

Consideración de las soluciones overlapping en la selección y parametrización de los operadores genéticos de un algoritmo evolutivo aplicado al flow-shop

Begoña González¹, Mariano Frutos^{2,3}, Daniel A. Rossit^{2,4} y Máximo Méndez¹

¹Instituto Universitario SIANI, Universidad de Las Palmas de Gran Canaria (ULPGC), Las Palmas de Gran Canaria (35017), España.

²Departamento de Ingeniería, Universidad Nacional del Sur (UNS), Bahía Blanca (8000), Argentina.

³Instituto de Investigaciones Económicas y Sociales del Sur (IIESS UNS-CONICET), Bahía Blanca (8000), Argentina.

⁴Instituto de Matemática de Bahía Blanca (INMABB UNS-CONICET), Bahía Blanca (8000), Argentina.

bego.landin@ulpgc.es - mfrutos@uns.edu.ar
daniel.rossit@uns.edu.ar - maximo.mendez@ulpgc.es

Abstract. En años recientes, se ha notado un crecimiento en el desarrollo de algoritmos para problemas en entornos de producción flow-shop. Nuestro estudio se enfoca en un análisis exhaustivo de un algoritmo evolutivo para el cual se utilizaron dos criterios de evaluación: makespan y total tardiness. Se analizan diferentes operadores genéticos para evaluar su impacto en diversidad y convergencia, prestando especial atención a las soluciones overlapping. Los resultados buscan mejorar la comprensión y proporcionar directrices prácticas para aplicaciones reales.

Keywords: Flow-shop, Optimización, Algoritmos Evolutivos, Overlapping.

1 Introducción

En los últimos años, se ha observado un aumento significativo en el desarrollo de algoritmos destinados a resolver problemas de flow-shop [1]. En este contexto, el presente trabajo se centra en realizar un estudio de un algoritmo evolutivo, empleando como criterios de evaluación el “makespan” [2] y el “total tardiness” [3]. El enfoque principal es analizar en detalle los operadores genéticos que dirigen la exploración del espacio de soluciones factibles. Se busca evaluar el impacto de estos operadores en términos de diversidad y convergencia, con el fin de comprender mejor su comportamiento y su influencia en el proceso de búsqueda. Específicamente, se dedica especial atención al fenómeno del overlapping [4], ya que este aspecto desempeña un papel crucial en la selección de los operadores óptimos para el problema y en la determinación de la configuración de parámetros más adecuada. Se espera que los resultados de este estudio contribuyan a mejorar la comprensión de los mecanismos subyacentes a los algoritmos evolutivos aplicados a problemas de flow-shop, así como a proporcionar directrices prácticas para la selección y configuración de estos algoritmos en aplicaciones reales.

2 Problema del flow-shop

El problema del flow-shop requiere que n trabajos (J_1, \dots, J_n) pasen secuencialmente por una serie de m máquinas (M_1, \dots, M_m). Cada trabajo debe ser procesado en cada máquina solo

una vez, en un orden establecido, primero M_1 , luego M_2 , y así hasta completar todas las máquinas. Cada máquina puede procesar un solo trabajo por vez. El problema consiste en identificar el orden en que deben secuenciarse los trabajos de manera de minimizar algún objetivo del sistema. En este trabajo se perseguirán de manera individual dos objetivos clásicos, el makespan [2] y el total tardiness [3]. El makespan se define como el tiempo que transcurre entre que el primer trabajo se inicia y el último que se completa. El total tardiness se refiere a la suma de los retrasos individuales de todos los trabajos en el flujo de producción.

Este problema tiene dos versiones: permutativa (PFS, permutation flow-shop) [5] y no permutativa (NPFS, non-permutation flow-shop) [6]. En la versión PFS, todas las tareas siguen una secuencia idéntica en todas las máquinas. Por otro lado, la versión NPFS permite que las secuencias de tareas sean diferentes en las distintas máquinas. Aunque la versión NPFS es más compleja que la PFS, es importante destacar que varios estudios han informado mejoras marginales, generalmente entre el 1% y el 3%, en el makespan al emplear la variante NPFS en comparación con la PFS [7]. La versión PFS fue la utilizada en este trabajo.

3 Algoritmo evolutivo y sus operadores genéticos

En este trabajo se utilizó un algoritmo evolutivo [8]. Estos son métodos de optimización y búsqueda inspirados en los procesos naturales de evolución y selección natural. Los algoritmos evolutivos operan mediante la creación de una población inicial de soluciones potenciales para el problema en cuestión. Cada solución potencial, conocida como "individuo", está representada de manera codificada y puede ser interpretada como un punto en el espacio de búsqueda del problema. Estos individuos son evaluados mediante una función de aptitud que determina qué tan buena es cada solución en términos de la calidad de la solución al problema.

En este trabajo nos centramos en la selección y parametrización de los operadores genéticos, cruzamiento y mutación, considerando las soluciones overlapping. Las soluciones overlapping ocurren cuando más de una solución en el espacio de decisiones se corresponde con la misma solución en el espacio de objetivos (figura 1). Esta situación amenaza la capacidad de exploración de los algoritmos evolutivos impidiéndoles tener una buena diversidad en su población. La influencia de las soluciones overlapping se intensifica en problemas combinatorios.

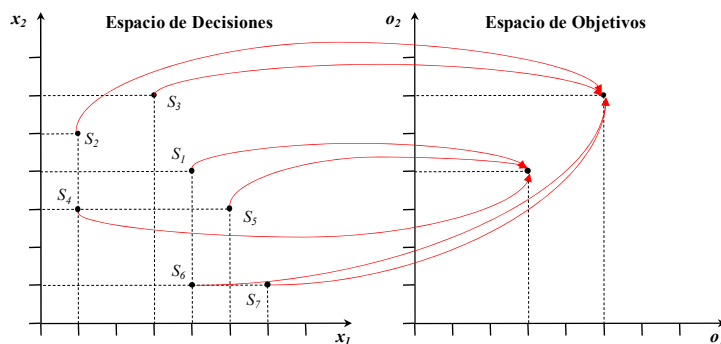


Fig. 1. Espacio de decisiones y espacio de objetivos.

4 Experimentaciones y resultados

Se experimentó con cuatro operadores de cruce (Cycle Crossover (CX), Modified-Cycle Crossover (CX2), Partially-Mapped Crossover (PMX) y Partially-Mapped Crossover 2

(PMX2)) y tres de mutación (Exchange, Insertion e Inversion). Para la probabilidad de cruce se consideraron los valores 0.80, 0.85 y 0.90, y para la de mutación 0.10, 0.15, 0.20, 0.25 y 0.30. La instancia que se tomó para el análisis fue de 50 trabajos y 10 máquinas (J50M10). Se realizaron 30 ejecuciones por cada ensayo. Se usó la selección por torneo y se tomó una población de 200 individuos. En la figura 2, se puede ver el número promedio de soluciones overlapping vs. las generaciones. Esto que se muestra fue generado para el makespan. Un análisis similar se realizó para el total tardiness.

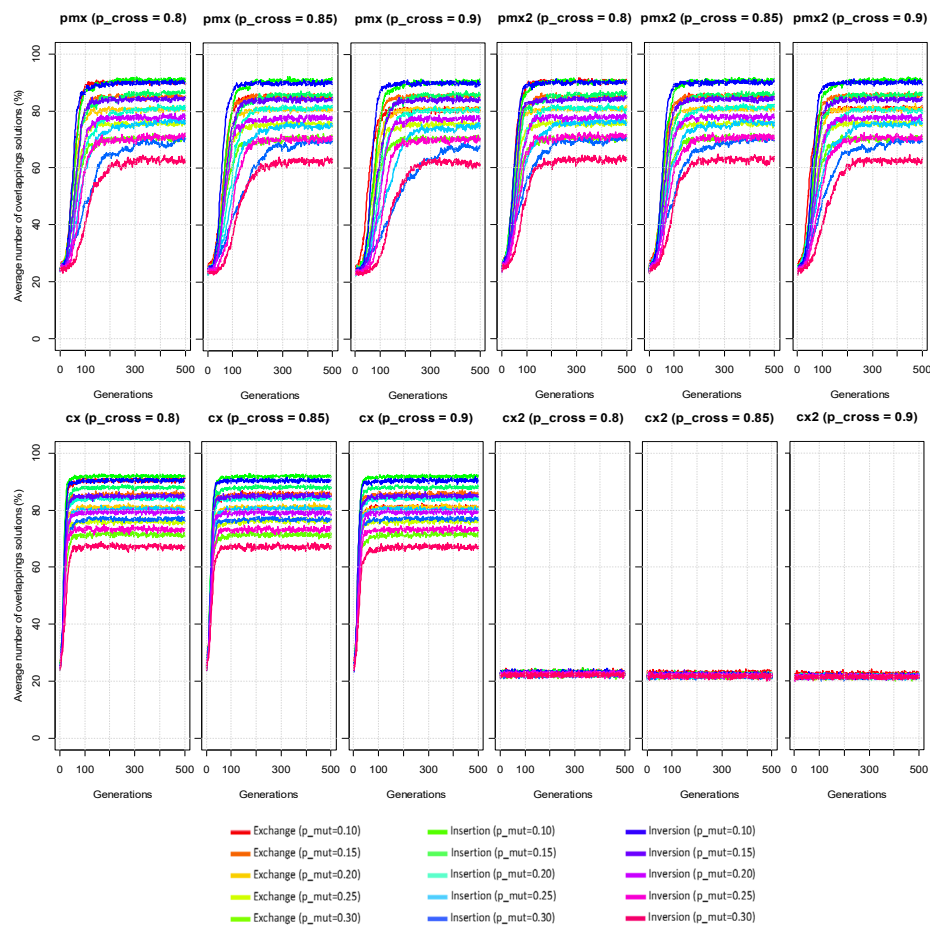


Fig. 2. Número promedio de soluciones overlapping vs. generaciones.

Se observa que el operador de cruce CX2 genera menos soluciones overlapping. También, se observa que la generación de soluciones overlapping está fuertemente relacionado con la probabilidad de mutación. En la figura 3, se concluye lo mismo a través de los diagramas boxplot. Todas estas observaciones fueron confirmadas con la aplicación del test de Kruskal-Wallis.

5 Conclusiones

En este trabajo se ha aplicado un algoritmo evolutivo al problema de flow-shop. Se analizaron diferentes operadores genéticos para evaluar su impacto en la diversidad, prestando

especial atención a las soluciones overlapping. Para el caso del makespan, la generación de soluciones overlapping se relaciona fuertemente al tipo de operador de cruce que se use y a la probabilidad de mutación. El operador CX2 fue el que generó menor número de soluciones overlapping. También, se concluyó que al aumentar la probabilidad de mutación el número de este tipo de soluciones disminuye. Actualmente, se están terminando el análisis para el total tardiness. Luego, se procederá a analizar la convergencia del algoritmo.

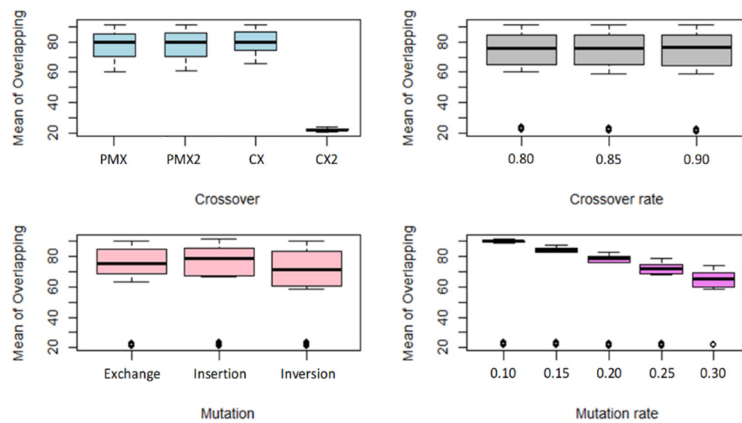


Fig. 3. Diagramas boxplot generados.

Acknowledgments. Los autores agradecen el soporte económico otorgado por la Consejería de Economía, Industria, Comercio y Conocimiento del Gobierno de Canarias a través de la subvención directa concedida a la ULPGC denominada "Apoyo a la actividad de I+D+i, Campus de Excelencia Internacional CEI CANARIAS-ULPGC. También, agradecen a la SGCyT de la Universidad Nacional del Sur por financiar parte de este trabajo a través de los proyectos PGI 24/ZJ50 y 24/J086.

Referencias

1. Umam, M.S., Mustafid, M., Suryono, S.: A hybrid genetic algorithm and tabu search for minimizing makespan in flow shop scheduling problem. *Journal of King Saud University, Computer and Information Sciences* **34**(9), 7459-7467 (2022).
2. Khatami, M., Salehipour, A., Cheng, T.C.E.: Flow-shop scheduling with exact delays to minimize makespan. *Computers & Industrial Engineering* **183**(1), 109456 (2023).
3. Koulamas, C.: The proportionate flow shop total tardiness problem. *European Journal of Operational Research* **284**(2), 439-444 (2020).
4. González, B., Rossit, D.A., Méndez, M., Frutos, M.: Objective space division-based hybrid evolutionary algorithm for handling overlapping solutions in combinatorial problems. *Mathematical Biosciences and Engineering* **19**(4), 3369-3401 (2022).
5. Zaied, A.N.H., Ismail, M.M., Mohamed, S.S.: Permutation flow shop scheduling problem with makespan criterion: literature review. *Journal of Theoretical and Applied Information Technology* **99**(4), 830-848 (2021).
6. Rossit, D.A., Tohmé, F., Frutos, M.: The non-permutation flow-shop scheduling problem: a literature review. *Omega* **77**(1), 143-153 (2018).
7. Benavides, A.J., Ritt, M.: Two simple and effective heuristics for minimizing the makespan in non-permutation flow shops. *Computers & Operations Research* **66**(1), 160-169 (2016).
8. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. *Machine Learning* **3**(1), 95-99 (1988).