UNIVERSIDADE FEDERAL DE MATO GROSSO - CAMPUS ARAGUAIA
INSTITUTO DE CIÊNCIAS EXATAS E DA TERRA - ICET
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Wesley Antonio Junior dos Santos

Problema n-rainhas com n=32, 64 e 128 relatório.

INTELIGÊNCIA ARTIFICIAL

Barra do Garças - MT
Setembro de 2024

1. Modelagem:

1.1. Hill-Climbing

Representação do Tabuleiro: O tabuleiro é representado por uma lista de inteiros, onde o índice representa a coluna e o valor no índice representa a linha da rainha naquela coluna. Por exemplo, se a lista for [0, 2, 1] para n=3, isso significa que a primeira rainha está na posição (0,0), a segunda na posição (1,2), e a terceira na posição (2,1).

Função de Avaliação (Heurística): A função de avaliação conta o número de pares de rainhas que estão se atacando, seja na mesma linha ou na mesma diagonal. Esta função é crucial, pois o algoritmo tenta minimizar este valor a cada iteração.

Operador de Vizinhança: A vizinhança é gerada através da troca de posição de duas rainhas escolhidas aleatoriamente. A ideia é que essa troca possa potencialmente reduzir o número de ataques, explorando soluções próximas da configuração atual.

Critério de Parada: O algoritmo para se encontrar uma solução sem conflitos ou ao atingir o número máximo de passos (max_steps), definido como 80.000. Este valor foi escolhido para fornecer um balanço entre tempo de execução e a chance de encontrar uma solução.

Pontos Fortes e Fracos: O Hill Climbing é simples e rápido, mas sofre de problemas relacionados a mínimos locais, onde o algoritmo pode ficar preso em uma solução subótima. Esse comportamento é particularmente problemático para valores maiores de n, onde o espaço de soluções é vasto.

1.2. Simulated Annealing

Representação do Tabuleiro: Similar ao Hill Climbing, uma lista de inteiros é usada para representar o tabuleiro, com cada índice representando uma coluna e o valor representando a linha.

Função de Avaliação (Heurística): O número de pares de rainhas em conflito é novamente utilizado como a função de avaliação.

Escala de Temperatura:

Temperatura Inicial (initial_temperature): A temperatura inicial é definida em 150.0 para permitir uma ampla aceitação de movimentos,

mesmo que eles aumentem o número de conflitos. Isso evita que o algoritmo fique preso em mínimos locais nas fases iniciais.

Taxa de Resfriamento (cooling_rate): A taxa de resfriamento é 0,97, o que significa que a temperatura diminui 4,5% a cada iteração. Esse valor foi escolhido para permitir uma desaceleração gradual do processo, permitindo ao algoritmo explorar o espaço de soluções amplamente antes de começar a focar em refinamentos.

Critério de Aceitação: O algoritmo aceita uma solução pior com uma probabilidade que diminui à medida que a temperatura cai. Essa probabilidade é determinada pela função $e^{-\Delta E/T}$, onde ΔE é a diferença no número de ataques e T é a temperatura. Um limite de 700 foi imposto para evitar overflow durante o cálculo exponencial.

Critério de Parada: O algoritmo para se uma solução livre de conflitos for encontrada ou se o número máximo de iterações (max_iterations) for atingido, o que foi definido como 80.000. Este limite foi escolhido para dar ao algoritmo tempo suficiente para explorar o espaço de soluções, especialmente para valores maiores de n.

Pontos Fortes e Fracos: O Simulated Annealing é eficaz para escapar de mínimos locais, mas depende muito da escolha adequada dos parâmetros de temperatura e resfriamento. Com parâmetros bem escolhidos, pode-se alcançar o mínimo global, mas isso vem ao custo de um tempo de execução maior em comparação ao Hill Climbing.

1.3. Algoritmo Genético

Representação do Cromossomo: Cada cromossomo é representado por uma lista de inteiros de tamanho n, onde cada inteiro representa a posição de uma rainha em uma coluna específica. Essa representação permite que cada cromossomo codifique uma solução potencial para o problema das n-rainhas.

Função de Avaliação (Fitness): O número de pares de rainhas em conflito é utilizado para determinar a "aptidão" de cada cromossomo. Um cromossomo com menos conflitos tem maior aptidão e, portanto, uma maior chance de ser selecionado para reprodução.

Operadores Genéticos:

Crossover: Um ponto de crossover é escolhido aleatoriamente entre as posições 1 e n−2 para recombinar genes dos pais, gerando dois novos cromossomos. Esta técnica permite a combinação de características dos pais, potencialmente criando descendentes mais aptos.

Mutação: Com uma probabilidade de 30% (mutation_rate = 0.3), uma mutação é aplicada onde uma rainha é movida para uma linha aleatória em sua coluna. Essa taxa relativamente alta foi escolhida

para manter a diversidade genética na população, evitando que o algoritmo converja prematuramente para uma solução subótima.

Seleção: A seleção dos pais para o crossover é feita de forma proporcional à aptidão, com os melhores indivíduos tendo uma chance maior de se reproduzir. Isso assegura que boas características tenham maior probabilidade de ser transmitidas à próxima geração.

Critério de Parada: O algoritmo para se uma solução sem conflitos for encontrada ou se o número máximo de gerações (generations) for atingido, definido como 80.000. Este valor foi escolhido para dar ao algoritmo uma quantidade suficiente de iterações para evoluir e refinar a população.

Pontos Fortes e Fracos: O Algoritmo Genético é poderoso na exploração de grandes espaços de soluções e pode escapar de mínimos locais através da mutação e recombinação genética. No entanto, esse poder vem com um custo computacional significativo, especialmente para grandes populações e muitas gerações.

2. Custo computacional:

- **Análise do desempenho para 32 rainhas:**

  Tempo médio Hill Climbing: 0.9439 segundos,
  Tempo médio Simulated Annealing: 0.2313 segundos,
  Tempo médio Genetic Algorithm: 11.3316 segundos,
  O algoritmo mais rápido foi: Simulated Annealing com tempo de 0.2313 segundos

- **Análise do desempenho para 64 rainhas:**

  Tempo médio Hill Climbing: 10.5696 segundos,
  Tempo médio Simulated Annealing: 0.6941 segundos,
  Tempo médio Genetic Algorithm: 31.5890 segundos,
  O algoritmo mais rápido foi: Simulated Annealing com tempo de 0.6941 segundos

- **Análise do desempenho para 128 rainhas:**

  Tempo médio Hill Climbing: 32.4189 segundos,
  Tempo médio Simulated Annealing: 1.4512 segundos,
  Tempo médio Genetic Algorithm: 63.5650 segundos,
  O algoritmo mais rápido foi: Simulated Annealing com tempo de 1.4512 segundos

3. Resultados obtidos:

O código foi estruturado para resolver o problema das n-rainhas utilizando Hill Climbing, Simulated Annealing e Algoritmo Genético. O Hill Climbing é uma busca local que tenta melhorar a solução atual a cada iteração, mas pode ficar preso em ótimos locais. Nos testes, para n = 32 ele encontrou o mínimo global na maioria das execuções, mas para n = 64 e n = 128 ele frequentemente encontrou mínimos locais, com uma média de 0,2 a 0,8 ataques. O Simulated Annealing, por outro lado, usa uma "temperatura" para aceitar temporariamente soluções piores e escapar de ótimos locais. Ele foi eficiente em encontrar soluções globais, especialmente para n = 32 e n = 64, com uma média de 0 a 0,4 ataques.

O Algoritmo Genético evolui uma população de soluções, utilizando seleção, crossover e mutação, o que o torna robusto em encontrar o mínimo global. Em todos os testes, para n = 32, 64 e 128, ele encontrou consistentemente soluções sem ataques, ou seja, o mínimo global, com uma média de ataques igual a zero. No entanto, isso teve um custo em termos de tempo de execução, especialmente para n maiores, onde o tempo médio foi significativamente mais alto que nos outros algoritmos. Em resumo, enquanto o Hill Climbing é rápido mas pode encontrar mínimos locais, o Simulated Annealing equilibra tempo e qualidade, e o Algoritmo Genético garante o mínimo global com maior custo de tempo.

4. Discussão sobre o comportamento dos métodos:

Os três métodos mostram como as estratégias de otimização diferem. O Hill Climbing, que é uma busca local, tem dificuldade em problemas maiores como n = 64 e n = 128, pois fica preso em soluções imperfeitas, chamadas de mínimos locais. O Simulated Annealing, por outro lado, é mais eficiente porque pode aceitar soluções piores temporariamente para escapar desses mínimos locais. Isso o torna mais robusto e rápido, exceto em alguns casos para n = 128. Já o Algoritmo Genético evita esses mínimos locais ao manter uma diversidade de soluções, mas é o mais lento dos três. No entanto, ele encontra soluções de alta qualidade, o que é útil quando a qualidade é mais importante que o tempo de execução.