

UNIVERSIDAD NACIONAL DE LA PAMPA

FACULTAD DE INGENIERÍA



PLAN DE TRABAJO CORRESPONDIENTE A LA TESIS
EN INGENIERÍA EN SISTEMAS

**Un algoritmo de evolución diferencial híbrido
para resolver el problema de planificación de
tareas flexible**

Autor

Franco Marcelo Morero

Fecha: abril, 2020

Director

Dra. Salto Carolina

1. Descripción del proyecto y justificación

Las técnicas de optimización juegan un rol importante en el campo de la ciencia y la ingeniería. En las últimas décadas, se han desarrollado varios algoritmos para resolver problemas de optimización complejos. La biología evolutiva o el comportamiento de enjambres inspiraron la mayoría de esos métodos. En este marco evolutivo o de inteligencia de enjambre se han propuesto varias clases de algoritmos que incluyen: algoritmos genéticos [1, 2], algoritmos meméticos [3], evolución diferencial (DE) [4], optimización de colonias de hormigas (ACO) [5], optimización por enjambre de partículas (PSO) [6], algoritmo de colonia de abejas artificiales (ABC) [7], entre otros.

La evolución diferencial es un algoritmo evolutivo desarrollado para resolver funciones de espacios continuos. Es un método de búsqueda estocástico. Comparado con otros algoritmos metaheurísticos basados en población, DE enfatiza más la mutación que la recombinación. DE muta un vector con una ayuda de pares de vectores elegidos aleatoriamente en la misma población. La mutación guía a los vectores hacia el óptimo global. La distribución de la diferencia entre vectores muestreados aleatoriamente se determina por la distribución de esos vectores. Como la distribución de esos vectores está principalmente determinada por una determinada función objetivo, los sesgos en los que DE intenta optimizar el problema coinciden con los de la función a optimizar.

DE ha sido recientemente usado para optimización global en una gran variedad de áreas de problemas tal como planificación [8], diseño ingenieril [9], problemas de optimización con restricción [10], entre otros [11, 12]. En particular, dentro de los ambientes de manufactura, nos encontramos con el problema de planificación job shop flexible (FJSSP). Está probado que este problema es NP-duro [13], debido a que se tiene que asignar cada operación de un trabajo a una máquina apropiada (problema de ruteo) y secuenciar las operaciones asignadas en cada máquina (problema de secuenciamiento). El objetivo que se persigue es minimizar el tiempo necesario para completar todos los trabajos. A medida que aumenta la cantidad de trabajos, por consiguiente la cantidad de operaciones, y la cantidad de máquinas disponibles, la complejidad del problema crece y por consiguiente la complejidad computacional para resolverlo. Esto trae aparejado tiempos de procesamiento altos. Una manera de hacer frente a estos tiempos de procesamiento alto es utilizar alguna técnica de paralelización que permita distribuir el trabajo en los distintos procesadores y/o cores de nuestro equi-

po informático. Una de las estrategias es utilizar una configuración paralela maestro-esclavos, utilizando una interfaz de programación de aplicaciones (API), como lo es openMP, para la programación multiproceso de memoria compartida.

2. Objetivos

El objetivo general de este trabajo es: diseñar un algoritmo DE híbrido para resolver el FJSSP utilizando estrategias de distribución del trabajo computacional.

Como objetivos específicos, para alcanzar el objetivo general, se plantean los siguientes:

- Realizar un análisis del estado del arte sobre algoritmos DE y propuestas paralelas.
- Realizar un análisis del estado del arte en la resolución del FSJJP con algoritmos evolutivos.
- Diseñar e implementar el DE híbrido para el FSJJP.
- Diseñar e implementar el DE híbrido paralelo para el FSJJP.
- Realizar la experimentación correspondiente y el análisis de los resultados obtenidos.

3. Asignaturas relacionadas

Para alcanzar los objetivos propuestos, se hará uso de los conocimientos obtenidos en las siguientes asignaturas:

- Estructura de Datos y Algoritmos
- Probabilidad y Estadística
- Sistemas Distribuidos I
- Sistemas Distribuidos II
- Sistemas Inteligentes

4. Metodología

A continuación se detalla la metodología que se adoptará para realizar este trabajo, detallando las fases en las que se divide.

La primera fase consistirá en la revisión de material bibliográfico relacionado con algoritmos evolutivos, haciendo especial énfasis en algoritmos de evolución diferencial, a fin de adquirir conocimientos necesarios sobre estas técnicas para, en etapas posteriores, poder concretar una propuesta algorítmica. Además, en esta etapa de revisión se estudiará el estado del arte en la problemática abordada FJSSP mediante la solución con DE y sus variantes híbridas. Esta fase permitirá identificar particularidades de las distintas alternativas de solución que resulten más prometedoras para resolver el FJSSP, y que se puedan integrar en una propuesta posterior.

En una segunda fase, será necesario el análisis y estudio de posibles paralelizaciones de metaheurísticas, con un enfoque centrado en paralelizar algoritmos de evolución diferencial, con el fin de poder lograr un algoritmo híbrido paralelo con grandes ventajas en tiempo y utilización de recursos computacionales.

En una tercera fase, se continuará con el diseño y la implementación del DE híbrido paralelo para resolver el FJSSP siguiendo el paradigma procedural, haciendo hincapié en la representación del problema.

Concretada la fase anterior, la siguiente consistirá en la puesta a punto del algoritmo desarrollado. Para lo cual se deben realizar ejecuciones preliminares y analizar los resultados de las mismas para determinar la mejor configuración de parámetros. Una vez definidos los mismos, se procederá al desarrollo de la experimentación completa para finalizar con el análisis de los resultados obtenidos y su estudio estadístico.

En la fase final se realizará la escritura del informe del proyecto final incluyendo el desarrollo realizado y el análisis de los resultados obtenidos.

5. Cronograma de trabajo

A continuación se detallan las distintas actividades, y su planificación en el tiempo, que surgen para la concreción de los objetivos parciales.

1. Estudiar y analizar el estado del arte en relación a algoritmo DE.
2. Estudiar y analizar el estado del arte en relación a posibles paralelizaciones de

metaheurísticas.

3. Diseñar la forma de paralelizar un algoritmo DE para resolver el FJSSP e implementar el algoritmo propuesto, evaluando alternativas de mejora.
4. Realizar la experimentación y posterior análisis de resultados.
5. Documentar el trabajo realizado.

Las duraciones estimadas (en horas) de las distintas actividades se muestran en la siguiente tabla:

Actividad	1	2	3	4	5
Duración	30	20	50	60	40

Tiempo total: 200 horas.

6. Resultados esperados

Obtener un software que permita resolver el problema de job shop flexible explotando el paralelismo en su ejecución

Referencias

- [1] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.
- [2] Holland J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press; 1992.
- [3] P. C3P Report. 826. Caltech Concurrent Computation Program; 1989. On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms.
- [4] Storn R., Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. 1997;11(4):341–359.

- [5] Colorni A., Dorigo M., Maniezzo V. Distributed optimization by ant colonies. Proceedings of the 1st European Conference on Artificial Life; 1991; Paris, France. pp. 134–142.
- [6] Eberhart R. C., Kennedy J. New optimizer using particle swarm theory. Proceedings of the 6th International Symposium on Micro Machine and Human Science; October 1995. pp. 39–43.
- [7] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, . 39(3):459–471, 2007.
- [8] Gnanavel Babu A., Jerald J., Noorul Haq A. , Muthu Luxmi V., Vigneswaralu T., Scheduling of machines and automated guided vehicles in FMS using differential evolution, *Int. J. Prod. Res.*, 48 ,2010, pp. 4683-4699.
- [9] De Melo V.V., Carosio G.L., Investigating multi-view differential evolution for solving constrained engineering design problems, *Expert Syst. Appl.*, 40, 2013, pp. 3370-3377.
- [10] Ali M., Zhu W.A, Penalty function-based differential evolution algorithm for constrained global optimization, *Comput. Optim. Appl.*, 54, 2013, pp. 707-739.
- [11] Juang J.-G., Yu S.-T., Disturbance encountered landing system design based on sliding mode control with evolutionary computation and cerebellar model articulation controller, *Appl. Math. Model.*, 39, 2015, pp. 5862-5881.
- [12] Yang Z., Yu Q., Dong W., Gu X., Qiao W., Liang X., Structure control classification and optimization model of hollow carbon nanosphere core polymer particle based on improved differential evolution support vector machine, *Appl. Math. Model.*, 37, 2013, pp. 7442-7451
- [13] Garey M. R., Johnson D. S., Sethi R., The complexity of flowshop and jobshop scheduling, *Math. Oper. Res.*, vol. 1, no. 2, 1976, pp. 117–129.