

Optimización de la redistribución de recursos en proyectos con retrasos utilizando algoritmos genéticos

Ing. Comp. **José Manuel Pérez Esquivel**

Centro Universitario UAEM Valle de Chalco

ORCID: <https://orcid.org/0009-0009-2611-4763>

jpereze010@alumno.uaemex.mx

Dr. **Marco Alberto Mendoza Pérez**

Centro universitario UAEM Valle de Chalco

ORCID: <https://orcid.org/0000-0003-4911-4757>

mamendozap@uaemex.mx

Dra. **Cristina Juárez Landín**

Centro Universitario UAEM Valle de Chalco

ORCID: <https://orcid.org/0000-0002-0988-3060>

cjuarezl@uaemex.mx

RESUMEN

La gestión eficiente de recursos es crucial en la ejecución de proyectos, especialmente en la industria de la construcción, donde los retrasos son comunes y pueden impactar significativamente los costos y la calidad. Este estudio se enfoca en evaluar la eficacia de un algoritmo genético para la redistribución óptima de recursos en proyectos con retrasos en su inicio usando un diseño experimental, se manipularon variables independientes determinadas por los parámetros del algoritmo genético para observar sus efectos sobre la eficacia y eficiencia en la redistribución de recursos. La población de estudio consistió en actividades simuladas de planificación de recursos, representativas de entornos de la industria de la construcción con una selección muestra de 10 actividades con una duración de 44 días laborables. La distribución inicial se generó utilizando la distribución normal, con una media (μ) de 22 y una desviación estándar (σ) de 8.8, posteriormente para simular retrasos, los valores de los primeros 10 días se ajustaron reflejando un retraso acumulado del 10%. Para evaluar la robustez y eficacia del algoritmo genético, se generaron múltiples instancias de actividades con diferentes distribuciones iniciales, simulando diversos escenarios para la implementación del algoritmo, se utilizó el lenguaje Python, usando técnicas como la selección por torneo, el cruzamiento de un punto y la mutación uniforme. La función de aptitud, basada en el Error Cuadrático Medio (ECM), permitió medir la calidad de la redistribución de recursos en términos de minimización del retraso acumulado y eficiencia en el uso de recursos.

Palabras clave: Algoritmos Genéticos, Gestión de Recursos, Optimización de Proyectos, Retrasos en el proyecto, Redistribución de Recursos.

ABSTRACT

Efficient resource management is crucial in project execution, especially in the construction industry, where delays are common and significantly impact costs and quality. This study focuses on evaluating the effectiveness of a genetic algorithm for optimal resource redistribution in projects with delays using an experimental design. Independent variables were manipulated based on genetic algorithm parameters to observe their effects on resource redistribution effectiveness and efficiency. This study population consisted of simulated resource planning activities, representative of construction industry environments, with a selected sample of 10 activities each lasting 44 working days. Initial distribution was generated using the normal distribution, with a mean (μ) of 22 and standard deviation (σ) of 8.8. To simulate delays, values for the first 10 days were adjusted to reflect a cumulative delay of 10%. To evaluate the robustness and effectiveness of the genetic algorithm, multiple instances of activities with different initial distributions were generated, simulating various scenarios. Python was used for algorithm implementation, employing techniques such as tournament selection, single-point crossover, and uniform mutation. The fitness function, based on the Mean Squared Error (MSE), measured the quality of resource redistribution in terms of minimizing cumulative delay and resource efficiency.

Keywords: Genetic Algorithms, Resource Management, Project Optimization, Project Delays, Resource Redistribution.

INTRODUCCIÓN

La gestión eficiente de recursos en proyectos es una de las principales preocupaciones en diversas industrias, especialmente en la construcción, donde los retrasos y desvíos en la planificación son comunes y pueden tener un impacto significativo en el costo y la calidad del proyecto (Mattos, 2019). La planificación de recursos adecuada es crucial para asegurar que los proyectos se completen a tiempo y dentro del presupuesto, manteniendo la calidad esperada. Sin embargo, la realidad del seguimiento y ejecución de proyectos frecuentemente presenta desafíos imprevistos que pueden causar retrasos y afectar la distribución de recursos planeada inicialmente.

En este contexto, la redistribución de recursos se convierte en una tarea esencial para mitigar los efectos negativos de los retrasos a lo que lleva a adoptar un enfoque prometedor para abordar este problema con el uso de algoritmos genéticos que, a diferencia de los programas convencionales, los algoritmos genéticos no se limitan a un conjunto reducido de soluciones, sino que exploran un universo mucho más vasto de posibilidades al abordar un problema (Holland, 1992). Los algoritmos genéticos son especialmente

adecuados para problemas complejos y no lineales, donde las soluciones óptimas no son evidentes y requieren una exploración extensa del espacio de búsqueda como en El-Shorbagy (2020) donde explora la resolución del sistema de ecuaciones no lineales mediante la aplicación de un algoritmo genético.

El objetivo principal de este estudio es evaluar la eficacia de un algoritmo genético para la redistribución óptima, o cercana a la óptima, de recursos en proyectos que han experimentado retrasos en el inicio de su ejecución. Para lograr esto, se diseñó un experimento que permite manipular variables independientes determinadas por los parámetros del algoritmo genético y observar sus efectos sobre las variables dependientes, como la eficacia y eficiencia de la redistribución de recursos.

El estudio se llevó a cabo en un entorno simulado que refleja características comunes en la gestión de proyectos reales observados en la industria de la construcción. No obstante, el algoritmo genético está diseñado para ser aplicable a diversas industrias. Se seleccionó una muestra representativa de 10 actividades simuladas, cada una con una duración de 44 días laborables y una distribución inicial de recursos calculada mediante la distribución normal, aunque las distribuciones muestrales presentan una amplia variedad de formas, entre las que se encuentran la distribución binomial, la distribución de Poisson, la distribución normal es más frecuentemente (Galli, 2021) y en la gestión de proyectos tienden a seguir una distribución aproximadamente normal.

Para simular los retrasos, se ajustaron los valores de los primeros 10 días de cada actividad, reflejando un retraso acumulado, para generar múltiples instancias de actividades con diferentes distribuciones iniciales para evaluar la robustez y eficacia del algoritmo genético bajo diversos escenarios. dicha implementación y evaluación del algoritmo se realizó utilizando el lenguaje Python, empleando técnicas como la selección por torneo, el cruzamiento de un punto y la mutación uniforme y como función de aptitud utilizada se adoptó el Error Cuadrático Medio (ECM) esta herramienta es útil para comparar diferentes modelos de regresión y seleccionar el que mejor se ajusta a los datos (James, 2023) permitiendo medir la calidad de la redistribución de recursos basándose en la minimización del retraso acumulado y la eficiencia en el uso de los recursos.

La validación del algoritmo genético se realizó mediante dos enfoques: validación interna y validación cruzada con respecto a la validación interna está consistió en ejecutar múltiples simulaciones para asegurar la consistencia y estabilidad del algoritmo, por otro lado, la validación cruzada, utilizando la técnica KFold, es un método para evaluar el rendimiento de un modelo de Machine Learning, se enfoca en dividir el conjunto de datos en k subconjuntos (o pliegues), en cada iteración, se utiliza un pliegue como conjunto de

prueba para evaluar el modelo, mientras que los $k-1$ pliegues restantes se emplean como datos de entrenamiento (Nti, 2021), con esto se permitió evaluar la robustez del algoritmo frente a diferentes escenarios de distribución inicial de recursos y niveles de retraso.

Los resultados obtenidos demuestran que el algoritmo genético es altamente eficaz en la redistribución de recursos, minimizando la desviación estándar y optimizando el uso de los recursos. La comparación entre el avance acumulado planeado y el optimizado por el algoritmo mostró que se logró mitigar el retraso y alinear el avance acumulado con el plan original hacia el final del período de ejecución. Aunque el estudio se centró en la industria de la construcción, el algoritmo está diseñado para ser aplicable a diversas industrias que enfrentan problemas similares de redistribución de recursos en proyectos con retrasos.

Este estudio no solo proporciona una solución práctica para la redistribución de recursos en proyectos con retrasos, sino que también abre el camino para futuras investigaciones y desarrollos en la aplicación de algoritmos genéticos en la gestión de proyectos. La integración de datos reales y la exploración de operadores genéticos más avanzados podrían mejorar aún más la robustez y eficacia del algoritmo, facilitando su uso en diversas industrias y contextos de proyectos.

METODOLOGÍA

La presente investigación se enmarca en un diseño experimental, cuyo objetivo principal es evaluar la eficacia de un algoritmo genético para una redistribución óptima, o cercana a la óptima, de recursos en proyectos con retrasos en el inicio de su ejecución, el diseño permite manipular variables independientes determinadas por los parámetros del algoritmo genético y observar sus efectos sobre las variables dependientes en la eficacia y eficiencia de la redistribución de recursos.

La población de estudio está compuesta por actividades de la planificación de recursos en entornos simulados, que reflejan características comunes en la gestión de proyectos reales observados en la industria de la construcción, aun así, cabe señalar que el algoritmo genético está diseñado para diversas industrias. Para este estudio, se seleccionó una muestra representativa de 10 actividades simuladas con una duración de 44 días laborables (aproximadamente 2 meses), cada una con una distribución inicial de recursos calculada por medio de la distribución normal, ya que es así como se obtiene el valor planeado o valores esperados de la ejecución de la actividad, por otra parte, se tienen valores observados o el valor ganado de las actividades de los primeros 10 días (representación de dos semanas) con datos en los que se identifican un retraso

Se utilizó la distribución normal para generar datos de las actividades simuladas ya que es adecuada para modelar la variabilidad de los recursos asignados en un proyecto, después de todo muchos fenómenos en la gestión de proyectos siguen una distribución aproximadamente normal. Para cada actividad, se generaron 44 valores en representación de porcentaje siguiendo una distribución normal con una media μ y una desviación estándar σ adecuadas de manera que los valores representen la distribución inicial planificada de recursos para cada día del proyecto. La media μ y la desviación estándar σ se eligieron basándose en datos históricos y escenarios típicos de proyectos similares. Por lo tanto, se utilizó una media $\mu=22$ y una desviación estándar $\sigma=8.8$. Para simular el retraso del 10%, se ajustaron los valores de los primeros 10 días multiplicándolos por 0.9, lo que refleja el retraso acumulado. En la Figura 1 se muestra un ejemplo del resultado de la creación de los valores que representan el valor planeado Figura 1.a y el valor ganado 1.b de las actividades

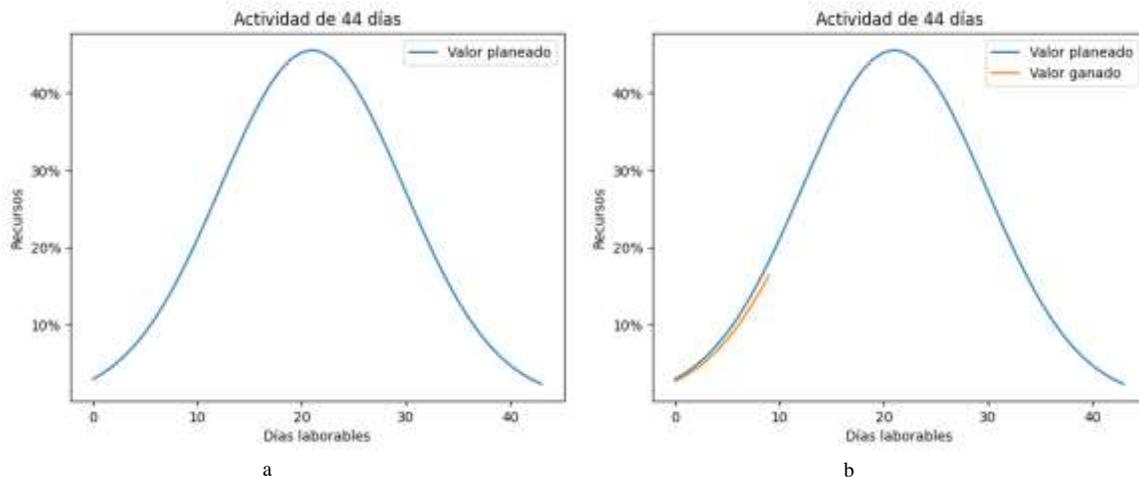


Figura 1. Distribuciones del valor planeado y el valor ganado.

Se generaron múltiples instancias de las actividades con diferentes distribuciones iniciales, simulando el valor planeado de la actividad, para evaluar la robustez y eficacia del algoritmo genético bajo diversos escenarios, para lograr esto se crean las distribuciones utilizando parámetros en rangos y en forma aleatoria, los parámetros son elegidos de acuerdo a las especificaciones de los organismo que determinan las buenas prácticas en la gestión de proyectos, se comenzara con los días asignando de 30 a 44 días laborables con números enteros y como consecuencia las media μ estará en el intervalo de 15 y 22 de igual forma representados con números enteros y la desviación estándar σ entre 5 y 9 considerando números decimales.

Para obtener el valor ganado de las actividades generadas anteriormente se crean nuevas instancias por cada una de las distribuciones de las actividades y se determina el avance de forma aleatoria, simulando con valores de atraso del 5 % al 10 %, rango en que el retraso es considerado para ajustar la distribución sin necesidad de aumentar los recursos, y de igual manera los días de avance son generados aleatoriamente en un rango de 10 a la media μ resultante del cálculo de distribución anterior.

Para implementar y evaluar el algoritmo genético, se desarrolló la codificación en Python. Con el análisis de la información se define una representación de la solución, ya que los datos de estas curvas se pueden representar como un arreglo de números reales que representan el avance de una actividad a lo largo del tiempo en la que se planifico, en otras palabras, se está tratando con el avance de la actividad a lo largo del tiempo, esto se puede representar como un cromosoma o individuo para el algoritmo genético, donde cada gen representa el avance de un día, por lo que si se tienen N días restantes de la tarea, como se menciona al principio el algoritmo genético buscara en el remanente del avance de la actividad es decir, el avance que falta por terminar la actividad, el cromosoma podría ser una secuencia de N genes con valores de números reales y su tamaño será definido por el remanente del avance de la actividad. La Figura 2 muestra cómo se obtienen los individuos (cromosomas) de una actividad generada con 39 días de duración y 10 días de avance reflejando un atraso del 10 %, por lo tanto, los 29 días por ejecutar serán los genes que serán asignados por números reales aleatorios.

Días	Valor Planeado	Valor Ganado
1	0.16 %	0.15 %
2	0.26 %	0.24 %
3	0.41 %	0.37 %
4	0.63 %	0.57 %
5	0.93 %	0.84 %
6	1.34 %	1.20 %
7	1.85 %	1.66 %
8	2.47 %	2.22 %
9	3.19 %	2.87 %
10	3.99 %	3.59 %
11	4.82 %	A01
12	5.62 %	A02
13	6.34 %	A03
14	6.90 %	A04
15	7.27 %	A05
16	7.39 %	A06
17	7.27 %	A07
18	6.90 %	A08
19	6.34 %	A09
20	5.62 %	A10
21	4.82 %	A11
22	3.99 %	A12
23	3.19 %	A13
24	2.47 %	A14
25	1.85 %	A15
26	1.34 %	A16
27	0.93 %	A17
28	0.63 %	A18
29	0.41 %	A19
30	0.26 %	A20
31	0.16 %	A21
32	0.10 %	A22
33	0.06 %	A23
34	0.03 %	A24
35	0.02 %	A25
36	0.01 %	A26
37	0.01 %	A27
38	0.01 %	A28
39	0.01 %	A29
	100.00 %	13.71 %

Días laborables:	39
Media:	16
Desviación estándar:	5.4
Días de avance:	10

A 10 días de avance	
Valor Planeado	15.23 %
Valor Ganado	13.71 %
Diferencia	-1.52 %
Atraso	-9.96 %

Tamaño del cromosoma: 29 alelos (A00).

Figura 2. Actividad generada aleatoriamente con una duración de 39 días.

La elección del operador de selección en un algoritmo genético depende en gran medida de la naturaleza del problema y de los objetivos específicos, en este sentido se elige la selección por torneo donde se seleccionan varios individuos al azar y el más apto de ellos se convierte en padre este proceso se repite hasta completar el número deseado de padres este método puede ser robusto y funcionar bien en la eficiencia del problema.

Con el cruzamiento es la esencia para combinar información genética de dos padres y generar descendencia y puede influir en la exploración del espacio de búsqueda y en la convergencia del algoritmo genético. Se ha elegido el cruzamiento de un punto (one point crossover), en el cual se intercambian las

partes de los cromosomas que están entre un punto de corte generado aleatoriamente, ya que puede ser útil si hay bloques de información genética que deben conservarse o intercambiarse en su totalidad para una distribución eficiente.

En los algoritmos genéticos (Agus) que trabajan con números reales o flota, la mutación juega un papel crucial para mantener la diversidad genética y evitar el estancamiento en soluciones locales, se usará la mutación uniforme donde cada gen en el individuo tiene una probabilidad igual de ser alterado de forma independiente de los demás genes al azar, una ventaja de esta técnica es mantener la diversidad genética sin introducir cambios drásticos en la población.

Se utilizan dos criterios de paro para este problema de optimización, en primer lugar, el criterio de mejor aptitud del individuo en el cual si el mejor individuo ha llegado a la aptitud deseada, en este caso 0.99995 de valor de aptitud, se interrumpe el algoritmo genético dando y se determina que la mejor solución ha sido encontrada. En segundo lugar, se elige el criterio de número máximo de generaciones donde se detiene la evolución una vez que se alcanza un número máximo predefinido de generaciones.

La función de aptitud es la encargada de medir la calidad de la redistribución de recursos basándose en la minimización del retraso acumulado y la eficiencia en el uso de los recursos, para lograr esto se utiliza el estimador Error Cuadrático Medio (ECM), para obtener el ECM se calcula la diferencia de los valores planeados restantes con los valores ganados creados aleatoriamente siendo estos los genes en el cromosoma o individuo, posteriormente se elevan al cuadrado cada uno de ellos para abrir paso a obtener la media haciendo uso de la biblioteca NumPy en Python, `mean`, la cual calcula la media aritmética de los elementos de un array o de una secuencia de valores como el objetivo es medir la diferencia entre una distribución de avance planeada con las distribuciones ajustadas de las soluciones candidatas se define la función de aptitud como una inversa (ya que se busca minimizar) de la media, quedando de la siguiente manera:

$$aptitud = \frac{1}{1 + \left(\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2\right)}$$

En la Figura 3 se muestra el pseudocódigo del algoritmo genético.

Algorithm 1: Algoritmo genético para redistribuir el presupuesto

```
procedure ALGORITMOGENETICO(distribución planeada, avance, tamaño  
de población, número de generaciones, probabilidad de reproducción,  
probabilidad de mutación)  
  InicializarPoblación(Tamaño de población, Tamaño de cromosoma)  
  EvaluarPoblación()  
  while No se cumplan los criterios de paro do  
    SeleccionarPadres()  
    CruzarPadres(Probabilidad de reproducción)  
    MutarDescendencia(Probabilidad de mutación)  
    EvaluarIndividuos()  
  end while  
  return Mejor individuo de la población final  
end procedure
```

Figura 3. Seudocódigo del algoritmo genético.

La validación del algoritmo genético se realizó mediante dos enfoques, una validación interna y una validación cruzada, con respecto a la primera validación se ejecutaron múltiples simulaciones para asegurar la consistencia y estabilidad del algoritmo consistiendo en pruebas con diferentes parámetros asignados aleatoriamente para el algoritmo donde se verifica que las soluciones convergen a valores óptimos, la selección de los parámetros está determinada en una primera fase por el tiempo de ejecución iniciando con rangos en los parámetros de la siguiente forma:

1. Criterio de paro por generaciones en 10, 000.
2. Población entre 20 y 200 individuos.
3. Probabilidad de cruce entre el 60 y 100 por ciento.
4. Probabilidad de mutación entre el 0.1 y 10 por ciento.

Por cada iteración el algoritmo genético se ejecutó en 255 ocasiones con el objetivo de refinar los parámetros los cuales servirán para generar los datos y con estos obtener una validación en forma gráfica, a modo de ejemplo se muestra en la Tabla 1 los datos con los que se obtiene una gráfica de convergencia.

Test	Fitness	Generations	Population	Crossover Rate	Mutation Rate	Time (s)
1	0.99995178	1700	90	0.8708	0.0318	0.4715
2	0.99995119	2369	60	0.8430	0.0219	0.4534
3	0.99995179	1191	69	0.9340	0.0315	0.2589
4	0.99994894	3100	66	0.8692	0.0348	0.6617
5	0.99994851	3100	47	0.8881	0.0166	0.4843
6	0.99995175	1504	84	0.9484	0.0151	0.3951
7	0.99995082	2996	41	0.8146	0.0103	0.4133
8	0.99995181	2736	51	0.7634	0.0237	0.4496
9	0.99995005	2934	34	0.7318	0.0334	0.3450
10	0.99995283	3047	99	0.7317	0.0081	0.9044
11	0.99992083	3100	95	0.7275	0.0072	0.8865
12	0.99996004	2504	45	0.9182	0.0269	0.3722
13	0.99995124	1872	92	0.7997	0.0231	0.5226
14	0.99995149	1894	48	0.8565	0.0168	0.2967
15	0.99995308	2068	85	0.7720	0.0162	0.5390
16	0.99995020	2706	98	0.8783	0.0220	0.7998
17	0.99995067	3081	102	0.9149	0.0079	0.9437
18	0.99995329	2176	90	0.9316	0.0251	0.5963
19	0.99992730	3100	74	0.7453	0.0264	0.7068
20	0.99995492	1978	77	0.8232	0.0260	0.4714
21	0.99995399	499	90	0.7780	0.0051	0.1368

Tabla 1. Datos obtenidos de 21 ejecuciones del algoritmo genético.

Para la gráfica de convergencia se observan los datos de la Tabla 1 y se obtiene la media del mejor fitness en la prueba en este caso la media de las 21 ejecuciones es el Test 14 con fitness 0.99995149 que concluyo con 1, 894 generaciones y con los siguientes parámetros: 48 individuos en la población con un 85.65% de posibilidad de cruza y una probabilidad de Mitación de 1.68%, con estos parámetros mencionados el algoritmo genético concluyo en 0.2967 segundos. La Figura 4 se tiene la gráfica de convergencia donde se da la valides en la evolución en la búsqueda de la solución óptima al problema.

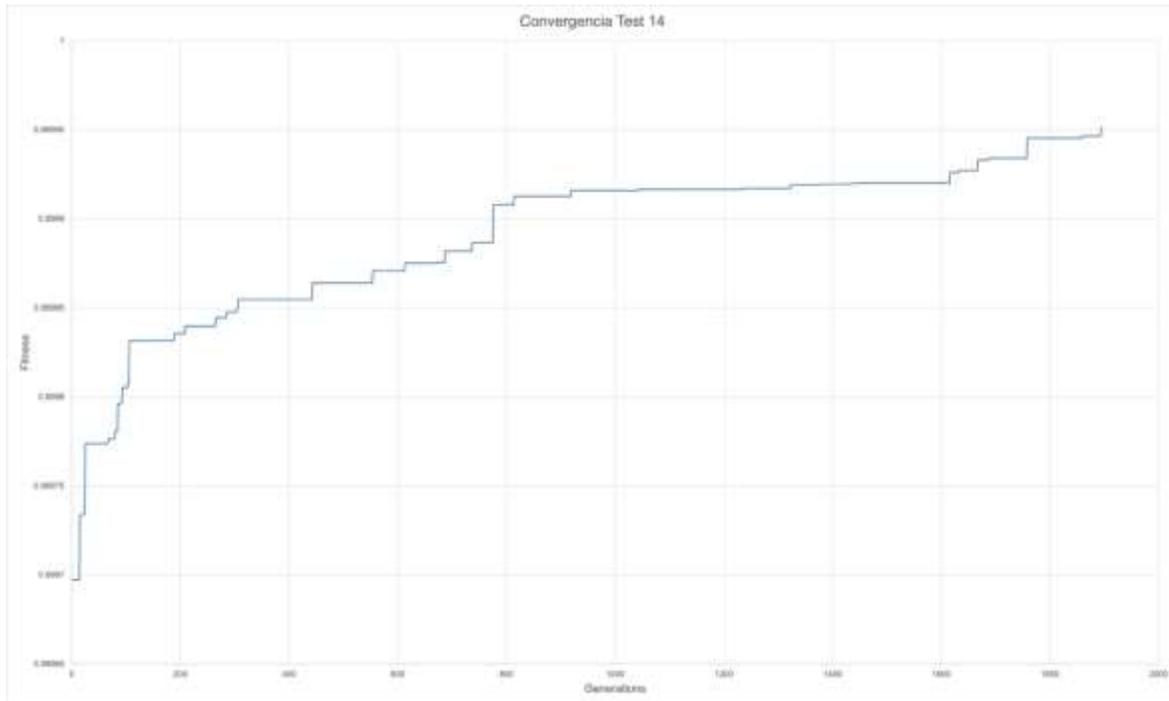


Figura 4. Grafica de convergencia resultante del Test 14.

Por otro lado, se realiza una validación cruzada con los parámetros de la Tabla 1 con el objetivo de verificar la robustez del algoritmo frente a diferentes escenarios de distribución inicial de recursos y niveles de retraso, el procedimiento para realizar la validación es ejecutar el algoritmo genético con todas las posibles combinaciones de los parámetros y el resultado será los mejores parámetros para ejecutar el algoritmo genético. La validación cruzada KFold de sklearn.model_selection en Python funciona para dividir los datos en 5 folds, ayudando a evaluar el rendimiento del algoritmo genético diseñado para redistribuir los recursos en un proyecto. En la Tabla 2 se muestran los valores seleccionados para dicha validación generando 400 ejecuciones.

Generations	Population	Crossover Rate	Mutation Rate
50	45	0.7275	0.0072
100	48	0.7317	0.0168
200	95	0.8565	0.0220
300	99	0.8783	0.0269
500		0.9182	

Tabla 2. Parámetros para la validación cruzada.

RESULTADOS Y DISCUSIÓN

Para abordar el problema de la redistribución de recursos en proyectos que presentan un retraso del 20%, se implementó el algoritmo genético con el objetivo de optimizar la planificación restante de los recursos en los días restantes del proyecto. Los resultados del algoritmo genético se evaluaron utilizando las métricas de desviación estándar, cumplimiento de objetivos y eficiencia de recursos. A continuación, se presentan los resultados promedios obtenidos para las mejores configuraciones de parámetros:

- Desviación Estándar Promedio: 0.012
- Cumplimiento de Objetivos Promedio: 1.0
- Eficiencia de Recursos Promedio: 1.437

El algoritmo genético demostró ser altamente eficaz en la redistribución de los recursos, minimizando la desviación estándar a un valor muy bajo (0.0120), lo que indica una distribución uniforme de los recursos, respecto al cumplimiento de objetivos alcanzó un valor perfecto (1.0), lo que significa que todas las soluciones generadas cumplieron con los requisitos mínimos de recursos necesarios para completar los proyectos.

La eficiencia de recursos promedio (1.437) sugiere que el algoritmo no solo redistribuye los recursos de manera efectiva, sino que también optimiza su uso, logrando una mayor eficiencia en comparación con la distribución inicial, por otro lado, configuración óptima de los parámetros del algoritmo genético, obtenida mediante la validación cruzada K-Fold, asegura que el modelo tiene una excelente capacidad de generalización

Los mejores parámetros encontrados fueron:

- Tamaño de la Población: 48
- Probabilidad de Cruce: 0.7317
- Probabilidad de Mutación: 0.0269
- Número de Generaciones: 500

En la Figura 5 se presenta la gráfica donde se muestra una comparación entre el avance acumulado planeado y el avance acumulado optimizado por el algoritmo genético para una tarea de 44 días con un retraso del 20%. A medida que la tarea progresa, la curva de la solución del algoritmo genético (naranja) comienza a diferir de la curva planeada (azul), esto es que el algoritmo ha redistribuido los recursos para

compensar el retraso y optimizar el avance acumulado, se puede observar que alrededor del día 20 en adelante, la curva optimizada se aproxima y se ajusta estrechamente a la curva planeada, lo que indica que el algoritmo genético ha logrado mitigar el retraso y alinear el avance acumulado con el plan original hacia el final del período.

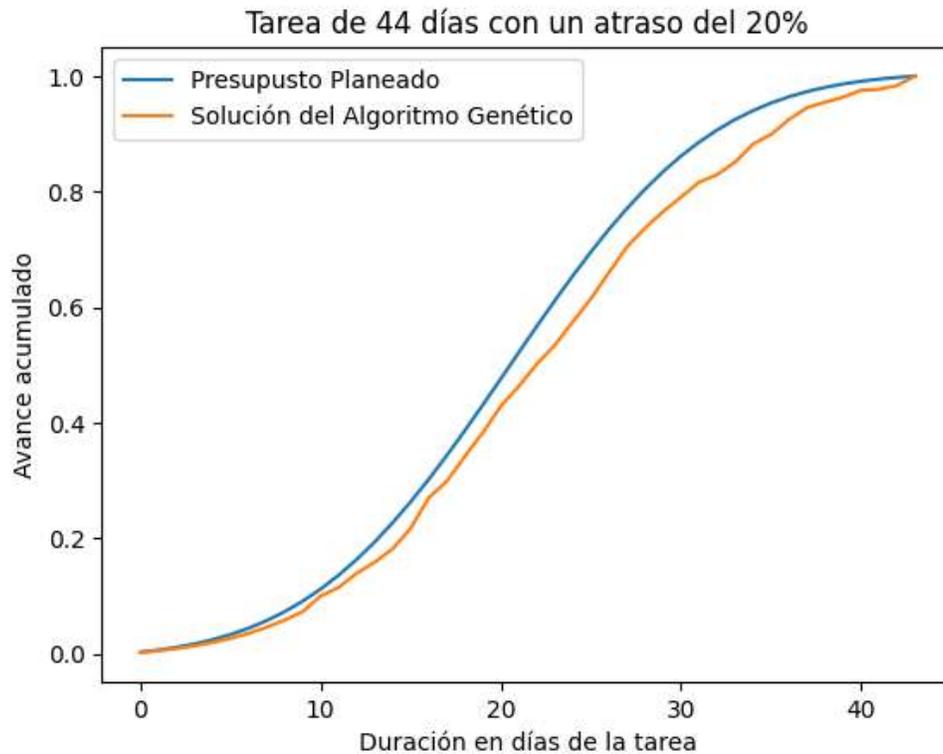


Figura 5. Curva S del plan original vs. redistribución optimizada por el algoritmo genético.

CONCLUSIONES

El presente estudio ha demostrado la eficacia de un algoritmo genético para la redistribución óptima, o cercana a la óptima, de recursos en proyectos con retrasos en el inicio de su ejecución, con la implementación de un diseño experimental que permitió evaluar los efectos de diferentes parámetros del algoritmo sobre la eficacia y eficiencia de la redistribución de recursos en entornos simulados representativos de la industria de la construcción.

El algoritmo genético demostró ser altamente eficaz, minimizando la desviación estándar a un valor muy bajo (0.0120), lo que indica una distribución uniforme donde se alcanzó un cumplimiento de objetivos

perfecto (1.0), lo que significa que todas las soluciones generadas cumplieron con los requisitos mínimos de recursos necesarios para completar los proyectos y con una eficiencia de recursos promedio (1.437) indicado que el algoritmo no solo redistribuye los recursos de manera efectiva, sino que también optimiza su uso, logrando una mayor eficiencia comprobada con una validación interna y una validación cruzada K-Fold que permitieron determinar la configuración óptima de los parámetros del algoritmo genético.

La implementación del algoritmo en Python, utilizando técnicas como la selección por torneo, el cruzamiento de un punto y la mutación uniforme, resultó en una robusta solución capaz de adaptarse a diversos escenarios de distribución de recursos y niveles de retraso, con el uso del Error Cuadrático Medio (ECM) como función de aptitud fue posible medir la calidad de la redistribución de recursos de manera precisa y eficiente.

La comparación entre el avance acumulado planeado y el optimizado por el algoritmo genético mostró que el algoritmo logró mitigar el retraso y alinear el avance acumulado con el plan original hacia el final del período de ejecución que se pudo apreciar de forma gráfica con la Curva S donde se demostró cómo el algoritmo ajusta la redistribución de recursos, compensando el retraso inicial y optimizando la ejecución del proyecto. Aunque el estudio se centró en la industria de la construcción, el algoritmo genético está diseñado para ser aplicable a diversas industrias que enfrentan problemas similares de redistribución de recursos en proyectos con retrasos.

Cabe señalar que el estudio se basó en entornos simulados para generar los datos de actividades, aunque estos entornos reflejan características comunes en la gestión de proyectos reales, pueden no capturar todas las variabilidades y complejidades de un entorno real en donde seguramente la complejidad computacional puede aumentar con la escala del proyecto, afectando el tiempo de ejecución y los recursos necesarios para ejecutar el algoritmo, además la función de aptitud se basa en el Error Cuadrático Medio (ECM), que es una métrica útil para medir la desviación, sin embargo, otros criterios de evaluación, como el costo adicional, recursos o el impacto en la calidad del proyecto, no fueron considerados.

La validación interna y la validación cruzada fueron fundamentales para determinar los parámetros óptimos del algoritmo genético, ayudando a mejorar la robustez y la generalización del algoritmo mostrando una gran capacidad de adaptabilidad, siendo capaz de ajustarse a diferentes escenarios de distribución de recursos y niveles de retraso.

Como trabajo a futuro está integrar datos reales de proyectos de diversas industrias para validar y mejorar la robustez del algoritmo genético, abriendo las posibilidades de explorar el uso de operadores genéticos más avanzados, como el cruzamiento multipunto, la mutación adaptativa y la selección elitista, para mejorar la exploración y explotación del espacio de soluciones, considerando desarrollar herramientas y plataformas de software que integren el algoritmo genético, facilitando su uso por parte de gestores de proyectos y profesionales en diferentes industrias para determinar cómo la colaboración y la comunicación entre los equipos de proyecto pueden influir en la eficacia del algoritmo genético.

REFERENCIAS BIBLIOGRÁFICAS

- El-Shorbagy, M. A., & El-Refaey, A. M. (2020). *Hybridization of grasshopper optimization algorithm with genetic algorithm for solving system of non-linear equations*. IEEE Access, 8, 220944-220961.
- Galli, B. J. (2021). Statistical tools and their impact on project management—how they relate. *The Journal of Modern Project Management*, 9(2).
- Gridin, I. (2021). *Learning Genetic Algorithms with Python: Empower the performance of Machine Learning and AI models with the capabilities of a powerful search algorithm (English Edition)*. BPB publications.
- Holland, J.H. (1992). Genetic algorithms. *Scientific American*, 267(1), 66–73.
- James, G., Witten, D., Hastie, T. & Tibshirani, R. (2023). *An introduction to statistical learning with applications in R*, New York, Springer Science and Business Media, 2013, eISBN: 978-1-4614-7137-7.
- Mattos, A. D., & de Valderrama, F. G. F. (2019). *Métodos de planificación y control de obras*. Reverté.
- Nti, I. K., Nyarko-Boateng, O., & Aning, J. (2021). *Performance of machine learning algorithms with different K values in K-fold cross-validation*. International Journal of Information Technology and Computer Science, 13(6), 61-71.