Fillipe Augusto de Siqueira Gomes
Inteligência Artificial - 2020/1 - Flexibilizado
14/10/20

Problema das n-Rainhas
Relatório

Atividade Proposta: Implementar o problema das n-rainhas com n = [32. 64, 128]

Organizar o tabuleiro de forma que as rainhas não se ataquem.

O problema foi solucionado usando 3 métodos: Hill-Climbing, Simulated Annealing e Algoritmo Genético.

Hill Climbing:

- Estado: refere-se a qualquer configuração do conjunto do tabuleiro N x N onde além disso permite que apenas exista uma rainha por coluna, nó algoritmo, o mesmo é definido através de um array de tamanho N).
- Vizinhos: Vizinhos de um estado são outros estados com configuração de tabuleiro diferente do estado atual, respeitando a ordem de haver apenas uma única rainha no tabuleiro.

O Hill Climbing consiste em começar a configuração do tabuleiro com um estado aleatório. Logo após definirmos um estado aleatório, verificamos todos os possíveis vizinhos do estado atual e vamos em direção ao vizinho com o maior valor objetivo, caso encontrado. Caso não haja um maior do que o estado atual, mas existam mais de um com o mesmo grau, decidimos aleatoriamente o próximo vizinho. Faremos esse processo de verificação dos vizinhos até que não exista nenhum mais alto do que o estado atual.

Após a busca local dos estados, temos a mínima local, já para a global, temos a possibilidade de a cada vez que adicionarmos um vizinho ou reiniciar a mínima local, aumentarmos nossas chances de encontra-lá. No caso do algoritmo das n-rainhas a mínima global acaba sendo 0, visto que queremos o mínimo possível de rainhas que se ataquem.

Durante os estudos de implementação foi possível perceber que usando o método do vizinho aleatório, temos uma pequena chance de encontrar a solução mínima, enquanto que reiniciando temos uma chance maior de solução ótima.

Qtd-rainhas	32	64	128
Tempo de Execução	37.5segundos	15min45s	
Função custo	-1.0	-1.0	
Qtd-Trocas	34	57	

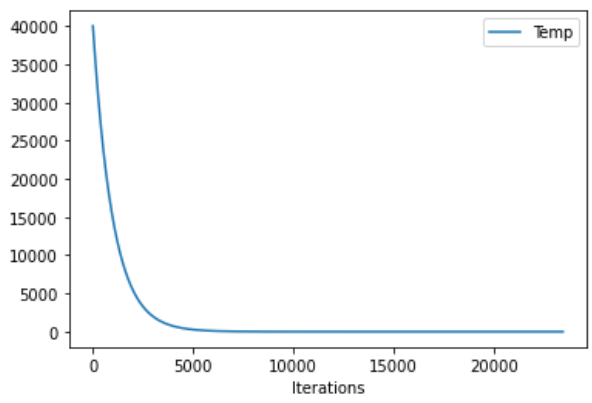
Pode ser percebido que o hill-climbing de 128 Rainhas se tornou inviável em tempo de execução. Mais detalhes sobre a implementação disponíveis na execução do código.

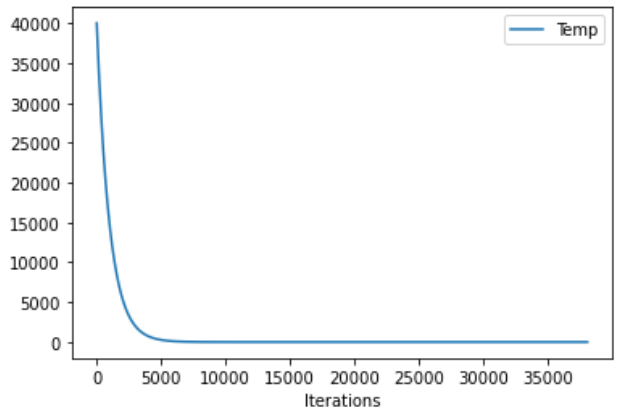
Simulated Annealing:

O método em questão consiste em fazer movimentos arriscados assim como mais simples em busca da solução. Definimos uma temperatura inicial que afeta o movimento randômico. Caso a temperatura seja alta, ocorrem os movimentos mais aleatórios e arriscados. Além disso temos o parâmetro de resfriamento, que definimos em como a temperatura será reduzida. Como é visto nas imagens, a medida que reduzimos a temperatura, ,aumenta-se as iterações.

Qtd-rainhas	32	64	128
Tempo de Execução	15.6segundos	3.4minutos	
Função custo	2.6	1.15	
Memoria	~110.07bytes	~125bytes	

Mais detalhes sobre a implementação disponíveis na execução do código.





Dentro do espaço de 1 hora não foi possível checar a uma solução para 128 rainhas. Assim como HillClimbing, não é recomendado para alto numero de indivíduos. Porém ainda assim, consome menos tempo mas ainda não garante uma solução ótima.

Algoritmo Genético:

No algoritmo genético levamos em consideração um espaço de solução aleatório definindo-se pela população inicial. E a partir disso vamos combinando os melhores indivíduos dessa população e gerando novas gerações de indivíduos (crossover).

Os cromossomos dos indivíduos são divididos em vários pontos de corte que podem assim serem escolhidos aleatoriamente. Semelhante a evolução das especies a cada vez que criamos uma nova geração, encontramos melhores soluções para o problema.

Ainda á um fator de mutação onde ele se aplica de maneira única para cada filho gerado, é bastante importante para garantir a convergência, visto que garante que não haja probabilidade zera numa busca.

Todavia no algoritmo desenvolvido não realizei o fator de mutação. Vale ainda salientar que com a aleatoriedade na execução, nem sempre serão gerado os melhores resultados e podem variar a quantidade de gerações para uma mesma quantidade de rainhas.

Qtd-rainhas	**32**	**64**	**128**
Qtd de gerações	27	99 (Valor limite)	99 (Valor limite)
Tempo de Execução	1,7 segundos	6.4 segundos	24 segundos
Sum Fitness	14647	60068	242703
Max Fitness	490	2004	8093
Memoria	40bytes	37bytes	26bytes

Podemos por fim perceber que o Algoritmo genético tem um desempenho extremamente superior ao hillclimbing e ao simulated annealing para a resolução do problema em questão.