

ENFOQUES HEURISTICOS PARA PROBLEMAS DE SCHEDULING ESTATICOS Y DINAMICOS

Esquivel S.C, Gallard R., Gatica C., Ferrero S., Leguizamón G., Zuppa F.

LIDIC¹

Facultad de Ciencias Físicas, Matemáticas y Naturales

Universidad Nacional de San Luis

{esquivel, rgallard, gaticac, swf, legui, fede}@unsl.edu.ar

Resumen

El presente trabajo describe una propuesta de investigación para atacar diferentes problemas de scheduling. Dicha tarea es realizada por investigadores de distintas líneas del LIDIC, y en consecuencia, con técnicas diferentes las cuales pueden usarse solas y/o combinadas.

En general, dentro del ámbito de scheduling, los modelos más estudiados fueron los modelos conocidos como clásicos o estáticos, es decir, donde las actividades, los recursos, los tiempos de procesamiento están predefinidos, no se modifican durante el proceso, y con un objetivo involucrando la minimización del tiempo de finalización y los costos de operación. Pero en los problemas del mundo real existen otra serie de decisiones que interactúan con el modelo clásico, si algunas de estas decisiones se adicionan a dicho modelo se obtienen los modelos de scheduling dinámicos.

Debido a la naturaleza de estos problemas, que pertenecen a la clase de problemas NP-duros, las tendencias más modernas tratan de hallar heurísticas con estrategias de explotación/exploración mixtas. Dentro de ellas cabe mencionar Tabu Search, Simulating Annealing, Colonia de Hormigas y Algoritmos Evolutivos.

1. INTRODUCCION

Un proceso de scheduling implica seleccionar y secuenciar actividades tal que ellas cumplan uno o más objetivos y satisfagan un conjunto de restricciones del dominio del problema. Durante este proceso se deberá seleccionar entre schedules (planes o planificaciones) alternativos 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, de manera que ciertas funciones objetivo (por ejemplo tardiness, makespan, etc.) sean minimizadas. Los problemas que pertenecen a esta categoría son conocidos como pertenecientes a la clase de los problemas NP-duros.

En general, dentro del ámbito de scheduling, los modelos más estudiados fueron los modelos estáticos, es decir, donde las actividades, los recursos, los tiempos de procesamiento están predefinidos, no se modifican durante el proceso, y con un objetivo involucrando la minimización del tiempo de finalización y los costos de operación.

En los problemas del mundo real existen otra serie de decisiones que interactúan con el modelo clásico, si algunas de estas decisiones se adicionan a dicho modelo se obtienen los modelos de scheduling dinámicos. Por ejemplo, algunas de estas decisiones pueden ser 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 bien, puede aumentar o disminuir la cantidad de operarios en el sistema, en distintos momentos del día, o uno o más recursos pueden deshabilitarse temporariamente por razones de falla o mantenimiento.

¹ Laboratorio de Investigación y Desarrollo en Inteligencia Computacional, Director Dr. Raúl Gallard

Los modelos de este tipo implican una reconfiguración interna dinámica del proceso de scheduling para adaptarlo a la nueva situación del contexto. También pueden existir causas externas que necesiten una reconfiguración, por ejemplo cambio, por parte de los clientes, en las fechas de entrega en función de sus stocks y demandas[1].

Para encarar los problemas de scheduling históricamente se usaron diferentes enfoques, entre ellos cabe mencionar:

Simulación por Computadoras: Entre los años 50 y 60, con la presencia de las computadoras se hizo posible representar el sistema a planificar por medio de modelos que permitían reflejar su estructura, los jobs y sus operaciones, los recursos y una serie de restricciones con algún nivel de detalle tal que, dados los datos de entrada y reglas de heurísticas simples (dispatching), en determinados puntos de decisión, el programa de simulación podía extrapolar un schedule a un costo relativamente bajo. La literatura al respecto es relativamente amplia puede recomendarse especialmente [2], [3], [4], [5], [6] y [7]. Si bien la simulación puede representar modelos muy realísticos a un costo computacional modesto tiene la desventaja de que los resultados obtenidos no son ni siquiera aproximados al óptimo y, en consecuencia, no resultan en buenos benchmarks para establecer cuán buenas son las soluciones encontradas o cómo encontrar otras más eficientes.

Modelos Matemáticos: En la década del 60 diferentes investigadores [8], [9] desarrollaron las técnicas de programación entera, aprovechando la mayor potencia computacional ya disponible en ese tiempo, que permite que problemas, más realistas fuesen formulados de manera que teóricamente permita su resolución exacta. Esto dio lugar a una serie de trabajos siguiendo esta dirección [10], [11], [12], [13] y [14].

También otra técnica ampliamente utilizada es la de branch-and-bound y su variante branch-and-prune. En los años 70 Srinivasen [15] y otros aplicaron la técnica de la programación dinámica para resolver problemas de secuenciamiento usando todos los subconjuntos posibles de jobs como elementos del espacio de estados.

Tanto en el caso de la programación entera como en el de la programación lineal mientras problemas muy pequeños pueden resolverse encontrando la solución óptima, esto es prioridades perfectas, ruteo y demás, los problemas complejos (de tamaño grande) permanecen intratables y, es de esperar que los problemas del mundo real permanezcan así, puesto que ellos pertenecen a la clase de los problemas NP-duros, lo cual expresado muy simplemente significa que no se conoce algoritmo que no crezca en tiempo exponencial acorde con el tamaño del problema [16].

Heurísticas: Las razones explicadas, en el párrafo anterior, dieron lugar al desarrollo de diferentes heurísticas para resolver, no ya de manera exacta sino aproximada, tales tipos de problemas.

Estas heurísticas pueden clasificarse dentro de las categorías de explorativas, explotativas o explorativas/explotativas.

Dentro de las heurísticas explotativas (llamadas así porque inspeccionan el espacio de búsqueda en el vecindario de una solución posible) las búsquedas en el vecindario son una de las técnicas usadas en este tipo de problemas [17]. Esta técnica es un ejemplo particular del método de programación no lineal conocido como hill-climbing.

Otra técnica, Lagrangian Relaxation [18] soluciona problemas de programación lineal simples a través de la eliminación de algunas restricciones y pagando penalidades proporcionales a las cantidades en que dichas restricciones se violan. Este método de relajación es semejante a la heurística dinámica de cuello de botella [19] que si bien son, a menudo, poderosas son a la vez complejas de usar y no son aun métodos de propósito general.

Dentro de las heurísticas explorativas que pueden ser pensadas que actúan, en determinado momento del proceso al no avanzar hacia nuevas soluciones, diciendo “es tiempo de un cambio” [20] lo cual se traduce en realizar un salto a otro punto del espacio de búsqueda (no del vecindario). Una de las primeras en ser utilizadas fue la conocida como muestra random.

Las tendencias más modernas tratan de hallar heurísticas con estrategias de explotación/exploración mixtas. Dentro de ellas cabe mencionar Tabu Search, Simulating Annealing, Colonia de Hormigas y Algoritmos Evolutivos.

Dentro de las heurísticas Evolutivas y de Colonia de Hormigas es que encuadra el desarrollo la presente investigación. Es importante destacar que los distintos problemas han sido estudiado por diferentes investigadores y desde diferentes disciplinas, especialmente en el modelo clásico o estático. Se han desarrollado diferentes algoritmos evolutivos abarcando distintos aspectos: representaciones directas vs. indirectas, enfoques puros vs. híbridos, algoritmos seriales vs. paralelos, operadores genéticos generales vs. específicos, y así siguiendo [22], [23], [24], [25], [26] y [27]]. Mientras que la heurística de la Colonia de Hormigas ha sido aplicada en pocos casos a problemas de scheduling. Dorigo y otros en [28,29] y Zwaan y otros[30] reportaron interesantes resultados para el problema de Job Shop Scheduling. Stützle en [31], desarrolló un algoritmo eficiente para resolver el problema de Flow Shop a través de la incorporación de búsqueda local. Otras aplicaciones incluyen diferentes algoritmos de Colonia de Hormigas para el problema de Single Machine Total Tardiness. En estas aplicaciones esta heurística es mejorada a través de formas alternativas de evaluación de feromona para guiar la búsqueda de las soluciones.

Con respecto al problema Scheduling Dinámico pocos investigadores lo han encarado usando algoritmos evolutivos. Dentro de los resultados publicados recientemente cabe mencionar [32], [33], [34]. Mientras que con respecto a la heurística de la Colonia de Hormigas el grupo no tiene conocimiento de que haya sido aplicada en ambientes no estacionarios.

El grupo tiene en su background un importante trabajo realizado sobre este tipo de problemas, esencialmente para los casos de Job Shop, Flow Shop y Máquinas Paralelas para el caso estático, atancado aspectos relacionados con la representación y multirecombinación que ha permitido obtener resultados interesantes. También se ha iniciado el estudio de esta heurística aplicada al problema de Flow Shop e hibridizadas con heurísticas greedy a fin de mejorar su performance [35].

2. PROPUESTA ACTUAL DE INVESTIGACION

La investigación actual se divide en las siguientes líneas:

1. **Job Shop:** Por un lado, se está trabajando en la búsqueda de soluciones dirigidas a determinadas regiones del espacio de búsqueda de todos los posibles schedules[36] usando un enfoque hibridizado de colonias de hormigas. Por otro lado, a partir de la experiencia obtenida [37], [38], [39] y [40], en el caso estático, se ha comenzado a encarar el diseño de heurísticas evolutivas alternativas para el problema del job shop dinámico adaptando las técnicas evolutivas aplicables a problemas generales de ambientes dinámicos, para ello se usarán distintas funciones objetivo y las características del problema inicialmente se restringen a: los jobs tienen distintos tiempos de arribo, cada job tiene asociado un peso y una fecha de entrega, las máquinas son reentrantes, no todos los jobs tienen la misma cantidad de operaciones y los jobs están agrupados en clusters que siguen las mismas rutas.
2. **Flow Shop:** En este momento se está trabajando en la optimización multiobjetivo, con un enfoque evolutivo, ámbito en el cual no hay mucho trabajo realizado.
3. **Máquinas Paralelas:** En este caso el problema más serio que se presenta es la falta de benchmarks y el no conocimiento de valores óptimos ni de upper bounds para instancias de tamaño de interés (medias y grandes), en consecuencia, si bien se trabaja en la incorporación de técnicas alternativas de multirecombinación, representación e hibridización simultáneamente se están implementando nuevas heurísticas más modernas y de ser posible algún método analítico que

permita obtener valores de referencia y óptimos, respectivamente, a los efectos de poder validar los resultados obtenidos.

3. REFERENCIAS BIBLIOGRÁFICAS:

- [1] Morton T.E., Pentico D.W. – Heuristic Scheduling Systems – John Wiley & Sons Inc., New York, 1993.
- [2] Baker, K. R. – Introduction to Sequencing and Scheduling – John Wiley & Sons Inc., New York, 1974.
- [3] Baker C.T y Dzienlinski B.P. – Simulation of a Simplified Job Shop – Management Science 6, págs. 311 – 323, 1960.
- [4] Moore J. M y Wilson R. C. - A Review Of Simulation Research in Job Shop Scheduling, Production and Inventory Management, 8, págs. 1 –10, 1967.
- [5] Pai A. R. y Mc Roberts K. L. – Simulation Research in Intercangeable Part Manufacturing, Management Science, 17, págs. 732 - 743, 1971.
- [6] Glaser R. u Hottenstein M. – Simulation Study of a Close-Loop Job Shop, Journal of Operations Management, 3, págs. 155 - 166, 1982.
- [7] Barret, R. T. y Barman S – A SLAM II Simulation of a Simplified Flow Shop, Simulation, 47, págs. 181 - 189, 1986.
- [8] Balas, E. - Discrete Programming by the Filter Method, Operations Research, 15, págs. 915 - 957, 1967.
- [9] Gomory, R. E. - Faces of an Integer Polyhedron, Proceedings of the National Academic of Science, 57, págs. 16 - 18, 1967.
- [10] Brown, A. y Lomnicki Z. A. Some Applications of the Branch and Bound Algorithm to the Machine Sequencing Problem, Operationional Research, 17, págs. 173 – 186, 1966.
- [11] Florian, M. P. et. al - An Implicit Enumeration Algorithm for the Machine Sequencing Problem, Managment Science, 17, págs. 782 – 792 , 1971.
- [12] Greenberg, H. - A Branch and Bound Solution to the General Scheduling Problem, Operations Research, 16, págs. 353 - 361, 1968.
- [13] McMahan, G. B. y Burton P. G. – Flow Shop Scheuling with the Branch and Bound Method, Operations Research, 15, págs. 473 - 481, 1967.
- [14] Schwimer, J. - On the n-Job, One Machine, Sequence Independent Scheduling Problem with Tardiness Penalties: A Branch and Boun Solution, Managment Science, 18, págs. 301 - 313, 1967.
- [15] Srinivasan, V. - A Hybrid Algorithm for the One Machine Sequencing Problem to Minimize Total Tardiness, Naval Research Logistics, 18, págs. 317 - 327, 1971.
- [16] Garey, M. R. - the complexity of Flowshop and Jobshop Scheduling, Matehematics of Operations Research, 1, págs. 117 - 129, 1976.
- [17] Wilkerson, L. J. e Irwin, J. D . – An Improved Algorithm for Scheuling Independent Tasks, AOOE Transactions, 3, págs. 239 - 245, 1971.
- [18] Van De Velde, S. L. - Machine Scheduling and Lagrangian Relaxation, PhD Thesis, Technische universiteit Eindhoven, 1991.
- [19] Glover, F. - Tabu Search: A Tutorial, Interfaces, 20, págs. 74 - 94, 1990.
- [20] Glover, F. y Laguna, M. – Target Analysis to Improve a Tabu Search Method for Machine Scheduling, Advanced Knowledge Research Group, US West Advanced Technologies, Boulder, CO., 1989.
- [21] Kirkpatrick, S.C. y Vecchi, M. P. – Optimazation by Simulated Annealing, Science, 220, págs. 671 – 680, 1983.
- [22] Whitley, et. al. – Sequence Scheduling ith Genetic Algorithms – New Directions for Operation Research in Manufacturing, Springer Verlag, 1991.

- [23] Hart, E, et. al – A Heuristic Cpbomatopm Method for solving Job Shop Scheduling Problems – Parallel Problem Solving from Nature V, Springer Verlag, 1998.
- [24] Lin, S., et. al. – Investigating Parallel Algorithms on Job Shop Scheduling Problems – VI Annual Conference on Evolutionary Programming, 1997.
- [25] Yamaha, T., et. al. – A Genetic Algorithm Applicable to Large Scale Job Shop Problems – Parallel Problem Solving from Nature II, 1992.
- [26] C. Bierwirth, et. al – Production Scheduling and Rescheduling with Genetic Algorithms – Evolutionary Computation, 7, N° 1, págs. 1 - 17, 1999.
- [27] Fang H. – Genetic Algorithms in Timetabling and Scheduling – Department of Artificial Intelligence, University of Edinburg, Scotland, 1994.
- [28] M. Dorigo and V. Maniezzo and A. Colorni - "Ant System for the job-shop scheduling". JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science, 1996.
- [29] M. Dorigo and V. Maniezzo and A. Colorni - "The ant system: Optimization by a colony of cooperating agents". IEEE Trans. Systems, Man, and Cybernetics. Part B, pp 29-41, 1996.
- [30] S. van der Zwaan and C. Marques - "Ant colony optimization for job job scheduling". Proceedings of the Third Workshop on Genetic Algorithms and Artificial Life (GAAL '99), 1999.
- [31] T. Stützle - "An ant approach for the flow shop problem". Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT '98), Vol 3, pp 1560-1564, 1998. Verlag, Aachen.
- [32] Vazquez M. y Whitley , D – A Comparision of Genetic Algorithms for the Dynamic Job Shop Scheduling Problem – Parallel Problem Solving from Nature VI, 2000.
- [33] Lin S., et. al – A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems – International Conference of Genetic Algorithms, 1997.
- [34] Whitley, D. y Kauth, G. – GENITOR: A Different Genetic Algorithm – Proceedings Rocky Mountain Conference on Artificial Intelligence, 1988.
- [35] Esquivel, Gallard, Leguizamón, Zuppa – “A Comparative Performance of Alternative Heuristics for the Flow Shop Scheduling Problem”, Evocop 2002, aceptado para su publicación en Lecture Notes in Computer Science, Springer Verlag.
- [36] G. Leguizamón and M. Schütz - "An improved ant colony algorithm for the job shop scheduling problem". 2002. En preparación.
- [37] Esquivel S., Ferrero S. y Gallard R. – “Multirecombined Evolutionary Algorithm to solve Multiobjective Job Shop Scheduling”, expuesto y publicado en los Anales del VI Congreso Argentino de Ciencias de la Computación, CACIC, Universidad Nacional de la Patagonia San Juan Bosco, 2-7 de Octubre de 2000, pag. 1331.
- [38] Esquivel S. C., Ferrero S.W., Gallard R.H. “A multiobjective optimization via multistage evolution for the job shop scheduling problem”. Second International ICSC Symposium of Intelligent Systems, EIS 2000, **presentado y publicado** June 27-30 , 2000, University of Paisley, Scotland U.K.
- [39] Esquivel S., Ferrero S., Gallard R., “Multirecombination in Multiobjective Evolutionary Algorithms for Job Shop Scheduling”, CEC’2000, Evolutionary Computation, San Diego, Usa, **IEEE Publishing Co**, aceptado para su publicación.
- [40] Esquivel S., Ferrero S., Gallard R. – “Multirecombination and Local Search in Multiobjctive Evolutionary Algorithms for the Job Shop Scheduling”, Third Iberoamerican Workshop on Distributed Artificial Intelligence and Multiagent Systems, 3WDAIMAS, **IEEE Publsishing Co.**, Sao Paulo, Brasil, November 2000, a aparecer.