

“Algoritmo Genético para resolver un problema de asignación de partes en celdas de manufactura”

Autor/Autores: José Alonso Reyes Samaniego (19984190), Rodrigo Alejandro Holguín Huari (20221466), Ariana Milagros Silva Flores (20220361)

Resumen - *El documento propone un algoritmo genético para asignar partes y máquinas en celdas de manufactura, optimizando recursos y minimizando excepciones. Utiliza casamiento uniforme y mutación para generar nuevas soluciones. Los resultados muestran eficiencia en problemas pequeños y medianos, aunque enfrenta desafíos en escalabilidad. Se sugiere explorar técnicas híbridas y considerar factores adicionales como tiempos de producción. Finalmente, se destacan implicancias éticas relacionadas con el empleo, sostenibilidad y privacidad de datos.*

1. Introducción

La manufactura celular es un proceso que organiza los trabajos y máquinas en grupos para poder mejorar la eficiencia, calidad y flexibilidad. Para ello, los productos se asignan a celdas que cumplen una parte del proceso.

Este proceso es relevante ya que ayuda a incrementar la calidad de la producción y maximizar la misma. Esto ayuda a las empresas a mantener la competitividad y adaptarse a las demandas del mercado.

Para poder optimizar este proceso, se ha propuesto la idea de utilizar algoritmos genéticos para la asignación de cada parte-máquina a una celda de manufactura correspondiente. El objetivo del problema es agrupar un conjunto de máquinas y partes en un número determinado de celdas para optimizar la utilización de recursos y minimizar los elementos excepcionales y vacíos.

2. Trabajos relacionados

1. “Genetic algorithm approach for solving a cell formation problem in cellular manufacturing”. El artículo presenta un algoritmo genético para resolver el problema de formación de celdas en manufactura celular, agrupando máquinas y partes en celdas de manera que se minimicen los **elementos excepcionales** (procesos fuera de las celdas asignadas) y los **vacíos** (espacios sin procesos dentro de las celdas). El modelo matemático es no lineal y NP-difícil, por lo que se utiliza un enfoque heurístico con operadores genéticos (cruza y mutación). El algoritmo es validado en 22 problemas benchmark, mostrando resultados superiores o comparables a métodos existentes como ZODIAC y GRAFICS. Este enfoque destaca por su eficiencia y su capacidad para manejar problemas de gran escala.

2. “Meta-heuristics in cellular manufacturing: A state-of-the-art review”. El documento es una revisión de metaheurísticas aplicadas a problemas de formación de celdas en manufactura celular, abordando enfoques como algoritmos genéticos, búsqueda tabú, recocido simulado y optimización por enjambre de partículas. Explica cómo estas técnicas han evolucionado para resolver problemas NP-completos, sus ventajas frente a métodos tradicionales y su capacidad para optimizar la asignación de máquinas y piezas en celdas. Se incluyen comparativas, estudios de caso y aplicaciones prácticas, resaltando la importancia de la hibridación y los desafíos futuros en el campo.

3. “Heuristic and meta-heuristic algorithms for solving medium and large scale sized cellular manufacturing system NP-hard problems: A comprehensive review”. El artículo revisa métodos heurísticos y metaheurísticos para resolver problemas NP-hard en sistemas de manufactura celular (CMS). Analiza algoritmos como genética, optimización por enjambre de partículas, recocido simulado y búsqueda tabú, comparándolos en eficiencia y eficacia. Identifica que los problemas incluyen formación de células, disposición de máquinas, asignación de recursos y costes. Concluye que los algoritmos híbridos y robustos son esenciales para abordar problemas de optimización múltiple en CMS. Finalmente, se resalta la necesidad de modelos multiobjetivo que incluyan costos de manejo y planificación de recursos para futuros desarrollos.

3. Metodología

■ Enfoque(s) propuesto:

- El problema posee de forma natural las siguientes variables:

P : que es el conjunto de partes.

M : conjunto de máquinas.

C : número de celdas.



r_{ij} : matriz de incidencia máquina-parte (1 si la parte i requiere de la máquina j , 0 en caso contrario).

$MIN ut_k$: utilización mínima de cada celda.

- Junto a ello, existe una serie de restricciones:
 - (1) Cada parte debe de estar asignada a una celda.
 - (2) Cada máquina debe de estar asignada a una celda.
 - (3) Cada celda debe tener un nivel mínimo de utilización ($MIN ut_k$).

El algoritmo sigue la estructura básica de un algoritmo genético: Primero se crea una población inicial. Para ello se utiliza un cromosoma que junta tanto las partes como las máquinas.

2 2 2 3 3 1 1 1 2 2 2 2 1 1 2 3 3 1 1 1

Imagen 1. Representación de cromosoma con 10 partes y 10 máquinas.

El valor presente en cada una de las partes/máquinas representa la celda en la que se encuentra. Para el caso representado en la imagen uno se ha utilizado un número total de 3 celdas, por lo que cada parte/máquina se encuentra en una de estas.

Luego de realizar la primera población (con un número determinado de casos 0000 para el tamaño de esta) se pasa al proceso de evaluación de fitness. En este, como se ha estipulado previamente, se busca minimizar los EE y los vacíos. Esto se puede realizar con la siguiente fórmula:

$$\sum_{i=1}^P \sum_{j=1}^M \sum_{k=1}^C (1 - 2r_{ij}) Z_{ik} Y_{jk}$$

Donde Z_{ik} es la relación entre las partes y las celdas y Y_{jk} la relación de las máquinas y las partes.

Después del fitness se realiza la ruleta, para poder obtener a los “padres” para los siguientes pasos del algoritmo genético.

Posterior a esto, se realiza tanto el proceso de casamiento como el de cruzamiento, cada uno de una forma específica:

- Casamiento uniforme: Por cada par aleatorio de padres elegidos, una proporción de genes elegidos de forma aleatoria se intercambian. (Ver Imagen 2)
- Operador de mutación: Se escoge un individuo de forma aleatoria de la población. Luego, un número r entre 1 y $M + P$ es elegido y se realiza la mutación correspondiente (alternando el valor mínimo y máximo del cromosoma en la posición r y sumando uno o restando uno de forma aleatoria en caso no sean el valor mínimo o máximo del cromosoma).

Estos métodos específicos proporcionan una mayor variedad de resultados y por ende una mayor oportunidad de encontrar la mejor solución de forma rápida y eficiente.

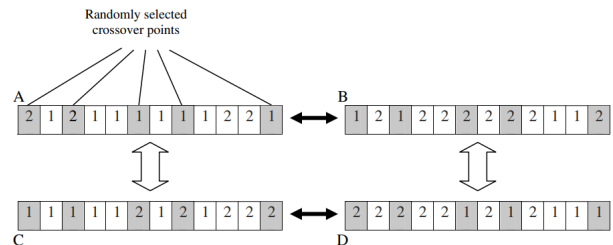


Imagen 2. Cruzamiento Uniforme. Recuperado de: <https://doi.org/10.1016/j.eswa.2008.07.054>

Finalmente, se crea la nueva población y se repiten los pasos anteriores hasta llegar al número de iteraciones solicitado al inicio del programa.

4. Experimentación y Resultados

■ Setup experimental:

- Se han utilizado varios datos experimentales, los cuales se presentan a continuación:

P	C	M	MIN_ut_k
7	2	5	0.65
8	2	6	0.55
20	3	8	0.6
15	3	10	0.7
15	7	12	0.5

Tabla 1. Datos experimentales.

- Una de las métricas principales es el tiempo de ejecución computacional. Para ello se debe de tomar en cuenta el tiempo que tardan otros algoritmos heurísticos y metaheurísticos y guiarse de ello para corroborar que el tiempo de este método sea óptimo.
- Otra métrica sería la aptitud. Los resultados deben de compararse con los resultados de otros algoritmos para corroborar la calidad del mismo.
- Para poder corroborar la productividad del algoritmo se utilizó una amplia variedad de datos, comenzando por tamaños pequeños de cromosomas hasta cada vez más grandes. A su vez, se trató de variar los tamaños de población para comprobar el manejo de memoria del mismo.

■ Resultados y Discusión:



- Se han analizado principalmente los tiempos de ejecución y se han obtenido los siguientes resultados: 3s, 4s, 20s, 19s y 48s.
- También se ha analizado el fitness de cada uno y se obtuvieron los siguientes resultados: -20, -32, -99, -129 y -182.

Podemos observar que conforme la cantidad de datos aumenta, los resultados obtenidos también son mayores, tanto para el fitness (tomando de referencia el valor absoluto) como para los tiempos de ejecución. Esto se debe a que al trabajar con iteraciones debe de realizar una mayor cantidad de pasos antes de poder continuar con el programa de forma eficiente. Es posible también decir que la población está “creciendo demasiado”, es decir, que la cantidad de datos es demasiada como para que la soporte el algoritmo, lo cual causa que tarde más en realizar la ejecución.

5. Conclusión

Se puede concluir que el algoritmo, si bien está funcionando de forma correcta, puede generar problemas cuando se utilizan datos grandes debido a que es demasiada información para que el programa la maneje. Sumado a ello, se observa una convergencia lenta hacia un valor en específico, ya que conforme se va probando con los datos se observa que estos cambian de forma variada.

6. Sugerencias de trabajos futuros

Para nuevos trabajos se recomienda buscar maneras en las cuales se deba tener en cuenta más condiciones para la asignación, como, por ejemplo, el tiempo que toma realizar una parte o si es que existe una mayor cantidad de rutas de fabricación de cada máquina. A su vez, también se puede ampliar el alcance del proyecto, implementando técnicas híbridas para el desarrollo de la solución.

7. Implicancias éticas

Una de las principales implicancias es el impacto en el empleo (debido a que trabajos que se realizaban antes por personas ahora serían realizados por una máquina).

Otra implicancia es la sostenibilidad ambiental, ya que al buscar la opción que sea más productiva (la que minimice mejor los espacios vacíos) no se toma en cuenta el impacto ambiental que puede generar a largo plazo.

Finalmente, una última implicancia importante es la de privacidad y uso de datos, ya que al requerir de una enorme cantidad de datos se puede poner en riesgo la privacidad y el uso indebido de datos por parte de terceros.

8. Link del repositorio del trabajo

<https://github.com/Puyuyel/Tarea-Academica-1IINF32/tree/main>

9. Declaración de contribución de cada integrante

- Ariana Silva: Redacción del Informe y del tema a desarrollar.
- Alonso Reyes: Desarrollo del código de la Tarea Académica.
- Rodrigo Holguin : Desarrollo del casamiento uniforme y grabación del video.

10. Referencias

Ejemplo:

- [1]. Mahdavi, I., Paydar, M. M., Solimanpur, M., & Heidarzade, A. (2009). Genetic algorithm approach for solving a cell formation problem in cellular manufacturing. *Expert Systems with Applications*, 36(4), 6598–6604. <https://doi.org/10.1016/j.eswa.2008.07.054>
- [2]. Ghosh, T., Sengupta, S., Chattopadhyay, M., & Dana, P. K. (2010). Meta-heuristics in cellular manufacturing: A state-of-the-art review. *International Journal of Industrial Engineering Computations*, 2(1), 87–122. <https://doi.org/10.5267/j.ijiec.2010.03.005>
- [3]. Kesavan, V., Kamalakannan, R., Sudhakarapandian, R., & Sivakumar, P. (2019). Heuristic and meta-heuristic algorithms for solving medium and large scale sized cellular manufacturing system NP-hard problems: A comprehensive review. *Materials Today: Proceedings*. <https://doi.org/10.1016/j.matpr.2019.05.363>