Instituto tecnológico de Culiacán

Unidad 2

Actividad:

Tarea 3: Resolver el problema 8 reinas con recocido simulado

Alumno:

Ivan Eduardo Ramírez moreno

Docente:

ZURIEL DATHAN MORA FELIX

Materia:

Tópicos de IA

Numero de control:

20170787

Semestre:

10

Introducción

El problema de las N reinas es uno de los problemas clásicos en el campo de la teoría de juegos y la inteligencia artificial. Consiste en colocar NN reinas en un tablero de ajedrez de tamaño N×NN \times N de forma que ninguna de ellas se ataque entre sí. Esto implica que no puede haber dos reinas en la misma fila, columna o diagonal. El problema tiene aplicaciones en diversas áreas como la optimización, la teoría de grafos y la planificación de tareas.

Resolver este problema mediante métodos tradicionales, como la búsqueda exhaustiva (backtracking), es factible para tamaños pequeños del tablero. Sin embargo, a medida que NN aumenta, el espacio de búsqueda crece exponencialmente, haciendo que encontrar una solución óptima sea computacionalmente costoso. Aquí es donde los algoritmos metaheurísticos, como el recocido simulado, se vuelven una herramienta valiosa al ofrecer soluciones aproximadas en tiempos razonables.

El recocido simulado se inspira en un proceso físico de la metalurgia llamado recocido, donde un material se calienta y luego se enfría lentamente para reducir defectos estructurales. En el contexto de la computación, este método se utiliza para escapar de mínimos locales y encontrar soluciones cercanas al óptimo global.

Desarrollo

Definición del Algoritmo

El recocido simulado es un algoritmo de optimización estocástica diseñado para encontrar una solución aproximada a problemas complejos. En el caso de las N reinas, el objetivo es minimizar el número de conflictos entre las reinas. El algoritmo consta de las siguientes etapas:

1. **Estado inicial**: Se inicia con una configuración del tablero donde las reinas están distribuidas aleatoriamente o según un estado inicial proporcionado por el usuario. Este estado es representado por una lista donde cada valor indica la posición de una reina en una fila.
2. **Función de costo**: Evalúa la calidad de un estado midiendo el número de conflictos. Los conflictos incluyen reinas en la misma fila, columna o diagonal. Un estado óptimo tendrá un costo de 0, indicando que no hay conflictos.
3. **Generación de vecinos**: Se crean estados vecinos modificando la posición de las reinas. Esto se hace intercambiando dos reinas aleatoriamente en el tablero. Los vecinos representan pequeños cambios en el estado actual y permiten explorar el espacio de búsqueda de manera local.
4. **Aceptación de nuevos estados**: La aceptación de un vecino depende de dos factores:

- **Mejora directa**: Si el costo del vecino es menor que el estado actual, se acepta inmediatamente.
- **Aceptación probabilística**: Incluso si el vecino tiene un costo mayor, puede aceptarse con una probabilidad que depende de la temperatura actual y de la diferencia de costo. Esto ayuda a escapar de mínimos locales.

5. **Enfriamiento**: La temperatura se reduce gradualmente durante las iteraciones usando una tasa de enfriamiento. Este proceso controla la exploración: al inicio permite movimientos más aleatorios y, hacia el final, se enfoca en la explotación de soluciones cercanas al mejor estado.

Implementación del Algoritmo

El algoritmo se implementa de manera que, después de un número definido de iteraciones o una temperatura extremadamente baja, se retorna el mejor estado encontrado. Las funciones clave son:

- cost: Evalúa la calidad de un estado.
- generate_neighbor: Genera un nuevo estado vecino.
- search: Controla el flujo principal del algoritmo, incluyendo el enfriamiento y la aceptación de nuevos estados.

El algoritmo es versátil y permite ajustes en parámetros como la temperatura inicial, la tasa de enfriamiento y el número máximo de iteraciones, lo que lo hace adecuado para diferentes tamaños de NN.

Ventajas del Recocido Simulado

- **Escalabilidad**: Es adecuado para tamaños grandes de NN, donde los métodos exactos son imprácticos.
- **Escape de mínimos locales**: La aceptación probabilística ayuda a evitar estancarse en soluciones subóptimas.
- **Facilidad de implementación**: La lógica del algoritmo es sencilla y flexible, permitiendo modificaciones según las necesidades del problema.

Limitaciones

- **Dependencia de parámetros**: El rendimiento del algoritmo depende en gran medida de la elección de parámetros como la temperatura inicial y la tasa de enfriamiento.
- **Resultados no garantizados**: Debido a su naturaleza probabilística, no siempre asegura encontrar la solución óptima.

- **Requiere ajuste manual**: A menudo es necesario experimentar con configuraciones diferentes para mejorar su desempeño.

Conclusión

El uso del recocido simulado para resolver el problema de las N reinas demuestra cómo los algoritmos metaheurísticos pueden abordar problemas de optimización complejos de manera eficiente. Aunque no siempre garantiza soluciones óptimas, su capacidad para escapar de mínimos locales y encontrar configuraciones cercanas al óptimo lo hace especialmente valioso en problemas donde el espacio de búsqueda es muy grande.

En este caso, el recocido simulado ofrece un balance adecuado entre simplicidad, eficiencia y efectividad. Sin embargo, su desempeño depende de la correcta selección de parámetros y del diseño adecuado de funciones como la generación de vecinos y la evaluación de costo. Para problemas más grandes o con restricciones adicionales, se pueden integrar estrategias avanzadas como enfriamiento adaptativo o estados iniciales inteligentes.

Este enfoque no solo resuelve el problema de las N reinas, sino que también puede adaptarse para abordar una amplia variedad de problemas de optimización combinatoria, consolidándolo como una herramienta esencial en inteligencia artificial y ciencias computacionales.

Referencias

- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, *220*(4598), 671–680. https://doi.org/10.1126/science.220.4598.671
- Pearl, J. (1984). *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Addison-Wesley.
- Russell, S., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach* (3rd ed.). Pearson.
- Reeves, C. R. (1993). Modern heuristic techniques for combinatorial problems. *Advanced Topics in Computer Science*. McGraw-Hill.
- Michalewicz, Z., & Fogel, D. B. (2000). *How to Solve It: Modern Heuristics*. Springer.