APLICACIÓN DEL SIMULATED ANNEALING AL PROBLEMA DE LAS N REINAS

RESUMEN

El Simulated Annealing, como técnica de optimización combinatorial, se usa para afrontar problemas de gran complejidad matemática; cuenta con una estrategia de aceptación para las nuevas configuraciones que permite salir de mínimos locales, y encontrar soluciones de muy alta calidad, dentro de las cuales eventualmente puede estar el óptimo global. El problema de las N Reinas es un problema clásico de búsqueda combinatorial que consiste en encontrar la ubicación de n reinas en un tablero de dimensiones n_xn, con la condición de que las reinas no se ataquen entre sí. En este artículo se muestra el algoritmo del Simulated Annealing, su manejo y su implementación para resolver el problema de las N Reinas.

PALABRAS CLAVES: Optimización combinatorial, Simulated Annealing, problema de las N Reinas.

JOHN FREDY FRANCO B.
Estudiante Maestría en Ingeniería Eléctrica. Área de Planeamiento.
jffb@utp.edu.co

POMPILIO TABARES E.
Profesor Titular
Universidad Tecnológica de Pereira
ptabares@utp.edu.co

Grupo de Planeamiento en Sistemas Eléctricos. Área de Investigación de Operaciones

ABSTRACT

Simulated Annealing, as combinatorial optimization technique, is used to deals with problems of great mathematical complexity; it counts with an acceptation strategy for the new configuration that allows escape from local minimums and find high quality solutions. N Queens problem is a classic problem of combinatorial search that requires to locate N queens on a board of n_xn dimensions with the condition that the queens must not attack each other. This paper shows the Simulated Annealing algorithm, the way to use it and its implementation to solve the N Quens problem.

KEYWORDS: *Combinatorial optimization, Simulated Annealing, N Queens problem.*

1. INTRODUCCIÓN

La técnica Simulated Annealing (SA) fue definida en el inicio de la década de los 80, como una nueva herramienta a ser empleada en la solución de grandes problemas combinatoriales. Surgió del campo de la termodinámica como consecuencia de la comparación de problemas formulados en este campo, con los del campo de la investigación operacional; es una metodología simple y de gran potencialidad para ser aplicada a una gran variedad de problemas.

El problema de las N reinas requiere encontrar la ubicación de n reinas en un tablero de dimensiones n_xn, con la condición de que las reinas no se ataquen entre sí. Es un problema clásico de búsqueda combinatorial que ha sido usado tradicionalmente para probar nuevas estrategias y algoritmos de búsqueda.

En un trabajo publicado en 1918, Ahrens [1] describió una solución analítica general para el problema de las N reinas. Estas soluciones presentan algunas limitaciones, dado que generan solamente configuraciones restringidas.

Esta limitación no aplica a la búsqueda propiamente, como una alternativa frente a las soluciones analíticas. Se han desarrollado métodos para resolver el problema que son conocidos como *backtracking search*, con la dificultad que tiene un incremento de tipo exponencial para grandes tableros.

Actualmente, existen algoritmos de búsqueda especializados que consiguen resolver el problema para grandes valores de N, como se muestra en [2].

En su primera parte, este trabajo presenta la técnica Simulated Annealing, sus fundamentos y los parámetros que la controlan. Como segunda parte, se describe el problema de las N Reinas y se plantea el modelo matemático. Finalmente se muestran los resultados de la implementación del Simulated Annealing al problema de las N Reinas.

2. SIMULATED ANNEALING.

El Simulated Annealing es una técnica de optimización combinatorial que se usa para afrontar problemas de gran complejidad matemática, de modo que se obtengan soluciones cercanas a la óptima.

La idea original que dio lugar al SA es llamada algoritmo de Metrópolis, el que a su vez está basado en el método

Fecha de Recepción: 31 Mayo de 2005
Fecha de Aceptación: 24 Noviembre de 2005

de Montecarlo, con el cual se estudian las propiedades de equilibrio en el análisis del comportamiento microscópico de los cuerpos.

Se fundamenta en el proceso físico de calentamiento de un sólido, seguido por un enfriamiento hasta lograr un estado cristalino con una estructura perfecta. Durante este proceso, la energía libre del sólido es minimizada. A una temperatura T, en el estado con nivel de energía Eì, se genera el estado siguiente j a través de de una pequeña perturbación. Si la diferencia de energía Ej – Ei es menor o igual a cero, el estado j es aceptado. Si la diferencia de energía es mayor que cero, el estado j es aceptado con una probabilidad dada por (1), donde K_B es la constante de Boltzmann. Si la disminución de la temperatura es hecha de manera paulatina, el sólido puede alcanzar el estado de equilibrio en cada nivel de temperatura.

$$e^{\frac{E_i - E_j}{k_B T}} \qquad (1)$$

El SA aplica la simulación de la secuencia de estados en el proceso de enfriamiento para la solución de problemas de minimización. El costo de la configuración se asocia a la energía y T corresponde al parámetro de control.

A partir del estado i con costo f(i), se genera el estado j con costo f(j). El criterio de aceptación determina si el nuevo estado es aceptado; para esto se calcula la probabilidad:

$$\Pr ob.\ Acep.\ j = \begin{cases} 1 & si\ f(j) \le f(i) \\ e^{\frac{f(i)-f(j)}{T}} & si\ f(j) > f(i) \end{cases} \qquad (2)$$

Con esta estrategia se evita el quedar atrapado en mínimos locales. Inicialmente cuando T es grande, se permiten nuevas configuraciones que deterioren la función objetivo. A medida que disminuye la temperatura, la probabilidad de aceptar soluciones peores que la que se tiene es cada vez menor.

Es clave la determinación de la temperatura inicial T_o, así como la velocidad de enfriamiento y la longitud N_k de la cadena de tentativas para cada nivel de temperatura T_k. Estos parámetros se calibran para adaptarse al tipo y tamaño del problema, de tal manera que se consigan soluciones satisfactoriamente con el SA.

2.1. Algoritmo simulated annealing.

A continuación se presenta el algoritmo en el que se implementa el Simulated Annealing. En la figura 1 se muestra el diagrama de flujo correspondiente.

inicializar T_0, N_0
generar configuración inicial Ri
inicializar k
repetir
  para h desde 1 hasta Nk
    generar Rj a partir de Ri
    si f(j)<=f(i)
      entonces Ri = Rj
    de lo contrario
      si exp{[(f(i)-f(j)]/T} > aleatorio [0,1]
        entonces Ri = Rj
      fin_si
    fin_si
  fin_para
  incrementar k
  cálculo de Nk
  cálculo de Tk
fin_ repetir hasta criterio de parada

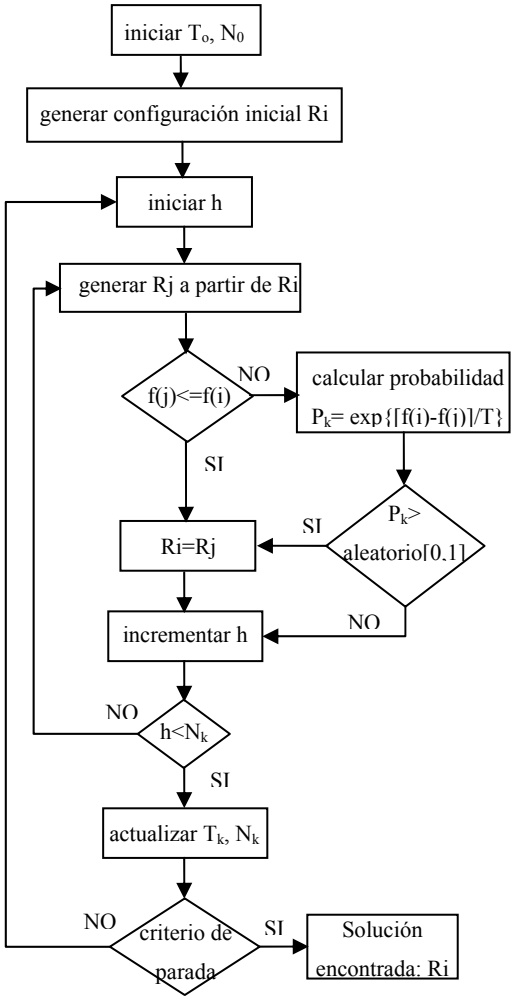

Figura 1. Algoritmo SA.

2.2. Temperatura.

La temperatura debe ser tal en el proceso que permita una adecuada exploración en las etapas del SA. Existen diferentes formas para calcular la temperatura inicial, siendo una de ellas simular el proceso para la primera cadena de Markov.

También es necesario definir la tasa *B* de disminución de la temperatura, para tener un enfriamiento controlado.

Tomándose $T_{k+1}= B\ T_k$. Se acostumbran usar tasas de 0.8 a 0.99.

La temperatura final de enfriamiento es aquella en la que ya es muy poco probable que se acepten soluciones que empeoren la actual.

2.3. Longitud de la cadena.

La longitud de la cadena N_k define el número de tentativas para el nivel de temperatura T_k. Como regla general se maneja una longitud creciente a medida que la temperatura disminuye, con el fin de permitir una exploración más intensa a temperaturas bajas. Por lo cual se toma $N_{k+1}= p\ N_k$, con $p\geq 1$.

Para definir la longitud inicial N_0 de la cadena, este se escoge proporcional al tamaño del problema, por ejemplo k veces el número de variables, $N_0=k_x$ #variables.

2.4. Criterio de parada

Se pueden usar varios criterios de parada:

- Se fija un número determinado de niveles de temperatura, usualmente entre 6 y 50.
- Si la función objetivo no mejora para varios niveles consecutivos, entonces se termina el proceso.
- Si no se cumple con un número mínimo de aceptaciones en el nivel de temperatura.

2.5. Generación de una nueva configuración.

Se debe definir una estructura de vecindad, de la cual se toma la nueva configuración a partir de la actual. La estructura de vecindad está conformada por aquellas configuraciones -no necesariamente todas- que se pueden generar mediante pequeñas modificaciones, como sacar un elemento activo, agregar uno, o intercambiar un elemento que está presente en la configuración por otro que no lo está.

La escogencia de la nueva configuración en la estructura de vecindad se hace aleatoriamente, pero se puede incrementar la probabilidad de selección para aquellas configuraciones que tienen un índice de sensibilidad que las identifica como atractivas para conseguir un mejoramiento en la función objetivo.

2.6. Adaptación al tipo de problema

En principio el algoritmo presentado está dirigido a solucionar problemas de minimización sin restricciones. Haciendo unas pequeñas modificaciones se puede adaptar a problemas de maximización o que tengan restricciones. La maximización es convertida en minimización multiplicando la función objetivo por -1.

Cuando se tienen restricciones, el modelo matemático puede ser solucionado de dos formas. La primera es resolver el modelo en su conjunto, función objetivo y restricciones. La segunda consiste en llevar las restricciones a la función objetivo a través de un factor de penalización.

3. APLICACIÓN DEL SA AL PROBLEMA DE LAS N REINAS

3.1. El problema de las n reinas

Consiste en ubicar en un tablero de ajedrez de tamaño $n_x n$, n reinas de tal manera que no existan ataques entre sí. Siempre es posible ubicar n reinas de acuerdo con esa condición.

Se puede plantear de dos maneras: ubicar n reinas minimizando las colisiones o maximizar el número de reinas en el tablero sujeto a la restricción de que no existan colisiones; siendo la primera la que se escogió en este trabajo.

El número de alternativas corresponde a n!, pero existen múltiples soluciones, incluyendo las simétricas. Por ejemplo, para N=8 hay 92 soluciones posibles dentro de las 40.320 diferentes alternativas.

Para este problema se pueden encontrar soluciones analíticas, pero son limitadas y construidas siguiendo cierta formación. En la figura 2 se tiene una configuración de este tipo que es solución del problema, es decir no se presentan ataques entre las 7 reinas del tablero 7_x7.

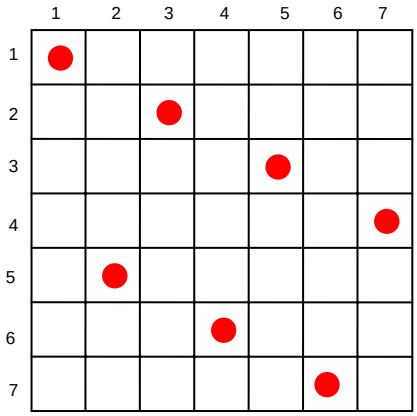

Figura 2. Una solución para el tablero 7_x7.

3.2. Representación del problema.

Debido a que las reinas no pueden ir en una misma fila, a cada reina le corresponde una fila que se le asocia. Así, la reina k se sitúa en la fila k. resta entonces definir la columna –diferente para cada una- en la que se ubica, lo que convierte el problema en una de permutación.

Si R(i) es el índice de la columna donde se localiza la reina i, entonces la configuración del tablero se describe con el vector R. Para el ejemplo del tablero 4_x4 de la figura 3, el vector de configuración R es [1, 4, 2, 3].

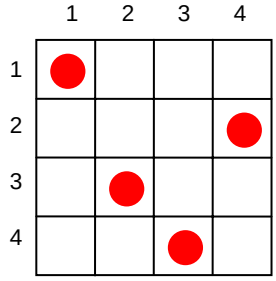

Figura 3. Disposición para un tablero 4_x4.

Para el tablero de tamaño 7_x7 de la figura 2, el vector de configuración R es [1, 3, 5, 7, 2, 4, 6].

Siendo que a cada reina se le asigna una fila y una columna que no comparte con otra, no existen colisiones entre reinas por filas o columnas; entonces debe analizarse la configuración para las diagonales.

En el tablero, las diagonales se clasifican como positivas y negativas. Una diagonal positiva se caracteriza porque la suma de sus índices (fila-columna) es constante a lo largo de la diagonal. Mientras que en una diagonal negativa se tiene que la resta es constante. Ver figura 4.

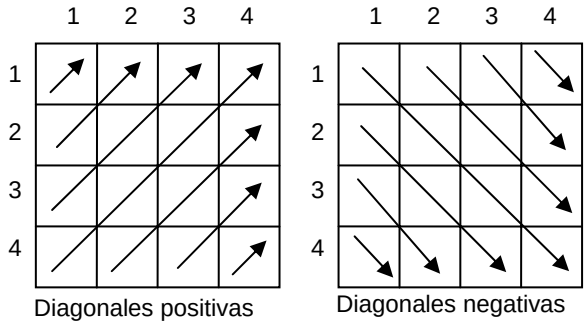

Figura 4. Diagonales en el tablero.

Si dos o más reinas comparten el mismo índice para una diagonal, entonces existe una colisión. El número de colisiones para esa diagonal es el número de reinas presentes menos uno.

3.3. Modelo matemático

Se pretende minimizar el número de colisiones, representadas por los elementos de un conjunto A. El modelo matemático del problema de las N reinas se plantea como:

$$\textit{minimizar Cardinalidad (A)}$$

$$A=\left\{\begin{array}{c}(i,j)/\ i-R(i)=j-R(j)\wedge i\neq j,\vee,\\ i+R(i)=j+R(j)\wedge i\neq j\end{array}\right\} \qquad (3)$$

$$i=1,\ldots,N \quad j=1,\ldots,N$$

$$R(i)\in[1,N],\wedge,R(i)\in Z$$

Donde: i, j son los índices que representan a las reinas

3.4. Configuración inicial

De la configuración inicial depende la rapidez para encontrar una solución. En [2] se plantea que para una inicialización hecha de forma aleatoria, se tiene que alrededor de la mitad de las reinas están en colisión. Existen estrategias para inicializar la configuración de tal manera que se reduzca ostensiblemente el número de colisiones al comienzo del proceso.

3.5. Implementación del algoritmo SA

Para solucionar el problema de las N reinas se usó el SA, construyendo las funciones necesarias en el paquete computacional Matlab.

La configuración inicial es creada de forma aleatoria y la temperatura inicial se calcula haciendo una prueba para una cadena de Markov de tamaño definido. Se genera una nueva configuración a través del intercambio de columnas entre dos reinas escogidas de forma aleatoria, pero dándole mayor probabilidad a las reinas que están en colisión de acuerdo con una lista que se va actualizando.

Se calcula el número de colisiones con una función que de forma eficiente no evalúa todo el tablero, sino que revisa las 8 diagonales que han cambiado usando la información de la configuración anterior. Esta información consiste en la posición de las reinas en las diagonales y las diagonales en las que hay colisión. Conociendo estos datos y las reinas que se intercambiaron, se encuentran los nuevos estados para las diagonales.

Los parámetros del SA dependen del tamaño del tablero a solucionar. En un principio para un tablero de tamaño N=200 se deben calibrar los parámetros de forma diferente que para uno de tamaño N=10.000. Dejándose de forma nominal B=0.8 y p=1.15. Asimismo la longitud inicial de la cadena se estableció como N_0=2N.

Los mejores resultados se obtuvieron cuando la probabilidad de crear una nueva configuración tomando una reina en colisión de la lista, era mucho mayor que tomar una reina para el intercambio de forma aleatoria.

4. RESULTADOS

Para probar la implementación del algoritmo construido se resolvió el problema de las N reinas para distintos valores de N. Para cada tablero se corrió 20 veces el programa. Se muestra en la tabla 1 el promedio del número de iteraciones en las que se alcanzó la solución del problema; los datos de la dimensión del tablero y el número de iteraciones están en miles. En la tabla 2 aparecen los datos para las veinte corridas, para N=1.000, 5.000, 10.000 y 20.000. Se observa que el número de iteraciones crece de forma lineal con el tamaño del tablero (figura 5). El tiempo de cómputo a su vez es proporcional al número de iteraciones; el tiempo para iteración se mantiene constante al aumentar la dimensión del tablero, debido a que se analizan 8 diagonales en cada iteración sin importar el tamaño del problema.

N	Iteraciones	N	Iteraciones
1	7,0	11	79,9
2	13,8	12	82,8
3	20,6	13	89,9
4	27,6	14	99,7
5	34,6	15	101,8
6	41,7	16	109,8
7	48,5	17	115,4
8	55,2	18	124,1
9	62,8	19	133,0
10	68,7	20	136,9

Tabla 1. Promedio del número de iteraciones.

N	1.000	5.000	10.000	20.000
1	7,8	35,0	69,3	134,7
2	6,8	34,4	69,3	134,6
3	7,8	33,3	71,0	138,4
4	7,4	36,5	66,3	138,6
5	7,0	35,4	67,3	136,4
6	6,9	33,9	69,6	138,8
7	6,8	36,0	67,7	134,9
8	6,8	34,4	68,3	135,5
9	7,0	35,6	670	135,6
10	7,8	32,6	72,9	140,4
11	6,2	34,1	70,3	137,9
12	7,3	32,2	68,0	134,1
13	6,7	34,7	68,8	136,8
14	6,9	31,7	69,1	141,2
15	6,5	36,0	68,2	138,7
16	6,2	35,8	68,5	137,9
17	7,3	34,1	69,1	136,1
18	6,6	36,5	64,7	138,9
19	7,2	34,9	69,4	135,1
20	6,7	34,9	68,8	134,1
Prom.	**7,0**	**34,6**	**68,7**	**136,9**
des. est.	0,5	1,4	1,7	1,4

Tabla 2. Número de iteraciones para 20 casos.

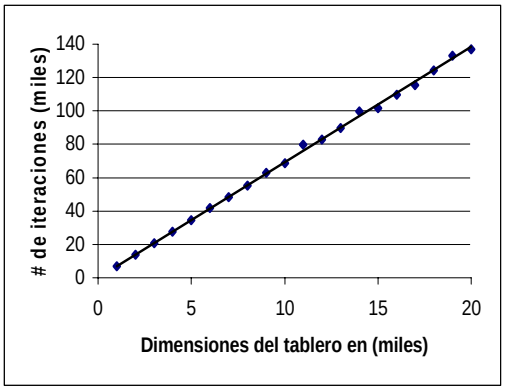

Figura 5. Gráfica del promedio del número de iteraciones.

5. CONCLUSIONES.

- Se presentó el Simulated Annealing como una técnica que permite la solución de problemas combinatoriales, mostrando su aplicación a un problema clásico como el de las N Reinas.

- Se encontró que el número de iteraciones necesarias para resolver el problema de las N Reinas es proporcional al tamaño del tablero usando el programa implementado.

- Gracias a la estrategia del SA siempre se encontró una solución para el problema, sin necesitar de reiniciar el programa, como se propone en [2].

- Es posible alcanzar tamaños más grandes del tablero programando en lenguajes tales como C, Delphi, Fortran.

6. BIBLIOGRAFÍA.

[1] AHRENS W. Mathematische Unterhaltungen Leipzig. 1918.

[2] SOSIČ Rok, GU Jun. Fast search algorithms for the N-Queens problem. IEEE Transactions on Systems Man and Cybernetics Vol. 21 No. 6, 1991.

[3] KALÉ L. V. An almost perfect heuristic for the N nonattacking queens problem. Information Processing Letters 34, 1990.

[4] GALLEGO Ramón, ESCOBAR Antonio, ROMERO Rubén. Métodos de Optimización combinatoriales, texto Maestría en Ingeniería Eléctrica, 2004

[5] AARTS Emile, KORST Jan. Simulated Annealing an Boltzmann Machines, 1989.

[6] The N Queens Problem. Página de Internet disponible en: http://ww w.durangobill.com/N_Queens.html