

Constructor Paralelo de Planificaciones

Garriga D., Huerga E., Printista M., Gallard R.

Proyecto UNSL-338403¹
Departamento de Informática
Universidad Nacional de San Luis
Ejército de los Andes 950
5700 - San Luis
Argentina
E-mail: {}@unsl.edu.ar
Tel.: ++54 652 20823
Fax: ++54 652 30224

Scheduling es un área activa de investigación en Inteligencia Artificial Aplicada. El interés en esta área es *planificar adecuadamente* la asignación de un número limitado de recursos a determinadas tareas que se deben desarrollar en el tiempo. Existen un número de técnicas heurísticas que han realizado contribuciones notables al problema de scheduling, entre ellas los algoritmos genéticos y otras variantes de la Computación Evolutiva. Estos algoritmos tienen la capacidad de descubrir *planificaciones* cercanas a las óptimas mucho más rápido debido a su paralelismo implícito. Varias implementaciones ya realizadas en esta dirección muestran que los algoritmos genéticos trabajan mejor cuando en la representación del cromosoma que utilizan se incorpora conocimiento específico del problema de modo tal que permita trabajar con operadores de recombinación avanzados. La elección de este tipo de representación indirecta del cromosoma requiere de un *decodificador* o "*schedulers builder*", el cual transforme la representación del cromosoma en una *planificación* legal. En otras palabras los *schedule builder* deben garantizar la factibilidad y consistencia de cada *planificación*.

En este trabajo, en primer lugar, se discute el diseño de **schedule builders** aplicado a un problema Clásico de scheduling: Job Shop Scheduling (JSP) y para el cual se ha considerado una representación de cromosoma indirecta de tres tipos diferentes. En el primer tipo, los cromosomas representan solamente el *ordenamiento* de jobs. En el segundo tipo, además del ordenamiento de jobs, los cromosomas incluyen un *plan de proceso* específico a seguir para cada orden de job. En el tercer tipo incluyen ordenes, un plan específico para cada orden y el *conjunto de recursos* utilizados en la ejecución de las operaciones definidas en el plan. La actividad del *schedule builders* va a depender de la cantidad de información codificada en la representación del cromosoma ya que en el primer y segundo tipo implicara una búsqueda local de información no suministrada en el cromosoma. En este contexto y para cada tipo de representación se muestra como se han implementado las tres funciones básicas que un Schedule Builder debe realizar: la Búsqueda local de Información, el Cumplimiento de Restricciones y la Construcción de Schedules.

¹ The Research Group is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

Por otro lado se consideró la relación cooperativa existente entre algoritmos genéticos y scheduler builder. El AG. requerirá servicio del schedule builder, enviando cromosomas. El schedule builder tomará los cromosomas, verificará su validez, obtendrá su función de evaluación y finalmente los retornará al algoritmo genético. La ejecución de esta secuencia, igual para todos los cromosomas recibidos, se realiza en forma paralela, tan pronto arriben los cromosomas al scheduler builder. En una situación, donde el AG requiera servicio del scheduler builder entregando n cromosomas, se reduciría el tiempo de computación a la n ésima parte del tiempo total que implicaría analizar los cromosomas en forma secuencial en un único procesador. Con el objetivo de lograr tiempos de computación más eficientes y desde luego más convenientes para esta relación cooperativa se implementó una paralelización del scheduler builder. Finalmente, este trabajo muestra la arquitectura que posibilita la ejecución Paralela del Scheduler Builder. Se discuten detalles de la implementación realizada sobre un Sistema de Computación Paralela, el PowerMouse de Parsystec. Finalmente se muestra un análisis estadístico y comparativo de la performance de la ejecución Paralela del scheduler builder

Palabras claves: *Schedulers Builders*, Algoritmos Genéticos, Sistemas Paralelos, Arquitectura del Sistema. Codificación Indirecta de Cromosoma.

BIBLIOGRAFÍA

1. Baker K.R.- Introduction to sequencing and Scheduling. New York. Wiley. 1974.
2. Conway R.W., Maxwell W.L. and Miller W.- Theory of Scheduling. Addison Wesley. 1993.
3. Davis L. – Handbooks of Genetic Algorithms. New York . Van Nostrand Reinhold. Foster Ian T. Designing and Building Parallel Programs- Addison Wesley. 1995.
4. Geist A., Benguelin A., Dongarra J., Jiang W., Manchek R., Sunderam V. -. PVM: Parallel Virtual Machine. The Mit Press . Cambridge. Massachusetts. 1994.
5. Geist A., Benguelin A., Dongarra J., Jiang W., Manchek R., Sunderam V. .- PVM3: User's Guide and Reference Manual Oak Ridge National Laboratory: Tennessee.
6. JáJá J.- An introduction to Parallel Algorithms. Reading Massachusetts. Addison Wesley. 1973
7. Kumar A., Morton T.E. - Dynamic Job Routing in a Heavily Loaded Shop. GSIA. Carnegie Mellon University. Pittsburgh.PA. 1991.
8. Morton T and Pentico D. - Heuristic Scheduling Systems - JOHN WILEY & SONS. 1991.
9. Pinedo M,; Scheduling - Theory, Algorithms, and Systems - PRENTICE HALL - 1995.
10. Stevens R.W.- Unix Networks Programming. Prentice Hall-Englewood Cliff. 1990.