PESQUIMAT , Revista de la F.C.M. de la
Universidad Nacional Mayor de San Marcos
Vol. XIV N°1, pp. 67-74, Lima - Perú, Julio 2011

REDUCIENDO EL ANCHO DE BANDA DE MATRICES DISPERSAS SIMÉTRICAS CON ALGORITMOS GENÉTICOS

Ricardo López Guevara[*]

Resumen: El presente trabajo propone la reducción del ancho de banda en matrices dispersas y simétricas, usando la Metaheurística Algoritmos Genéticos y un software desarrollado en MS. Visual Studio 6.0. Existen numerosas aplicaciones tanto en las ciencias como en la ingeniería que requieren la solución de la reducción del ancho de banda de matrices dispersas y simétricas. La solución de grandes sistemas de ecuaciones algebraicas lineales con dispersión, estructuradas y con coeficientes simétricos, involucra tener grandes cantidades de espacio de almacenamiento y de tiempo computacional. El espacio de búsqueda es N! en la cual N es la dimensión de la matriz, la cual usualmente es bastante grande. Este problema consiste de encontrar una permutación de filas y columnas de una matriz dispersa y simétrica dada, la cual mantenga los elementos diferente de cero en una banda tan cercana como sea posible a la diagonal principal.
Palabras clave: Metaheurísticas, Ancho de Banda, Algoritmo Genético, Explosión combinatoria.

REDUCING THE BANDWIDTH OF SYMMETRICAL DISPERSED WOMBS WITH GENETIC ALGORITHMS

Abstract: This work proposes the minimization of bandwidth in sparse symmetric Matrix, using genetic algorithms and an own software, developed in MS Visual Studio 6.0. There are very numerous applications both in Science and Engineering that requires the resolution of the minimization of bandwidth in symmetric sparse matrix. The solution of great systems of linear algebraic equations with dispersion, structured and symmetric coefficients, involves having great amounts of storage space and a large amount of computational time. The search space is N! in which N is the dimension of the matrix, which is usually quite large. This problem consists of finding a permutation of the rows and columns of a given sparse symmetric matrix, which keeps the nonzero elements in a band that is as close as possible to the main diagonal.
Key words: Metaheuristic, Bandwidth, Genetic Algorithmic, Combinatory Explosion.

1. INTRODUCIÓN

Hoy en día las máquinas resuelven problemas mediante algoritmos que tienen como máximo una complejidad o costo computacional polinómico, es decir, la relación entre el tamaño del problema y su tiempo de ejecución es polinómico. Estos problemas son agrupados en el conjunto P. Los problemas con costo factorial o costo combinatorio están agrupados en el conjunto NP. Estos problemas no tienen una solución práctica, es decir, una máquina no puede resolverlos en un tiempo razonable.

La importancia de una reducción en el ancho de banda de una matriz, radica en que disminuirá significativamente el tiempo de cálculo computacional para la descomposición de la matriz. Se ha probado que el problema de reducción de Ancho de Banda de una Matriz es un problema NP-Completo. Papadimitriou [7] (1976) y Garey (1978) mostraron que este es un problema NP - completo aun cuando el grafo de entrada sea un árbol cuyo máximo grado de vértice es 3. El espacio de búsqueda de soluciones es N!, donde N es la dimensión de la matriz, que usualmente es muy grande[3].

Un gran problema de la optimización es el fenómeno llamado "Explosión Combinatoria", que significa, que cuando crece el número de variables de decisión del problema, el número de decisiones factibles y

[*]UNMSMS, Facultad de Ciencias Matemáticas, e-mail: rlopezg@unmsm.edu.pe

el esfuerzo computacional crecen en forma exponencial, es decir el crecimiento de forma exponencial del tamaño de espacio de búsqueda en relación con las variables y su dominio. Cuando el espacio de búsqueda es muy grande, los métodos completos son incapaces de encontrar una solución aceptable, por lo que se opta por utilizar heurísticas y / o Metaheurísticas. Sin embargo no todos los problemas combinatorios son tan complejos de resolver; existen algunos para los cuales hay algoritmos que resuelven esos problemas con un esfuerzo computacional que crece de manera polinomial con el tamaño del problema.

1.1. SISTEMAS DE ECUACIONES LINEALES

La solución de sistemas de ecuaciones lineales es un tema de gran interés en el mundo de la computación actual. Son muchos los problemas que pueden resolverse utilizando estas técnicas, y en áreas tan variadas como se quiera. Por eso es de gran importancia el estudio de algoritmos que optimicen el uso de recursos computacionales como la memoria física y el tiempo de ejecución, a la vez que se encuentren soluciones correctas de manera óptima. Al estudiar las matrices generadas por los sistemas de ecuaciones se puede observar ciertas características y propiedades, que pueden permitirnos en algunas ocasiones un tratamiento especial del problema mejorando los algoritmos tradicionales al explotar la estructura y las propiedades de la matriz[4].

En nuestra investigación particular se pretende trabajar con nuevos métodos, como los Algoritmos Genéticos, para la solución de sistemas de ecuaciones con matrices asociadas simétricas, de estructura dispersa y de cualquier dimensión. Los métodos de Algoritmos Genéticos simulan la evolución de los procesos y los cambios genéticos en las estructuras de los cromosomas tomados de la Biología, y se busca la solución tanto en procesamiento secuencial como paralelo. Este tipo de sistemas (ecuaciones lineales) se encuentran en diversas áreas como optimizaciones lineales y no lineales, análisis de elementos finitos, problemas inversos de eingenvalores y muchas más. En general se debe usar algún tipo de pivoteamiento para resolver correctamente estos sistemas y el reto es mantener una estabilidad computacional y la estructura dispersa en las matrices.

De aquí la importancia de los Algoritmos Genéticos que tienen un método sistemático para la resolución de problemas de búsqueda y optimización, los cuales se aplican imitando los mismos métodos de la evolución biológica: selección basada en la población, reproducción sexual y mutación.

Además, un Algoritmo Genético consiste de lo siguiente: encontrar los parámetros con los cuales cuenta el problema, codificarlos en un cromosoma, y aplicar los métodos de evolución: selección y reproducción sexual con intercambio de información y alteraciones que causan diversidad.

2. FORMULACIÓN DEL PROBLEMA

2.1. Formulación

El sistema lineal de ecuaciones a ser considerado es de la forma:

$$Ax = b, \tag{1}$$

donde A es una matriz dispersa simétrica $N \times N$. Los elementos de A serán señalados como a_{ij} donde i es el índice fila y j es el índice columna. Usaremos el valor máximo de $\max\{|i-j|;\ a_{ij} \neq 0\}$ como la medida del ancho de banda de A, ver [5]. Es decir $\min\{\max\{|i-j|;\ a_{ij} \neq 0\}\}$ con Algoritmos Genéticos. Este problema NP-completo, también puede ser formulado etiquetando los vértices de un grafo, donde las aristas son los elementos diferentes de cero de la correspondiente matriz simétrica. Muchos algoritmos de reducción del ancho de banda han sido desarrollados desde 1960 y se han aplicado a ingeniería estructural, dinámica de fluidos, análisis de redes y muchos campos mas de la ingeniería.

2.2. Esquema del algoritmo Genético

Los Algoritmos Genéticos, AG, son algoritmos de búsqueda paralela que imitan la selección natural de los procesos de evolución. Un AG usa el mecanismo natural de selección para guiar la búsqueda. En el tiempo, la selección natural ha producido un amplio rango de estructuras robustas (formas de vida)

que eficientemente ejecutan un amplio rango de funciones. Parecido a la selección natural, los AG son métodos de búsqueda robustos que requieren poca información para buscar efectivamente en espacios pobremente entendidos.

Un AG es un proceso que imita la manera en que trabaja la evolución biológica. Como la evolución, este opera sobre una población de individuos los cuales representan soluciones potenciales para un problema dado. Buscan producir los mejores (fitness) individuos (soluciones) combinando los mejores de los existentes (crías). Usando la "táctica de sobrevive el mejor" ello separa los malos y tiende a producir mas de los buenos individuos. No solo produce más de las buenas soluciones sino mejores y mejores soluciones. Esto es porque combina los mejores rasgos de los padres de los individuos para producir hijos superiores. Este operador de combinación es llamado crossover o cruce. El término algoritmo genético viene del hecho que los individuos son representados como strings o cadenas de bits análogos a los cromosomas y genes. Adicionalmente para recombinar por cruce, también probamos con mutaciones aleatorias de esas cadenas de bits tantas como se pueda. Esto mantiene al AG atrapado en una buena solución pero no la óptima [6].

Nuestro modelo de AG codifica un diseño candidato (matriz) en una cadena binaria. Un conjunto generado aleatoriamente de tal cadena forma la población inicial (matriz) desde los AG inician su búsqueda. Tres operadores genéticos básicos guían este trabajo: Selección, Cruce y Mutación. El proceso de búsqueda es iterativo, evaluativo, seleccionando y recombinando cadenas en la población durante cada iteración llamada generación hasta que alcance una condición de parada.

2.3. Estructura de un Algoritmo Genético

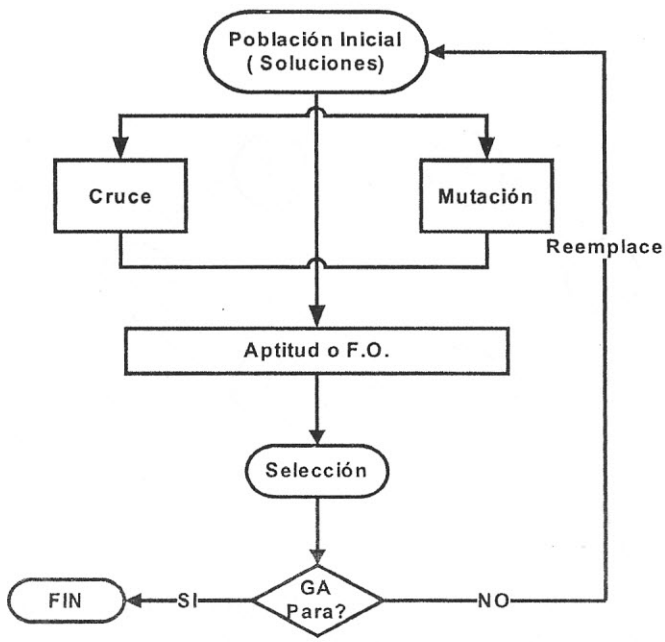


Figura 1

3. METODOLOGIA

3.1. Metodología del AG

Para llevar a la práctica el esquema anterior y establecer explícitamente en el algoritmo, se debe especificar los siguientes elementos:

- Una representación cromosómica.
- Una población inicial

- Una medida de evaluación (fitness o f.o.)
- Un criterio de selección de los cromosomas.
- Una o varias operaciones de recombinación.
- Una o varias operaciones de mutación.
- Los parámetros de entrada del AG para guiar la evolución.

3.2. Una representación cromosómica

En este trabajo proponemos primero una representación binaria, para lo cual obtenemos una matriz generada aleatoriamente. La metodología usada para ello es usar MS Excel para generar y simular la matriz con una probabilidad de elementos diferente de cero, y obtenemos la matriz inicial, para iniciar y generar el grafo de cromosomas equivalente. Por ejemplo, esta matriz de 6×6

	1 2 3 4 5 6
1	**1 1 1 0 0 0**
2	**1 1 1 1 0 0**
3	**1 1 1 0 1 0**
4	**0 1 0 1 1 1**
5	**0 0 1 1 1 1**
6	**0 0 0 1 1 1**

Y su grafo equivalente sería:

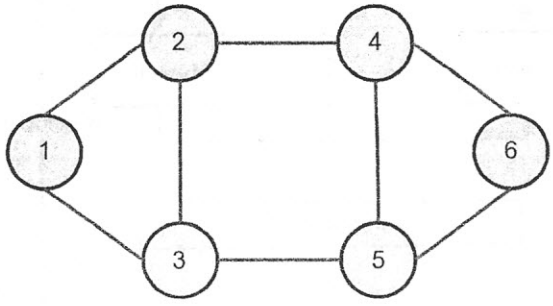


Figura 2

3.3. Una población inicial

Con la matriz y grafo inicial, se genera aleatoriamente los cromosomas, ejemplo: Un grafo cromosoma puede ser → (123456) y otros cromosomas aleatoriamente pueden ser

(1 5 6 3 2 4)
(5 4 6 2 1 3)
(6 4 5 2 3 1)

Hasta completar n cromosomas en la población (para este ejemplo, se usa el parámetro de la población igual a 20). Luego generamos la matriz equivalente y obtenemos el ancho de banda y seguimos así.

3.4. Una medida de evaluación (fitness)

El fitness o medida de evaluación es la capacidad del cromosoma, es decir el ancho de banda, el máximo entre la columna y la fila con valor 1. Este procedimiento básico es permutar filas y columnas de la matriz dada para minimizar el ancho de banda.

3.5. Un criterio de selección de los cromosomas

Usamos como criterio de selección el procedimiento conocido de la Ruleta. Algunos cromosomas son seleccionados más de una vez, esto es en concordancia con el teorema de los esquemas; los mejores cromosomas obtienen mas copias, el promedio permanece y desaparece el peor.

Supongamos lo siguiente para el mismo ejemplo:

Población cromosoma 1 (1) = 2 3 1 6 4 5
Población cromosoma 1 (2) = 4 5 1 6 3 2
Población cromosoma 1 (3) = 4 5 1 3 2 6
Población cromosoma 1 (4) = 2 3 1 6 4 5
Población cromosoma 1 (5) = 1 5 4 3 6 2
Población cromosoma 1 (6) = 3 6 4 5 1 2
Población cromosoma 1 (7) = 2 3 1 6 4 5
Población cromosoma 1 (8) = 6 5 3 4 1 2
Población cromosoma 1 (9) = 3 1 4 2 5 6
Población cromosoma 1 (10) = 4 5 1 6 3 2
Población cromosoma 1 (11) = 6 5 3 4 1 2
Población cromosoma 1 (12) = 2 3 1 6 4 5
Población cromosoma 1 (13) = 2 3 1 6 4 5
Población cromosoma 1 (14) = 3 1 4 2 5 6
Población cromosoma 1 (15) = 6 3 5 4 2 1
Población cromosoma 1 (16) = 6 5 2 1 4 3
Población cromosoma 1 (17) = 5 4 6 2 3 1
Población cromosoma 1 (18) = 6 3 5 4 2 1
Población cromosoma 1 (19) = 3 1 4 2 5 6
Población cromosoma 1 (20) = 4 5 1 6 3 2

3.6. Una o varias operaciones de recombinación

Ahora estamos listos para aplicar el operador de cruce o recombinación para los nuevos individuos de la población. Usamos el parámetro pc para definir la probabilidad de cruce. Este parámetro nos da el pc.size_pop, que es el número esperado de cromosomas a ser cruzados. En este trabajo usamos el método de cruce de punto simple.

Genera un valor aleatorio entre 0 y 1 para el cruce:

$r(1) = .821119904518127$
$r(2) = .613891899585724$
$r(3) = .494059801101685$
$r(4) = .765604436397552$
$r(5) = .152325510978699$
$r(6) = .57559734582901$
$r(7) = .992976665496826$
$r(8) = .366754233837128$
$r(9) = .305174946784973$
$r(10) = .498152554035187$
$r(11) = .332000494003296$
$r(12) = .279614388942719$
$r(13) = 2.67995595932007E-02$
$r(14) = .202663481235504$
$r(15) = .860313415527344$
$r(16) = .019138514995575$
$r(17) = 8.48385095596313E-02$
$r(18) = .448712646961212$
$r(19) = .11166787147522$
$r(20) = .546569287776947$

En este ejemplo, utilizamos para la selección solo a pc.=0.25
Cromosoma seleccionado para el cruce

Cromosoma 1 $\Longrightarrow$ (2 3 1 6 4 5): no
Cromosoma 2 $\Longrightarrow$ (4 5 1 6 3 2): no
Cromosoma 3 $\Longrightarrow$ (4 5 1 3 2 6): no
Cromosoma 4 $\Longrightarrow$ (2 3 1 6 4 5): no
Cromosoma 5 $\Longrightarrow$ (1 5 4 3 6 2): yes
Cromosoma 6 $\Longrightarrow$ (3 6 4 5 1 2): no
Cromosoma 7 $\Longrightarrow$ (2 3 1 6 4 5): no
Cromosoma 8 $\Longrightarrow$ (6 5 3 4 1 2): no
Cromosoma 9 $\Longrightarrow$ (3 1 4 2 5 6): no
Cromosoma 10 $\Longrightarrow$ (4 5 1 6 3 2): no
Cromosoma 11 $\Longrightarrow$ (6 5 3 4 1 2): no
Cromosoma 12 $\Longrightarrow$ (2 3 1 6 4 5): no
Cromosoma 13 $\Longrightarrow$ (2 3 1 6 4 5): yes
Cromosoma 14 $\Longrightarrow$ (3 1 4 2 5 6): yes
Cromosoma 15 $\Longrightarrow$ (6 3 5 4 2 1): no
Cromosoma 16 $\Longrightarrow$ (6 5 2 1 4 3): yes
Cromosoma 17 $\Longrightarrow$ (5 4 6 2 3 1): yes
Cromosoma 18 $\Longrightarrow$ (6 3 5 4 2 1): no
Cromosoma 19 $\Longrightarrow$ (3 1 4 2 5 6): yes
Cromosoma 20 $\Longrightarrow$ (4 5 1 6 3 2): no

3.7. Una o varias operaciones de mutación

Del mismo modo que en la operación de cruce. el operador de mutación usa el parámetro de probabilidad pm. Para este operador se hace lo mismo que en el caso anterior.

3.8. Los parámetros de entrada para guiar la evolución

En los programas NP-completo se podría evaluar todas las soluciones alternativas y elegir la mejor. Si el total de soluciones es 1000000, se evaluaría todo, pero usando AG se puede usar solamente un tamaño de población menor a esos 1000000 como 20, 30, 100, 200, ... cualquier valor que el investigador considere. No esta definido ni normado una regla exacta, la idea es que el valor sea diferente al valor real. Del mismo modo se trata el número de generaciones, el usuario define el valor de este parámetro. El parámetro para la probabilidad de cruce puede estar alrededor de 0.25, es decir se cruzara el 25 % del tamaño de la población, después se pueden probar otros valores. En el caso del parámetro para la probabilidad de la mutación es recomendado que este alrededor del 0.01. En conclusión, los parámetros pueden ser dados por el usuario porque no hay reglas para cada uno de ellos.

4. RESULTADOS

4.1. Programa principal en Visual Basic

Main Program
$t = 0$
Call startpob
Do While ($t <$ Parameter.numgen)
$t = t + 1$
Call selection
Call crossover
Call mutation
Call evaluate
Loop

Figura 3: Ventana principal del programa

4.2. Resultados y Discusión

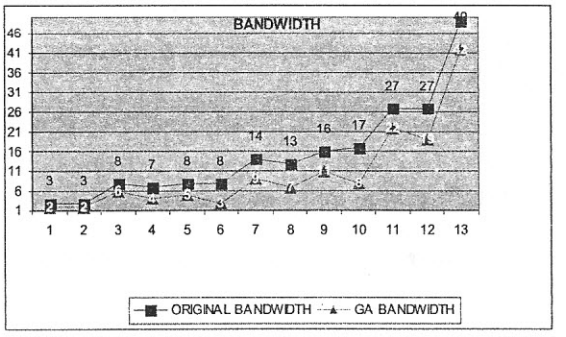
Nuestro programa en MS-Visual.net se inicia con la ventana de la Figura 3, y requiere los parámetros iniciales, el resto lo hace el AG. Finalmente, un conjunto de problemas fue corrido, para lo cual las posiciones de los elementos diferentes de cero en las matrices dispersas simétricas fueron generadas usando el generador de números aleatorios de MS Excel, para garantizar su aleatoriedad. La densidad, se refiere al promedio de elementos diferentes de cero en la matriz, fue obtenida ingresando una probabilidad deseada.

Tabla 1 – Experimento Computacional

Matriz inicial de Tamaño N	6	6	9	9	12	12	15	15	20
Tamaño de la población	20	20	20	25	30	30	20	25	25
Probabilidad de Cruce	25%	30%	25%	25%	30%	25%	25%	20%	20%
Probabilidad de Mutación	1%	2%	1%	1%	2%	1%	2%	1%	1%
Cantidad de generaciones	100	100	100	120	120	100	100	120	120
Probabilidad de No ceros (sparse)	60%	80%	55%	75%	70%	85%	65%	83%	80%
Cantidad de UNOS (1s) Densidad	10	8	30	22	30	12	62	30	56
Ancho de Banda Problema Inicial	3	3	8	7	8	8	14	13	16
Ancho de Banda Reducido	2	2	6	4	5	3	9	7	11
CPU TIEMPO – SEGUNDOS (aproximado)	2	2.1	3.2	4	5.6	4	3.8	6	7
ESPACIO ALAMCENAMIENTO (kb)	1674	1684	2113	3156	4784	3975	3331	4924	8087

Tabla 2 – Experimento Computacional

Matriz inicial de Tamaño N	30	30	50
Tamaño de la población	25	25	20
Probabilidad de Cruce	25%	25%	25%
Probabilidad de Mutación	1%	1%	1%
Cantidad de generaciones	100	100	100
Probabilidad de No ceros (sparse)	70%	85%	70%
Cantidad de UNOS (1s) Densidad	225	107	740
Ancho de Banda Problema Inicial	**27**	**27**	**49**
Ancho de Banda Reducido	**22**	**19**	**42**
CPU TIEMPO – SEGUNDOS (aproximado)	7.4	7.2	11
ESPACIO ALAMCENAMIENTO (kb)	9413	9407	15940



5. CONCLUSIONES

En este trabajo, se reporta sobre el uso de un AG simple que permite cubrir con una evaluación computacional muy costosa la función objetivo. Los resultados de nuestro experimento computacional revelan que las estrategias implementadas con un procedimiento de un AG simple es capaz de desempeñarse competitivamente con otras Metaheurísticas como Tabú Search o Búsqueda Tabú. Aunque se debe admitir que el procedimiento no es satisfactorio en algunos casos cuando crece el tamaño de la matriz inicial ya que consume mas tiempo de CPU. Por otro lado, aquí no se ha usado el método de cruce de doble punto y puede ser una oportunidad para experimentar nuevas formas de mejorar el método. Otro aspecto importante es tener que correr con otras matrices de diferentes tamaños a las probadas aquí y también el uso de otros parámetros.

Bibliografía

[1] Almeida P., Franco J., On Reducing Bandwidth of Matrices in the Go-Away Algorithm for Regular Grids.

[2] Cutchill E., Mckee. J. (1969), Reducing the bandwidth of sparse symmetric matrices. Proceedings 24th National of the ACM, pages 157–172.

[3] Díaz, A., Glover F., Ghaziri H., Gonzalez J., Laguna M., Moscato P., Tseng, F. (1996), Optimización Heurística y Redes Neuronales, Paraninfo, Madrid.

[4] Jaramillo, D., Vidal, A. (2006), Métodos directos para la solución de sistemas de ecuaciones lineales simétricos, indefinidos, dispersos y de gran dimensión. Universidad Eafit. Colombia.

[5] Marti, R., Laguna, M., Glover, F., Campos, V. (2001), Reducing the Bandwidth of Sparse Matrix with Tabu Search, European Journal of Operational Research, 135(2), pp.

[6] Mangano, S. (1996), An Introduction to Genetic Algorithm Implementation, Theory, Application, History and Future Potential. Man Machine Interfaces Inc.

[7] Papadimitriou, c. (1976), The np-completeness of the bandwidth minimization problem. Computing, 16:263-270.

[8] Reeves, C. (1995), Modern Heuristic Techniques for Combinatorial Problems, ed. McGraw-Hill, UK.