

EVALUACIÓN DE RENDIMIENTO Y PRESIÓN DE METAHEURÍSTICAS HÍBRIDAS PARALELAS EN UN CLUSTER MULTICORE

Miguel Méndez-Garabetti^{a,b}, Javier Rosenstein^a, Ailin Carribero^a, Maria Murazzo^c,
Nelson Rodríguez^c, Miguel Guevara^c, y Pablo Gómez^c

^a*Instituto de Investigaciones, Facultad de Informática y Diseño, Universidad Champagnat, Mendoza, Argentina, <http://www.uch.edu.ar/>*

^b*Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), <http://www.conicet.gob.ar/>*

^c*Departamento de Informática – FCFyN, UNSJ - CUIM (Complejo Islas Malvinas), San Juan, Argentina, <http://www.unsj.edu.ar/>*

Palabras Clave: Metaheurísticas, Metaheurísticas Híbridas, HPC, Multicore.

Resumen. En los últimos años las tendencias más relevantes para lograr el diseño de metaheurísticas eficientes han estado acompañadas por estrategias de hibridación y paralelismo. Si bien existen diversos tipos de problemas que actualmente son resueltos de forma aceptable mediante la utilización de algoritmos secuenciales. A medida que la complejidad incrementa, dichas implementaciones suelen volverse ineficientes e incluso obsoletas. Siendo necesario utilizar herramientas que permitan ofrecer resultados más eficientes. En este tipo de escenarios, la computación paralela se ha convertido en la forma tradicional de resolver problemas con altas cargas de procesamiento. Desde el punto de vista de exactitud o precisión, cuando trabajamos con problemas del tipo NP-duros, los algoritmos exactos suelen tener tiempos de ejecución que crecen de forma exponencial en relación al tamaño del problema haciéndolos inoperables antes este tipo de escenarios. Para este tipo de problemas, las metaheurísticas suelen ser una buena elección ya que ofrecen resultados de buena calidad en tiempos razonables. En este trabajo se evalúan -en términos de rendimiento y precisión- diferentes estrategias de hibridación colaborativa utilizando combinaciones de las siguientes metaheurísticas: Algoritmos Genéticos, Evolución Diferencial, Optimización por Cúmulo de Partículas y Optimización por Colonia de Hormigas. Las implementaciones han sido llevadas a cabo en un cluster multicore utilizando programación paralela basada en MPI y OpenMP.