RESOLUCIÓN DE PROBLEMAS DE PLANIFICACIÓN DE TAREAS EN AMBIENTES DE MÁQUINAS PARALELAS USANDO ALGORITMOS EVOLUTIVOS

Ferretti E., Esquivel S., Gallard R.
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)[1]
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950 - Local 106
(5700) - San Luis - Argentina
e-mail: {ferretti, esquivel}@unsl.edu.ar
TE: +54 2652 420823
Fax: +54 2652 430224

Resumen

La planificación en un ambiente de máquinas idénticas en paralelo (P_m) implica efectuar un cierto número de tareas (jobs) que utilizan varios recursos (un banco de máquinas en paralelo) por un cierto periodo de tiempo. Un modelo sencillo que consiste de m máquinas y n tareas independientes es la base de modelos más complejos. En este modelo, las tareas son asignadas de acuerdo con la disponibilidad de los recursos existentes siguiendo alguna regla de despacho determinada. El tiempo de finalización de la última tarea que abandona el sistema, conocido como makespan (C_{max}), es uno de los objetivos más importantes a ser minimizado debido a que usualmente implica una alta utilización de los recursos; sin embargo, deben ser considerados otros objetivos tratados comúnmente en la literatura [9, 11], los cuales además de tener importancia teórica son de relevancia práctica. Muchos de estos problemas son NP-Hard para $2 \leq m \leq n$ y por ende, se han desarrollado heurísticas convencionales, Algoritmos Evolutivos (AEs) y otras heurísticas poblacionales para proveer planes o planificaciones aceptables como soluciones.

Este trabajo, expone las conclusiones obtenidas en la resolución de problemas de planificación de tareas sin restricciones de precedencia, en ambientes de planificación de 2 y 5 máquinas idénticas en paralelo, mediante el uso de AEs que implementan distintas técnicas de recombinación y utilizan distintas representaciones indirectas de cromosoma. La performance lograda por los distintos AEs implementados, se compara con los resultados obtenidos por un conjunto de heurísticas convencionales que se aplican usualmente a diferentes problemas de planificación de máquinas idénticas en paralelo.

[1] El LIDIC es financiado por la Universidad Nacional de San Luis y por la ANPCYT (Agencia Nacional para Promover la Ciencia y Tecnología).

1. Introducción

Los problemas de planificación ocurren en áreas muy diversas como sistemas de manufactura, sistemas de producción, diseños de computadoras, logística, sistemas de comunicación entre otros. Una característica común de los problemas que se presentan en estas áreas, es que no existen algoritmos que brinden soluciones óptimas en tiempos de orden polinomial. En particular, el problema de encontrar un plan o planificación, para $m > 2$ procesadores de igual capacidad, que minimice el tiempo completo de procesamiento de un conjunto de tareas independientes, se ha probado que pertenece a la clase de problemas NP-Hard [11]. En la literatura, la minimización de C_{max} ha sido estudiada por diferentes investigadores y un gran número de benchmarks pueden ser encontrados, para distintos problemas de planificación considerando este objetivo. Este no es el caso de otros objetivos tratados comúnmente en la literatura [9, 11], los cuales además de tener importancia teórica son de relevancia práctica, entre los que se pueden mencionar: la tardanza máxima (T_{max}), la tardanza media (T_{avg}), la tardanza ponderada (T_{wt}), el número de jobs tardíos (N_t) y el número ponderado de jobs tardíos (N_{wt}). Para la mayoría de ellos, su complejidad computacional se mantuvo sin resolver por muchos años, hasta que se estableció como NP-Hard en 1989 [11]. Varias heurísticas convencionales se han desarrollado para la resolución de problemas de planificación, en distintos ambientes, cuando se consideran los objetivos citados precedentemente. Algunas de ellas son: longest processing time (LPT), weighted longest processing time (WLPT), shortest processing time (SPT), weighted shortest processing time (WSPT), earliest due date (EDD), least slack (Slack), Hodgson´s algorithm (HDG), weighted Hodgson (WTD HDG) y Rachamadugu y Morton (R&M).

En trabajos previos se ha estudiado el uso de AEs en la resolución del problema de planificación (P_m| |C_{max}) [5], como así también la influencia que tiene la representación del cromosoma en la misma [8]. En la actualidad, nuestro trabajo se ha extendido a (P_2| |Obj) y (P_5| |Obj) donde Obj es alguno de los objetivos relacionados con las fechas de entrega (due-dates) mencionados previamente. Los resultados logrados con los AEs implementados han sido contrastados con los obtenidos por las heurísticas convencionales.

Como no es usual encontrar publicados Benchmarks para instancias de tamaño de interés (medias y grandes), para problemas de planificación en máquinas paralelas, construimos nuestro propio conjunto de datos de trabajo con 20 instancias de 40 tareas, seleccionadas de la OR-Library [1], una colección de conjuntos de datos de prueba para problemas de planificación y otros campos de interés de investigación operativa. La idea subyacente es que los AEs implementados, trabajen con datos que hayan sido propuestos para distintos problemas de planificación basados en las fechas de entrega. De esta manera se prueba la consistencia de las fechas de entrega, tiempos de arrivo y tiempos de procesamiento de cada instancia de trabajo.

Este conjunto de datos también sirvió como entrada para las heurísticas convencionales. Para evaluar las heurísticas, se usó PARSIFAL [9], un software provisto por Morton y Pentico para resolver diferentes problemas de planificación. Los mejores resultados obtenidos, se utilizaron como cotas superiores (Benchmarks) en los problemas de planificación que se deseaban resolver.

2. Técnicas de multirecombinación

Se implementaron AEs con técnicas de recombinación simple, donde el operador de crossover se aplica sólo una vez a dos padres y produce dos descendientes (SCPC) y con técnicas de multirecombinación. A continuación se describen las principales características de las mismas. Multiple Crossovers per Couple (MCPC) [6] y Multiple Crossovers on Multiple Parents (MCMP) [7] son métodos con multirecombinación, que mejoran la performance de los AEs reforzando y balanceando la exploración y explotación realizada en el proceso de búsqueda. En particular, MCMP es una extensión de MCPC donde se introduce la propuesta de múltiples padres, realizada por Eiben

[2, 3, 4]. Resultados obtenidos en diversos problemas de optimización mono-objetivo y multi-objetivo, indican que el espacio de búsqueda es eficientemente explotado por las múltiples aplicaciones del operador de crossover y eficazmente explorado por el mayor número de muestras provistas por el conjunto de padres.

MCMP-SRI es una variante de MCMP, donde se trata de mantener un mejor equilibrio entre la exploración del espacio de posibles soluciones y la explotación de soluciones ya encontradas, combinando el material genético de un súper individuo, denominado *stud*, con cromosomas inmigrantes creados aleatoriamente [10]. De la vieja población se selecciona por medio de selección proporcional al *stud* y se lo inserta en el pool de apareamiento. Se completa el pool de apareamiento con n_2-1 padres creados aleatoriamente (*random immigrants*). El *stud* se aparea con todos los otros padres, las parejas se cruzan mediante el operador de crossover que se esté utilizando y se crean 2*(n_2-1) hijos. El mejor de esos 2*(n_2-1) hijos es almacenado en un pool de hijos temporario. La operación de crossover se repite n_1 veces con probabilidad p_c, para diferentes puntos de corte cada vez, hasta que se complete el pool de hijos. Los hijos no están expuestos a mutación. Finalmente, el mejor hijo creado desde n_2 padres con n_1 crossovers, es insertado en la nueva población.

Una novedosa técnica de multirecombinación, que opera de manera muy similar a como lo hace MCMP-SRI es MCMP-SRSI. La única diferencia existente entre las mismas, es que la última, permite la inserción de semillas (conocimiento específico del problema). Una semilla es una planificación, codificada de acuerdo con la representación de cromosoma que se esté usando, y la misma ha sido obtenida, utilizando alguna heurística convencional para la cual hayan pruebas teóricas o datos empíricos que demuestren, que dicha heurística es buena para resolver instancias del problema que se esté tratando.

3. Representación de soluciones y operadores genéticos

En los AEs, desde la perspectiva de la representación del cromosoma, existen muchas alternativas para el problema general de scheduling. Con respecto a la representación de las soluciones, las formas de codificación pueden ser categorizadas en primera instancia, como representaciones directas e indirectas.

En el caso de la representación indirecta de soluciones, el algoritmo trabaja con una población de soluciones codificadas. Debido a que la representación no provee directamente una planificación, se necesita un planificador para transformar un cromosoma en un plan, validarlo y evaluarlo. El planificador garantiza la factibilidad de la solución y su trabajo depende de la cantidad de información incluida en la representación. Se trabajó con tres alternativas distintas de representaciones indirectas; la primera utiliza permutaciones y las dos restantes emplean decodificadores.

Representación basada en Permutaciones

Las permutaciones, describen una lista de prioridades de tareas. El planificador toma la primera tarea de la lista y la asigna a un procesador disponible. Si dos o más procesadores están libres, aquél con el índice más bajo será el encargado de procesar la tarea.

Representación basada en un decodificador de prioridades de despacho de procesadores

En esta representación, se codifica una planificación en el cromosoma de manera que, la tarea indicada por la posición del gen, es asignada al procesador indicado por su alelo correspondiente. De esta manera, el cromosoma da instrucciones a un decodificador acerca de cómo construir una planificación factible. Para este caso la decodificación se lleva a cabo como se describe a

continuación. Se recorre el cromosoma de izquierda a derecha y se asigna la tarea correspondiente al procesador indicado, tan pronto como el mismo esté disponible. De esta manera una prioridad se establece, indicando a qué procesador se le asigna una tarea determinada. Aún cuando existan otros procesadores disponibles la tarea deberá esperar hasta que el procesador que tiene asignado quede libre.

Representación basada en un decodificador de prioridades en una lista de tareas

En esta representación el cromosoma es un vector de *n* componentes, donde la *i*-ésima componente es un número entero perteneciente al rango 1..(*n*–*i*+1). El cromosoma es interpretado como una estrategia para extraer tareas de una lista *L*, de tareas ordenadas (por ejemplo, canónicamente) y construir a partir de ella una permutación. Brevemente se explica cómo funciona el decodificador. Dado un conjunto de *n* tareas representadas por la lista *L* y un cromosoma *C*, los valores de los genes en el cromosoma *C* indican las posiciones en la lista *L*. Recorriendo de izquierda a derecha el cromosoma, el decodificador construye una lista de prioridad *L'*, tomando de *L* aquellos elementos cuya posición es indicada por el valor del gen. Esos elementos son puestos en *L'* y son borrados de *L*. Se continúa con este proceso hasta que no queden elementos en *L*, esto debe ocurrir conjuntamente al alcanzar el final del cromosoma. La Fig. 1, muestra la lista *L*, el cromosoma *C* y la permutación resultante en *L'*.

L =	1	2	3	4	5	6
C =	3	2	2	1	1	1
L' =	3	2	4	1	5	6

Fig.1. L es la lista de tareas, L' es la lista de prioridad de tareas

Operadores genéticos

La clase de representación de cromosoma usada impone restricciones acerca de los operadores genéticos a utilizar. Cuando se usaron permutaciones, el operador de crossover empleado en las técnicas de recombinación SCPC, MCPC, MCMP-SRI y MCMP-SRSI fue *Partial Mapped Crossover* (PMX). Para MCMP el operador de crossover utilizado fue *Controlled Uniform Scanning Crossover* (CUSX). La técnica de mutación que se empleó fue Swap. De esta manera nos aseguramos de que los hijos resultantes del intercambio genético fueran permutaciones válidas.

Las representaciones basadas en decodificadores son más flexibles y no necesitan operadores especializados para producir hijos válidos. Los hijos resultantes del entrecruzamiento de padres basados en decodificadores siguen siendo decodificadores. El operador de crossover empleado en las técnicas de recombinación SCPC, MCPC, MCMP-SRI y MCMP-SRSI fue *One Point Crossover* (OPX). Para MCMP el operador de crossover utilizado fue *Uniform Scanning Crossover* (USX). La técnica de mutación que se empleó fue Big Creep.

4. Conclusiones y trabajo futuro

De las tres representaciones con las que se trabajó, sin duda con la que se obtuvieron los mejores resultados es la representación basada en *permutaciones de las tareas*, la cuál, parece ser la más intuitiva cuando se tienen que codificar soluciones en problemas de planificación.

MCMP-SRSI y MCMP-SRI han sido los métodos que han obtenido mejores resultados, y enfoques como MCPC y MCMP han tenido también una buena performance sin llegar a ser tan costosos como los dos mencionados primeramente. Cuando la representación elegida es adecuada

para el problema, como eran en nuestro caso las permutaciones, con una técnica de recombinación simple como SCPC se obtienen muy buenos resultados sobre todo desde un punto de vista beneficio/costo.

Considerando las tres representaciones de cromosoma utilizadas y todos los objetivos estudiados, en general todos los métodos de recombinación implementados tuvieron mejor rendimiento en P_5 con respecto de P_2. Podríamos conjeturar que esto sucede dado que en P_5, con una mayor cantidad de recursos, el número de soluciones alternativas de buen y similar fitness es mayor lo que favorece el proceso de búsqueda.

En el futuro se seguirá trabajando en ambientes de planificación de máquinas idénticas en paralelo, con instancias de trabajo de 100 jobs o mayores, con parametrización estática vs. dinámica, con el mismo conjunto de funciones objetivo que se ha estudiado y con una representación basada en permutaciones. También, se contempla la posibilidad de estudiar problemas de planificación parcial o totalmente dinámicos.

5. Referencias

[1] Beasley J.E., *"Single Machine Total Weighted Tardiness Problem"*, OR-Library, http://mscmga.ms.ic.ac.uk/jeb/pub/wt40.txt

[2] Eiben A.E., Raué P.E., and Ruttkay Z., "*Genetic algorithms with multi-parent recombination*", *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature*, Springer-Verlag, 1994, number 866 in LNCS, pp. 78-87.

[3] Eiben A.E., Van Kemenade C.H.M., and Kok J.N., "*Orgy in the computer: Multi-parent reproduction in genetic algorithms*", Proceedings of the *3rd European Conference on Artificial Life*, Springer-Verlag, 1995, number 929 in LNAI, pp. 934-945.

[4] Eiben A.E. and. Bäck T., "*An empirical investigation of multi-parent recombination operators in evolution strategies*", Evolutionary Computation, 5(3):347-365, 1997.

[5] Esquivel S., Gatica C., Gallard R., "*Evolutionary Approaches with Multirecombination for the Parallel Machine Scheduling Problem*", XX Conferencia Internacional de la Sociedad Chilena de Ciencias de la Computación, Noviembre 2000, Santiago, Chile. IEEE Publishing Co, pp. 1–6.

[6] Esquivel S., Leiva A., Gallard R., "*Multiple Crossover per Couple in Genetic Algorithms*", Proceedings of the *Fourth IEEE Conference on Evolutionary Computation (ICEC'97)*, Indianapolis, USA, April 1997, pp. 103-106.

[7] Esquivel S., Leiva A., Gallard R., "*Multiple crossovers between multiple parents to improve search in evolutionary algorithms*", Proceedings of the *Congress on Evolutionary Computation (IEEE)*, Washington DC, 1999, pp. 1589-1594.

[8] Esquivel S., Gatica C. y Gallard R., "*Performance of Evolutionary Approaches for Parallel Task Scheduling under Different Representations*", Lecture Notes in Computer Science, LNCS 2279, pp. 41-50, Springer Verlag, April 2002.

[9] Morton T., Pentico D., *"Heuristic scheduling systems"*, Wiley series in Engineering and technology management, John Wiley and Sons, INC, 1993.

[10] Pandolfi D., De San Pedro M., Villagra A., Vilanova G., Gallard R., "*Studs mating immigrants in evolutionary algorithm to solve the earliness-tardiness scheduling problems*", En Cybernetics and Systems del Taylor and Francis Journal, (U.K.), pp. 391-400, June 2002.

[11] Pinedo M., *"Scheduling: Theory, Algorithms and System"*, Prentice Hall, First edition, 1995.