Centro de Investigación en Matemáticas, A.C.

Paisaje de búsqueda en el problema de planificación de producción tipo taller

Tesis que presenta

Juan Germán Caltzontzin Rabell

para obtener el Grado de

Maestro en Ciencias con Especialidad en Computación y Matemáticas Industriales

Director de Tesis

Dr. Carlos Segura González

Guanajuato, Gto. Julio de 2021

Resumen

En este trabajo se analiza el efecto de distintas metodologías de modificación del paisaje de búsqueda del problema de planificación de producción tipo taller (JSP por sus siglas en inglés). Existen una gran cantidad de tipos de problemas de planificación, el JSP es un tipo específico de problema que ha atraido considerable atención por su dificultad. En general estos problemas consisten en encontrar una planificación que minimice el tiempo requerido para completar ciertos procesos.

Aunque existen algoritmos exactos para encontrar la solución óptima JSP[5] el tiempo que requieren problemas grandes es demasiado largo como para que sean prácticos o inclusive factibles. La complejidad de este problema lo hizo un candidato ideal para utlizar métodos aproximados para conseguir una solución aceptable en un tiempo razonable. Actualmente los mejores resultados para las instancias de prueba más difíciles se han encontrado mediante metaheurísticas, en especfico la conocida como búsqueda tabú.

Las metaheurísticas son convenientes por su relativa simplicidad algorítmica frente a otros métodos sin embargo, el tiempo requerido para obtener los resultados del estado del arte ha comenzado a crecer también, por lo que es importante dar un paso atrás y replantear un poco la forma de aplicar estos métodos para evitar complicarlos de más.

En el campo de las metahurísticas hay varios conceptos fundamentales que tienen un gran impacto en su desempeño, en este trabajo se analizan tres de ellos que juntos componen lo que se conoce como paisaje de búsqueda el cual está estrechamente relacionado con la dificultad con el desempeño de las metaheurísticas de trayectoria.

Los conceptos analizados en este trabajo son : representación de las soluciones, definición de vecindad de una solución, función de aptitud o fitness. Para evaluar el impacto de cada uno de ellos se utliza una de las metaheurísticas de trayectoria más sencillas; la búsqueda local iterada.

Se tomaron las instancias dmu que son uno de los conjuntos de prueba más utilizados actualmente y que aun no ha sido resuelto por completo para hacer experimentos computacionales y se encontró que de las tres áreas en las que se plantearon modificaciones, la más importante fue la de la representación de soluciones.

Abstract

En este trabajo se analiza el efecto de distintas metodologías de modificación del paisaje de búsqueda del problema de planificación de producción tipo taller (JSSP por sus siglas en inglés) (En inglés)

Índice general

Capítulo 1

Introducción

Sumario

1.1. Antecedentes y Motivación

Actualmente los resultados del estado del arte para este problema se han obtenido mediante algoritmos meméticos con búsqueda tabú y con ejecuciones de entre 24 y 48 hrs paralelas.
La literatura reciente se ha centrado en hacer más eficiente la búsqueda tabú sin considerar el paisaje de búsqueda.

1.2. Objetivo

El objetivo de este trabajo es plantear modificaciones que permitan el uso de mecanismos más simples y rápidos para encontrar soluciones comparables al estado del arte. Dentro del diseño de las metaheurísticas se

1.3. Propuesta

1. Se propone utilizar la búsqueda local iterada (ILS por sus siglas en inglés) por ser un algoritmo simple y rápido.

2. Se propone una extensión a la vecindad N7

3. Se planteó una función de fitness que no solo toma en cuenta el makespan sino también los tiempos de finalización de todas las máquinas.

1.4. Contribuciones

1.5. Panorama General

Capítulo 2

Marco teórico

Sumario

2.1. Optimización

La optimización es una herramienta que nos ayuda a encontrar *la mejor* entre diferentes opciones elegibles. En nuestra vida diaria a menudo nos encontramos en este tipo de situaciones, por ejemplo al elegir entre diferentes rutas para llegar a algún lugar.

Formalmente, un problema de optimización consiste en hallar el mínimo o máximo de una función $f : X \to \mathbb{R}^n$ (llamada función objetivo) en un conjunto de soluciones X. De modo que el problema consiste en hallar:

$$\min_{x \in X} f(x) \tag{2.1}$$

Es importante mencionar que cualquier problema de maximización puede transformarse en un problema equivalente de minimiación con el reemplazo $f(x) \leftarrow -f(x)$, por lo que se considera solo el caso de minimización sin pérdida de generalidad.

2.2. Metaheurísticas

Es muy común que en nuestra cotidianidad nos enfrentemos a problemas tan difíciles o para los que tengamos tan poco tiempo de decisión que no podamos hacer un análisis riguroso, en estos casos es muy común que utilicemos algún método (posiblemente basado en la experiencia) que nos permita hallar una solución aceptable, por ejemplo, es común que reemplacemos el problema por uno más simple que sí podemos responder y cuya respuesta está relacionada con nuestro problema original.[1]

En el contexto de la optimización una metaheurística es una metodología de alto nivel que combina diferentes heurísticas y puede aplicarse para resolver de manera aproximada una gran cantidad de problemas. En la práctica existen numerosas metaheuristicas que pueden ser muy diferentes entre sí por lo que no hay un sistema de clasificación universalmente aceptado aunque se han propuesto diferentes criterios de clasificación [9] así como características como:

- De trayectoria vs discontinua. Una metaheurística de trayectoria consiste en, dada una solución inicial, mejorarla de manera iterativa mediante algún operador que «mueve» a la solución a través del espacio de búsqueda.

- basadas en población vs basadas en una sola solución. En las metaheurísticas basadas en población se mantiene un conjunto de soluciones candidatas.

- basadas en búsqueda local vs constructivas. Como se explicará más adelante, en la búsqueda local, el proceso de mejora implica la evaluación de soluciones muy parecidas a una solución inicial dada mientras que en las constructivas se crean nuevas soluciones de acuerdo a una heurística o algoritmo preestablecido.

- Con uso de memoria vs sin uso de memoria. El uso de memoria consiste en almacenar información que nos ayude a explorar el espacio de búsqueda.

2.2.1. Vecindad

La definicion de vecindad es crucial para las metaheurísticas de trayectoria y las basadas en una sola solución. Formalmente, una vecindad es un mapeo $N : X \leftarrow 2^X$ que le asigna a cada solución $x \in X$ un subconjunto de soluciones en X. Intuitivamente podemos pensar que es una

[1] No podemos predecir con certeza si lloverá durante el día pero sí podemos responder si el cielo está plagado de nubes oscuras

forma de definir a las soluciones que «rodean» a otra. Se dice que la solución y es un vecino de x si $y \in N(x)$.

A partir de la definición de vecindad podemos también definir un operador de movimiento cuyo efecto al aplicarlo a una solución sea transformarla en una que pertenezca a su vecindad, i.e. este operador selecciona a un vecino de la solución inicial.

2.2.2. Función de aptitud o fitness

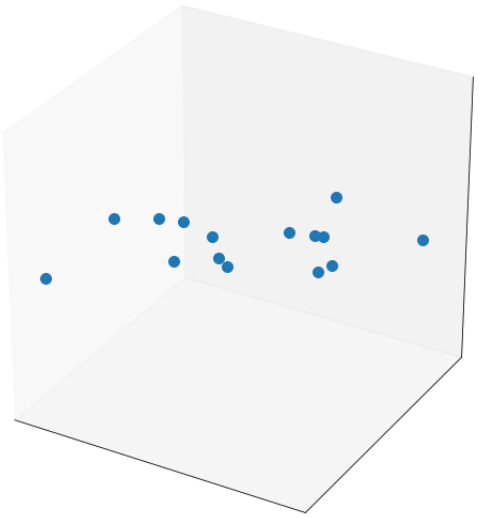
Aunque para un problema de optimización ya se tiene definida una función objetivo que se quiere minimizar, no siempre tenderemos el mejor desempeño de las metaheurísticas con solo esta función por lo que resulta benéfico plantear una nueva función a minimizar con la que tengamos mejor desempeño. Por ejemplo puede suceder que aunque dos soluciones tengan asociado el mismo valor de la función objetivo, una de ellas sea un mejor punto de partida para una metaheuristica de trayectoria.

Esta función debe asociar a cada solución un elemento de un espacio donde esté definido un ordenamiento total. En esencia esta función define un operador de comparación entre soluciones.

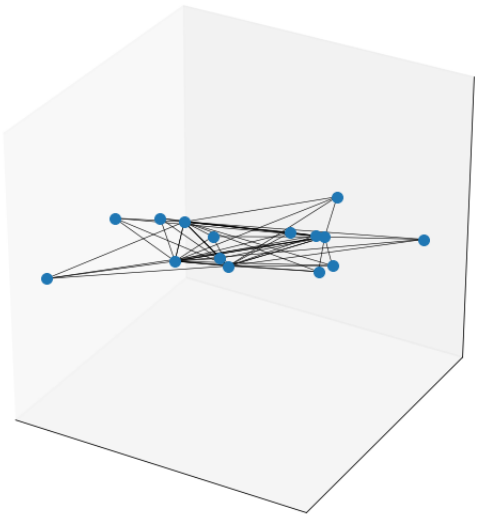
2.2.3. Paisaje de búsqueda

Una vez que tenemos el espacio de búsqueda y operadores de cambio para generar nuevas soluciones a partir de otras, se define el espacio de búsqueda como un grafo dirigido en el que los nodos son las soluciones al problema y una solución x está conectada a otra y si podemos generar a y aplicando los operadores de cambio a x.

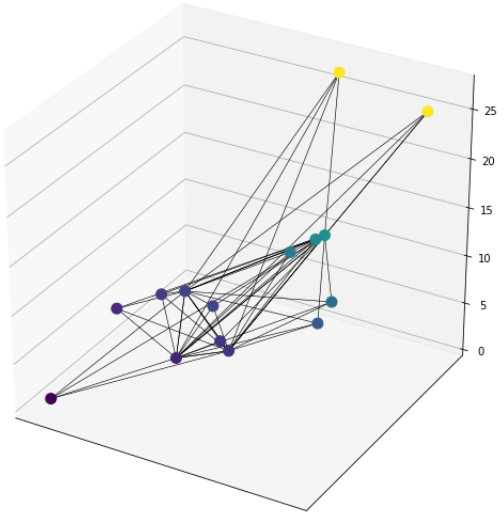
Podemos asociar a cada solución en el espacio un valor de aptitud o fitness que mide la calidad de dicha solución. La adición de esta función de aptitud al espacio de búsqueda genera al paisaje de búsqueda.



(a) Soluciones



(b) Relaciones inducidas por los operadores de cambio



(c) Adición de la función de fitness

Figura 2.1: Creación del paisaje de búsqueda

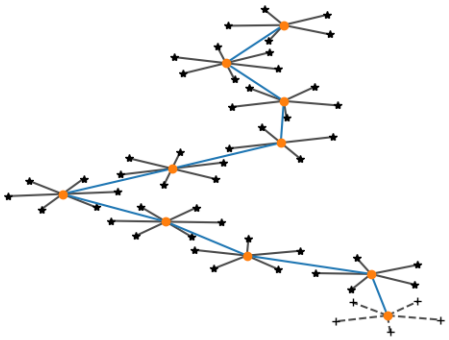


Figura 2.2: Ilustración de una metaheurística de trayectoria.
En cada paso se selecciona un vecino mejor que la solución actual hasta que se cumple algún criterio de paro

2.2.4. Búsqueda Tabú

Esta metaheurística de trayectoria es la que actualmente ha obtenido mejores resultados por lo que se describe a continuación:

2.3. Problema del planificación de producción tipo taller (JSSP)

Una instancia del JSSP consiste en n trabajos diferentes constituidos cada uno por m operaciones que deben procesarse por un tiempo determinado en m máquinas en una secuencia predeterminada.
El objetivo es hallar la planificación que minimiza el tiempo que toma terminar todos los trabajos dado que cada máquina puede procesar solo un trabajo a la vez.

Una planificación consiste en asignar tiempos de inicio y fin a cada operación respetando el orden requerido para cada trabajo. El tiempo que toma terminar todos los trabajos se conoce como makespan y la secuencia de trabajos que toma el mayor tiempo en completarse se conoce como ruta crítica. La ruta crítica puede verse como una serie de bloques críticos que consisten en las secuencias de operaciones de la ruta crítica que se ejecutan de forma adyacente en la misma máquina. Una planificación puede tener una o varias rutas críticas.

En general se considera que el tiempo requerido para procesar cada operación puede expresarse como un entero.

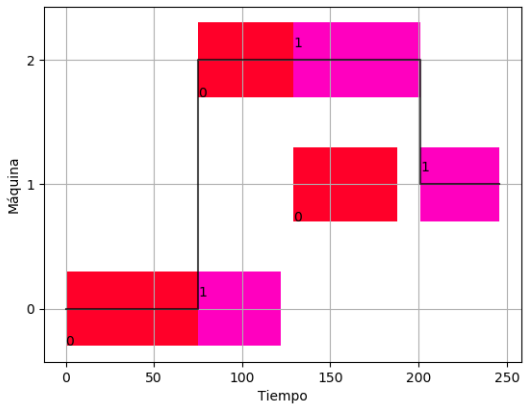
Ejemplo

Se muestra un ejemplo de una instancia con 3 máquinas y 2 trabajos.

Tabla 2.1: Instancia simple con 3 maquinas y 2 trabajos

Trabajo	Secuencia de procesamiento (máquina, tiempo)		
0	0, 75	2, 54	1, 59
1	0, 47	2, 72	1, 45

La siguiente es una posible planificación para la instancia de ejemplo, visualizada mediante un diagrama de gantt. En negro se marca los trabajos que conforman la ruta crítica.



2.4. Representación de planificaciones

Existen varias formas de representar las planificaciones, en este trabajo se utilizaron dos: el grafo disyuntivo y las reglas de prioridad.

Modelo de grafo disyuntivo

En este modelo las planificaciones se representan con un grafo dirigido $G = (V, A, E)$ en el que V es un conjunto de nodos que representa las operaciones, las aristas A representan la secuencia que deben seguir las operaciones dentro de un mismo trabajo y E es otro conjunto de aristas que indica el orden de procesamiento en cada una de las máquinas. Es importante mencionar que con este modelo podemos representar planificaciones no factibles, esto se da cuando el grafo G contiene un ciclo.

Reglas de prioridad

En esta representación una planificación se construye al aplicar un proceso de simulación en el que para cada maquina se construye una cola con las operaciones cuyas dependencias ya han sido procesadas. Inicialmente se tienen en las colas solo las operaciones iniciales de cada trabajo. Una vez que se tiene esto se utiliza una regla de prioridad para elegir qué operación debe planificarse en qué máquina. Se actualizan las colas para las máquinas que lo requieran y se continua con este proceso hasta completar la planificación (vaciar las colas)

Tipos de planificaciones

Dentro de el conjunto de planificaciones factibles se pueden distinguir dos subconjuntos de interés para el presente trabajo: el conjunto de las planificaciones óptimas que está conformado por las planificaciones con el menor makespan posible y el conjunto de las planificaciones activas. Estas últimas se definen como las planificaciones en las que no es posible disminuir el tiempo de inicio de ninguna operación sin aumentar el tiempo de inicio de otra. Es conocido que las planificaciones óptimas representan un subconjunto de las activas.

2.5. Metaheurísticas aplicadas al JSP

Dada la complejidad del JSP se han aplicado y desarrollado multiples metaheutrísticas para hallar soluciones razonables. Actualmente las más exitosas son algoritmos meméticos que combinan técnicas poblacionales con metaheurísticas de trayectoria. La metaheurística más exitosa y ampliamente usada se conoce como búsqueda Tabú

2.6. Vecindades previamente propuestas

Se han propuesto varias estructuras de vecindad al JSP, a continuación se describen las más importantes a la fecha:

- N1 [3] Consiste en considerar todas las soluciones que se crean al intercambiar cualquier par de operaciones adyacentes que pertenecen a un bloque crítico. Esta vecindad es muy grande y considera muchos cambios que no mejoran el makespan.

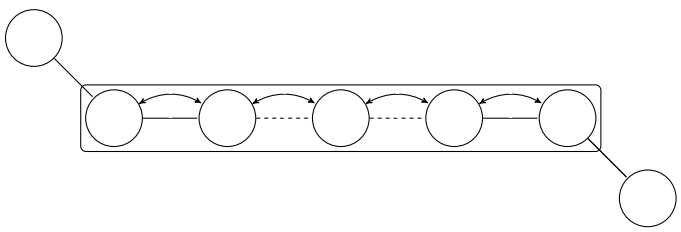


Figura 2.3: Movimientos de la vecindad N1

- N4 [6] Esta vecindad se propuso como un refinamiento y extensión de la vecindad N1 y toma como base el concepto de bloque crítico. Consiste en llevar operaciones internas del bloque crítico al inicio o final.

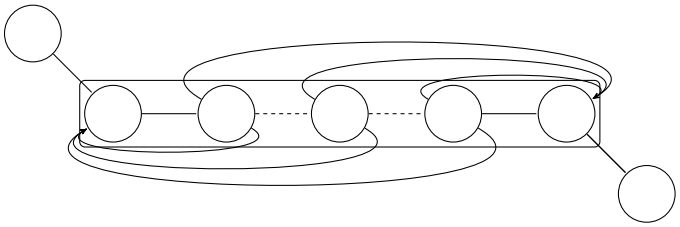


Figura 2.4: Movimientos de la vecindad N4

- N5 [7] Consiste en intercambiar solo las operaciones adyacentes a la final o inicial de un bloque crítico.

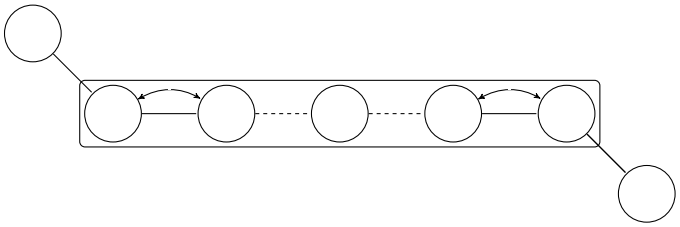


Figura 2.5: Movimientos de la vecindad N5

- N6 [2] Los autores utilizan varios teoremas para identificar pares (u, v) de operaciones dentro de un bloque crítico que puedan llevar a mejorar la solución y a su vez identificar si se tiene que mover a u justo antes de v(forward) o bien a v justo antes de u (backward).

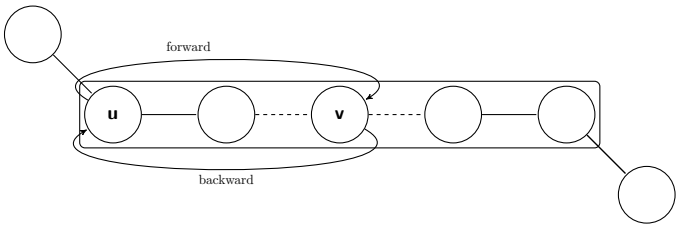


Figura 2.6: Los dos tipos de movimientos para un par (u, v)

- N7 [10] Esta vecindad se plantea como una extensión de la N6 en la cual no solo se lleavan operaciones internas del bloque al inicio o final sino que las operaciones inicial y final también se llevan al interior del bloque.

Capítulo 3

Propuestas

3.1. Extensión a vecindad N7

Como punto de partida se planteó agregar movimientos a la vecindad N7 con la cual se han obtenido los resultados del estado del arte. Los movimientos que plantea esta vecindad solo tienen que ver con pares de operaciones en la ruta crítica por lo que una extensión sencilla consiste en considerar movimientos de operaciones que pueden no pertenecer a la ruta crítica. Es importante resaltar que si no se planteara también una función de fitness que no tome solo en cuenta el makespan estos movimientos nunca llevarían a una mejora [3]. Los movimientos planteados se basan en observar que en alguna solución encontrada por una búsqueda local para cada maquina pueden existir periodos de tiempo en la que está inactiva pero existe una operación que podría comenzar a procesarse en este periodo y que se procesa después.

3.2. Vecindad basada en soluciones activas

3.3. Grafo disyuntivo

Inicialmente se trabajó con la representación del grafo disyuntivo [1]. Una solución factible se representa como un grafo dirigido acíclico en el que las aristas marcan el orden en el que se procesan las operaciones dentro de las máquinas. Esta representación se ha usado ampliamente en otros trabajos, siendo particularmente útil en aquellos que plantean métodos exactos de solución basados en enumeración completa[4].

Formalmente en una instancia del JSP se representa cada operación como un nodo, se agregan dos nodos de control que sirven como el nodo inicial (del que dependen todos los trabajos) y final (que depdende de todos los trabajos), las restricciones de precedencia dentro de cada trabajo se representan como aristas dirigidas fijas y las operaciones que deben procesarse en una misma máquina se unen mediante aristas disyuntivas, una solución factible o planificación se obtiene al elegir la dirección para cada arista disyuntiva de modo que no se generen ciclos.

3.4. Llaves aleatorias

La representación propuesta se basa en asignar a cada operación una número real entre 0 y 1 el cual sirve para definir un orden entre operaciones en una misma máquina mediante un proceso de decodificación un planteamiento similar puede encontrarse en [**?**].

Para decodificar la solución a partir de las llaves para cada operación se utiliza el algoritmo de Giffler & Thompson [8] con el cual se generan soluciones activas.

Capítulo 4

Validación experimental

Capítulo 5

Conclusiones y Trabajos a Futuro

Conclusiones

Bibliografía

[1] Balas, E. Machine sequencing via disjunctive graphs: an implicit enumeration algorithm. *Operations research 17*, 6 (1969), 941–957.

[2] Balas, E., and Vazacopoulos, A. Guided local search with shifting bottleneck for job shop scheduling. *Management Science 44*, 2 (1998), 262–275.

[3] Błażewicz, J., Domschke, W., and Pesch, E. The job shop scheduling problem: Conventional and new solution techniques. *European journal of operational research 93*, 1 (1996), 1–33.

[4] Brucker, P. *Due-Date Scheduling*. 2001.

[5] Brucker, P., Jurisch, B., and Sievers, B. A branch and bound algorithm for the job-shop scheduling problem. *Discrete Applied Mathematics 49*, 1-3 (1994), 107–127.

[6] Dell'Amico, M., and Trubian, M. Applying tabu search to the job-shop scheduling problem. *Annals of Operations research 41*, 3 (1993), 231–252.

[7] Eugeniusz Nowicki, C. S., and To. A Fast Taboo Search Algorithm for the Job Shop Problem. *Manage. Sci. 42*, 6 (2003), 797–813.

[8] Giffler, B., and Thompson, G. L. Algorithms for Solving Production-Scheduling Problems, 1960.

[9] Stegherr, H., Heider, M., and Hähner, J. Classifying Metaheuristics: Towards a unified multi-level classification system. *Natural Computing 0* (2020).

[10] Zhang, C. Y., Li, P. G., Guan, Z. L., and Rao, Y. Q. A tabu search algorithm with a new neighborhood structure for the job shop scheduling problem. *Computers and Operations Research 34*, 11 (2007), 3229–3242.