

Propuesta de Proyecto de investigación

Maestría en Ciencias y Tecnologías de la Información

1. Nombre del Proyecto

Soporte para el Elitismo en Algoritmos Evolutivos Multiobjetivo Paralelos

2. Responsables

Dr. Antonio López Jaimes
Departamento de Matemáticas Aplicadas y Sistemas,
Universidad Autónoma Metropolitana Cuajimalpa

Dra. Elizabeth Pérez-Cortés
Departamento de Ingeniería Eléctrica
Universidad Autónoma Metropolitana Iztapalapa

3. Perfil deseable del alumno

El alumno(a) debe tener conocimientos básicos de algoritmos genéticos, buenas habilidades de programación concurrente y facilidad para la lectura de documentos en inglés.

4. Presentación del contexto e identificación de la problemática

La mayoría de los problemas de optimización del mundo real involucran dos o más objetivos (véase por ejemplo, [1][2]) que se tienen que optimizar simultáneamente y que se encuentran en conflicto (i.e., mejorar uno de ellos, empeora algún otro). Como resultado, a diferencia de la optimización de un solo objetivo, en la optimización multiobjetivo no hay una solución óptima sino un conjunto de soluciones compromiso óptimas (llamadas *frente de Pareto*). Estas soluciones son óptimas en el sentido que no es posible mejorar un objetivo sin deteriorar otro.

Los algoritmos evolutivos (AEs) fueron diseñados para resolver problemas de optimización del mundo real en los que las técnicas clásicas de programación matemática tienen un desempeño pobre o incluso no pueden aplicarse [3][4]. Entre las dificultades que los algoritmos evolutivos tratan de resolver se encuentran las siguientes:

- Problemas con espacios de búsqueda no convexos o discontinuos.
- Problemas con varios óptimos locales.
- Problemas con objetivos para los cuáles no hay una expresión explícita sino que son el resultado de una simulación. En algunos casos incluso, se utiliza el dispositivo de hardware mismo para evaluar las soluciones (por ejemplo [5]).
- Presencia de ruido estocástico en la evaluación de los objetivos [6].

Los algoritmos evolutivos son particularmente adecuados para resolver problemas multiobjetivo ya que mantienen simultáneamente un conjunto de soluciones para obtener una muestra del frente de Pareto. Por su parte, los algoritmos de

programación matemática necesitan varias ejecuciones para obtener varios elementos del frente de Pareto y los AEs son menos susceptibles a la forma y continuidad del frente de Pareto [3]. Los AEs diseñados para resolver problemas con varios objetivos son denominados algoritmos evolutivos multiobjetivo (AEMO).

El elitismo es el mecanismo para evitar perder las mejores soluciones encontradas durante la ejecución de un AE debido a efectos estocásticos. Este concepto juega un papel importante en AE modernos ya que junto con la mutación, garantiza convergencia global [3]. En optimización multiobjetivo, la implementación de elitismo es más compleja que en la optimización mono-objetivo puesto que el conjunto de soluciones óptimas puede ser muy grande o infinito en varios casos. Dado que contamos con recursos limitados tanto de memoria y tiempo, si hay más soluciones del frente de Pareto de las que podemos costear, entonces algunas de ellas tienen que ser descartadas.

Existen dos enfoques principales para implementar el elitismo. Uno de ellos combina la población anterior y la nueva, y posteriormente utiliza una selección determinista para mantener a las mejores soluciones en la siguiente generación. De este modo, la población actual mantiene constante su tamaño. En el otro enfoque se mantiene un conjunto externo de soluciones llamado “archivo” que mantiene las mejores soluciones encontradas durante la búsqueda. Si bien usualmente el archivo es más grande que la población actual, su tamaño se restringe debido al tiempo necesario para actualizarlo cuando hay nuevas soluciones y para mantenerlo en memoria.

La estructura de datos utilizada para mantener el archivo es de vital importancia pues cada que el AE genera una solución candidata a pertenecer al frente de Pareto, es necesario comprobar que, o bien alguna solución del archivo es mejor que ésta y por ende debe ser desechada, o bien no es así y es necesario integrarla al archivo. En este último caso debe garantizarse eliminar del archivo aquellas soluciones que son menos buenas que la nueva solución integrada.

Actualmente por limitaciones de tiempo, el número máximo soluciones en el archivo se mantiene constante con un tamaño similar al de la población principal. Sin embargo, en este caso se puede presentar el fenómeno de deterioro del archivo [12]. El deterioro ocurre cuando soluciones que están en el archivo pudieron ser superadas por soluciones que fueron mantenidas anteriormente en el archivo. Por otro lado, en aplicaciones del mundo real donde la evaluación de cada solución representa una inversión de tiempo considerable, es importante no descartar ninguna buena solución ya evaluada. En este escenario y dados los recursos de cómputo paralelo actuales (e.g., computadoras multi-núcleo o *clusters* con miles de procesadores) es factible diseñar un algoritmo de archivado paralelo para mantener un número de soluciones prácticamente ilimitado.

Existen múltiples esfuerzos para encontrar una estructura de datos que permita el mantenimiento eficiente del archivo en un entorno centralizado [7-11] y, en este proyecto, estamos interesados en hacer lo propio para un entorno paralelo. En otras palabras, estamos interesados en diseñar un algoritmo de archivado paralelo de manera que pueda contener un gran número de soluciones ($\geq 5 \times 10^5$).

5. Objetivos generales y específicos

Objetivo general

Diseñar un algoritmo paralelo de archivado capaz de mantener de manera eficiente un gran número de soluciones ($\geq 5 \times 10^5$).

Objetivos específicos

- Identificar los algoritmos de archivado propuestos actualmente que puedan implementarse en paralelo directamente.
- Conocer la eficiencia relativa de los algoritmos de archivado identificados.
- Proponer un algoritmo de archivado paralelo (posiblemente basado en una versión secuencial existente) para ejecutarse en un multiprocesador débilmente acoplado (e.g., un *cluster*).
- Poner el algoritmo implementado a disposición de la comunidad de Computación Evolutiva.

6. Metodología propuesta

1. Análisis de la literatura para compilar y comprender las estructuras de datos (ED) para el soporte del elitismo en AEs.
2. Identificación de los algoritmos de archivado que se puedan adaptar para ejecutarse en paralelo.
3. Implementación del algoritmo de archivado paralelo más eficiente, mismo que servirá como algoritmo de referencia.
4. Diseño e implementación de un nuevo algoritmo de archivado apto para ejecutarse en computadoras multiprocesador.
5. Planteamiento de los escenarios de evaluación de los algoritmos de archivado.
6. Evaluación del algoritmo de archivado propuesto comparándolo con el algoritmo de referencia.
7. Puesta a disposición de la comunidad de CE del algoritmo de archivado propuesto y la documentación pertinente.

7. Calendarización de actividades

Actividad	Trimestre 1			Trimestre 2			Trimestre 3		
Análisis de la literatura para compilar y comprender las ED para el soporte del elitismo en AEs.	■	■							
Identificación de los algoritmos de archivado que se puedan adaptar para ejecutarse en paralelo.		■							
Implementación del algoritmo de archivado paralelo más eficiente.			■	■					
Elaboración del reporte del primer trimestre			■						

Diseño e implementación de un nuevo algoritmo de archivado apto para ejecutarse en computadoras multiprocesador.									
Planteamiento de los escenarios de evaluación de los algoritmos de archivado.									
Evaluación del algoritmo de archivado propuesto.									
Elaboración de reporte del 2o. trimestre									
Puesta a disposición de la comunidad de CE del algoritmo de archivado propuesto y la documentación pertinente.									
Elaboración de la primera versión de la idónea comunicación de resultados.									

8. Infraestructura necesaria y disponible

Se requiere de una computadora con acceso a Internet disponible en el laboratorio de Cómputo Paralelo y Sistemas Distribuidos T-169.

9. Lugar de realización

Laboratorio de Cómputo Paralelo y Sistemas Distribuidos.

10. Referencias

1. Carlos A. Coello Coello y Gary B. Lamont, editores. Applications of Multi-Objective Evolutionary Algorithms. World Scientific, Singapur, 2004. ISBN 981-256-106-4.
2. Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, y Roman Slowinski, editores. Multiobjective Optimization. Interactive and Evolutionary Approaches. Springer. Lecture Notes in Computer Science Vol. 5252, Berlín, Alemania, 2008.
3. Carlos A. Coello Coello, Gary B. Lamont, y David A. Van Veldhuizen. Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, Nueva York, segunda edición, septiembre 2007. ISBN 978-0-387-33254-3.
4. Kalyanmoy Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. En Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, y Jacques Periaux, editores, Evolutionary Algorithms in Engineering and Computer Science, capítulo 8, páginas 135–161. John Wiley & Sons, Ltd, Chichester, Reino Unido, 1999.

5. Piotr Wozniak. Preferences in multi-objective evolutionary optimisation of electric motor speed control with hardware in the loop. *Applied Soft Computing*, 11(1):49 – 55, 2011. ISSN 1568-4946.
6. Y. Jin y J. Branke. Evolutionary optimization in uncertain environments - a survey. *IEEE Transactions On Evolutionary Computation*, 9(3): 303–317, 2005.
7. Mostaghim, S. y Teich, J.: Quad-trees: A Data Structure for Storing Pareto Sets in Multiobjective Evolutionary Algorithms with Elitism. En Abraham, A., Jain, L., Goldberg, R., eds.: *Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing*. Springer Berlin Heidelberg (2005) 81–104.
8. Schütze, O.: A New Data Structure for the Nondominance Problem in Multiobjective Optimization. En proceedings of the 2nd international conference on Evolutionary multi-criterion optimization. EMO'03, Berlin, Heidelberg, Springer-Verlag (2003) 509–518.
9. Fieldsend, J.E., Everson, R.M. y Singh, S.: Using Unconstrained Elite Archives for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* 7(3) (2003) 305–323.
10. Sun, M. y Steuer, R.E.: Quad-Trees and Linear Lists for Identifying Nondominated Criterion Vectors. *INFORMS Journal on Computing*, otoño 1996(8) (1996) 367–375.
11. Mostaghim, S., Teich, J. y Tyagi, A.: Comparison of data structures for storing Pareto-sets in MOEAs. En proceedings of the Congress on Evolutionary Computation, 2002. CEC '02.. Volumen 1. (2002) 843 –848.
12. Thomas Hanne. Global Multiobjective Optimization with Evolutionary Algorithms: Selection Mechanisms and Mutation Control. En Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, y David Corne, editores, *First International Conference on Evolutionary Multi-Criterion Optimization*, páginas 197–212. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.