generaciones. Teniendo en cuenta estos resultados se realizó la obtención de resultados con parámetros de entrada del algoritmo Población=20 individuos y Generaciones=50.

## 8. Resultados

Para evaluar la efectividad del algoritmo genético propuesto en el capítulo 7 se seleccionaron aleatoriamente de la librería PSPLIB 10 instancias de prueba j30 con 30 actividades y 10 instancias j60 con 60 actividades, cada instancia es ingresada en el algoritmo y se obtuvo como solución al problema una línea base para cada una. Posterior a esto para efectos de un análisis comparativo del resultado arrojado, se abordó un método de solución exacto cuyas características (Ortiz & Diaz, 2019).

Dicho método de solución está basado en un modelo de programación lineal entera y se encuentra codificado en lenguaje de programación GAMS, a partir de este, se obtiene como solución al problema una línea base igual que en algoritmo propuesto y los resultados hallados se encuentran en el apéndice E.

Para llevar a cabo la comparación de los resultados entre líneas base generadas por cada método de solución, se realizó un proceso de simulación de Montecarlo de 10000 simulaciones a

través de un sistema generador de secuencias en paralelo, el cual usa como regla de prioridad, a la hora de programar una actividad, el orden de actividades propuesto por la línea base y tiene en cuenta la probabilidad de ocurrencia de riesgos que puedan interrumpir la ejecución. Los resultados obtenidos se pueden hallar en los apéndices G y H y a partir de estos se hallaron los indicadores de valor esperado del makespan y las medidas de robustez.

### **8.1. Aplicación de Riesgo**

Dado que este proceso de simulación se realiza con el fin de acercarse a la realidad lo más posible en cualquier tipo de proyecto, se estableció aleatoriamente 10 actividades de instancias j30 que estarán sujetas a una probabilidad de ocurrencia de 1, 2 tipos de riesgos o ambos y para instancias j60 un total de 20 actividades las cuales también pueden ser interrumpidas por 2 tipos de riesgos. Cada vez que se materialice el riesgo asociado a cada actividad definido en la tabla 15 para instancias j30 y la tabla 16 para instancias j60, la duración de las actividades se verá afectada incrementalmente. Para efectos de tener mayor aleatoriedad en la duración de las actividades en el SGS paralelo programado, se halla la duración de la actividad por medio de la distribución de probabilidad beta explicada anteriormente en el algoritmo genético y además de esto, sumarle la duración del riesgo materializado. Para la duración del riesgo 1 se usó como criterio un 50% de la duración estimada para cada tipo de instancia y para el riesgo 2 un 60% de la duración estimada de la actividad igualmente.

Por lo tanto, la duración de la actividad en cada escenario simulado para cada instancia estará determinada por la ecuación 9.1 que muestra la sumatoria de la duración estimada por la distribución beta más la duración del riesgo 1 y la duración del riesgo 2 en caso de cumplirse la probabilidad de ocurrencia.

$$DAesc = Db + (P1 * Dr1) + (P2 * Dr2) \quad (12)$$

**Tabla 15***Riesgos Instancias j30*

Actividad	Probabilidad	Riesgo	Probabilidad	Riesgo
	1	1	2	2
2	0,5	1	0	0
3	0,5	1	0	0
6	0,5	1	0	0
11	0,5	1	0	0
12	0	0	0,7	1
14	0	0	0,7	1
20	0	0	0,7	1
25	0,5	1	0,7	1
27	0,5	1	0	0
29	0	0	0,7	1

**Tabla 16***Riesgos Instancias j60*

Actividad	Probabilidad	Riesgo	Probabilidad	Riesgo
	1	1	2	2
2	0,5	1	0	0
3	0,5	1	0	0
6	0,5	1	0	0
11	0,5	1	0	0
12	0	0	0,7	1
14	0	0	0,7	1
20	0	0	0,7	1

*Continuación tabla 16 (Riesgos instancias j60)*

Actividad	Probabilidad	Riesgo	Probabilidad	Riesgo
	1	1	2	2
25	0,5	1	0,7	1
27	0,5	1	0	0
29	0	0	0,7	1
30	0,5	1	0	0
33	0,5	1	0	0
38	0	0	0,7	1
41	0,5	1	0	0
43	0,5	1	0,7	1
45	0,5	1	0	0
50	0	0	0,7	1
51	0	0	0,7	1
56	0	0	0,7	1
61	0,5	1	0	0

### 8.1.1. Valor Esperado

Resulta del promedio de los makespan generados de los 10000 escenarios simulados en el SGS en paralelo, y este indica la menor duración del proyecto si se llevase a cabo en la vida real de acuerdo con el orden de prioridad establecido por la línea base.

$$E(x) = \frac{\sum_1^{esc} makespan}{esc} \quad (13)$$

### 8.1.2. Medidas de Robustez

Con el fin de evaluar el desempeño las soluciones generadas y posteriormente compararlas con las instancias j30 y j60, se toma en cuenta dos medidas de robustez: Robustez de la calidad y

Robustez de la solución, estos indicadores permiten analizar qué tan apta es la línea base para soportar demoras que puedan comprometer los plazos fijados para la finalización del proyecto al igual que sus recursos.

**8.1.2.1. Robustez de la calidad.** Indica que tan alejado está el makespan de la línea base generada  $S_n^{LB}$  por el algoritmo genético con respecto a los makespan de las simulaciones  $S_n^{esc}$ , es decir, es el resultado de la diferencia entre la duración total del proyecto y a la duración total esperada en cada escenario.

$$QR = \frac{\sum_1^M |S_n^{LB} - S_n^{esc}|}{m} \quad (14)$$

**8.1.2.2. Robustez de la solución.** Indica que tan robusta es la línea base propuesta por el algoritmo genético y el método exacto en cuanto soportar interrupciones en los tiempos de inicio de cada actividad (Ortiz Pimiento, 2019). Para hallar el indicador es necesario obtener la desviación media de cada tiempo de inicio  $\Delta S_i$  de las actividades de la línea base  $S_i^{LB}$  con los tiempos de inicio de cada simulación  $S_i^m$ , posteriormente se calcula el SR calculando la sumatoria de las desviaciones media de las n actividades de la instancia de proyecto.

$$\Delta S_i = \frac{\sum_1^n |S_i^{LB} - S_i^m|}{m} \quad (15)$$

$$SR = \sum_{i=1}^n \Delta S_i \quad (16)$$

## 8.2. Análisis Según Indicadores

En el siguiente análisis se toma como indicador más relevante al valor esperado, puesto que representa la menor duración total del proyecto, por lo tanto, la línea base de menor valor esperado será la mejor. Seguidamente se considera la robustez de la calidad representada en la variación de las duraciones totales del proyecto y la esperada, y por último la robustez de la solución que muestra la sumatoria de la varianza entre los tiempos de inicio del proyecto y esperados de las actividades. Para ambos indicadores se tiene en cuenta el menor valor arrojado para línea base tanto la del algoritmo genético (AG) como la del método exacto (ME). Para efectos comparativos se escogió el color verde para las menores medidas y el color rojo para las mayores.

**Tabla 17**

*Valor Esperado Instancias j30*

<b>Instancia</b>	<b>AG E(x)</b>	<b>ME E(x)</b>
j301_2	80,69	70,86
j301_10	91,95	79,58
j306_3	84,61	70,56
j3014_3	92,01	80,11
j3019_5	99,97	82,97
j3020_7	74,94	63,86
j3023_9	112,30	125,42
j3037_8	120,36	100,45
j3044_9	124,12	116,12
j3047_6	102,05	87,32

Según los resultados de la tabla 17 del valor esperado para instancias j30, se puede apreciar que el 90% de los resultados del método exacto (ME) arrojaron un mejor makespan esperado que

las soluciones encontradas en el algoritmo genético (AG), solo en un 10% de las soluciones del algoritmo genético se logra tener un mejor valor esperado.

**Tabla 18**

*Valor Esperado Instancias j60*

<b>Instancia</b>	<b>AG E(x)</b>	<b>ME E(x)</b>
j601_2	122,97	120,19
j601_6	123,61	107,20
j603_5	140,87	124,14
j604_6	142,08	125,36
j607_7	148,88	136,03
j6011_8	115,54	102,20
j6012_10	137,98	121,40
j6014_2	126,26	106,73
j6017_6	138,27	114,90
j6022_10	122,07	107,56

Para las instancias j60 según la tabla 18 se obtuvo como resultado que el 100% de las soluciones del método exacto son mejores a las del algoritmo genético, es decir que sus makespan esperados son menores en todos los casos. Por lo tanto, según el orden de la línea base arrojada por el método exacto, programará menores soluciones en cuanto a la duración del proyecto.

**Tabla 19**

*Resultados Robustez de la Calidad Instancias j30*

<b>Instancia</b>	<b>AG QR</b>	<b>ME QR</b>
j301_2	37,60	47,08

*Continuación tabla 19 (Resultados Robustez de la Calidad Instancias j30)*

<b>Instancia</b>	<b>AG QR</b>	<b>ME QR</b>
j301_10	26,89	39,17
j306_3	45,26	59,07
j3014_3	67,16	79,03
j3019_5	35,79	52,46
j3020_7	23,92	34,39
j3023_9	53,48	40,33
j3037_8	83,40	103,30
j3044_9	19,59	26,82
j3047_6	68,53	83,22

A partir de la tabla 19 se infiere que la robustez de la calidad de las instancias j30 en el AG es mejor en un 90% que la del método exacto, por lo tanto, las líneas base generadas por el AG presentan una mejor robustez.

## **Tabla 20**

*Resultados Robustez de la Calidad Instancia j60*

<b>Instancia</b>	<b>AG QR</b>	<b>ME QR</b>
j601_2	112,97	115,74
j601_6	44,59	60,98
j603_5	66,12	82,83
j604_6	29,28	45,44
j607_7	64,37	77,20
j6011_8	69,29	82,54
j6012_10	49,61	65,97
j6014_2	79,55	99,03
j6017_6	62,76	86,02

*Continuación tabla 20 (Resultados Robustez de la Calidad Instancias j60)*

<b>Instancia</b>	<b>AG QR</b>	<b>ME QR</b>
j6022_10	75,06	89,56

Según la tabla 20 para la instancia j60 se observa que la robustez de la calidad en las líneas base arrojadas por el AG son mejores en un 100% con respecto a las del ME.

**Tabla 21***Resultados Robustez de la Solución Instancias j30*

<b>Instancia</b>	<b>AG SR</b>	<b>ME SR</b>
j301_2	596,01	143,89
j301_10	482,45	158,00
j306_3	925,13	458,20
j3014_3	1172,89	185,81
j3019_5	746,38	160,16
j3020_7	539,28	174,01
j3023_9	724,82	548,85
j3037_8	1652,38	209,70
j3044_9	303,25	494,80
j3047_6	115,12	307,29

Según la tabla 21 de los indicadores de robustez de la solución para instancias j30 se encontró que 80% de las soluciones del ME presentan una mejor robustez que el algoritmo genético porque este solo presentó un menor valor de robustez de la solución en 2 de las 10 líneas base simuladas.

**Tabla 22***Resultados Robustez de la Solución Instancias j60*

<b>Instancia</b>	<b>AG SR</b>	<b>ME SR</b>
j601_2	3.694,02	695,11
j601_6	1.846,60	739,01
j603_5	2.062,77	444,84
j604_6	1.777,83	1.653,29
j607_7	2.188,85	636,53
j6011_8	2.027,52	603,88
j6012_10	1.681,53	605,53
j6014_2	2.637,93	704,25
j6017_6	2.848,09	772,53
j6022_10	2.174,85	402,34

De la robustez de la solución para las instancias j60 según la tabla 22 se encontró que el algoritmo tuvo peores resultados ya que el 100% de estos son mayores en comparación con los resultados del ME.

Para finalizar es importante resaltar que, de los dos métodos de solución evaluados, el método exacto presenta mejores soluciones en para el valor esperado y la robustez de la solución por lo menos en 80% en cada uno de los 2 tipos de instancias analizadas; por otra parte, se evidenció que el algoritmo genético arrojó la menor robustez de calidad como mínimo en 90% de los casos simulados.

### 8.3. Análisis Según Tiempo de Ejecución

Por otra parte, se analizó el esfuerzo computacional que representa hallar solución al problema por cada uno de los métodos propuestos, teniendo en cuenta el tiempo de procesamiento en el computador.

**Tabla 23**

*Tiempo de procesamiento computacional para instancias j30 y j60*

<b>Instancia</b>	<b>TC AG</b>	<b>TC ME</b>
<b>j301_2</b>	36,73	0,063
<b>j301_10</b>	37,3	0,016
<b>j306_3</b>	36,08	0,031
<b>j3014_3</b>	36,49	0,031
<b>j3019_5</b>	38,57	0,015
<b>j3020_7</b>	36,31	0,016
<b>j3023_9</b>	37,83	0,032
<b>j3037_8</b>	40,68	0,032
<b>j3044_9</b>	39,9	0,031
<b>j3047_6</b>	40,75	0,032
<b>j601_2</b>	95,74	0,109
<b>j601_6</b>	92,41	0,062
<b>j603_5</b>	95,22	0,062
<b>j604_6</b>	90,75	0,093
<b>j607_7</b>	96,24	0,125
<b>j6011_8</b>	94,57	0,093
<b>j6012_10</b>	97,38	0,093
<b>j6014_2</b>	96,18	0,109
<b>j6017_6</b>	102,38	0,078
<b>j6022_10</b>	111,07	0,078

A partir de la tabla 23 se observa que el tiempo de ejecución de las instancias j30 y j60 del algoritmo genético, es significativamente mayor al tiempo de ejecución obtenido por el método exacto, sin embargo, es de resaltar que el método exacto no puede ser procesado en GAMS cuando el número de actividades de una instancia es mayor a 90 debido a que el esfuerzo computacional es muy alto, sin embargo, una instancia j90 si podría ser procesada en un tiempo razonable con el método metaheurístico usado, por tanto, se puede inferir que el algoritmo genético es la mejor opción cuando un gran número de actividades requieren ser programadas.

## 9. Conclusiones

De acuerdo con la revisión de literatura se evidenció que el problema de programación de proyectos no determinístico con recursos restringidos es un tema de gran importancia dentro de la industria a la hora de desarrollar nuevos productos, el investigador Deumelemester E. es quien más ha realizado aportes al tema, y es en el territorio asiático donde hay más actividad investigativa al respecto. Se registran 148 artículos desde el año 1996 hasta el 2021 los cuales en mayoría pertenecen al área de la ingeniería y la ciencia de la computación.

Se realizó la codificación de un proceso de Simulación de Montecarlo incluida en el algoritmo genético con el fin ingresar aleatoriedad al algoritmo en las duraciones de las actividades de proyecto gracias a una distribución de probabilidad beta, se llevaron a cabo 1.000 simulaciones de las duraciones y posteriormente se promediaron, a partir de esto, crea el primer individuo del algoritmo correspondiente a una Línea Base de actividades del proyecto.

Respecto al algoritmo genético desarrollado, gracias a un análisis de un diseño de experimentos 2k con 5 réplicas se realizó una validación del comportamiento de los parámetros de entrada que permiten arrojar un menor makespan de proyecto y se encontró que el factor población es el más significativo para solución del algoritmo con un nivel bajo de 20 individuos. En cuanto al factor generación presentó ser significativo en instancias de proyecto de 30 actividades y no significativo en instancias j60 y además presentó mejor solución para el algoritmo con un nivel alto de 50 generaciones. Por lo tanto, se concluyó que el algoritmo genético presenta un mejor comportamiento en cuanto a la variable respuesta makespan con un nivel bajo de población igual a 20 individuos y un nivel alto del factor generación igual a 50.

De las 10.000 simulaciones usadas para comparar las líneas bases del algoritmo genético y del método exacto, se concluyó que para las instancias j30 observadas, el método exacto propuesto por N Ortiz obtiene mejores resultados en cuanto al makespan esperado en un 90% de los casos, por el contrario, si analizamos la robustez de la calidad de las líneas base, se observó mejores resultados en el algoritmo genético propuesto en un 90% de las instancias, por último, para la robustez de la solución el método exacto arrojó soluciones más robustas en un 80% respecto al algoritmo propuesto.

Para el caso de las instancias de las j60 se evidencio un comportamiento similar en todos los criterios de evaluación, el makespan esperado del método exacto es superior en la totalidad de los casos en comparación al algoritmo genético. La solución de la calidad del algoritmo genético es más robusta en todos los casos respecto al método exacto y finalmente, en la robustez de la solución, el método exacto obtuvo en todas las instancias simuladas mejores soluciones que el algoritmo genético.

Teniendo en cuenta que el algoritmo genético propuesto presentó el mejor valor de indicador de robustez de la calidad en 19 de las 20 instancias probadas, se encontró que dicho método de solución presenta soluciones más robustas ya que las duraciones del proyecto son más cercanas al valor esperado del makespan.

## **10. Recomendaciones**

Teniendo en cuenta la gran aplicabilidad del problema de programación de proyectos con recursos restringidos en su forma determinística y no determinística, se recomienda para futuras investigaciones donde se abordé métodos de solución metaheurística como un algoritmo genético, búsqueda tabú o algoritmos híbridos se puedan ingresar la ocurrencia de riesgos para que así este tipo de soluciones proactivas se acerquen más a la realidad en cuanto a la ejecución. Por otra parte, para aumentar la variabilidad de las soluciones se recomienda usar otro tipo de reglas de decisión como el rango de mayor peso posicional (GRPW) en el sistema generador de secuencias en serie, dichas reglas pueden usarse en conjunto con una probabilidad de ocurrencia para cada regla de decisión.

En cuanto a la utilización de algoritmos genéticos es importante tener en cuenta que este tipo de metaheurística aporta gran variedad de métodos de para generar las soluciones, el mejor método a usar depende de la estrategia de búsqueda que se desea del algoritmo por lo tanto se recomienda tener en cuenta otros tipos de métodos de selección de padres, ya que con un método de alta presión de selección se centran la búsqueda en la mejores soluciones, por el contrario con

un presión de selección menor amplía el espacio de búsqueda lo cual da mayor variedad de soluciones. Respecto al operador de mutación se sugiere optar por mayor variabilidad en la probabilidad de mutación para los individuos de una población.

**Referencias Bibliográficas**

- Ahmadpour, S., & Ghezavati, V. (2019). Modeling and solving multi-skilled resource-constrained project scheduling problem with calendars in fuzzy condition. *Journal of Industrial Engineering International*, 19.
- Alipouri, Y., Ardeshir, A., Chan, W., & Hassam, M. (2019). Solving the FS-RCPSP with hyper-heuristics: A policy-driven approach. *Journal of the Operational Research Society*, 18.
- Alipouri, Y., Hassan, M., Ardeshir, A., & Fazel, M. (2020). A mixed-integer linear programming model for solving fuzzy stochastic resource constrained project scheduling problem. *Operational Research*, 21.
- Azofeifa, C. E. (2008). *Aplicación de la Simulación de Monte Carlo en el Cálculo del Riesgo Usando Excel, Tecnología en Marcha*. Costa Rica.
- Ballestin, F. (2007). When it is worthwhile to work with the stochastic RCPSP? *Springer Science+Business Media*, 158.
- Balouka, N., & Cohen, I. (2019). A robust optimization approach for the multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 14.
- Brčić, M., Katić, M., & Hlupić, N. (2019). Planning horizons based proactive rescheduling for stochastic resource-constrained project scheduling problems. *European Journal of Operational Research*, 9.
- Buitrago, W., & Romero, E. (2019). *Modelo de solución al problema de programación de proyectos de desarrollo de nuevos productos con recursos restringidos, inserción de tareas y duración aleatoria*. Universidad Industrial de Santander .

- Chakrabortty, R., & Ryan, M. (2019). An uncertainty tolerant approach for stochastic resource constrained project scheduling problems. *Technology and Engineering Management Conference*, 6.
- Chakrabortty, R., Abbasi, A., & Ryan, M. (2019). A Risk Assessment Framework for Scheduling Projects With Resource and Duration Uncertainties. *IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT*, 15.
- Chakrabortty, R., Rahman, H., & Ryan, M. (2020). Efficient priority rules for project scheduling under dynamic environments: A heuristic approach. *Computers and Industrial Engineering*, 18.
- Chase, R., Jacobs, R., & Aquilano, N. (2009). *Administración de Operaciones*. Mc Graw Hill.
- Clarivare. (2021). Base de datos. *Web of Science* [Figura].
- Creemers, S. (2019). The preemptive stochastic resource-constrained project scheduling problem. *European Journal of Operational Research*, 10.
- Davari, M., & Demeulemeester, E. (2019a). The Proactive and Reactive Resource-Constrained Project Scheduling Problem. *Journal of Scheduling*, 27.
- Davari, M., & Demeulemeester, E. (2019b). Important classes of reaction for the proactive and reactive resource-constrained project scheduling problem. *Annals of Operations Research*, 24.
- Elsevier, (2021), Base de datos *Scopus* [Figura].
- Faria, J., Araujo, M., Demeulemeester, E., & Tereso, A. (2020). Project management under uncertainty: using flexible. *European J. Industrial Engineering*, 33.
- Federico, L. M. (2007). *Tesis de Grado en Ingeniería Industrial, Simulación y Análisis Económico de la Logística de Distribución de una Empresa*. Instituto Tecnológico de Buenos Aires.

- Franco, A. (2013). *Los métodos de Montecarlo*. Procedimientos numéricos. [http://www.sc.edu.es/sbweb/fisica\\_/numerico/montecarlo/montecarlo.html#:~:text=En%20la%20pr%C3%A1ctica%2C%20las%20pruebas,variable%20aleatoria%20discreta%20o%20continua.](http://www.sc.edu.es/sbweb/fisica_/numerico/montecarlo/montecarlo.html#:~:text=En%20la%20pr%C3%A1ctica%2C%20las%20pruebas,variable%20aleatoria%20discreta%20o%20continua.)
- García, E. (2020). *Planeación y control de proyectos*. Escuela de Liderazgo Aprendizaje Experiencial. <http://equipodeliderazgomaseficaz.over-blog.es/2020/03/planeacion-y-control-de-proyectos.html>
- González, M., Alba, E., & Odieres, M. (2014). *Ingeniería de Proyectos*. Dextra.
- Haider, S., Ahnd, M., Anwer, S., Aslam, M., & Asadullah, M. (2019). Efecto de la coordinación relacional en las intenciones de rotación de empleados de satisfacción laboral: el uso de ecuaciones estructurales y la simulación de montecarlo. *ESIC Market. Economic and Business Journal*, 9.
- Hillier, F. S. (2010). *Introducción a la investigación de operaciones*. Mc Graw Hill.
- Hillier, F., y Lieberman, G. (2010). *Introducción a la Investigación de Operaciones*. Mc Graw Hill.
- Khalilzadeh, M., Hosseini, S. H., & Ghaeli, R. (2020). A genetic algorithm-based method for solving multi-mode resourceconstrained project scheduling problem in uncertain environment. *Journal of Project Management*, 8.
- Klein, R. (2000). Bidirectional Planing: improving priority rule-based heuristic for scheduling resource-constrained projects. *European Journal of Operational Research*.
- Kolisch, R. (1995). Serial and Parallel Resource-Constrained Project Scheduling Methods Revisited: Theory and Computation. *European Journal of Operational Research*, 3.
- Kolisch, R., & Hartmann, S. (1996). *Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis*. Springer.

- Kong, M. (2013). *Investigación de Operaciones: programación lineal, problemas de transporte. Análisis de redes*. Fondo editorial de la pontificia Universidad del Perú.
- Kucuksayacigil, F., & Ulusoy, G. (2020). A Hybrid Genetic Algorithm Application for a Bi-Objective, Multi-Project,. *Frontiers of Engineering Management* , 38.
- Kuhleemann, S., Sellmann, M., & Tierney, K. (2019). Exploiting Counterfactuals for Scalable Stochastic Optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 18.
- Lima, R., Tereso , A., & Faria, J. (2019). Project management under uncertainty: resource flexibility visualization in the schedule. *Procedia Computer Science*, 8.
- Ma, Z., Demeulemeester, E., He, Z., & Wang, N. (2019). A computational experiment to explore better robustness measures for project scheduling under two types of uncertain environments. *Computers and Industrial Engineering*, 9.
- Maroto, C. (2013). *Investigación Operativa en Administración y Dirección de Empresas*. Universidad Politecnica de Valencia.
- Maroto, C., Alcaraz, J., Ginestar, C., y Segura, M. (2013). *Investigación Operativa en Administración y Dirección de Empresas*. Editorial de la Universidad Politecnica de Valencia.
- Moradi, M., Hafezalkotob, A., & 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*, 10.
- Morillo Torres, D. (2017). *Eficiencia Energética en la Programación de Tareas con Recursos Restringidos*. Universidad Politecnica de Valencia.

- Niño, K., & Peña, J. (2019). A Based-Bee Algorithm Approach for the Multi-Mode Project Scheduling Problem. *Procedia Manufacturing*, 8.
- Ortiz, N. R. (2019). *Modelo de solución al problema de programación de proyectos de desarrollo de nuevos productos con recursos restringidos, inserción de tareas y duración aleatoria*. Universidad Nacional de Colombia.
- Ortiz, N. R., & Diaz, F. J. (2018). The Project Scheduling Problem with Non-deterministic Activities Duration: A literature review. *Journal of Industrial Engineering and Management*, 19.
- Páez, R., y Moreno, E. D. (2017). *Plataforma logística regional de distribución de contenedores para la eficiencia de los buques que pasan por el puerto Corozal* (Tesis de pregrado). <https://repository.ucatolica.edu.co/bitstream/10983/15237/1/PLATAFORMA%20LOG%20C3%8DSTICA%20REGIONAL%20DE%20DISTRIBUCI%C3%93N%20DE%20CONTENEDORES%20PARA%20LA%20EFICIENCIA%20DE%20LOS%20BUQUES%20QU.pdf>
- Pérez, L. M. (2015). *Aplicación del Método de Monte Carlo al Análisis de Falla de Placas Laminadas, Bajo Carga Puntual Constante en su Centro*. Universidad Tecnológica de Bolívar Cartagena.
- Román, P. (2017). *Cálculo de probabilidades I*.
- Spiegel, M., Schiller, J., y Srinivasan, A. (2012). *Probabilidad y Estadística*. McGraw-Hill Interamericana.
- Subulan, K. (2019). An interval programming based approach for fully uncertain resourceconstrained project scheduling problem considering project manager's. *Pamukkale University Journal of Engineering Sciences*, 17.

- Sun, W., Bocchini, P., & Davison, B. (2020). Model for Estimating the Impact of. *JOURNAL OF INFRASTRUCTURE SYSTEMS* , 20.
- Tackenberg, S., & Duckwitz, S. (2019). Actor-Oriented Optimization Model For Maintenance Tasks. *Proceedings of the 2018 Winter Simulation Conference*, 12.
- Taha, H. (2017). *Investigación de Operaciones*. Pearson.
- Taibo , A. (2002). *Investigación de Operaciones para los no Matematicos*. Instituto Politécnico Nacional.
- Torres, Z. H. (2014). *Administración de Proyectos*. Grupo editorial Patria.
- Van, N. J. & Waltman, L. (2021). *VOSviewer* (Versión 1.6.16). Windows. Leiden: Centro de Estudios de Ciencia y Tecnología (CWTS). [Figura].
- Velasco, J. (2015). *Gestión de Proyecto en la Empresa: planificación, programación y control*. Pirámide.
- Wang, J., Hu, X., Demeulemeester, E., & Zhao, Y. (2019). A bi-objective robust resource allocation model for the RCPSP considering resource transfer costs. *International Journal of Production Research*, 22.