

N Queens valorado utilizando computação evolucionária

Vinicius Gasparini¹

¹Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina (UDESC)
Centro de Ciências Tecnológicas – Joinville – SC – Brasil

v.gasparini@edu.udesc.br

Resumo. O presente relatório busca propor uma solução algorítmica ao problema das N rainhas valorado. Será apresentado esta variação do problema clássico – arranjar N rainhas em um tabuleiro de dimensão $N \times N$ com o acréscimo de um valor lucro – a codificação dos indivíduos e modelagem do *fitness*. Será também apresentado parâmetros e métricas obtidas nos testes realizados, além dos resultados obtidos e análises.

1. Problema

O problema básico das N rainhas tem como base o jogo de xadrez. A rainha é uma peça que pode se locomover tanto ortogonalmente quanto diagonalmente dentro do tabuleiro. O problema portanto consiste em dispor N rainhas em um tabuleiro de tamanho $N \times N$ de modo que nenhuma delas possa ser atacada por outra rainha. A Figura 1 apresenta uma solução possível para o problema quando $N = 8$.

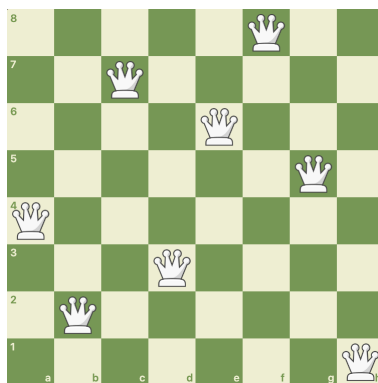


Figura 1. Solução possível para o problema quando $N = 8$.

A variação do problema que é objeto desse estudo se dá ao incluir uma nova dimensão a solução. Além das colisões serem um fator de satisfazibilidade da solução, agora cada célula do tabuleiro recebe um valor, este valor é descrito pelo seguinte processo:

- Cada célula recebe um valor inteiro de 1 até N conforme exemplificado na Figura 2
- Cada linha ímpar terá o valor de sua célula substituído pela raiz quadrada do valor presente anteriormente.
- Cada linha par terá o valor de sua célula substituído pelo logaritmo base 10 do valor presente anteriormente.

Ao final, o tabuleiro valorado terá a forma conforme exemplificado na Figura 3.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

Figura 2. Distribuição inicial do tabuleiro quando $N = 8$

1	1.41	$\sqrt{3}$	$\sqrt{4}$	2.82
0.95	1	$\log_{10}(11)$	$\log_{10}(12)$	1.20
4.12	4.24	$\sqrt{19}$	$\sqrt{20}$	4.89
1.40	1.41	$\log_{10}(27)$	$\log_{10}(28)$	1.50
5.74	5.83	$\sqrt{35}$	$\sqrt{36}$	6.32
1.61	1.62	$\log_{10}(43)$	$\log_{10}(44)$	1.68
7	7.07	$\sqrt{51}$	$\sqrt{52}$	7.48
1.75	1.76	$\log_{10}(59)$	$\log_{10}(60)$	1.80

Figura 3. Distribuição valorada do tabuleiro quando $N = 8$

1.1. Objetivo

O objetivo portanto dessa variante do problema é encontrar um posicionamento válido para as N rainhas de modo que não haja colisões e com lucro máximo. Lucro de uma solução é calculado pelo somatório dos valores das células onde há uma rainha. Para a solução proposta na Figura 1 temos:

$$lucro_i = \sqrt{6} + \log_{10}(11) + \sqrt{21} + \log_{10}(31) + \sqrt{33} + \log_{10}(44) + \sqrt{50} + \log_{10}(64) \quad (1)$$

Sendo assim, uma observação interessante a se fazer é que o escopo do problema é multimodal, pois apesar de diversas soluções, existem diversos atratores locais mas um máximo global.

2. Modelagem

2.1. Codificação

A codificação de um individuo deverá representar a distribuição das N rainhas no tabuleiro. Deste modo, opta-se por utilizar uma lista de inteiros permutados de 0 a $N - 1$. Cada índice desta lista – ou vetor – representa uma linha do tabuleiro e o valor inteiro contido representa a coluna em que a rainha esta sendo posicionada. A codificação da solução apresentada na Figura 1 seria portanto $[5, 2, 4, 6, 0, 3, 1, 7]$. Esta modelagem garante que uma restrição – de que duas rainhas não podem ocupar a mesma linha – seja incorporada já no cromossomo do individuo.

2.2. Fitness

O *fitness* de um individuo será dado pela relação entre suas colisões e seu lucro. Sendo o ótimo um individuo sem colisões e lucro o mais próximo possível de seu máximo.

O lucro máximo (Eq. 2) é obtido através do posicionamento de todas as N rainhas na última linha onde o operador é a raiz quadrada, isso se dá pelo fato de que a curva de \log_{10} é sempre menor que a curva de raiz quadrada – demonstração omitida porém trivial.

$$lucro_{max} = \sum_{i=0}^{N-1} tab[N-1][i] \quad (2)$$

Quanto as colisões, a quantidade máxima de colisões (Eq. 3) se dá quando todas as rainhas estão alinhadas, seja verticalmente seja diagonalmente – horizontalmente é impossível devido restrição estar incluída na codificação. Para cada par de indivíduos, a colisão entre A e B é a mesma de B e A, logo, o máximo de colisões do tabuleiro será a combinação N por 2.

$$colisao_{max} = C_{N,2} = \frac{N!}{2! \times (N-2)!} \quad (3)$$

Estes valores de máximo são necessários para que o lucro e a colisão de um indivíduo sejam normalizados. Por fim, temos que o *fitness* é dado pela subtração do lucro normalizado pela quantidade de colisões normalizada, ou seja, colisões são penalidades.

2.3. Parametrização

O critério de parada a ser utilizado será quantidade de gerações. Cada dimensão do problema contará com uma quantidade de gerações pré-definidas. O método de seleção utilizado será do tipo torneio estocástico, haverá *crossover* do tipo PMX, operador de mutação do tipo *swap* e elitismo. Além disso, foi incluída a rotina de *generation gap* iniciando com $g = 0.5$ e incrementando em 0.125 a cada $1/5$ da quantidade máxima de gerações, causando que o último quinto execute com $g = 1$. Cada instância do problema contará com 10 execuções com o mesmo conjunto de parâmetros, assim sendo possível métricas como desvio padrão entre as diversas tentativas.

3. Análise dos resultados obtidos

3.1. Quando $N = 8$

Dado a configuração abaixo, atingimos os seguintes resultados:

- DIM = 8
- COD = INT_PERM
- LIM = [1,8]
- GEN = 750
- POP = 30
- PM=0.05
- PC=0.8

Onde, *DIM* é a dimensão N do problema, *COD* o tipo de codificação usada, neste caso permutação de inteiros, *LIM* sendo os limites da codificação, *GEN* a quantidade de gerações utilizadas na simulação, *POP* o tamanho da população, *PM* a probabilidade de mutação em porcentagem e *PC* a probabilidade de *crossover* em porcentagem.

Na Figura 4 vemos algumas informações. Acima das curvas tomamos conhecimento do *fitness* alcançado pelo descrito indivíduo. No caso dessa instância, foi realizado a busca exaustiva pelo ótimo através da geração dos 40320 indivíduos permutados e computado seu *fitness*. O indivíduo ótimo encontrado pelo algoritmo genético coincide com

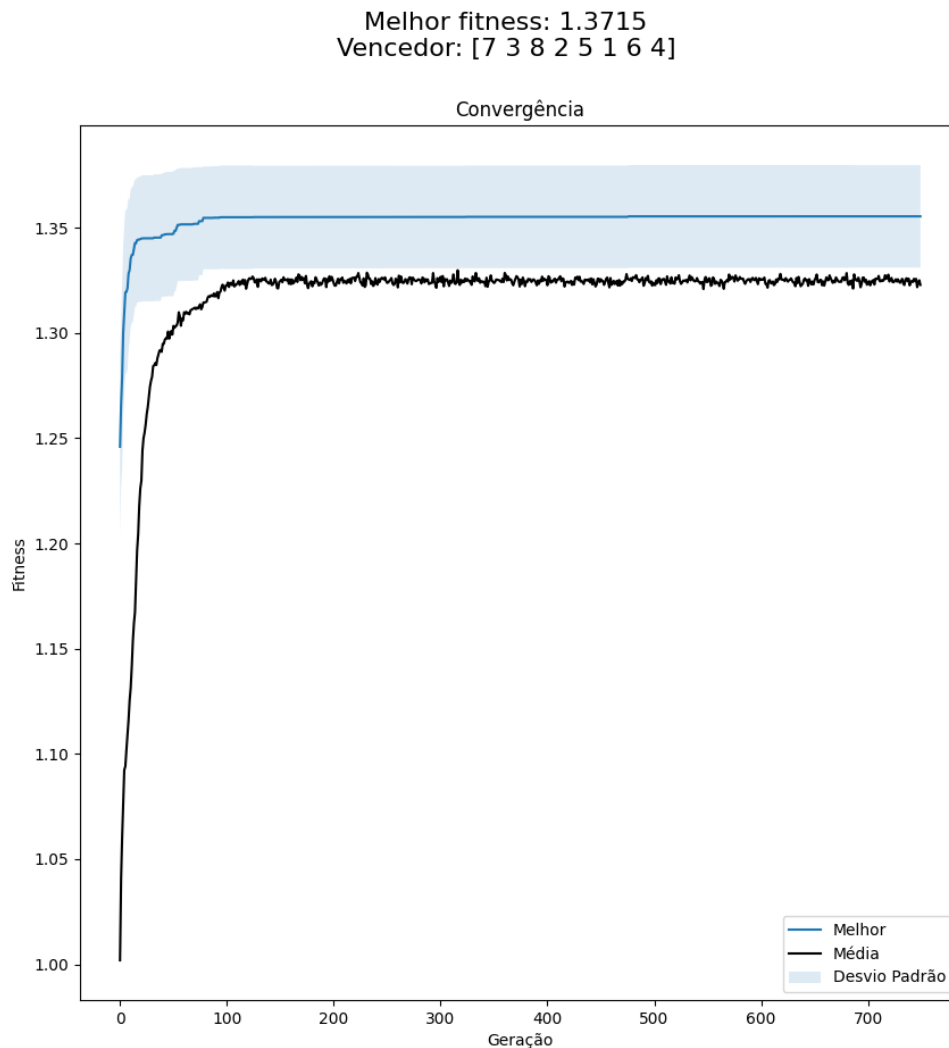


Figura 4. Gráfico de convergência quando $N = 8$.

a melhor solução apresentada pela busca exaustiva. Sendo assim, o algoritmo se provou eficaz. A taxa de lucro para esse indivíduo foi de 26.823264.

Ainda neste figura, é possível observar a curva do melhor *fitness* – em azul – crescente de maneira a se assemelhar a curva logarítmica. Em preto temos a linha do valor médio entre os indivíduos. A crescente deste dado demonstra que está ocorrendo a seleção dos melhores e as gerações futuras possuem desempenho melhor que seus antepassados.

Na Figura 5 foram plotados os gráficos que relacionam 4 execuções do algoritmo e os compara lado a lado. Vemos que a curva de piores soluções apesar de caótica, geram poucas soluções significativamente ruins, na média a espécie está progredindo. Nesta execução de exemplo o ótimo não foi alcançado por nenhuma das instâncias de teste.

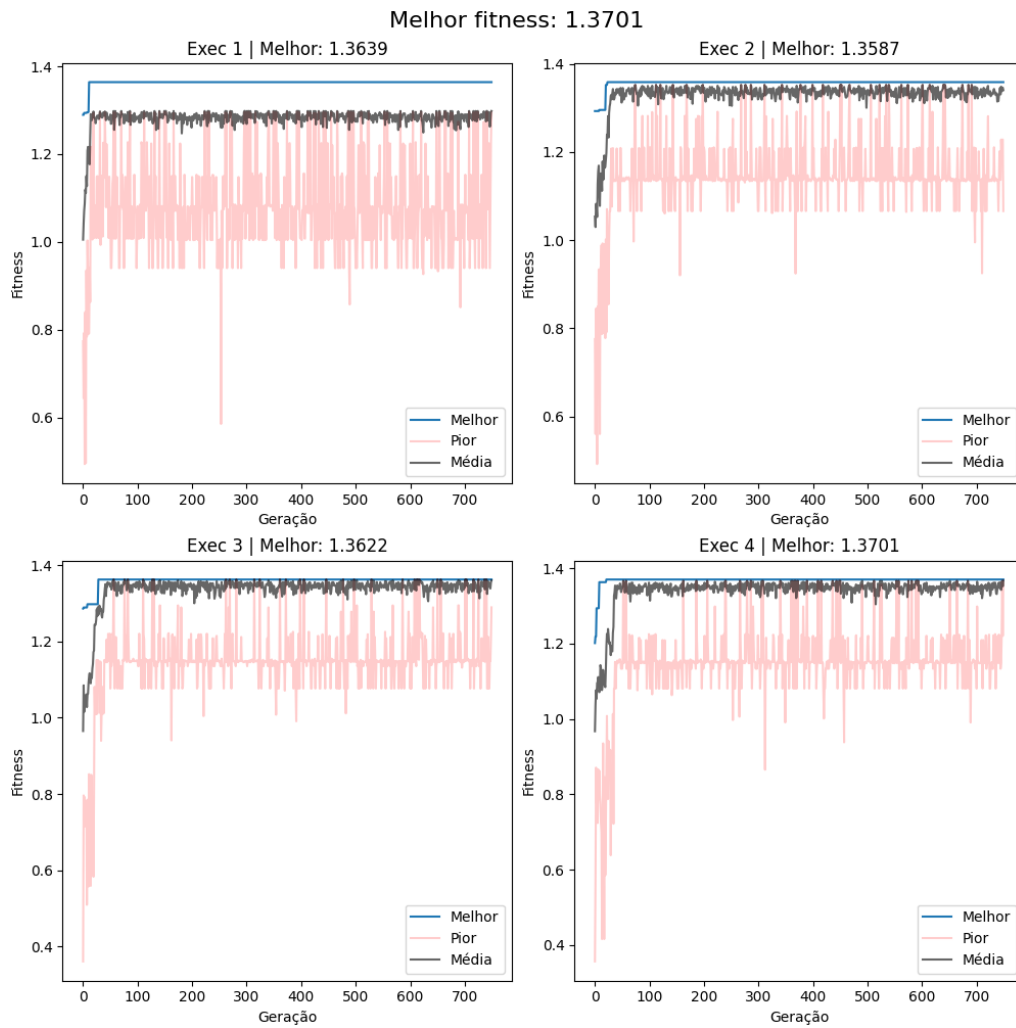


Figura 5. Gráfico de 4 execuções para $N = 8$.

3.2. Quando $N = 16$

Dado a configuração abaixo, atingimos os seguintes resultados:

- DIM = 16
- COD = INT_PERM
- LIM = [1,16]
- GEN = 1500
- POP = 30
- PM=0.15
- PC=0.9

As principais alterações de parâmetros em relação ao teste anterior se deu pelo impacto que a rotina de *generation gap* proporcionou. Vemos na Figura 6 uma curva de convergência menos acentuada no início das gerações, isso demonstra que houve desacelero da velocidade de convergência.

A taxa de lucro para o melhor indivíduo encontrado foi de 100.570555. Como as dimensões do problema escalam de maneira fatorial, não foi possível determinar qual é o ponto ótimo para 16 rainhas. Mas a solução encontrada é válida – isto é, não possui

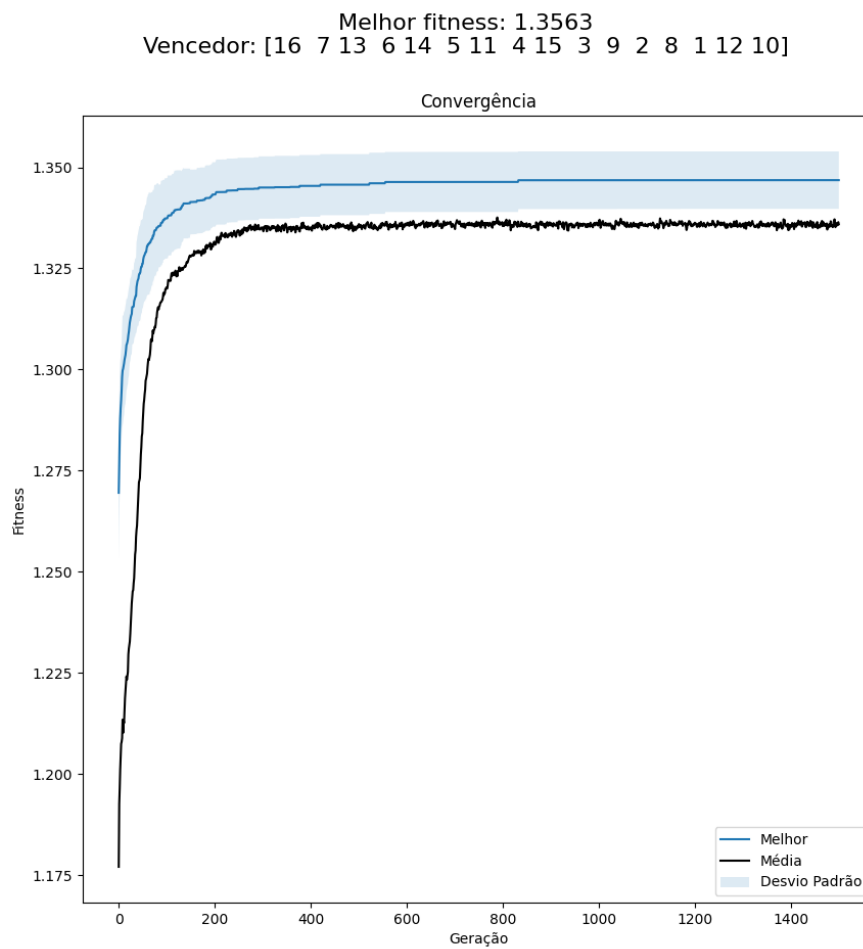


Figura 6. Gráfico de convergência quando $N = 16$.

nenhuma colisão. Computado a uma taxa de 2500 gerações / segundo, levou-se menos de 2 minutos para computar todas as 10 instâncias de teste. Com isso, decidiu-se extrapolar para tabuleiros maiores. Mantendo as configurações de parâmetros somente aumentando a quantidade máxima de gerações.

3.3. Demais tabuleiros

Vemos na Figura 7 o gráfico de convergência para algumas outras instâncias de teste. Na Tabela 1 vemos o resumo dos resultados referentes às execuções do algoritmo em diversas bases. Na Tabela 2 vamos os cromossomos dos indivíduos com melhor desempenho.

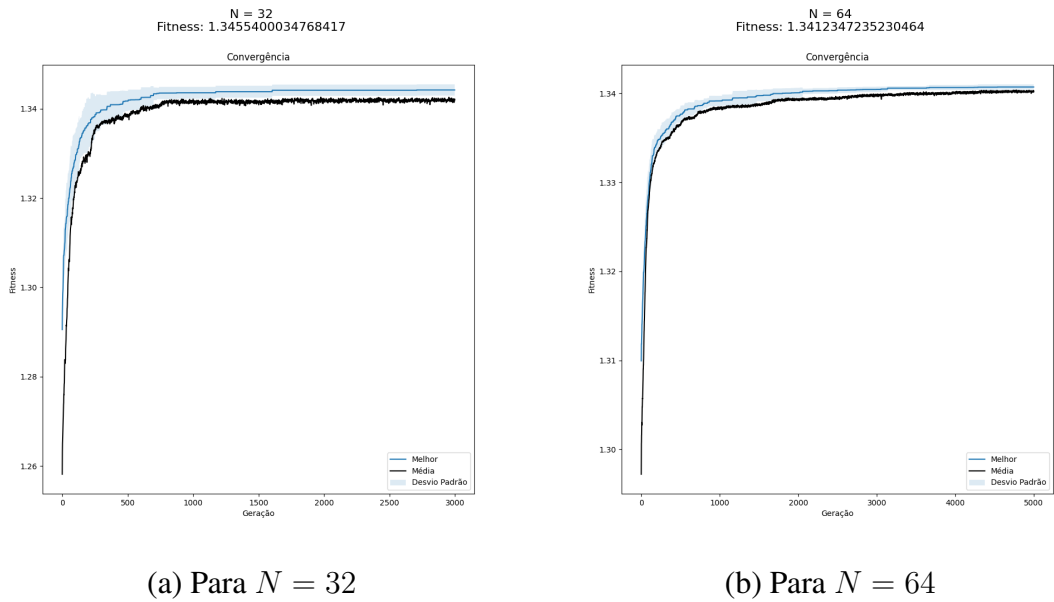


Figura 7. Experimento com tabuleiros maiores

Parâmetros				Resultado		
N	PM	PC	GEN	Fitness	Lucro	Válido
8	9%	80%	750	1.371497	26.823264	Sim
16	15%	90%	1500	1.356280	100.570555	Sim
32	15%	90%	3000	1.345540	378.737369	Sim
64	15%	90%	5000	1.341235	1460.120778	Sim

Tabela 1. Resumo dos resultados

N	Cromossomo
8	[7 3 8 2 5 1 6 4]
16	[16 7 13 6 14 5 11 4 15 3 9 2 8 1 12 10]
32	[26 24 13 6 20 3 10 4 32 9 23 16 27 21 31 7 25 12 14 2 22 28 15 5 30 11 17 8 18 1 29 19]
64	[62 36 48 43 54 21 60 33 63 39 44 16 19 10 61 46 26 42 38 3 56 15 45 9 28 5 53 12 29 18 58 7 41 11 17 34 24 49 57 6 51 31 64 22 50 4 59 8 30 14 37 2 25 27 20 13 47 52 35 40 32 23 55 1]

Tabela 2. Resumo dos cromossomos vencedores

4. Conclusão

A estratégia de utilizar um algoritmo bio-inspirado para encontrar soluções em problemas complexos apresentou resultados bem satisfatórios. A modelagem do problema de

maneira a incluir restrições já na codificação do indivíduo é um dos grandes motivos de sucesso. Assim se torna possível colapsar uma dimensão do problema e reduzir significativamente o espaço de busca. A escolha do método de seleção por torneio se apresentou eficiente conforme resultados demonstram. A parametrização mesmo que pouco refinada apresentou excelentes comportamentos de evolução. A rotina de *generation gap* foi um fator de boa melhora para instâncias maiores que 8.

Para trabalhos futuros, recomenda-se a adoção de avaliação online da diversidade da população e assim interferir em parâmetros de mutação e seleção de maneira mais assertiva.