

Planificación de tareas mediante el uso de algoritmos bioinspirados

Daniel Cruz, *dcruzg1900@alumno.ipn.mx*, Rebeca Lopez, *rlopez11901@alumno.ipn.mx*,
y M. Alejandro Pérez, *mperezn1900@alumno.ipn.mx*

Instituto Politécnico Nacional
Escuela Superior de Cómputo, Inteligencia Artificial

Abstract—En este artículo se aborda el problema de la planificación de tareas o también conocido como *Job Scheduling Problem* buscando minimizar lo más posible el tiempo de convergencia hacia soluciones subóptimas para el problema. El algoritmo hace uso del algoritmo genético en el cual se cruza, muta y se seleccionan los individuos de manera elitista por medio de la selección por torneo. Se verificó la efectividad del algoritmo propuesto por medio de pruebas que garantizan la viabilidad y la validez estadística de los resultados.

Index Terms—Planificación de tareas, algoritmos bioinspirados, makespan

I. INTRODUCCIÓN

LOS algoritmos bioinspirados son técnicas de optimización de problemas que se basan en principios que son observados en la naturaleza, como por ejemplo los algoritmos genéticos que se basan en la evolución natural. Estas estrategias se destacan por la capacidad que tienen para explorar grandes espacios de solución de manera eficiente, especialmente problemas que tienen restricciones complejas o un cierto grado de incertidumbre.

Por otro lado, está el problema de la planificación de tareas, también conocida como *Job Scheduling Problem*, este consiste en optimizar asignaciones de trabajos en sistemas compartidos; tiene como objetivo principal maximizar las prioridades de las tareas o minimizar el tiempo total necesario para completar un conjunto de operaciones, este concepto se denomina **makespan**. Para lograr esto se utilizan estructuras como las colas de prioridad, ya que estas mantienen un seguimiento del estado de las tareas según sus tiempo de finalización y sus prioridades, además, a medida que las condiciones cambian, como la finalización de un trabajo o la incorporación de nuevos, el planificador ajusta el orden de ejecución. Para abordar el problema de la planificación de tareas, se han desarrollado diversas soluciones utilizando enfoques innovadores y sofisticados, entre estas técnicas destacan la programación dinámica, los algoritmos genéticos y el algoritmo de la colonia de hormigas, cada uno de

los cuales ofrece ventajas particulares en términos de eficiencia y capacidad de exploración de soluciones óptimas.

Este problema ha sido estudiado desde diferentes perspectivas, es decir, como un problema monoobjetivo y como un problema multiobjetivo. En particular, el enfoque multiobjetivo ha ganado un interés significativo en los últimos años debido a su aplicabilidad a escenarios complejos del mundo real, donde es necesario optimizar múltiples criterios simultáneamente, como tiempo, costo y recursos.

A lo largo del tiempo se han propuesto soluciones diversas e interesantes que resuelven el problema para situaciones particulares; algunas de ellas hacen uso de algoritmos altamente conocidos para la optimización de problemas. A continuación se describen algunos artículos en donde se proponen soluciones para la planificación de tareas con diferentes formas de abordar el problema.

A. Solving Multi objective Job shop scheduling Problems using Artificial Immune System Shifting Bottleneck Approach

El enfoque que se utiliza en este artículo es el de Sistema Inmunológico Artificial con Enfoque de Cuellos de Botella Cambiantes o AISSB por sus siglas en inglés para resolver la planificación de tareas multiobjetivo. Este enfoque ha tomado relevancia desde hace unas décadas, el cual se basa en el sistema inmune de los seres vivos, pues este sistema está compuesto por componentes simples los cuales desempeñan una tarea pequeña y específica para lograr objetivos complejos, convirtiéndose en un grande, complejo y altamente coordinado sistema de comunicación [1].

En esta propuesta, la secuencia inicial de soluciones es procesada con el Algoritmo del Sistema Inmunológico (ASI) y, posteriormente, se refina con el enfoque de Cuellos de Botella Cambiantes (CBC). Para el ASI se usaron dos principios base: principio de selección clonal y principio de maduración de afinidad, el primero explica

cómo el sistema inmunológico identifica y combate los antígenos, y el segundo describe el proceso mediante el cual el sistema inmunológico mejora progresivamente la capacidad de los anticuerpos para unirse a antígenos específicos. De esta manera, el CBC es utilizado para realizar ciclos de reoptimización que refinan la solución para alcanzar soluciones óptimas o casi óptimas [2].

Los resultados obtenidos mediante la aplicación de este enfoque fueron la minimización en las métricas (makespan, mean flow-time y mean tardiness) en comparación con otros algoritmos, obteniendo resultados prometedores. Con esto se concluyó que el algoritmo que fue propuesto es competente y eficaz para resolver problemas con múltiples criterios [2].

B. An Improved Ant Colony Algorithm is Proposed to Solve the Single Objective Flexible Job-shop Scheduling Problem

En este artículo se tiene como objetivo de optimización la minimización del tiempo máximo de finalización, proponiendo una mejora al algoritmo de colonia de hormigas para su resolución, haciendo modificaciones en las reglas de actualización de feromonas y las reglas de transición de estado, y también se establece un modelo de grafo disyuntivo para el problema [3].

El algoritmo de colonia de hormigas es superior a otros algoritmos en términos de calidad de la solución y velocidad de convergencia al abordar problemas de programación simples, pero la velocidad de convergencia se ve comprometida cuando la escala del problema es grande, debido al gran número de hormigas y al hecho de que cada una construye la solución de manera independiente. Por en esta propuesta, se pretendía mejorar la capacidad de búsqueda global y la velocidad de convergencia del algoritmo [3].

Mediante las mejoras propuestas para el algoritmo de colonia de hormigas, se logró que el algoritmo converja rápidamente hacia la solución óptima, verificando la efectividad del algoritmo mediante los resultados experimentales para el problema Flexible de Programación Job-shop de Objetivo Único [3].

C. An Improved Grey Wolf Optimizer for Flexible Job-shop Scheduling Problem

El algoritmo propuesto es una versión mejorada de lobo gris (IGWO, por sus siglas en inglés) para la solución del problema flexible de programación Job-shop. Así mismo, se introduce una estrategia de búsqueda local para equilibrar eficazmente las capacidades de exploración y explotación del algoritmo [4].

Por medio de un estudio de caso, se pudieron obtener soluciones satisfactorias en poco tiempo, mostrando que

el algoritmo propuesto es efectivo en términos de calidad de solución y velocidad de ejecución, ya que la mejora incorporada al algoritmo de lobo gris equilibra las capacidades de búsqueda global y local, evitando soluciones prematuras en óptimos locales [4].

II. SOLUCIÓN MEDIANTE ALGORITMO GENÉTICO

Para la solución del problema se propone el uso de un algoritmo genético como se explica a continuación:

Algorithm 1 Algoritmo Genético para la planificación de tareas

```

1: Entrada: TamPoblacion, numGeneraciones, TasaMutacion, TasaCruce
2: Salida: Mejor planificación de tareas
3: function ALGORITMOGENETICO(TamPoblacion, numGeneraciones, TasaMutacion, TasaCruce)
4:   Generar población inicial
5:   for cada generación  $i$  de 1 a  $numGeneraciones$ 
6:     do
7:       Seleccionar mejores individuos por torneo
8:       if random < TasaCruce then
9:         Operador de cruce
10:      end if
11:      if random < TasaMutacion then
12:        Operador de mutación a la nueva población
13:      end if
14:      Seleccionar los mejores individuos
15:    end for
16:  end function

```

Con este algoritmo se logra mejorar la calidad de los individuos en la población con cada generación que pasa al cruzar únicamente a los individuos con menor valor en la función objetivo y posteriormente mutando con un porcentaje bajo a los nuevos individuos generados. También se implementó al final una estrategia elitista para seleccionar a los mejores n individuos para formar la nueva generación y asegurar que el algoritmo convergerá sin perder a los mejores individuos de cada generación.

III. PRUEBAS

Para evaluar el rendimiento del algoritmo desarrollado, se realizarán pruebas utilizando el análisis de varianza (ANOVA), con el objetivo de determinar si las diferencias observadas entre las medias de los grupos son estadísticamente significativas. En estas pruebas se emplearán cinco configuraciones distintas de los parámetros clave del algoritmo. Para cada configuración, se llevarán a cabo 10 ejecuciones independientes, empleando diferentes semillas en la generación de números aleatorios para garantizar la variabilidad y la validez estadística de los resultados.

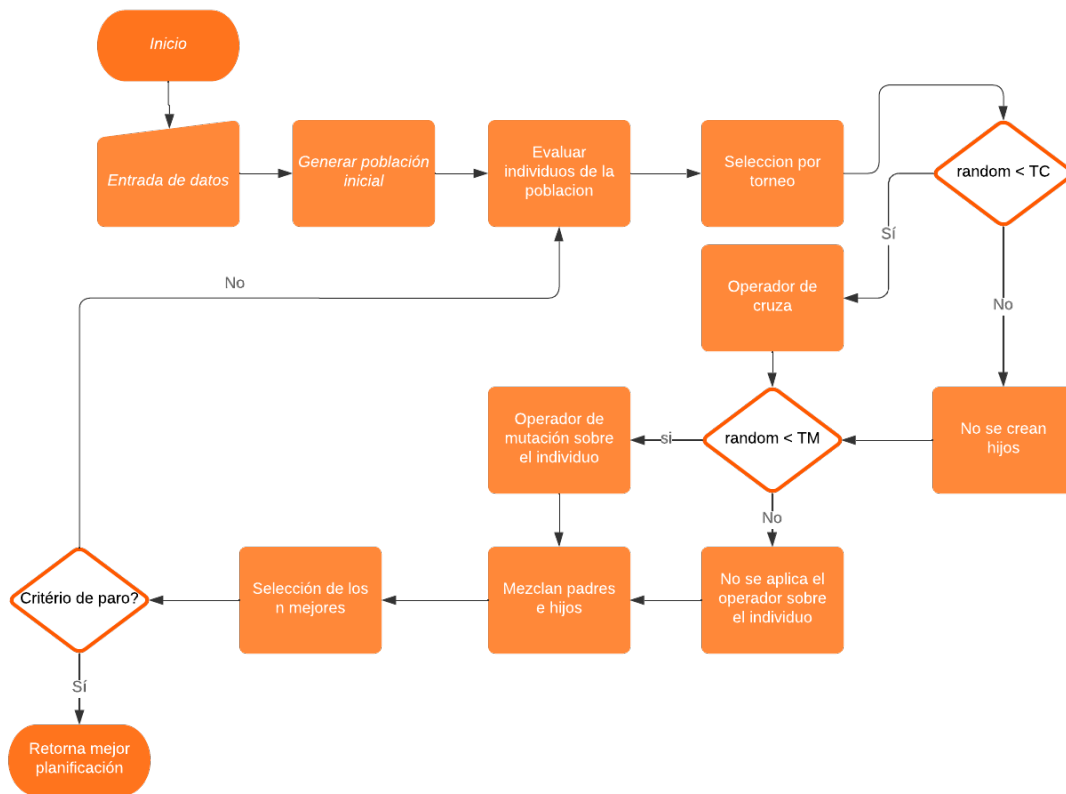


Fig. 1. Diagrama de flujo del algoritmo genético

TABLE I
RESULTADOS DE LAS CONFIGURACIONES CON 200 GENERACIONES

Configuración	Tamaño Población	Tasa de mutación	Tasa de cruce	Promedio	Desviación estándar
1	50	0.3	0.85	61.5	7.2
2	50	0.3	0.6	65.3	14.2
3	50	0.1	0.85	63.18	14.2
4	50	0.1	0.6	63.7	13.5
5	100	0.3	0.85	51.3	5.9
6	100	0.3	0.6	41.6	7.5
7	100	0.1	0.85	52.4	9.2
8	100	0.1	0.6	44.9	5.4

IV. RESULTADOS

Se llevó a cabo un análisis de varianza (ANOVA) para evaluar el efecto de las configuraciones del algoritmo genético en el promedio del desempeño obtenido en la resolución del Job Scheduling Problem. En este análisis, se consideraron tres factores principales: tamaño de la población, tasa de mutación y tasa de cruce. Los resultados detallados de las configuraciones más significativas probadas se resumen en la Tabla I.

Por medio de este análisis se observó que el tamaño de población tiene un efecto significativo en el desempeño

del algoritmo, como se observa en la Tabla I, donde las configuraciones con un tamaño de población de 50 obtuvieron mejores resultados promedio en comparación con las de tamaño 100, esto indica que este parámetro juega un papel crucial en la capacidad del algoritmo para explorar y explotar el espacio de soluciones de manera eficiente. Por otro lado, la tasa de mutación no mostró un efecto significativo, y los resultados promedio fueron consistentes entre las configuraciones con tasas de 0.1 y 0.3, indicando que este factor no es determinante dentro de los niveles evaluados. De manera similar, la

tasa de cruce tampoco tuvo un impacto significativo en el desempeño, con una variación mínima entre las configuraciones con tasas de 0.6 y 0.85.

V. CONCLUSIÓN

En este artículo, se presentó un algoritmo basado en técnicas de algoritmos genéticos para abordar el problema de planificación de tareas, y se implementó una estrategia elitista para garantizar la selección de las mejores soluciones a lo largo del proceso evolutivo. A través de las pruebas realizadas y el análisis estadístico mediante ANOVA, se identificó que el tamaño de la población tuvo un impacto significativo en los resultados obtenidos; sin embargo, no se lograron los resultados esperados, ya que las soluciones generadas no alcanzaron un desempeño óptimo. Estos resultados, aunque limitados, permitieron identificar áreas de mejora que podrían contribuir al desarrollo de un algoritmo más eficiente y alineado con los objetivos planteados para la resolución del problema de planificación de tareas.

VI. TRABAJO FUTURO

El trabajo resultante se podría beneficiar con algunas mejoras en los mecanismos de búsqueda local; aunque el algoritmo genético demuestra ser eficaz en la convergencia en las iteraciones iniciales, cuando se generan nuevas planificaciones en ocasiones se identifican muchos espacios en blanco o brechas en el plan generado. Una línea de investigación futura puede centrarse en el desarrollo de una estrategia de búsqueda local que permita identificar y asignar tareas de manera más eficiente en las áreas menos saturadas del espacio de solución.

Este enfoque incluiría la implementación de diversas formas de evaluar que analicen el estado actual de la asignación y que puedan generar ajustes locales para minimizar espacios desocupados y mejorar la distribución de carga entre las tareas; además de implementar estas técnicas, se pueden establecer métricas que evalúen la calidad de los ajustes que se hacen a las planificaciones y de esta manera se podría mejorar iterativamente mientras se generan las poblaciones y no exista una gran variación entre los individuos, así se ofrecería una solución más robusta.

REFERENCES

- [1] R. Monroy, R. Saab, and F. Godínez, "On modelling an immune system," *Computación y Sistemas*, vol. 7, no. 4, pp. 249–259, 2004. [Online]. Available: http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1405-55462004000200004&lng=es&tlng=en.
- [2] K. Palanikumar, *192.1 introduction*, 2015. [Online]. Available: <https://app.knovel.com/hotlink/khtml/id:kt00UC0DO1/advances-in-materials/solving-mu-introduction>.
- [3] M. huang, D. guo, X. liang, and X. liang, "An improved ant colony algorithm is proposed to solve the single objective flexible job-shop scheduling problem," in *2020 IEEE 8th International Conference on Computer Science and Network Technology (ICCSNT)*, 2020, pp. 16–21. DOI: 10.1109/ICCSNT50940.2020.9305005.
- [4] Y. Jieran, W. Aimin, G. Yan, and S. Xinyi, "An improved grey wolf optimizer for flexible job-shop scheduling problem," in *2020 IEEE 11th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT)*, 2020, pp. 213–217. DOI: 10.1109/ICMIMT49010.2020.9041184.