



# APLICACIÓN DE TÉCNICAS DE PROGRAMACIÓN PARALELA PARA LA ESTIMACIÓN EFICIENTE DE RECURSOS MINERALES

**Milenis Fernández Díaz, José Gabriel Espinosa Ramirez, Armando Ortíz Cabrera**

Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños km 2 ½ La Lisa, La Habana, Cuba, [mfdez@uci.cu](mailto:mfdez@uci.cu)

## RESUMEN

El cálculo de las reservas y recursos minerales es de capital importancia para las empresas geomineras, tanto para evaluar las reservas como para optimizar el rendimiento económico de la explotación de un yacimiento determinado. La estimación de los recursos minerales se puede realizar utilizando diversas técnicas, predominando los métodos geoestadísticos, los cuales tienen en cuenta las características de variabilidad y de correlación espacial de los datos originales. El kriging constituye el estimador utilizado en los métodos geoestadísticos; su objetivo es encontrar el Mejor Estimador Lineal Insesgado a partir de los datos disponibles, los cuales generalmente son insuficientes debido al costo de su obtención. El proceso de estimación de recursos minerales a través la interpolación de kriging ordinario suele ser ineficiente en cuanto al tiempo de ejecución. La razón de los altos tiempos de ejecución del kriging ordinario encuentra explicación en las grandes cantidades de datos, la complejidad de los cálculos y el no aprovechando en su totalidad de las capacidades de cómputo instaladas. La rapidez y la precisión son cualidades deseables en las operaciones de estimación de reservas minerales. El presente trabajo tuvo como resultado un algoritmo de kriging ordinario que permite una mayor rapidez en la estimación de reservas minerales, a partir del aprovechamiento eficiente de los recursos computacionales. Este algoritmo fue implementado usando C++11 como lenguaje de programación, Qt Creator 3.0.1 como entorno de desarrollo, Armadillo 4.2 como biblioteca de álgebra lineal para los cálculos matriciales; así como técnicas de programación paralela en memoria compartida a través de OpenMP 4.8.2.

## ABSTRACT

The calculation of mineral reserves and resources is crucial for geominig companies, both to evaluate reserves as to maximize the economic performance for the exploitation of a mineral deposit. The mineral resource estimation may be performed using various techniques, predominating geostatistical methods, which take into account the characteristics of variability and spatial correlation of the original data. The kriging estimator is used in geostatistical methods; its goal is to find the Best Linear Unbiased Estimator based on the available data, which are generally insufficient because of the cost of obtaining it. The process of estimating mineral resources through ordinary kriging interpolation is often inefficient in terms of runtime. The reason for the high execution times of ordinary kriging is explained by the large amounts of data, the complexity of the calculations and not taking full advantage of the computing capabilities installed. Speed and accuracy are desirable qualities in the operations of estimating mineral reserves. This work had as result an ordinary kriging algorithm that allows faster in the estimation of mineral reserves, through the efficient use of computing resources. This algorithm was implemented using C++11 as programming language, Qt Creator 3.0.1 as development environment, Armadillo 4.2 as linear algebra library for matrix calculations; and parallel programming techniques in shared memory through OpenMP 4.8.2.

## INTRODUCCIÓN

Los recursos minerales son la base de la sociedad moderna, y su demanda ha aumentado aceleradamente con el desarrollo tecnológico. Los recursos minerales son finitos y, en su mayoría, no renovables, haciendo que la estimación de sus depósitos sea un elemento primordial, tanto para evaluar las reservas como para optimizar el rendimiento económico para la explotación de un yacimiento determinado. La estimación de los recursos minerales es un factor clave a la hora de decidir la ejecución de un proyecto minero, con el fin de asegurar el éxito. Los cálculos que se realizan deben asegurar un alto grado de certeza, pues las inversiones requeridas son muy costosas.



El cálculo de los recursos y reservas minerales se puede realizar utilizando diversas técnicas que son agrupadas en métodos clásicos o métodos modernos. En los métodos modernos predominan los métodos geoestadísticos, teniendo como premisa fundamental la realización de estimaciones a partir de las características de variabilidad y de correlación espacial de los datos originales (Cuador Gil, 2005). Se basan en técnicas de interpolación espacial de alta confiabilidad las cuales están respaldadas por modelos matemáticos, y además proporcionan el error derivado del proceso de estimación.

El kriging constituye el estimador utilizado en los métodos geoestadísticos; su objetivo es encontrar el Mejor Estimador Lineal Insesgado a partir de los datos disponibles, los cuales generalmente son insuficientes debido al costo de su obtención. El krigeado atribuye un peso  $\lambda_i$  a la ley de cada muestra  $z(x_i)$ , donde los pesos altos corresponden a las muestras cercanas y los pesos débiles a las alejadas. Estos pesos  $\lambda_i$  se calculan de manera que la varianza del error cometido sea mínima y considerando las características geométricas del problema. Los valores de los pesos asociados a cada uno de los puntos se calculan mediante un sistema de ecuaciones lineales, el cual se compone de  $N+1$  ecuaciones con  $N+1$  incógnitas (Alfaro Sironvalle, 2007).

Varios autores confirman la complejidad y alto costo computacional de estos algoritmos (Kerry, y otros, 1998) (O'Sullivan, y otros, 2002) (Lloyd, 2006) (Vasan Srinivasan, y otros, 2010) (Pesquer, y otros, 2011). En (Vasan Srinivasan, y otros, 2010) se plantea que el algoritmo por cada punto de interés tiene una complejidad cúbica, lo que conduce a una complejidad de  $O(MN^3)$ , siendo  $N$  el número de muestras y  $M$  el número de bloques. Cuando  $M \approx N$  la complejidad computacional puede considerarse  $O(N^4)$  lo cual no es favorable cuando se trabaja con yacimientos complejos que precisan de grandes volúmenes de datos.

La complejidad de los cálculos matemáticos utilizados en la interpolación de kriging ordinario, hace imprescindible la utilización de medios automatizados para el proceso de estimación. Aún así, el proceso de estimación de recursos minerales a través de las técnicas de kriging suele ser ineficiente en cuanto al tiempo de ejecución. La razón de los altos tiempos de ejecución del kriging ordinario encuentra explicación en las grandes cantidades de datos, la complejidad de los cálculos y el no aprovechamiento en su totalidad las capacidades de cómputo instaladas.

La rapidez y la precisión son cualidades deseables en las operaciones de estimación de reservas minerales. Una alternativa viable para la solución rápida de problemas computacionalmente grandes y con muchos datos, se puede encontrar en las técnicas de computación paralela. Esta se define como el uso de múltiples núcleos o procesadores simultáneamente para mejorar la velocidad de las aplicaciones (Campbell, y otros, 2011). La principal ventaja de las técnicas de computación paralela radica en la aceleración de los procesos de cómputo a partir de la utilización óptima de los recursos computacionales.

Este trabajo tiene como objetivo: la aplicación de técnicas de programación paralela al algoritmo de kriging ordinario para lograr una mayor rapidez en la estimación de reservas minerales. Haciendo una mejor utilización de las capacidades de cómputo instaladas, se podrá agilizar la interpolación de los datos en la ejecución del kriging ordinario.

## MATERIALES Y MÉTODOS

Las tendencias actuales en el desarrollo de hardware están encaminadas a la incorporación de más núcleos en los procesadores, en lugar del incremento de la velocidad del reloj de los procesadores. Por tal motivo, el algoritmo propuesto está dirigido al aprovechamiento eficiente de las capacidades de procesamiento de los sistemas multinúcleos, adaptándose a las características del hardware subyacente. Esta adaptación permite obtener mejores rendimientos mientras mayores sean las capacidades de procesamiento paralelo de los equipos (Campbell, y otros, 2011). La estrategia de



paralelización del algoritmo consistió en la subdivisión de los datos y aplicarle a cada subgrupo de datos el algoritmo, obteniéndose resultados parciales que tributan a la solución general del problema. Esta variante puede ser muy útil en el procesamiento de grandes cantidades de datos.

El algoritmo de kriging ordinario fue implementado usando C++11 como lenguaje de programación y Qt Creator 3.0.1 como entorno de desarrollo. Para los cálculos matriciales se utilizó la biblioteca de álgebra lineal Armadillo, caracterizada por ser rápida y fácil de usar. Esta biblioteca permitió simplificar el cálculo de operaciones matriciales como el determinante, la matriz inversa, la matriz transpuesta, así como la multiplicación de matrices. También se aplicaron técnicas de programación paralela en memoria compartida a través de la biblioteca OpenMP 4.8.2.

OpenMP es una API (*Application Program Interface*) destinada a máquinas de memoria compartida, para programar explícitamente en aplicaciones paralelas multihilos. Esta API está compuesta por un conjunto de directivas de compilador que el usuario utiliza para especificar qué regiones de código van a ser paralelizadas. La implementación de OpenMP utiliza multiplicidad de hilos para obtener el paralelismo, mediante un modelo fork-join. Este modelo plantea la división del hilo maestro en hilos esclavos que se ejecutan concurrentemente, distribuyéndose las tareas sobre estos hilos. Estos hilos acceden a la misma memoria, aunque es posible gestionar estos accesos generando espacios de memoria privada (Chapman, y otros, 2008).

Se realizó la evaluación experimental del algoritmo en computadoras con diferentes prestaciones de hardware variando el tamaño de los datos a utilizar y el número de hilos. Como parte del experimento se realizaron mediciones a los tiempos de ejecución de diferentes corridas del algoritmo (en sus variantes secuencial y paralela), para luego establecer comparaciones entre los resultados obtenidos. El reloj de la computadora sirvió como instrumento de medición. A través de la experimentación se constató la reducción de los tiempos de ejecución del algoritmo a partir de su implementación utilizando técnicas de computación paralela.

## RESULTADOS

Como parte de la investigación se realizaron 2 experimentos considerando el tamaño de entrada del algoritmo ( $n$ ) y el número de procesadores ( $p$ ). En función de estos parámetros, los algoritmos fueron evaluados atendiendo a la Ganancia de Velocidad (Speed Up) y la Eficiencia ( $E$ ). A la Ganancia de Velocidad también se le conoce como Aceleración, y consiste en la relación entre el tiempo de ejecución sobre un procesador secuencial y el tiempo de ejecución sobre múltiples procesadores. Por Eficiencia entiéndase el porcentaje de tiempo empleado en proceso efectivo (Müller, 2011); este indicador mide el grado de utilización de un sistema multiprocesador.

Las versiones del algoritmo se ejecutaron en una computadora personal con 4 GB de RAM procesador Intel Core i5-2430M (Intel, 2015) que cuenta con 2 núcleos y 4 subprocesos, con el sistema operativo Microsoft Windows 7 Home Premium Edition. Para observar el comportamiento de las implementaciones se utilizaron 3 juegos de datos generados aleatoriamente y 4 asignaciones distintas de hilos, obteniéndose las mediciones que se pueden apreciar en las figuras 1 y 2, y en la tabla I.

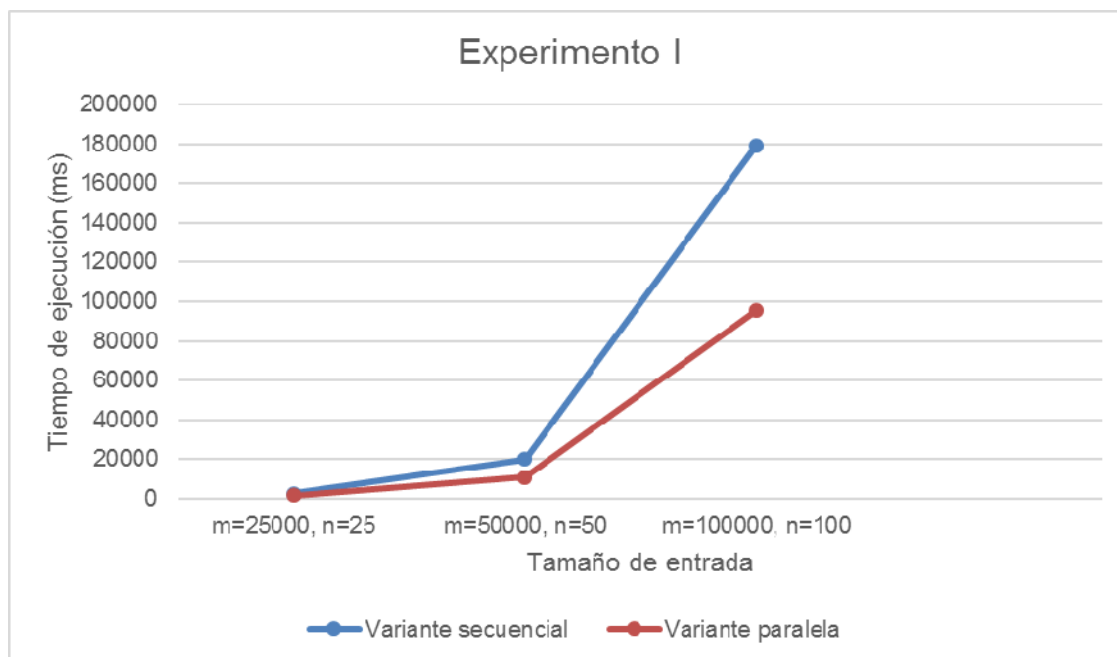


Figura 1. Comparación de los tiempos del algoritmo, en sus variantes secuencial y paralela, ejecutándose sobre diferentes números de hilos. El tamaño de entrada está dado por la cantidad de bloques ( $m$ ) y la cantidad de muestras por bloque ( $n$ ).

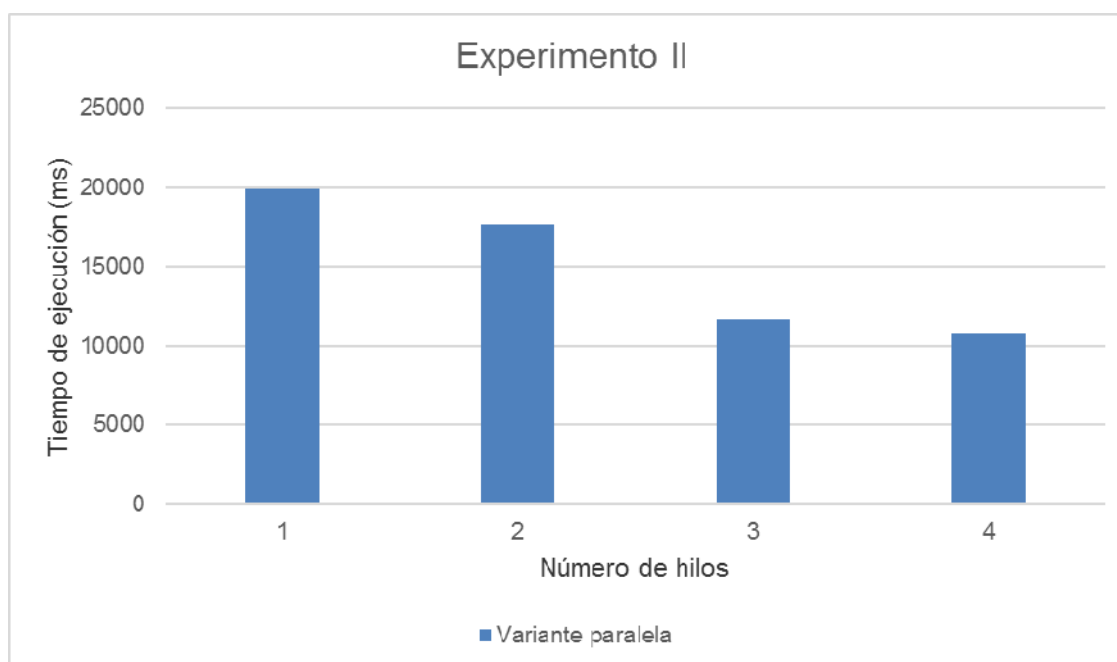


Figura 2. Mediciones de los tiempos de ejecución del algoritmo de kriging ordinario (en su variante paralela) aplicado a la estimación de 50 000 bloques, cada uno con 50 muestras.



Tabla I. Mediciones de los tiempos de ejecución (T) del algoritmo de kriging ordinario aplicado a la estimación de los recursos minerales, para diferentes cantidades de bloques y muestras. Incluye los indicadores ganancia de velocidad (Sp) y eficiencia (E).

Tamaño de entrada	V. Secuencial	V. Paralela
m = 25000 n = 25	T = 2658 ms	T = 1713 ms Sp = 1.5517 E = 0.3879 %
m = 50000 n = 50	T = 19992 ms	T = 10752 ms Sp = 1.8594 E = 0.4649 %
m = 100000 n = 100	T = 179155 ms	T = 95391 ms Sp = 1.8781 E = 0.4695 %

## DISCUSIÓN

Las técnicas de programación paralela han sido aplicadas a la interpolación de kriging ordinario usando diferentes tecnologías y estrategias de paralelización. Diversas son las soluciones reportadas en la Web, con una tendencia hacia el cómputo de propósito general en unidades de procesamiento gráfico, GPU (*Graphics Processing Unit*). Algunas de las soluciones más recientes se pueden encontrar en (Allombert, y otros, 2014) y (Gutiérrez de Ravé, y otros, 2014). Estas unidades se encuentran optimizadas para cálculos, en particular operaciones de Álgebra Lineal. Sin embargo, la utilización de este tipo de dispositivos de hardware obliga a programar para fabricantes específicos. Además se debe tener presente que no todas las estaciones de trabajo disponen de estos dispositivos. Por tales motivos, el algoritmo propuesto está dirigido a explotar los equipos con procesadores multinúcleos, teniendo en cuenta la tendencia a la incorporación de más núcleos en los procesadores.

Los datos arrojados por la evaluación experimental del algoritmo evidencian una disminución notable de los tiempos requeridos en la interpolación de kriging ordinario con la aplicación de técnicas de programación paralela. Como se puede observar en la Tabla I, el indicador ganancia de velocidad incrementa ligeramente al aumentar el número de hilos. Por otro lado, los valores de eficiencia indican un aprovechamiento de casi la mitad de las capacidades de cómputo. Estos resultados se alcanzaron simplemente dividiendo los bloques entre los núcleos de procesamiento. Si la estrategia de paralelización se hubiese enfocado en agilizar los cálculos matriciales considerados como los más pesados, los resultados obtenidos indicarían un mejor rendimiento computacional. Si se logra una distribución equitativa de los bloques en los hilos también se favorecerá el rendimiento.

## CONCLUSIONES

Una vez concluida la investigación se arriban a las siguientes conclusiones:

- La capacidad de adaptarse a la cantidad de procesadores disponibles permite al algoritmo obtener mejores rendimientos mientras mayores sean las capacidades de procesamiento paralelo del equipo.
- La aplicación de técnicas de programación paralela en memoria compartida usando OpenMP ofrece mayor eficiencia en la estimación de recursos minerales, al reducir el tiempo de cómputo de las operaciones requeridas en la interpolación de kriging ordinario.



- La subdivisión de los bloques en diferentes hilos es muy útil cuando se trabaja con grandes cantidades de bloques, pero si no se logra una distribución equitativa de estos, el aprovechamiento de los núcleos de procesamiento no será óptimo.

## BIBLIOGRAFÍA

- Alfaro, M.A., 2007. *Estimación de Recursos Mineros*.
- Allombert, V., D. Michea, F. Dupros, C. Bellier, B. Bourguine, H. Aochi, S. Jubertie, 2014. *An out-of-core GPU approach for accelerating geostatistical interpolation*, 29 : 888-896.
- Campbell, C., A. Miller, 2011. *Parallel programming with Microsoft Visual C++. Design patterns for decomposition and coordination on multicore architectures*. ISBN 978-0-7356-5175-3.
- Chapman, B., G. Jost, A. R. Van, And Ruud, 2008. *Using OpenMP Portable Shared Memory Parallel Programming*. (London) : The MIT Press. ISBN-13: 978-0-262-53302-7.
- Cuador, J. Q., 2005. *Estudios de estimación y simulación geoestadística para la caracterización de parámetros geólogo-industriales en el yacimiento laterítico Punta Gorda*. (Moa), ISSN 02585979.
- Gutiérrez de Ravé, E., F. J. Jimenez, A. B. Ariza, J. M. Gómez, 2014. *Using general-purpose computing on graphics processing units (GPGPU) to accelerate the ordinary kriging algorithm*. 1-6.
- Intel. 2015. Intel® Core™ i5-2430M Processor (3M Cache, up to 3.00 GHz). *Sitio oficial de Intel*. [En línea]. [http://ark.intel.com/es/products/53450/Intel-Core-i5-2430M-Processor-3M-Cache-up-to-3\\_00-GHz](http://ark.intel.com/es/products/53450/Intel-Core-i5-2430M-Processor-3M-Cache-up-to-3_00-GHz) 31 de enero de 2015
- Kerry, K. E. y K.A. Hawick, 1998. *Kriging interpolation on high-performace computers*. (Australia) : Universidad de Adelaide
- Lloyd, C. D. 2006. *Local models for spatials analysis*. Belfast : CRC Press.
- Müller, L., 2011. *Evaluación del rendimiento de algoritmos paralelos y/o concurrentes*.
- O'Sullivan, D. y D. Unwin, 2002. *Geographic Information Analysis*. (New Jersey) : John Wiley & Sons Hoboken.
- Pesquer, L., A. Cortés, X. Pons, 2011. *Parallel ordinary kriging interpolation incorporating automatic variogram fitting*, 37:464–473.
- Vasan, B., R. Duraiswami, R. Murtugudde, 2010. *Efficient kriging for real-time spatio-temporal interpolation*.