

Algoritmos Evolucionarios aplicados al Problema de Secuenciamiento de Flow Shop

Daniel Pandolfi

LabTEM - Unidad Académica Caleta Olivia
Universidad Nacional de la Patagonia Austral
Caleta Olivia– Santa Cruz - Argentina
dpandolfi@uaco.unpa.edu.ar

Guillermo Leguizamon

LIDIC – Facultad de Cs. Fco. Matemáticas y Naturales
Universidad Nacional de San Luis
San Luis, Argentina.
legui@unsl.edu.ar

Resumen

Los algoritmos de Estimación de Distribuciones (EDAs - *Estimation of Distribution Algorithms*) son una clase de algoritmos basados en el paradigma de Computación Evolutiva (CE) que sustituyen los mecanismos de variación (cruce y mutación) utilizados tradicionalmente por Algoritmos Evolutivos (AEs). La población de nuevas soluciones se genera a través de la simulación de una estimación de probabilidad producida por la información de las soluciones generadas en iteraciones pasadas

Por su parte, el problema de secuenciamiento de *Flow Shop* y conocido como FSSP (*Flow Shop Sequencing Problem*) ha convocado la atención de muchos investigadores en los últimos años. En FSSP, un conjunto de tareas deben seguir el mismo orden en una rutina de procesamiento para un conjunto de máquinas con el objetivo de optimizar alguna variable de performance (*makespan, tardiness, lateness*). En el caso de *makespan*, se trata de minimizar el tiempo de salida de la última tarea en la última máquina. Para máquinas mayores e iguales a tres el problema se transforma en *NP-hard*, conforme se incrementa el número de tareas.

Este trabajo propone la exploración de distintos tipos de algoritmos evolucionarios aplicados a la resolución del problema de secuenciamiento de *Flow Shop*.

1. INTRODUCCIÓN

Los problemas de planificación abarcan una variedad de problemas de optimización en campos tales como operaciones de producción y despacho en la industria manufacturera, sistemas distribuidos y paralelos, logística y tráfico. Algunos de ellos pueden incluirse dentro de la clase general de problemas de *scheduling* (Garey, 1979). En general, el *scheduling* consiste en la asignación de tareas, a través del tiempo, cuando la disponibilidad de recursos es limitada, donde ciertos objetivos deben ser optimizados y varias restricciones deben ser satisfechas.

Los problemas de *scheduling* son de aplicación en las organizaciones y en la industria y en consecuencia tiene un fuerte impacto económico y social. El estudio de estos problemas data aproximadamente de 1950 donde investigadores de ingeniería industrial, investigación operativa, y administradores desarrollaron nuevos enfoques y algoritmos que tienen como objetivo principal la reducción de los costos de producción en la industria (Leung J., 2004).

Muchos algoritmos eficientes han sido desarrollados para encontrar soluciones óptimas, aunque para tamaños pequeños de este tipo de problema. Por ejemplo, se pueden mencionar los trabajos de Jackson (1955), Johnson (1954). Con el advenimiento de la teoría de complejidad (Cook S., 1971), muchas investigaciones sobre dicha temática se han desarrollado debido a la inherente dificultad para resolver esta clase de problemas de forma exacta. Muchos de los problemas de *scheduling* son computacionalmente complejos y el tiempo requerido para calcular una solución óptima se incrementa con el tamaño del problema (Morton y Pinedo 1993; Pinedo M. 1995). Además, se ha

demostrado, por cierto, que muchos problemas de scheduling pertenecen a la clase de NP-Hard (Brucker P., 2004; Lenstra J. et al, 1978).

En el problema de secuenciamiento de *Flow Shop* existen m máquinas en serie y n tareas (jobs) que deben ser procesadas en cada una de las m máquinas. Todos los jobs tienen el mismo routing, es decir, primero tienen que procesarse en la máquina 1, luego debe procesarse en la siguiente máquina y así sucesivamente. Luego de haberse completado en una máquina, una tarea se pone en cola de la próxima, en general la disciplina de cada cola es FIFO.

La Computación Evolutiva es un campo de investigación emergente que provee nuevas metaheurísticas para la resolución de problemas de optimización donde los enfoques tradicionales hacen al problema computacionalmente intratable. Reflejando la relevancia industrial de estos problemas se han reportado en la literatura una variedad de métodos basados en algoritmos evolutivos de resolución de problemas de Scheduling (Syswerda G.; 1991; Branke J. y Mattfeld D. 2000; Burke et al, 2001; Cowling P. et al, 2002).

2. SCHEDULING EVOLUCIONARIO

Algoritmos Evolutivos

Los Algoritmos Evolutivos (AEs) son una de las metaheurísticas más ampliamente difundidas y estudiadas (Bäck T., 1996). Estas, como muchas otras metaheurísticas poblacionales, pueden ser mejoradas en su diseño a fin de realizar una exploración más eficiente del espacio de búsqueda. En el caso de los AEs, un adecuado desempeño de los mismos, depende en gran medida de los operadores y/o mecanismos de exploración involucrados y que adecuadamente implementados, pueden dar lugar a versiones más eficientes.

El enfoque denominado canónico de un AE, está basado en representación binaria y el proceso de recombinación es operado sobre una par de soluciones que son seleccionadas desde una población. Tal enfoque es conocido como SCPC (*Single Crossover per Couple*).

Este enfoque, ha sido exitosamente aplicado para resolver problemas de *Flow Shop*, sustituyendo la representación binaria por una representación basada en permutaciones. Tsujimura et al (1995) muestra la calidad de los AEs en contraste con otras heurísticas convencionales, aplicando operadores de cruce ampliamente conocidos en representaciones basadas en permutaciones, tales como *Partially-Mapped Crossover -PMX-* (Goldberg y Lingle; 1987), *Order Crossover -OX-* (Davis L., 1991) y *Cycle Crossover -CX-* (Oliver et al. 1986).

Debido a que el problema de *Flow Shop* es esencialmente un problema de secuenciamiento de permutaciones, una permutación puede ser usada como un esquema de representación de los cromosomas, lo cual representa una forma natural de expresar la solución del problema. La representación, también llamada representación por orden, puede producir un descendiente ilegal si se opera con un cruce tradicional como OPX (*One Point Crossover*). Consecuentemente, se han propuesto varios operadores de cruce que permiten a partir de dos soluciones codificadas como permutaciones producir otras soluciones válidas, tales como los mencionados anteriormente PMX, OX, y CX entre otros.

Reeves (1995) propone un enfoque híbrido, el cual inserta una semilla en la población inicial, generada por la heurística NEH (Morton y Pentico, 1993). También ha mostrado en su implementación un operador de cruce llamado *one-cut-point crossover* (OCPX) que es capaz de operar con soluciones codificadas como permutaciones. Luego de unas pocas experimentaciones Reeves observó que el enfoque evolutivo propuesto producía resultados comparables a los producidos por una heurística como ser *Simulated Annealing* (SA) para FSSP en los *benchmarks* de Taillard (1993). Los resultados reportados mostraron que el enfoque evolutivo se comportaba sensiblemente mejor para problemas de gran tamaño, produciendo muy buenas soluciones y de forma rápida.

Otra forma de resolver el problema de FSSP es presentada por Grefenstette (1991), representando las soluciones como decodificadores (*decoders*), conocida como representación ordinal. Bajo este enfoque un cromosoma da información a un decodificador sobre la manera de construir una solución factible. Si bien, los decodificadores son principalmente usados en problemas con restricciones, presentamos este esquema basado en representación ordinal porque es fácil de implementar y produce soluciones factibles aplicando diferentes operadores de cruce convencionales (tales como OPX) y haciendo innecesario el uso de mecanismos de reparación (como los embebidos en PMX, OX y CX). Aquí un cromosoma es un vector- n donde la i -ésima componente es un entero en el rango $1..(n-i+1)$. El cromosoma es interpretado como una estrategia para extraer items desde una lista ordenada L para construir una permutación. Por ejemplo si $L=(1,2,3,4)$, luego el cromosoma [3121] y [2211] son interpretados como las permutaciones [3142] y [2314] respectivamente.

Algoritmos de Estimación de Distribuciones.

Los EDAs, fueron introducidos originalmente por Mühlenbein y PaaB (1996) como una extensión de AEs, donde no se requiere de los operadores de variación tradicionales de cruce y mutación para la generación de nuevas soluciones. La nueva población de soluciones se obtiene de la simulación de una distribución de probabilidad, la cual es estimada a partir de la información generada por las soluciones generadas a partir poblaciones en iteraciones pasadas.

La motivación en la utilización de este enfoque esta dada por dos aspectos principales. El primero, se refiere a la dificultad de los EAs en trabajar con problemas de decepción y no separabilidad, y el segundo se refiere a que a la implícita búsqueda llevada a cabo por los operadores de variación de los EAs, se le puede añadir información a cerca de la correlación entre las variables del problema. (Larrañaga y Lozano 2002; Mühlenbein et al 1999).

Esta nueva clase de algoritmos es conocida también en la literatura como *Probabilistic Model Building Genetic Algorithms* –PMBGA- (Pelikan et al 1999) e *Iterated Density Estimators* *Iterated Density Estimators Evolutionary Algorithms* –IDEAS- (Bosman y Thierens, 1999)

Los EDAs pueden ser clasificados de acuerdo al tipo de interacción entre las variables permitidas en el modelo de distribución de probabilidad (Larrañaga y Lozano 2002). Así, podemos distinguir distintos enfoques en los EDAs, sin dependencias entre variables, con interacciones de a pares, o con múltiples interacciones.

Los EDAs sin dependencias, es la forma más simple de estimar una distribución, donde se asume que las variables son independientes unas de otras. Algunos ejemplos de estos algoritmos son *Univariate Marginal Distribution Algorithm* -UMDA- (Mühlenbein y Voigt, 1996) y *Population Based Incremental Learning algorithm* –PBIL- (Bajula 1994).

Los EDAs con dependencias bivariantes asumen que las variables en un problema no son independientes entre sí. Estos algoritmos simulan la distribución de probabilidad considerando que existe una dependencia entre dos variables. En este tipo de algoritmos podemos mencionar, *Mutual Information Maximization Algorithm for Input Clustering* -MIMIC- (De Bonet et al., 1997) y *Bivariate Marginal Distribution Algorithm* -BMDA- (Pelikan y Mühlenbein 1999).

La aplicación los EDAs a problemas de optimización han reportado en muchos de los resultados mejoras a la aplicación de los enfoques evolutivos, tales como algoritmos genéticos (Larrañaga y Lozano 2002). Sin embargo, los EDAs presentan su principal debilidad en el costo computacional que insumen a comparación de algoritmos genéticos más clásicos.

3. LINEA DE INVESTIGACIÓN Y TRABAJOS FUTUROS

Esta línea de trabajo tiene como principal objetivo el estudio, desarrollo e implementación de Técnicas en Computación Evolutiva básicas y avanzadas (Algoritmos Genéticos, y Algoritmos Basados en Estimación de Distribuciones) y su aplicación a la resolución de problemas de

secuenciamiento *Flow Shop*. Ambos enfoques tratados en este trabajo, son lo suficientemente flexibles para incorporar conocimiento desde distintas fuentes de información.

Trabajos futuros, tienen como objetivo explorar y combinar distintos mecanismos propios de los algoritmos AEs y EDAs para lograr una mejora importante respecto de la eficiencia en la exploración del espacio de búsqueda en problemas de secuenciamiento.

4. AGRADECIMIENTOS

El primer autor agradece a la Universidad Nacional de la Patagonia Austral, de la cual recibe financiamiento y apoyo continuo y además, a la cooperación de los integrantes del LabTEM que continuamente proveen de nuevas ideas y críticas constructivas. El segundo autor agradece el constante apoyo brindado por la Universidad Nacional de San Luis y la ANPCyT que financian sus actuales investigaciones, así como de los integrantes del LIDIC del cual recibe continuo apoyo.

REFERENCIAS

- Bäck T. *Evolutionary Algorithms in theory and practice*. New York:Oxford University Press, (1996).
- Bajula, Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning, *Technical Report CMU-CS*, pp. 94-163, Carnegie Mellon University, (1994).
- Bosman A.N. and Thierens. D.; Linkage information processing in distribution estimation algorithms. *In Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, 1, pp 60–67. Morgan Kaufmann Publishers, San Francisco, LA. (1999).
- Branke J., Mattfeld D., Anticipation in Dynamic Optimization: The Scheduling Case. *Proceedings of VI PPSN*, pp 253-262, (2000).
- Brucker P., *Scheduling Algorithms*, 3rd ed. Springer-Verlag New York, (2004).
- Burke E.K., De Causmaecker P., Petrovic S., Vanden Berghe G., Fitness Evaluation for Nurse Scheduling Problems, *Proc Congress on Evolutionary Computation, CEC2001, Seoul, IEEE Press*, pp 1139-1146, (2001).
- Campbell H., Dudek R., Smith M., “A heuristic algorithm for the n job m machine sequencing problem”, *Management Science* 16, pp. 630-637 (1970).
- Cook S.A. The complexity of theorem-proving procedures, *Proceedings of 3rd Annual ACM Symposium on Theory of Computing, Association for Computing Machinery, New York*, pp 151-158 (1971).
- Cowling P. Kendall G. Han L., An Investigation of a Hyperheuristic Genetic Algorithm Applied to a Trainer Scheduling Problem, *Proc Congress on Evolutionary Computation, CEC2002, Hawaii, IEEE Press*, pp 1185-1190, (2002).
- Davis L., *Handbook of Genetic Algorithms*, New York: Van Nostrand Reinhold Computer Library, (1991).
- De Bonet J. S., Isbell C. L., and Viola P.; MIMIC: Finding optima by estimating probability densities. *Advances in Neural Information Processing Systems*, 9. (1997).
- Garey R., Johnson D.; *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman & Co., San Francisco, CA, (1979).
- Goldberg, D. and R. Lingle, Alleles, loci and the traveling salesman problem, *in Proceeding of the First International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, pp. 154-159, Hillsdale, NJ, (1987).

- Grefenstette J. J., Gopal R., Rosmaita B., Van Gutch D.; Genetic Algorithm for the TSP; *Proceedings of the 1st Int. Conf. on Genetic Algorithms*, Pittsburgh , PA. (1991).
- Gupta J., “A functional heuristic algorithm for the flowshop scheduling problem”, *Operational Research Quarterly* 22, pp. 39-48 (1971).
- Jackson J. R.; Scheduling a production line to minimize maximum tardiness, *Research Report 43, Management Science Research Project, University of California; Los Angeles*, (1955).
- Johnson S. M. Optimal two and three stage production; *Naval Research Logistics Quarterly*, **1**, pp 61-67, (1954).
- Larrañaga P. and Lozano J.A.; Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. *Kluwer Academic Publishers*, (2002).
- Lenstra J. K., Rinnooy Kan A. H., Computational complexity of scheduling under precedence constraints, *Operations Research*, **26**, pp 22-35, (1978).
- Leung Joseph. Handbook of Scheduling: Algorithms, Models and Performance Analysis, *Chapman & Hall/CCR Computer and Information Sciences Series* (2004).
- Madera J., Dorronosoro B.; Estimation of distribution algorithms, Metaheuristic procedures for training neural networks; *Springer Science and Business Media*, (2006).
- Morton T., Pentico D., “Heuristic scheduling systems”, *Wiley series in Engineering and technology management. John Wiley and Sons, INC* (1993).
- Mühlenbein H., Mahnig T., and Ochoa A.. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, **5**, pp 215–247, (1999).
- Mühlenbein H. and Paaß G.; From recombination of genes to the estimation of distributions I. Binary parameters. *In Lecture Notes in Computer Science 1411:Parallel Problem Solving from Nature - PPSN IV*, pp 178–187 (1996).
- Mühlenbein H. and Voigt H.M.; Gene pool recombination in genetic algorithms. *Metaheuristics: Theory and applications*, pp 53–62, (1996).
- Nawaz M., Ensore E., Ham I., “A heuristic algorithm for the m-machine n-job flow shop sequencing problem”. *Omega* vol **II**, pp 11-95 (1983).
- Oliver, I., D. Smith, and J. Holland, A study of permutation crossover operators on the traveling salesman problem, *in European Journal of Operational Research*, pp. 224-230 (1986).
- Palmer D., “Sequencing jobs through a multistage process in the minimum total time- A quick method of obtaining a near optimum”, *Operational Research Quarterly* **16**, pp 101-107 (1965).
- Pinedo Michael – “Scheduling- Theory, Algorithms, and Systems. *Prentice Hall International in Industrial and System Engineering* (1995).
- Reeves C., A genetic algorithm for flow shop sequencing, *Computers and Operations Research*, **22**, pp 5-13 (1995).
- Syswerda G.; Schedule optimization using genetic algorithms, *Handbook of Genetic Algorithms* , Van Nostrand Reinhold, New York, **21**, pp 332-349. 1991.
- Taillard, E. Benchmarks for basic scheduling problems, *European Journal of Operational Research*, **64**, pp 278-285 (1993).
- Tsujimura Y., Gen M., Kubota E., Flow shop scheduling with fuzzy processing time using genetic algorithms. *The 11th Fuzzy Systems Symposium* pp 248-252. Okinawa (1995).