

INCORPORANDO BÚSQUEDA LOCAL A UN ALGORITMO ACO PARA EL PROBLEMA DE SCHEDULING DE TARDANZA PONDERADA

Lasso M., de San Pedro M.
Laboratorio de Tecnologías Emergentes (LabTEm)
Proyecto UNPA-29/B084/1¹
Unidad Académica Caleta Olivia - Universidad Nacional de La Patagonia Austral
(9011) Caleta Olivia – Santa Cruz - Argentina
e-mail: {mlasso, edesanpedro}@uaco.unpa.edu.ar

Leguizamón, G.
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Departamento de Informática
Universidad Nacional de San Luis
(5700) San Luis – Argentina
e-mail: legui@unsl.edu.ar

RESUMEN

La meta-heurística ACO está inspirada en el comportamiento de las hormigas reales; se caracteriza por ser un método de búsqueda distribuida, estocástica y basada en la comunicación indirecta de una colonia artificial de hormigas, transmitida por trayectos artificiales de feromona. Estos trayectos sirven como información usada por las hormigas para construir probabilísticamente soluciones al problema bajo consideración. Las hormigas modifican los trayectos de feromona durante la ejecución del algoritmo para reflejar su experiencia de búsqueda.

En la mayoría de las metaheurísticas aplicadas a problemas de planificación, se han incorporado diferentes procesos de búsqueda local para mejorar la calidad de las soluciones.

En este informe se presentan dos alternativas aplicadas al problema de Tardanza Total Ponderada en entornos de máquina única para comparar bondades de uno sobre otro, y establecer ventajas y desventajas de aplicar uno u otro algoritmo.

1 INTRODUCCIÓN

Las tareas que deben entregarse en un sistema de producción, usualmente son ponderadas de acuerdo a la relevancia y requerimientos del cliente. En el intento de brindar mayor satisfacción al usuario, las tendencias en manufactura están enfocadas a la producción de políticas que enfatizan la mínima tardanza ponderada.

Los problemas de *scheduling* para máquina única son de interés para entornos de máquina más complejos. Es importante tener heurísticas que brinden buenas planificaciones con un esfuerzo computacional razonable debido a que *Branch & Bound* y otros métodos basados en enumeración parcial consumen tiempo extremadamente alto aún con 20 tareas.

Una meta heurística es un conjunto de conceptos algorítmicos que pueden ser usados para definir métodos heurísticos aplicables para un gran conjunto de problemas diferentes. En otras palabras,

¹ El Grupo de Investigación cuenta con el apoyo de la Universidad Nacional de La Patagonia Austral.

una meta heurística puede ser vista como una estructura algorítmica general que puede ser aplicada a diferentes problemas de optimización con relativamente pocas modificaciones para adaptarlos a un problema específico. Todas tienen en común que intentan evitar la generación de soluciones de pobre calidad introduciendo mecanismos generales que extiendan el problema específico; las diferencias tienen que ver con las técnicas empleadas para evitar que se estancan en soluciones sub-óptimas y el tipo de trayectoria seguida en el espacio de cualquiera de las soluciones, parcial o total.

Existen diferentes meta-heurísticas [8], con diferencias entre ellas; una de estas diferencias, tiene que ver con la técnica de construcción de las soluciones: unas están basadas en la evolución de la población, como es el caso de los Algoritmos Evolutivos, y otras utilizan un método de construcción iterativa como la meta-heurística ACO (*Ant Colony Optimization*). Aunque las meta-heurísticas constructivas y basadas en la población pueden ser usadas sin recurrir a la búsqueda local, muchas veces su rendimiento puede ser mejorado enormemente si ésta se incluye.

La meta-heurística ACO se ha introducido como un nuevo paradigma de optimización que está inspirado por el rastro siguiendo el comportamiento de las colonias de hormigas reales [4]. Esta meta-heurística se ha implementado con resultados muy promisorios para el problema bien conocido del TSP (*Traveling Salesman Problem*) [3] y [11], y está actualmente entre los mejores algoritmos disponibles para otros problemas como el Problema de Asignación Cuadrática [6], [7] y [10], el Problema de Ordenamiento Secuencial [6], Problemas de Ruteo de Vehículos [5], y problemas de ruteo en entornos dinámicos [2].

En los algoritmos de la meta-heurística ACO, una colonia de hormigas construye iterativamente soluciones para el problema bajo consideración usando rastros de feromona que están asociadas con componentes de solución definidas apropiadamente e información heurística. Cada hormiga decide iterativamente cuál tarea agregar a la secuencia. Este proceso de selección está influenciado por la información heurística que junto con el trayecto de feromona, determinan cuan bueno es elegir una tarea determinada sobre otra.

En la sección 2 se define el problema abordado y características de la metaheurística ACO; la sección 3 se describen características de las alternativas propuestas para la meta-heurística ACO; en la sección 4 se presentan las líneas de investigación y trabajos futuros.

2 ACO APLICADO A PROBLEMAS DE PLANIFICACIÓN DE MÁQUINA ÚNICA

El problema de planificación en Máquina Única abordado, es el *Total Weighted Tardiness* [9] y [1], en donde n tareas deben ser procesadas secuencialmente sobre una única máquina, sin interrupción; cada tarea tiene un tiempo de procesamiento p_j asociado, una ponderación w_j , y una fecha estimada de terminación d_j , y todas las tareas están disponibles para el procesamiento en tiempo cero.

La tardanza de la tarea j está definida como:

$$T_j = \max \{ 0, CT_j - d_j \}$$

donde CT_j es su tiempo de completación en la secuencia, de la tarea actual. El objetivo es encontrar una secuencia de tareas, que minimicen la suma de tardanzas ponderadas, dado por

$$\sum_{j=1}^n w_j T_j$$

En los algoritmos ACO, una colonia de hormigas (artificiales) construye iterativamente soluciones para el problema bajo consideración usando rastros de feromona (artificial) que están asociadas con componentes de solución definidas apropiadamente, e información heurística. Las hormigas sólo se comunican indirectamente modificando los trayectos de feromona durante la ejecución del algoritmo.

Cuando se aplica ACO a problemas de planificación de máquina única, cada hormiga comienza con una secuencia vacía y agrega iterativamente una tarea no planificada a la secuencia parcial construida hasta el momento. Cada hormiga decide cuál tarea agregar a la secuencia, influenciada al momento de la selección, por la información heurística que junto con el trayecto de feromona, determinan cuán bueno es elegir una tarea determinada sobre otra. La actualización del rastro de feromona se realiza en dos etapas: “después de cada iteración”, donde se agrega rastro de feromona a los componentes de la mejor solución global encontrada hasta el momento; y “paso a paso”, inmediatamente después que una hormiga ha agregado una nueva tarea a la secuencia parcial.

Debido a que las soluciones construidas necesitan no ser localmente óptimas con respecto a pequeñas modificaciones, en muchos de los algoritmos ACO de mejor rendimiento, las hormigas mejoran sus soluciones aplicando un algoritmo de búsqueda local.

3 ALTERNATIVAS PARA MEJORAR LAS SOLUCIONES

Con el objetivo de mejorar las buenas soluciones obtenidas en cada iteración, se pensaron en dos alternativas diferentes para lograrlo, utilizando en ambos casos, la generación de vecinos aplicando dos operadores diferentes, de adyacencia y de inversión, a la mejor solución de cada iteración.

3.1 Búsqueda local

La búsqueda local comienza desde una secuencia inicial (la mejor secuencia encontrada en la iteración) e intenta repetidamente mejorar la secuencia actual reemplazándola con soluciones vecinas. Si en el vecindario de la secuencia actual π se encuentra una mejor solución, ésta reemplaza a π y la búsqueda local continúa desde la nueva solución. El algoritmo de búsqueda local, aplica estos pasos repetidamente hasta que se dé una de las siguientes condiciones: a) no se encuentra una mejor secuencia vecina, b) después de cuatro iteraciones que generan soluciones con la misma secuencia de tareas, ó c) después de diez iteraciones con soluciones diferentes, y mejores o iguales a la anterior.

3.2 Exploración del vecindario

En este caso, no se realiza un proceso de búsqueda local, sino que se explora el vecindario. Se generan soluciones para cada iteración, donde la mejor de esas soluciones es tomada como secuencia inicial. A partir de ella se construyen vecinos que son comparados con la mejor solución antes encontrada, reemplazándola en caso de haberse obtenido mejores resultados.

Tanto para la búsqueda local como para la exploración del vecindario, se consideraron los siguientes operadores para generar los vecinos:

- (1) *Operador de adyacencia*: genera un vecino cambiando la secuencia original, seleccionando una posición i de la secuencia elegida de manera aleatoria, e intercambiándola por cualquiera de sus posiciones adyacentes j (izquierda o derecha) con la misma probabilidad.
- (2) *Operador de inversión*: genera un vecino intercambiando dos posiciones de la secuencia, i y j elegidas de manera aleatoria, considerando que $i \neq j$

4 LÍNEAS DE INVESTIGACIÓN Y TRABAJOS FUTUROS

Las dos propuestas de mejora del algoritmo ACO utilizan búsqueda local, una a través de búsqueda iterativa sobre buenas soluciones encontradas, y la otra a través de exploración del vecindario mediante la generación de vecinos. En ambos casos se logran mejores resultados que cuando se utiliza el algoritmo ACO tradicional, aunque requiere mayor esfuerzo computacional. Comparando

estas dos alternativas, se comprobó que utilizando búsqueda local iterativa se mejoran las soluciones para todos los casos.

Para optimizar la metaheurística ACO tradicional, se están probando diferentes alternativas que permitan al algoritmo encontrar buenas soluciones en menor tiempo, no solamente a través de buenos resultados sino con menor esfuerzo computacional. Para lograr esto se evaluarán nuevas alternativas que analicen la cantidad de evaluaciones que realiza el algoritmo, para las diferentes alternativas propuestas hasta el momento.

En trabajos futuros se realizará un estudio del comportamiento del algoritmo, variando los parámetros de control, cantidad de hormigas y ciclos. Se analizará de manera detallada la calidad de los resultados, teniendo en cuenta las evaluaciones que realiza el algoritmo y considerando el criterio de parada cuando se agrega búsqueda local. Se aplicará ésta meta-heurística en otros problemas de planificación y para otros tamaños de problemas.

AGRADECIMIENTOS

Agradecemos a la Universidad Nacional de la Patagonia Austral por su apoyo al grupo de investigación, la cooperación y las críticas constructivas proporcionadas por el mismo.

REFERENCIAS

- [1] Chen, T. and Gupta, M., “Survey of scheduling research involving due date determination decision”, *European Journal of Operational Research*, 38 156-166 (1989).
- [2] Di Caro, G. and Dorigo, M., “AntNet: Distributed stigmergetic control for communications networks”, *Journal of Artificial Intelligence Research*, 9: 317-365, (1998).
- [3] Dorigo, M. y Gambardella L.M., “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, *IEEE Transactions on Evolutionary Computation*, 1(1): 53-66, (1999).
- [4] Dorigo, M. y Stützle, T. *Ant Colony Optimization*, The MIT Press (2004).
- [5] Gambardella, L.M., Taillard E.D. y Dorigo, M., “Ant colonies for the quadratic assignment problem”, *Journal of the Operational Research Society*, 50:167–176, (1999b).
- [6] Gambardella, L.M., Taillard, E. y Agazzi G.. “MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows”. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, 63–76. McGraw Hill, (1999a).
- [7] Maniezzo, V., “Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem”. *INFORMS Journal on Computing*, 11(4): 358–369, (1999).
- [8] Morton T., Pentico D., “Heuristic scheduling systems”, *Wiley series in Engineering and technology management*. John Wiley and Sons, INC, 1993.
- [9] Pinedo, M., *Scheduling: Theory, Algorithms and System*, First edition Prentice Hall (1995).
- [10] Stützle T. y Dorigo M., “ACO algorithms for the quadratic assignment problem”. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, 33–50. Mc-Graw Hill, (1999).
- [11] Stützle, T., “Local Search Algorithms for Combinatorial Problems – Analysis, Improvements, and New Applications”, PhD thesis, Darmstadt University of Technology, Department of Computer Science, (1998).