

METAHEURÍSTICA ACO APLICADA A PROBLEMAS DE PLANIFICACIÓN EN ENTORNOS DINÁMICOS

de San Pedro M., Pandolfi D., Lasso M., Villagra A.

Laboratorio de Tecnologías Emergentes (LabTEem)

Proyecto UNPA-29/B084/1¹

Unidad Académica Caleta Olivia - Universidad Nacional de La Patagonia Austral
(9011) Caleta Olivia – Santa Cruz - Argentina

e-mail: {edesanpedro,dpandolfi,mlasso,avillagra}@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

Las organizaciones industriales frecuentemente están sujetas a diferentes tipos de cambios como pueden ser: que se agregue una nueva tarea, la cancelación de una tarea, la ruptura de una máquina, como así también cambios en los tiempos de procesamiento o de la fecha estimada de terminación de la tarea. Debido a su naturaleza dinámica, los problemas de planificación real son computacionalmente complejos y el tiempo requerido para encontrar una solución óptima se incrementa exponencialmente con el tamaño del problema. Los problemas de planificación se pueden clasificar en: estáticos, donde todas las tareas son conocidas antes del comienzo de la planificación; y dinámicos, en donde sólo el tiempo de comienzo de la tarea es desconocido (dinamismo parcial) ó donde todas las propiedades de las tareas son desconocidas (dinamismo total).

Los problemas de planificación (*scheduling*) estáticos se han abordado a través de distintas metaheurísticas (*Simulating Annealing*, *Tabu Search*, Algoritmos Evolutivos y *Ant Colony Optimization*), pero los problemas de *scheduling* dinámicos han sido encarados principalmente con Algoritmos Evolutivos, para problemas de *job shop* [1], [2], [3], [4], [5] y para problemas de máquina única [6], [7], [8], [9], [10] y [11].

Una metaheurística particularmente exitosa está inspirada para el comportamiento de las hormigas reales, conocida como la metaheurística *Ant Colony Optimization* (ACO). Numerosos enfoques algorítmicos basados sobre las mismas fueron desarrollados y aplicados con éxito para una variedad de problemas de optimización [14], [15], [16], [17], [18], [19] y [20].

Se pretende a través de esta línea de investigación, realizar un análisis comparativo de los trabajos realizados hasta el momento sobre *scheduling* dinámico con algoritmos evolutivos para diferentes problemas de máquina única (*Weighted Tardiness*, *Average Tardiness*, *Weighted Number of Tardy Job*), pero abordado con otra metaheurística diferente como es el caso de ACO.

1. INTRODUCCIÓN

Cuando estamos ante un proceso de *scheduling* de actividades, es necesario seleccionarlas para construir una secuencia de las mismas, de manera que cumplan uno o más objetivos, y

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

satisfagan un conjunto de restricciones del dominio del problema. Durante este proceso se deberá seleccionar entre planificaciones (*schedules*) alternativas y asignar recursos y tiempos a cada actividad, de manera tal que dichas asignaciones respeten las restricciones temporales de las actividades (*jobs*) y las capacidades limitadas de un conjunto de recursos compartidos, para que puedan ser minimizadas ciertas funciones objetivo (por ejemplo *tardiness*, *makespan*, etc.).

Dentro del ámbito de *scheduling*, los modelos más estudiados han sido siempre los modelos conocidos como estáticos, en donde las actividades, los recursos, los tiempos de procesamiento están predefinidos, es decir que no se modifican durante el proceso. Pero en los problemas del mundo real existen otra serie de decisiones que interactúan con el modelo clásico, como por ejemplo cambiar la cantidad y/o configuración interna de los recursos mientras el proceso de *scheduling* está en avance para balancear los cambios en la carga de los *jobs* que arriban al sistema; o puede aumentar o disminuir la cantidad de operarios en el sistema, en distintos momentos del día; o uno o más recursos pueden deshabilitarse temporalmente por razones de falla o mantenimiento. Si los modelos contemplan alguna de estas decisiones se obtienen los modelos de dinámicos.

Ante la presencia de un modelo dinámico, existen diferentes criterios para categorizar las clases de ambientes dinámicos: frecuencia del cambio, severidad del cambio, ¿cuán predecible es el cambio?, o ¿hay un ciclo repetitivo en los cambios? [13]. Además de las posibles clasificaciones y/o alternativas de cambios en el entorno, existe una cuestión muy importante a tener en cuenta que es la manera en que se evalúan los algoritmos propuestos para un problema dinámico, ya que las comparaciones no son tan directas como lo podrían ser en un entorno estático, y por ende el análisis de la calidad de los algoritmos propuestos debe ser muy cauteloso.

Para el caso de problemas de *scheduling* dinámico, se pueden considerar diferentes aspectos que tienen que ver con los problemas reales en general, como son el cambio de fecha límite de terminación de las tareas (*due dates*) o cambios en el tiempo de procesamiento de una tarea. Sin embargo existen otras alternativas que son más específicas de los problemas de *scheduling*, como puede ser que se agreguen nuevas tareas durante el proceso de asignación de recursos, o por el contrario, eliminación de tareas, rotura de máquinas, etc. Aunque estos tipos de cambios son más complejos de manejar, son más cercanos a situaciones encontradas en el mundo real.

En la siguiente sección se describen características generales de diferentes metaheurísticas, en la sección 3 se detallan aspectos particulares de la metaheurística ACO y finalmente en la sección 4 se presentan las líneas de investigación y trabajos futuros.

2. CARACTERÍSTICAS GENERALES DE LAS METAHEURÍSTICAS

Todas las metaheurísticas tienen en común que intentan evitar la generación de soluciones de pobre calidad [12], introduciendo mecanismos generales que extiendan el problema específico, algoritmos de una corrida como heurísticas de construcción, o búsqueda local de mejora iterativa. Las diferencias entre las metaheurísticas disponibles tienen que ver con las técnicas empleadas para evitar que se estanque en soluciones sub-óptimas y el tipo de trayectoria seguida en el espacio de cualquiera de las soluciones, parcial o total.

Una primera distinción que puede hacerse entre metaheurísticas, es si ellas se basan en búsqueda constructiva o local. Otra distinción importante está dada en que en cada iteración, ellas manipulan una solución simple o una población de soluciones. Aunque las metaheurísticas 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 es incluida. Este es el caso para ACO y AE.

Otra clasificación de metaheurísticas tienen que ver con el uso de memoria, es decir, aquellas que explotan memoria para dirigir la búsqueda futura: *Tabu Search* memoriza explícitamente, soluciones encontradas previamente o componentes de soluciones vistas previamente; *Guided Local Search* (GLS), almacena penalidades asociadas con componentes soluciones para modificar la función de evaluación de soluciones; y *Ant Colony System* (ACO) usa feromona para mantener una memoria de experiencias pasadas.

Es interesante notar que, para todas las metaheurísticas, existe un criterio de terminación no general. En la práctica, se usan un número de reglas generales: el máximo tiempo transcurrido en CPU, el número máximo de soluciones generadas, el porcentaje de desviación desde un *lower/upper bound* al óptimo, y el máximo número de iteraciones sin mejora en la calidad de la solución, son ejemplos de tales reglas. En algunos casos, se pueden definir reglas generales dependientes de la metaheurística.

ACO tiene varias características que en su combinación particular lo hace un enfoque único: usa una población (colonia) de hormigas que construye soluciones explotando una forma de memoria indirecta llamada feromona artificial.

3. METAHEURÍSTICA ACO PARA AMBIENTES DINÁMICOS

Considerando las características de los ambientes dinámicos, sería deseable contar con un algoritmo de optimización que sea capaz de ir adaptando continuamente la solución a los cambios del entorno, re-usando la información obtenida en el pasado. Los Algoritmos Evolutivos (AEs) parecen ser los candidatos apropiados ya que tienen mucho en común con la evolución natural, y la adaptación en la naturaleza es un proceso continuo.

El problema principal con los AEs, es que ellos convergen eventualmente a un óptimo y pierden así su diversidad que es necesaria para explorar eficientemente el espacio de búsqueda. Así, una vez que la población del algoritmo evolutivo converge, ésta también pierde su habilidad para adaptarse a un cambio en el entorno cuando tal cambio ocurre. En consecuencia, se requiere de mecanismos adicionales que provean permanente diversidad en la población sin perturbar el proceso de búsqueda.

Por otro lado, la metaheurística ACO está inspirada en el comportamiento de las hormigas reales. Puede ser aplicada a cualquier problema de optimización, para que un procedimiento de construcción de una solución pueda ser realizado. ACO 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 [12].

La aplicación de ACO es particularmente interesante para: problemas NP-duros, que no pueden ser resueltos eficientemente por muchos algoritmos tradicionales; problemas dinámicos del camino más corto, en el cual algunas propiedades de la representación del grafo del problema, cambian concurrentemente en tiempo extra con el proceso de optimización; problemas en los cuales la arquitectura computacional está distribuida espacialmente. La versatilidad de la metaheurística ACO ha sido mostrada en varias aplicaciones.

4. LÍNEAS DE INVESTIGACIÓN Y TRABAJOS FUTUROS

Hasta este momento la investigación en el grupo ha estado orientada al estudio de ambientes dinámicos para problemas de máquina única, abordados en todos los casos con Algoritmos Evolutivos [8], [9], [10], [11], [21]. Con el objetivo de diseñar e implementar diferentes heurísticas alternativas, se ha iniciado el estudio en primer lugar del comportamiento de la metaheurística ACO. En este sentido se prevé además el diseño e implementación de

experimentos que permitan poder realizar la evaluación de la heurística y analizar los resultados obtenidos, a fin de poder hacer un análisis comparativo con los resultados ya obtenidos para ambientes dinámicos con AEs.

A futuro se realizará el estudio de otras metaheurísticas desarrolladas y se analizarán los resultados obtenidos para cada una de ellas a los fines de poder realizar una comparación de resultados que nos indique cuál es la mejor heurística a utilizar para el modelo dinámico planteado, suponiendo para este caso que solo exista una.

Además de analizar el comportamiento de las distintas metaheurísticas en el mismo ambiente dinámico estudiado con algoritmos evolutivos, se prevé hacer un análisis más profundo sobre los diferentes modelos dinámicos en la planificación.

Todos estos problemas han sido desarrollados sin restricciones, por lo que en enfoques futuros se incorporarán a estos problemas, algunas restricciones que hasta el momento no han sido analizadas tales como *preemption vs nonpreemption*, inserción vs no inserción de tiempo ocioso, *set up* dependientes de la secuencia, dependencia entre jobs (*precedence constraints*) entre otros.

AGRADECIMIENTOS

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

BIBLIOGRAFÍA

- [1] C. Bierwirth, et. al – *Production Scheduling and Rescheduling with Genetic Algorithms* – Evolutionary Computation, 7, N° 1, págs. 1 - 17, 1999.
- [2] Fang H. – *Genetic Algorithms in Timetabling and Scheduling* – Department of Artificial Intelligence, University of Edinburg, Scotland, 1994.
- [3] Vazquez M. y Whitley , D – *A Comparision of Genetic Algorithms for the Dynamic Job Shop Scheduling Problem* – Parallel Problem Solving from Nature VI, 2000.
- [4] Lin S., et. al – *A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems* – International Conference of Genetic Algorithms, 1997.
- [5] Whitley, D. y Kauth, G. – *GENITOR: A Different Genetic Algorithm* – Proceedings Rocky Mountain Conference on Artificial Intelligence, 1988.
- [6] Madureira A., Ramos C., do Carmo Silva, Silvio – *A Genetic Approach to Dynamic Scheduling for Total Weighted Tardiness Problem*, PLANSIG'99, 18th Workshop of the UK Planning and Scheduling Special Interest Group, Manchester, UK, 1999.
- [7] Mao W., Kincaid R. y Rifkin, A. – *On-line Algorithms for a Single Machine Scheduling Problem*, en Impact of Emerging Technologies in Computer Science and Operation Research, págs. 157 – 173, 1995.
- [8] Lasso, M., Pandolfi D., De San Pedro M., Villagra A., Vilanova G., Gallard, R. – *Algorithms to Solve the Dynamic Weighted Tardiness Problems*, VIII Congreso Argentino de Ciencias de la Computación, Bs. As., pág. 609 – 616, Octubre 2002.
- [9] Lasso, M., Pandolfi D., De San Pedro M., Villagra A., Gallard, R. – *Heuristics to Solve Dynamic W-T Problems in Single Machine Environments*, Proceedings of the International Conference on Computer Science, Software Engineering Information Technology, e-Business and Applications, págs. 432 –437, Rio de Janeiro, June 2003, Brazil.

- [10] De San Pedro M., Lasso, M., Villagra A., Pandolfi D., Gallard, R. – *Solutions to the Dynamic Average Tardiness Problem in Single Machine Environmets*, IX Congreso Argentino de Ciencias de la Computación, La Plata, págs. 729 – 739, Octubre 2003.
- [11] Lasso, M., Pandolfi D., De San Pedro M., Villagra A., , Gallard, R. – *Solving Dynamic Tardiness Problems in Single Machine Environments*, IEEE Congress on Evolutionary Computation, Vol. I, págs. 1143 – 1149, Portland, USA, 2004.
- [12] Dorigo, M. y Stützle, T. – *Ant Colony Optimization*, The MIT Press, 2004.
- [13] Branke J. – *Evolutionary Optimization in Dynamic Environments (Genetic Algorithms and Evolutionary Computation)*. Kluwer Academic Publishers (KAP), 2002.
- [14] Colorni, A., Dorigo, M., Maniezzo V., Trubian, M. – *Ant System for Job Shop Scheduling*, JORBEL – Belgian Journal of Operations Research, Statistics and Computer Science, Vol. 34, N°19, págs. 39 – 53.
- [15] Bauer, A., Bullheimer, B., Hartl R. F., Strauss, C. – *Minimizing Total Tardiness on a Single Machine using Ant Colony Optimization*, Central European Journal of Operations Research and Economics, Vol 8, N° 2, págs. 125-141, 2000.
- [16] Blum C. – *ACO Applied to Group Shop Scheduling: A Case Study on Intensification and Diversification*, Proceedings of ANTS 2002, Lecture Notes in Computer Science Series, N° 2463, Springer-Verlag, 2002.
- [17] Pfahringer, B. – *Multi-agent Search for Open Shop Scheduling. Adapting the Ant_Q Formalism*, Technical Report TR-96-09, Australian Research Institute for Artificial Intelligence, Vienna, 1996.
- [18] Gagné C., Price, W. L., Gravel, M. – *Comparing an ACO Algorithm with other Metaheuristics for the Single Machine Scheduling Problem with Sequence-dependent setup times*, Journal of Operational Research Society, Vol. 53, págs. 895 – 906, 2002.
- [19] Merkle, D., Middendorf, M. – *On Solving Permutation Scheduling Problems with Ant Colony Optimization*, Technical Report 415, Institute AIFB, University of Karlsruhe, 2002.
- [20] Guntsch, M., Middendorf, M. – *Applying Population Based ACO to Dynamic Optimization Problems*, Proceedings of ANTS2002, Lecture Notes in Computer Science Series N° 2463, págs. 111 – 122, Springer-Verlag, 2002.
- [21] de San Pedro M., Pandolfi D., Lasso M., Villagra A. - *Dynamic Scheduling Approaches to solve Single Machine Problems* – ASC 2005 - The Ninth IASTED International Conference on Artificial Intelligence and Soft Computing – pp 275-280, Benidorm, España, 2005