Inteligencia Artificial

Estado del Arte: Problema Car Sequencing Problem

Ian Rossi A.

June 8, 2024

Evaluación

Resumen (5%):	_____
Introducción (5%):	_____
Definición del Problema (10%):	_____
Estado del Arte (35%):	_____
Modelo Matemático (20%):	_____
Conclusiones (20%):	_____
Bibliografía (5%):	_____
Nota Final (100%):	_____

Abstract

Estado del Arte para Car Sequencing Problem, haciendo una descripción general del problema, su modelamiento, soluciones estudiadas anteriormente y la posibilidad de usar un algoritmo Greedy basado en Simulated Annealing para proponer una solución alternativa.

1 Introducción

El propósito de este artículo preliminar es presentar el problema base de manera sencilla para determinar el camino correcto a tomar con el fin de desarrollar una nueva solución mediante una versión Greedy del algoritmo Simulated Annealing. Primero se define a grandes rasgos el problema, junto con sus variables, restricciones, espacio de búsqueda y, más importante aún, su función objetivo, lo que se quiere maximizar o minimizar. El Car Sequencing Problem, referido de aquí en adelante como CSP, fue introducido por primera vez por Parrello, Kabat, and Wos (1986)[16] como una variación del Job Shop Problem, trata de modelar una linea de ensamblaje en la cual se producirán varios automóviles donde cada uno de estos se personaliza con un conjunto específico de opciones, buscando una secuencia que satisfaga las condiciones. Una forma fácil de entender la importancia de estudiar este problema es notar que los métodos de resolución aplican no solo para este caso particular acerca de automóviles, sino para cualquier tipo de linea de ensamblaje.

2 Definición del Problema

La línea atraviesa por varias estaciones de trabajo para la instalación de estas opciones. Los vehículos no se pueden colocar al azar ya que cada una de estas estaciones de trabajo tiene una capacidad limitada y necesitan tiempo para configurar las opciones mientras la línea de montaje se mueve. Estas restricciones de capacidad se formalizan utilizando restricciones de la forma m por N, lo que indica que la estación de trabajo puede instalar la opción en m por cada grupo de N automóviles. El problema consiste en determinar una secuencia de los automóviles en la línea que satisfaga las restricciones de demanda para cada conjunto de opciones, de aquí en adelante referido como clase y las restricciones de capacidad para cada estación de trabajo.

Como se puede apreciar, el problema no involucra optimización, ya que no hay función objetivo que busque maximizar o minimizar cierto parámetro, pero tomando el hecho de que no necesariamente existen soluciones que satisfagan todas las restricciones [16, 9], se puede elegir entre las no óptimas minimizando la cantidad de restricciones no cumplidas. De manera conceptual, esto se puede entender como que el movimiento de la linea de producción se pause para que la estación de trabajo que está trabajando más allá de su capacidad se pueda poner a la par. Para poder aplicar esto al modelo, se puede definir el costo a minimizar como la suma de la cantidad de subsecuencias de tamaño q_i, que tienen un número de vehículos que requieren la opción o_i, mayor a la capacidad p_i de la estación de trabajo correspondiente, para toda opción. Esto puede parecer confuso, pero es fácil entenderlo como la cantidad de veces que una máquina se vería sobrepasada en su capacidad, teniendo que detener la línea.

El CSP es una variación del Job-Shop Scheduling Problem (JSSP), el cual es un problema que se sabe es NP-Hard [12], por lo que también se puede decir que CSP es NP-Hard [6]. Esto es muy útil ya que mediante Transformación Polinomial es posible hacer uso de datos y estudios provenientes de otros problemas para este informe.

Variantes conocidas y o estudiadas anteriormente de este problema:

- Car Sequencing Problemwith Buffer (CSPwB) [21]:

  Introduce un buffer a la linea de ensamblaje para crear un balance entre la salida de una estación de trabajo y la entrada de otra. Existen distintos tipos de buffer, entre ellos: Pull-out buffer, Ring buffer, AS/RS, Parallel buffer.

- Robust Car Sequencing Problem(RCSP) [10]:

  Esta versión del problema se basa en la filosofía robusta de producción, que necesita ser usada sin modificación una vez iniciada, incluso si elementos de la linea de ensamblaje fallan, básicamente no se puede detener.

- Mixed-Model Assembly Line Sequencing Problem (MMALSP) [14, 2]:

  Esta versión del problema involucra múltiples modelos de autos, cada uno con su conjunto de clases particular. Estos modelos comparten algunas estaciones de trabajo dentro de su proceso y al igual que el problema base, busca una secuencia ideal.

3 Estado del Arte

El asignar trabajos a máquinas o procesadores, se estudia como *Deterministic Scheduling Theory* desde al menos la segunda mitad del siglo pasado [11, 17], referido entonces como *Line Balancing Problem*, el cual hacía referencia a una línea de producción que solo contiene una clase de bien a ser fabricado. Esto evolucionó a la notación moderna usada para el Job Shop Problem, introducida por *Ronald Graham, Eugene Lawler, Jan Karel Lenstra and Alexander Rinnooy Kan*[8] durante la década del 70. Luego, como fue mencionado anteriormente, el problema CSP como tal se introdujo en los años 80 [16]. Desde este punto se ha seguido estudiando, particularmente por el hecho de que los espacios de búsqueda pueden llegar a ser muy grandes[16, 9]. El hecho

de que el CSP sea una variación de otro problema, crea además el hecho de que no exista una formulación o notación estándar para éste. Se pueden revisar avances, modelos anteriormente formulados, y resultados de estos en [9]. Las técnicas utilizadas anteriormente incluyen optimización por colonia de hormigas [18], búsqueda del vecino mas cercano [4], algoritmo genético [20], simulated annealing [3], branch and bound [7] y programación en Prolog [5, 1]. Muchas de las investigaciones en cuanto al CSP han sido posibles gracias a *CSPLib*[15, 13], una librería de problemas de prueba creada con el propósito de estimular el estudio de esta área. Como fue mencionado en la sección anterior, en muchos casos el CSP se considera un *Constraint Satisfaction Problem* (referido de aqui en adelante como *Limitation Satisfaction Problem* o LSP para evitar confusión), es decir no existe una función objetivo como tal que sea parte del problema de manera absolutamente consistente. Una de las que usualmente se agrega al problema es la de minimizar costos, este puede ser interpretado como el tiempo total de trabajo.

4 Modelo Matemático

Se utilizará como base el modelo matemático establecido por la *Société française de Recherche Opérationnelle et Aide à la Décision* (ROADEF) para su desafío 2005 [19], que trata de entregar una algoritmo de solución para ciertas instancias del CSP. Cada instancia del problema se entrega como una tupla con la siguiente forma (V, O, p, q, r), donde:

- $V = v_1, v_2, ..., v_n$ es el conjunto de vehículos o clases a ser producidos.
- $O = o_1, o_2, ..., o_m$ es el conjunto de opciones que pueden o no ser elegidas para cada clase.
- $p_i : O \to \mathbb{N}$ y $q_i : O \to \mathbb{N}$ son la capacidad relativa y absoluta respectivamente de la estación de trabajo asociada a la opción $o_i \in O$.
- $r_i j : V \times O \to 0, 1$ corresponde a la matriz que indica si una opción o_i debe ser instalada en un vehículo v_j.

Dos vehículos pueden compartir su configuración de opciones a instalar, en este caso ambos pertenecen a la misma clase. V puede ser dividido en j subconjuntos que definen las clases existentes, de manera que $V = V_1 \cup V_2 \cup ...V_j$, con todos los miembros de un mismo subconjunto comparten configuración.

Cada estación de trabajo asociada a un o_i tiene una capacidad máxima determinada por $\frac{p_i}{q_i}$, la cual indica que puede instalar la opción o_i en p_i de cada q_i vehículos.

Para poder representar dentro del modelo el concepto del costo o tiempo extra que se requiere cuando una estación se ve sobrepasada, se crea la constante T_i que indica el tiempo que se tarda en instalar la opción asociada.

Todo lo anterior dicho queda aquí formalizado:

4.1 Constantes

- $V = \{v_1, v_2, ..., v_n\}$ es el conjunto de tipos de vehículos o clases a ser producidos.
- $O = \{o_1, o_2, ..., o_m\}$ es el conjunto de opciones que pueden o no ser elegidas para cada clase.
- $p_i : O \to \mathbb{N}$ y $q_i : O \to \mathbb{N}$ son la capacidad relativa y absoluta respectivamente de la estación de trabajo asociada a la opción $o_i \in O$.
- $r_{ij} : V \times O \to \{0, 1\}$ corresponde a la matriz que indica si una opción o_i debe ser instalada en un vehículo de clase v_j.
- $d_j : V \to \mathbb{N}$ es la demanda de una clase j.

- $N \in \mathbb{N}$ es la cantidad total de vehículos presentes en la línea para una instancia dada del problema.
- $T_i : O \to \mathbb{R}$ el tiempo que se tarda en instalar la opción o_i.

4.2 Variables

- $X_i = \{1, 2, ..., j\}$ corresponde a la clase del i-ésimo vehículo en la secuencia.
- $D_i^j = \{0, 1\}$ toma el valor 1 si es que una opción j debe ser instalada en el i-ésimo vehículo, toma 0 en caso contrario.

4.3 Restricciones

- Establece que no existan subsecuencias s de tamaño q_i que sobrepasen a p_i para toda opción.

$$\sum_{i \in O} \sum_{s \in X} \sum_{j \in s} r_{ij} - q_i > 0, \forall o_i \in O \tag{1}$$

- La demanda de cada clase se debe cumplir siempre

$$d_j = \sum_{j \in V} \sum_{i \in O} r_{ij}, \forall j \in V \tag{2}$$

4.4 Función Objetivo

Se busca minimizar la cantidad de tiempo invertido en completar una instancia dada

$$min \left(\sum_{j=1}^{|V|} \sum_{i=1}^{|O|} T_i \cdot r_{ij} \right)$$

4.5 Espacio de Búsqueda

El espacio de búsqueda asociado a este modelo está dado por las permutaciones de las diferentes configuraciones posibles, es decir que para un conjunto V compuesto por k clases, el espacio de búsqueda es:

$$\frac{|V|!}{|V_1|! \cdot |V_2|! \cdot ... \cdot |V_k|!}$$

5 Conclusiones

Sacar conclusiones generales sobre este problema resulta ser un desafío no menor, esto se debe a que no esta muy bien definido en un punto general, cada grupo que intenta desarrollar una solución crea una formulación bastante distinta a cualquier otra, hay algunos que lo toman como un LSP simple y otros como un problema de optimización. Aquellos lo toman como un problema de optimización tienden a encontrarse rápidamente con que el espacio de búsqueda y por consecuencia el tiempo requerido en encontrar una solución deja de ser viable. Teniendo en cuenta que es un problema NP-Hard, no hay mucho que se pueda hacer en cuanto a esto por ahora, solo mejorar las técnicas que ya se usan. Por ejemplo, en el *ROADEF'2005 Challenge* mencionado en la sección del modelo matemático, el equipo ganador venció a los demás usando un algoritmo de búsqueda local, solo siendo mucho más rápido y eficiente que los demás. Se propone dirigir el trabajo futuro en desarrollar heurísticas para agilizar el desarrollo de soluciones pero usando técnicas clásicas.

References

[1] Abderrahmane Aggoun, Mehmet Dincbas, Alexander Herold, Helmut Simonis, and Pascal Van Hentenryck. The chip system. Technical report, Technical Report TR-LP-24, ECRC, Munich, Germany, 1987.

[2] Onur Serkan Akgündüz and Semra Tunalı. An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem. *International Journal of Production Research*, 48(17):5157–5179, 2010.

[3] Olivier Briant, Denis Naddef, and Grégory Mounié. Greedy approach and multi-criteria simulated annealing for the car sequencing problem. *European Journal of Operational Research*, 191(3):993–1003, 2008.

[4] Andrew Davenport, Edward Tsang, Chang Wang, and Kangmin Zhu. Genet: A connectionist architecture for solving constraint satisfaction problems by iterative improvement. *Proceedings of the National Conference on Artificial Intelligence*, 1, 08 1994.

[5] Mehmet Dincbas, Helmut Simonis, and Pascal Van Hentenryck. Solving the car sequencing problem in constraint logic programming. pages 290–295, 08 1988.

[6] Bertrand Estellon and Frédéric Gardi. Car sequencing is np-hard: A short proof. *Journal of the Operational Research Society*, 64, 10 2013.

[7] Malte Fliedner and Nils Boysen. Solving the car sequencing problem via branch and bound. *European Journal of Operational Research*, 191(3):1023–1042, 2008.

[8] R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G.Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In P.L. Hammer, E.L. Johnson, and B.H. Korte, editors, *Discrete Optimization II*, volume 5 of *Annals of Discrete Mathematics*, pages 287–326. Elsevier, 1979.

[9] Marc Gravel, Caroline Gagné, and Wilson Price. Review and comparison of three methods for the solution of the car sequencing problem. *Journal of the Operational Research Society*, 56, 03 2005.

[10] Andreas Hottenrott, Leon Waidner, and Martin Grunow. Robust car sequencing for automotive assembly. *European Journal of Operational Research*, 291(3):983–994, 2021.

[11] James R. Jackson. A computing procedure for a line balancing problem. *Management Science*, 2(3):261–271, 1956.

[12] A.S. Jain and S. Meeran. Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research*, 113(2):390–434, 1999.

[13] Christopher Jefferson, Ian Miguel, Brahim Hnich, Toby Walsh, and Ian P. Gent. CSPLib: A problem library for constraints, 1999.

[14] H. Mosadegh, S.M.T. Fatemi Ghomi, and G.A. Süer. Stochastic mixed-model assembly line sequencing problem: Mathematical modeling and q-learning based simulated annealing hyper-heuristics. *European Journal of Operational Research*, 282(2):530–544, 2020.

[15] Gent I. P. and Walsh T. Csplib. 1999.

[16] Parrello, B.D., Kabat, W.C., and Wos L. Job-shop scheduling using automated reasoning: A case study of the car-sequencing problem. *Computational Intelligence*, pages 1–42, 1986.

[17] M. E. Salveson. The Assembly-Line Balancing Problem. *Transactions of the American Society of Mechanical Engineers*, 77(6):939–947, 07 2022.

[18] C. Solnon. Ants can solve constraint satisfaction problems. *IEEE Transactions on Evolutionary Computation*, 6(4):347–357, Aug 2002.

[19] Christine Solnon, Van Dat Cung, Alain Nguyen, and Christian Artigues. The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the roadef'2005 challenge problem. *European Journal of Operational Research*, 191(3):912–927, 2008.

[20] Terry Warwick and Edward Tsang. Tackling car sequencing problems using a generic genetic algorithm. *Evolutionary Computation - EC*, 3:267–298, 09 1995.

[21] Haida Zhang and Wensi Ding. A decomposition algorithm for dynamic car sequencing problems with buffers. *Applied Sciences*, 13(12), 2023.