

Algoritmo genético aplicado a uma variação do problema das N rainhas

1st Christopher Renkavieski
Departamento de Ciência da Computação
Universidade do Estado de Santa Catarina
Joinville, Brasil
chris.renka@gmail.com

Resumo—Este trabalho apresenta a aplicação de algoritmos genéticos a uma variação do problema das N rainhas, em que um fator lucro é adicionado às suas soluções. O problema é apresentado em detalhes, juntamente com a codificação utilizada para os indivíduos do algoritmo, e do cálculo utilizado para o fitness. São apresentados todos os testes realizados, com as rotinas e parâmetros utilizados, além dos resultados obtidos e as suas análises.

Index Terms—Computação Evolutiva, Algoritmo Genético, N Rainhas

I. INTRODUÇÃO

Tendo como inspiração a teoria Neo-Darwinista, Algoritmos Genéticos são uma classe de algoritmos bio-inspirados pertencentes à área de Computação Evolutiva, e têm como objetivo “evoluir” uma solução ótima para um problema, a partir de uma população de possíveis soluções.

Este trabalho, desenvolvido para a disciplina de Computação Evolucionária, sob orientação do professor Rafael Parpinelli [1], apresenta a aplicação da ferramenta apresentada em [2] a uma variação do problema das N rainhas, que adiciona um fator lucro às configurações do tabuleiro.

A seção II apresenta em detalhes o problema a ser solucionado. A seção III explica a codificação utilizada para os indivíduos do Algoritmo Genético. Na seção IV, é mostrado como é feito o cálculo do fitness de um indivíduo. A seção V apresenta a ferramenta usada, em conjunto com os parâmetros utilizados para o problema. A seção VI mostra todos os testes realizados, em conjunto com seus resultados, e a seção VII apresenta dois testes adicionais realizados, avaliando o efeito da rotina de generation gap no algoritmo. Na seção VIII, estes resultados são analisados e, por fim, a seção IX apresenta as considerações finais do autor.

II. PROBLEMA

O problema básico das N rainhas consiste em dispor, em um tabuleiro de tamanho $N \times N$, N rainhas de modo que nenhuma delas possa atacar outra rainha, ou seja, nenhum par de rainhas pode estar na mesma linha horizontal, vertical ou diagonal no tabuleiro. No resto deste relatório, será utilizado o termo “colisão” para se referir a duas rainhas que possam se atacar, ou seja, estas rainhas se colidem.

A figura 1 apresenta um exemplo de solução para o problema das N rainhas, com $N = 8$.

	0	1	2	3	4	5	6	7
0				X				
1							X	
2			X					
3								X
4		X						
5					X			
6	X							
7						X		

Figura 1. Exemplo de solução para o problema das N rainhas, com $N = 8$.

Como este problema apresenta múltiplas soluções válidas, foi adicionado um fator extra a ele, de modo que cada solução apresente um “lucro” diferente, e deseja-se encontrar a solução válida (sem colisões) e de maior lucro.

Para se determinar esse lucro, primeiro é fornecido um valor a cada célula do tabuleiro. Este valor é calculado em duas etapas:

A primeira etapa é dar um valor inteiro, entre 1 e N^2 , para cada célula. No tabuleiro de 8×8 , as células da primeira linha recebem valores de 1 a 8, as da segunda linha, de 9 a 16, e assim por diante, até o valor 64, para a célula na última linha e última coluna. Para tabuleiros de outros tamanhos, o raciocínio é o mesmo.

A segunda etapa é aplicar uma transformação a esses valores. Começando a linha 1, as células nas linhas ímpares têm seu valor substituído pela sua raiz quadrada, e os valores nas linhas pares têm seu valor substituído pelo seu logaritmo de base 10. Assim, as células da primeira linha, no tabuleiro 8×8 , recebem a raiz quadrada de 1 a 8, as da segunda linha recebem o logaritmo dos valores de 9 a 16, e assim por diante, até a última linha.

Por fim, o lucro de uma solução é determinado como o somatório dos valores das células onde há uma rainha. Para a solução apresentada na figura 1, por exemplo, o lucro é:

$$\begin{aligned} \text{lucro} = & \sqrt{4} + \log_{10} 15 + \sqrt{19} + \log_{10} 32 + \sqrt{34} \\ & + \log_{10} 45 + \sqrt{49} + \log_{10} 62 \end{aligned}$$

Tabela I
COMPARAÇÃO ENTRE TAMANHO DO TABULEIRO E ESPAÇO DE BUSCA
PARA TABULEIROS DE LADO IGUAL A 8, 16, 32 E 64.

Tabuleiro	Espaço de busca
8x8	4,0320e4
16x16	2,0923e13
32x32	2,6313e35
64x64	1,2689e89

Desta forma, cada solução possível para o problema apresenta um lucro diferente, de modo que, mesmo entre as soluções válidas, existe uma ótima, com o maior lucro. O objetivo deste trabalho é encontrar esta solução ótima.

III. CODIFICAÇÃO

Para aplicar o algoritmo genético a este problema, faz-se com que cada indivíduo represente uma possível distribuição das N rainhas no tabuleiro.

Para isso, cada indivíduo é representado por um vetor de tamanho N, onde cada um de seus elementos é um valor entre 0 e N-1, sem repetições. Cada índice do vetor representa uma linha, e o valor do índice indica a coluna onde a rainha está localizada nesta linha.

Assim, por exemplo, a solução apresentada na figura 1 é representada pelo vetor [3, 6, 2, 7, 1, 4, 0, 5].

Esta codificação garante que nenhuma rainha será posicionada na mesma linha ou coluna, de modo que o espaço de busca é reduzido em relação a uma codificação que permita que as rainhas ocupem qualquer célula no tabuleiro.

Para um tabuleiro de NxN, o tamanho do espaço de busca, com esta codificação, é um arranjo de N em N, que é igual a N!. Para tabuleiros de lado igual a 8, 16, 32 e 64, o tamanho do espaço de busca é mostrado na tabela I.

Como o tamanho do espaço de busca cresce a uma taxa factorial em relação ao tamanho do tabuleiro, fica claro como este problema pode ser complexo para altas dimensionalidades.

IV. FITNESS

Para o cálculo do fitness de um indivíduo, são considerados dois fatores: a quantidade de colisões e o lucro, interpretando o lucro como um objetivo que deve ser maximizado, e as colisões como uma restrição, que geram uma penalidade. Para que a aplicação da penalidade seja possível, ambos os valores calculados são normalizados entre 0 e 1.

O lucro é normalizado em relação ao maior lucro possível para uma configuração de rainhas no tabuleiro, que é obtido posicionando todas as rainhas na penúltima linha. Assim, o valor do lucro normalizado é igual ao lucro calculado para um indivíduo, dividido pelo maior lucro possível.

Já para as colisões, primeiro deve-se estabelecer como as colisões são contadas. Considera-se que, para duas rainhas A e B, a colisão de A com B é a mesma que de B com A. Assim, cada par de rainhas que se colidem conta como uma única colisão. Deste modo, o número máximo de colisões possíveis, em um tabuleiro NxN, é uma combinação de N elementos em grupos de tamanho 2, ou seja, $C_{N,2} = \frac{N!}{2!(N-2)!}$. Deste modo,

a quantidade de colisões normalizada de um indivíduo é igual ao seu número de colisões dividido por $C_{N,2}$.

Por fim, o fitness total de um indivíduo é igual ao seu lucro normalizado, subtraído pela sua quantidade de colisões normalizada. Deste modo, indivíduos com alto lucro que têm colisões terão seu fitness penalizado.

V. FERRAMENTA E PARÂMETROS

Foi utilizada a ferramenta apresentada em [2], com a adição das rotinas de escalonamento linear e generation gap [1], e com o uso da medida de diversidade apresentada em [3], que é uma modificação da apresentada em [4].

Todos os testes com o algoritmo genético foram feitos usando uma população de 40 indivíduos, seleção por roleta, crossover tipo PMX, com probabilidade de 80%, mutação do tipo swap, com probabilidade de 3%, e com elitismo.

A seleção do tipo roleta foi escolhida devido ao uso do escalonamento linear, que não tem efeito na seleção por torneio. Para o escalonamento linear, o valor de c varia linearmente de 1.2 a 2.0 durante as primeiras 80% gerações, e se mantém como 2.0 durante as últimas 20%.

Foi utilizada também a rotina de generation gap nos testes. O valor de g é inicializado em 0.5, e a cada 1/6 das gerações ele é incrementado em 0.1, de modo que as últimas 1/6 gerações são feitos com g = 1.0. Deste modo, mantém-se uma diversidade maior nas primeiras gerações.

VI. TESTES E RESULTADOS

Foram realizados testes em tabuleiros de tamanho 8x8, 16x16 e 32x32. Para cada um dos tamanhos, foram realizados 10 testes, sendo que estes são feitos paralelamente. Todos os resultados apresentados ao fim são a média das 10 execuções.

Também foram realizados testes equivalentes utilizando busca aleatória, para comparação. Estes testes foram realizados com a mesma quantidade de avaliação de função dos testes com algoritmo genético, ou seja, com uma população de tamanho P e G gerações para o AG, na busca aleatória foram gerados P*G indivíduos aleatoriamente. Da mesma forma que nos testes com AG, foram realizadas 10 execuções da busca, e os resultados apresentados são a média destas execuções.

Para facilitar a comparação entre os métodos, os gráficos da busca aleatória foram gerados no mesmo formato dos gráficos do algoritmo genético. Assim, cada ponto no gráfico é a média de um número de indivíduos igual ao tamanho da população do AG.

As subseções seguintes apresentam os resultados obtidos para cada tamanho de tabuleiro.

A. 8x8

Os testes para o tabuleiro de 8x8 foram executados com um máximo de 500 gerações. Com 40 indivíduos na população, isso significa 20.000 aplicações de função, tanto para o algoritmo genético quanto para a busca aleatória.

1) *Algoritmo genético*: As 10 execuções do algoritmo genético levaram 5 segundos para ser completadas. O melhor indivíduo encontrado foi [6, 2, 7, 1, 4, 0, 5, 3], com fitness de 0,4629 e lucro de 26,8233. Este indivíduo não apresenta colisões, ou seja, é uma solução válida.

A figura 2 apresenta a evolução do fitness para o algoritmo genético aplicado ao tabuleiro de tamanho 8x8. Nela, são mostrados a evolução do melhor fitness, do fitness médio, e do fitness médio com a adição e subtração do desvio padrão da população em cada geração.

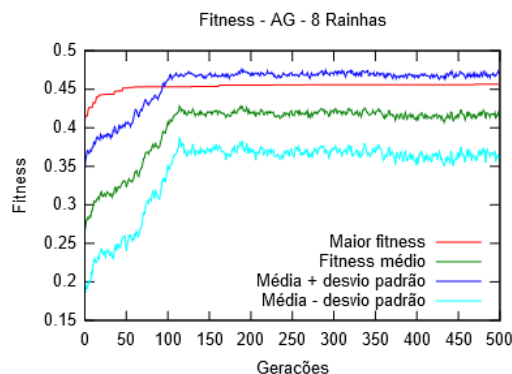


Figura 2. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para o algoritmo genético aplicado ao tabuleiro 8x8.

Na figura 2, é difícil observar a variação do desvio padrão em relação às gerações. Devido a isto, foi gerada a figura 3, que mostra a evolução do desvio padrão neste teste.

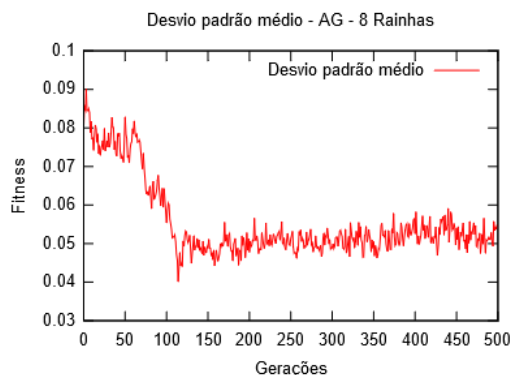


Figura 3. Evolução do desvio padrão da população para o algoritmo genético aplicado ao tabuleiro 8x8.

A figura 4 apresenta a evolução da diversidade genética da população ao longo das gerações.

2) *Busca aleatória*: As 10 execuções da busca aleatória levaram 3,4 segundos para ser completadas. O melhor indivíduo encontrado foi [6, 2, 7, 1, 4, 0, 5, 3], o mesmo encontrado pelo algoritmo genético. Seu fitness é de 0,4629, e seu lucro é 26,8233. Este indivíduo não apresenta colisões, ou seja, é uma solução válida.

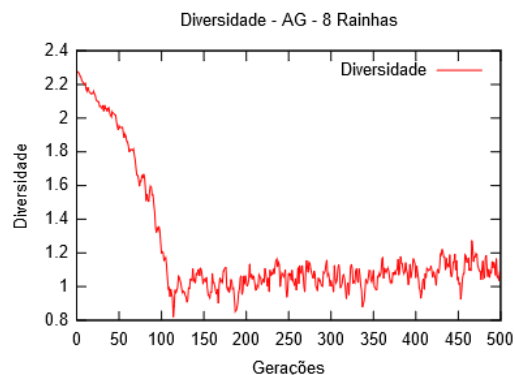


Figura 4. Evolução da diversidade genética da população para o algoritmo genético aplicado ao tabuleiro 8x8.

A figura 5 apresenta a evolução do fitness para a busca aleatória (Random Search) aplicada ao tabuleiro de tamanho 8x8. Por mais que o conceito de “geração” não exista na busca aleatória, ele foi usado nos gráficos para deixá-los no mesmo formato dos gráficos do algoritmo genético, como explicado no início desta seção.

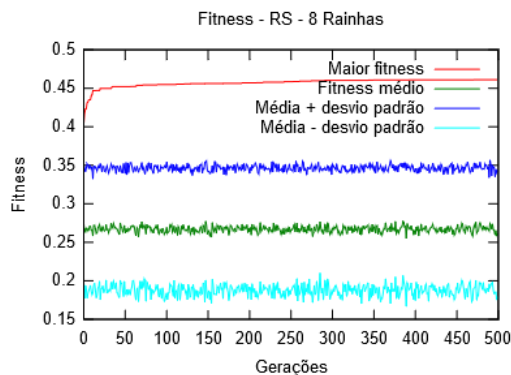


Figura 5. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para a busca aleatória aplicada ao tabuleiro 8x8.

Na figura 6, é mostrada a variação do desvio padrão ao longo das gerações, para a busca aleatória aplicada ao tabuleiro 8x8.

B. 16x16

Os testes para o tabuleiro de 16x16 foram executados com um máximo de 2.000 gerações. Com 40 indivíduos na população, isso significa 80.000 aplicações de função, tanto para o algoritmo genético quanto para a busca aleatória.

1) *Algoritmo genético*: As 10 execuções do algoritmo genético levaram 38,9 segundos para ser completadas. O melhor indivíduo encontrado foi [12, 9, 13, 2, 5, 3, 15, 7, 10, 4, 14, 0, 8, 6, 11, 1], com fitness de 0,4106 e lucro de 100,1592. Este indivíduo não apresenta colisões, ou seja, é uma solução válida.

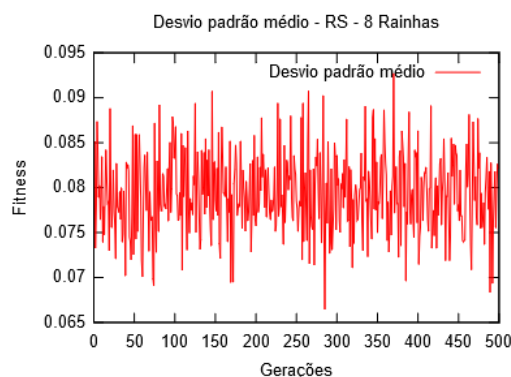


Figura 6. Evolução do desvio padrão da população para a busca aleatória aplicada ao tabuleiro 8x8.

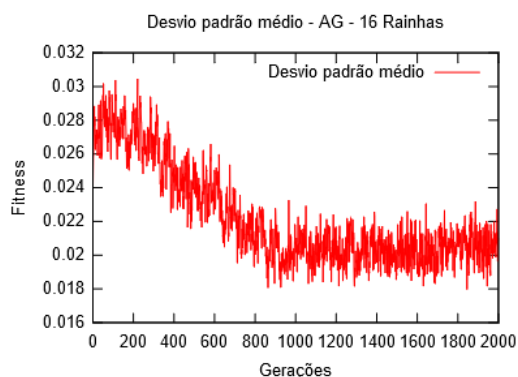


Figura 8. Evolução do desvio padrão da população para o algoritmo genético aplicado ao tabuleiro 16x16.

A figura 7 apresenta a evolução do fitness para o algoritmo genético aplicado ao tabuleiro de tamanho 16x16. Nela, são mostrados a evolução do melhor fitness, do fitness médio, e do fitness médio com a adição e subtração do desvio padrão da população em cada geração.

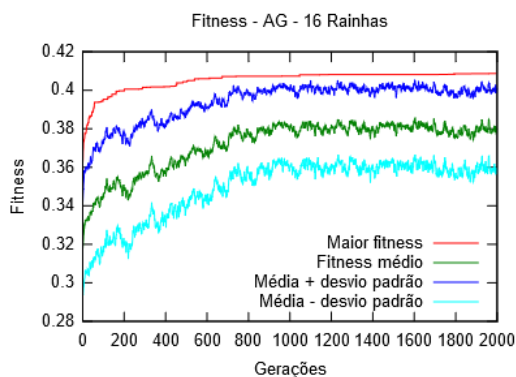


Figura 7. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para o algoritmo genético aplicado ao tabuleiro 16x16.

As figuras 8 e 9 apresentam, respectivamente, a evolução do desvio padrão e da diversidade genética ao longo das gerações para este teste.

2) *Busca aleatória*: As 10 execuções da busca aleatória levaram 23,5 segundos para ser completadas. O melhor indivíduo encontrado foi [13, 9, 7, 1, 4, 12, 5, 0, 11, 6, 14, 3, 8, 15, 2, 10]. Seu fitness é de 0,4062, e seu lucro é 99,0969. Este indivíduo não apresenta colisões, ou seja, é uma solução válida.

A figura 10 apresenta a evolução do fitness para a busca aleatória (Random Search) aplicada ao tabuleiro de tamanho 16x16.

Na figura 11, é mostrada a variação do desvio padrão ao longo das gerações, para a busca aleatória aplicada ao tabuleiro 16x16.

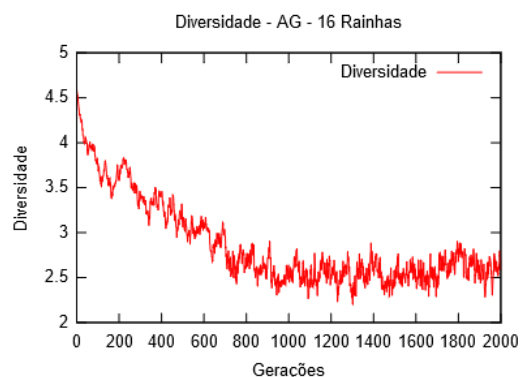


Figura 9. Evolução da diversidade genética da população para o algoritmo genético aplicado ao tabuleiro 16x16.

C. 32x32

Os testes para o tabuleiro de 16x16 foram executados com um máximo de 5.000 gerações. Com 40 indivíduos na população, isso significa 200.000 aplicações de função, tanto para o algoritmo genético quanto para a busca aleatória.

1) *Algoritmo genético*: As 10 execuções do algoritmo genético levaram 3 minutos e 53 segundos para ser completadas. O melhor indivíduo encontrado foi [26, 9, 19, 5, 24, 11, 28, 6, 23, 16, 8, 3, 30, 17, 27, 12, 21, 2, 29, 10, 15, 1, 31, 0, 25, 18, 22, 13, 4, 7, 20, 14], com fitness de 0,3796 e lucro de 379,5345. Este indivíduo não apresenta colisões, ou seja, é uma solução válida.

A figura 12 apresenta a evolução do fitness para o algoritmo genético aplicado ao tabuleiro de tamanho 32x32. Nela, são mostrados a evolução do melhor fitness, do fitness médio, e do fitness médio com a adição e subtração do desvio padrão da população em cada geração.

As figuras 13 e 14 apresentam, respectivamente, a evolução do desvio padrão e da diversidade genética ao longo das gerações para este teste.

2) *Busca aleatória*: As 10 execuções da busca aleatória levaram 2 minutos e 55 segundos para ser completadas. O

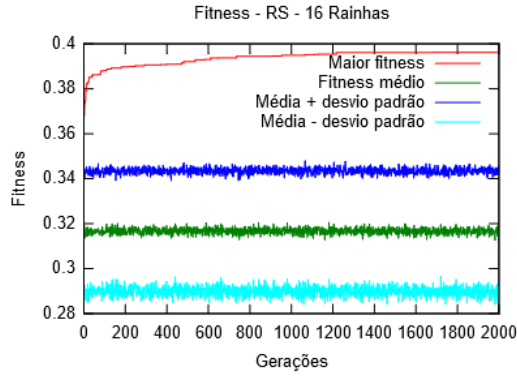


Figura 10. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para a busca aleatória aplicada ao tabuleiro 16x16.

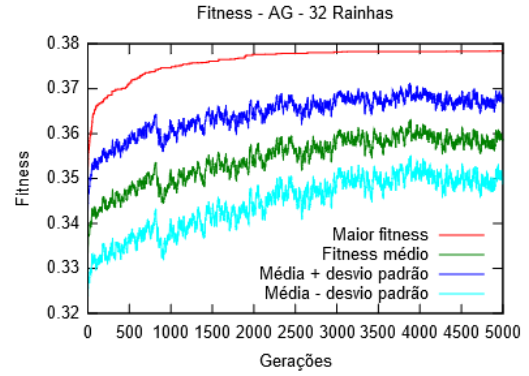


Figura 12. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para o algoritmo genético aplicado ao tabuleiro 32x32.

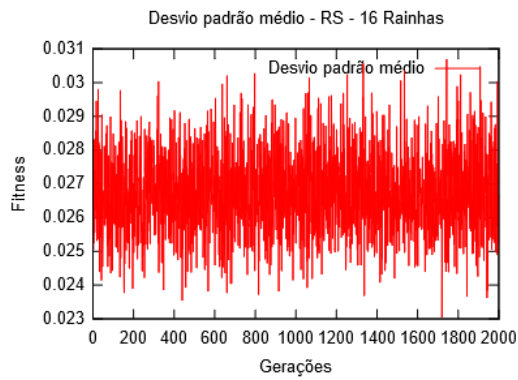


Figura 11. Evolução do desvio padrão da população para a busca aleatória aplicada ao tabuleiro 16x16.

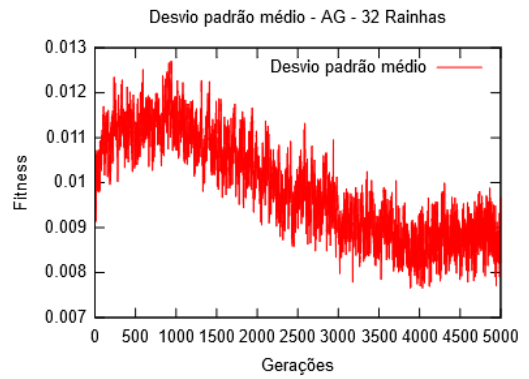


Figura 13. Evolução do desvio padrão da população para o algoritmo genético aplicado ao tabuleiro 32x32.

melhor indivíduo encontrado foi [13, 3, 12, 0, 8, 5, 14, 1, 18, 29, 22, 20, 23, 30, 26, 7, 11, 6, 19, 25, 27, 4, 2, 16, 9, 15, 31, 28, 24, 17, 21, 10]. Seu fitness é de 0,3680, e seu lucro é 376,0369. Este indivíduo apresenta 4 colisões, portanto não é uma solução válida ao problema.

A figura 15 apresenta a evolução do fitness para a busca aleatória (Random Search) aplicada ao tabuleiro de tamanho 32x32.

Na figura 16, é mostrada a variação do desvio padrão ao longo das gerações, para a busca aleatória aplicada ao tabuleiro 32x32.

VII. TESTES ADICIONAIS

Foram realizados dois testes com valores fixos para a variável g , com o objetivo de avaliar o efeito da rotina de generation gap na evolução.

Ambos os testes foram feitos em redes de tamanho 32x32, e foram utilizados os mesmos parâmetros dos testes anteriores. O primeiro deles foi feito com $g = 1.0$, e o segundo com $g = 0.5$.

A. $g = 1.0$

Com $g = 1.0$, 100% dos indivíduos são substituídos a cada geração, o que é equivalente a retirar a rotina de generation gap.

As 10 execuções deste teste levaram 4 minutos e 13 segundos para ser completadas. O melhor indivíduo encontrado foi [26, 5, 30, 11, 24, 14, 29, 17, 15, 7, 28, 4, 13, 8, 20, 2, 27, 3, 12, 31, 25, 1, 22, 6, 16, 21, 23, 9, 19, 10, 18, 0], com fitness de 0,37976 e lucro de 379,7461. Este indivíduo não apresenta colisões, ou seja, é uma solução válida.

A figura 17 apresenta a evolução do fitness para o algoritmo genético aplicado ao tabuleiro de tamanho 32x32 e $g = 1.0$. Nela, são mostrados a evolução do melhor fitness, do fitness médio, e do fitness médio com a adição e subtração do desvio padrão da população em cada geração.

As figuras 18 e 19 apresentam, respectivamente, a evolução do desvio padrão e da diversidade genética ao longo das gerações para este teste.

B. $g = 0.5$

Com $g = 0.5$, 50% dos indivíduos são substituídos a cada geração, e os outros 50% são mantidos.

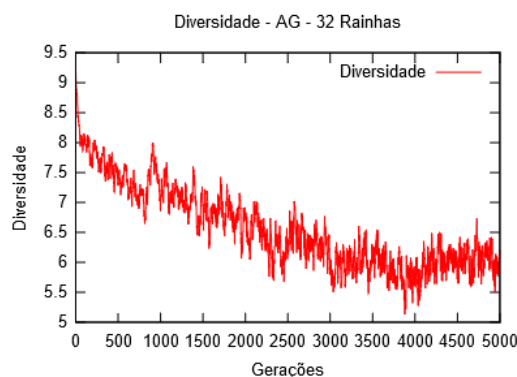


Figura 14. Evolução da diversidade genética da população para o algoritmo genético aplicado ao tabuleiro 32x32.

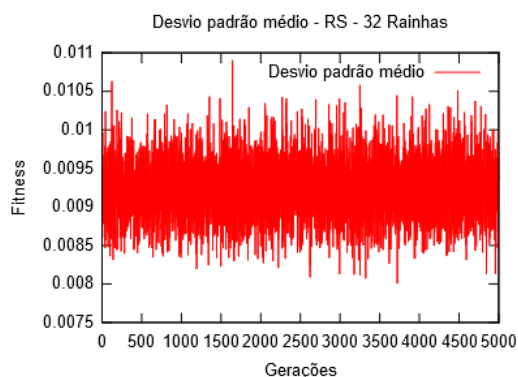


Figura 16. Evolução do desvio padrão da população para a busca aleatória aplicada ao tabuleiro 32x32.

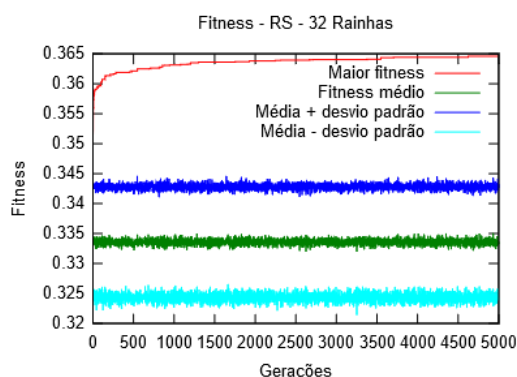


Figura 15. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para a busca aleatória aplicada ao tabuleiro 32x32.

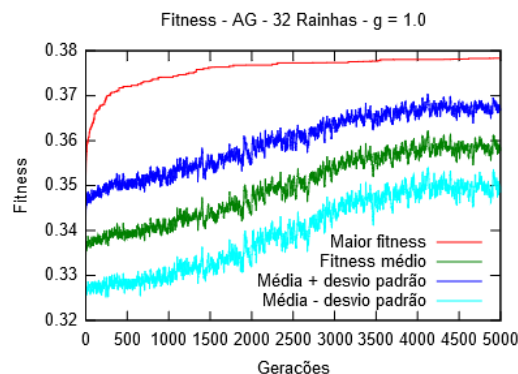


Figura 17. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para o algoritmo genético aplicado ao tabuleiro 32x32, com $g = 1.0$.

As 10 execuções deste teste levaram 3 minutos e 27 segundos para ser completadas. O melhor indivíduo encontrado foi [28, 6, 25, 7, 12, 3, 23, 2, 29, 10, 24, 14, 19, 4, 27, 0, 31, 9, 20, 16, 30, 1, 11, 22, 18, 15, 26, 5, 21, 17, 13, 8], com fitness de 0,37975 e lucro de 379,7367. Este indivíduo não apresenta colisões, ou seja, é uma solução válida.

A figura 20 apresenta a evolução do fitness para o algoritmo genético aplicado ao tabuleiro de tamanho 32x32 e $g = 0.5$. Nela, são mostrados a evolução do melhor fitness, do fitness médio, e do fitness médio com a adição e subtração do desvio padrão da população em cada geração.

As figuras 21 e 22 apresentam, respectivamente, a evolução do desvio padrão e da diversidade genética ao longo das gerações para este teste.

VIII. ANÁLISE

Todos os testes mostraram uma diferença significativa entre o Algoritmo Genético e a busca aleatória, com o primeiro apresentando resultados melhores e mais consistentes.

Esta diferença fica clara ao se comparar a variação do fitness médio e do desvio padrão entre os dois métodos: enquanto o algoritmo genético apresenta uma melhora gradual destas

variáveis a longo das gerações, na busca aleatória elas somente oscilam em torno de um mesmo valor, mostrando que não há progresso acontecendo.

Já os testes feitos com valores fixos de generation gap apresentaram resultados inesperados, com a evolução mantendo 50% dos indivíduos entre gerações apresentando convergência melhor que a evolução onde 100% dos indivíduos são substituídos. Uma possível hipótese para este comportamento é que, a cada geração, os indivíduos novos tendem a se sobressair em relação aos da geração anterior, acelerando a convergência. Estes resultados merecem uma melhor investigação.

IX. CONCLUSÃO

Este relatório apresentou a aplicação de um algoritmo genético a uma variação do problema das N rainhas, e comparou os seus resultados com a busca aleatória aplicada ao mesmo problema.

Estes resultados confirmam as conclusões obtidas no relatório anterior [2], confirmando o poder computacional dos algoritmos genéticos em um problema mais complexo que os apresentados anteriormente.

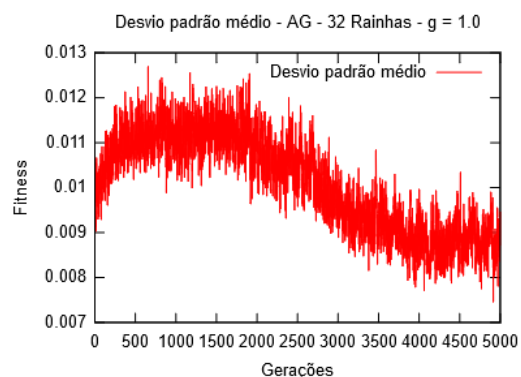


Figura 18. Evolução do desvio padrão da população para o algoritmo genético aplicado ao tabuleiro 32x32, com $g = 1.0$.

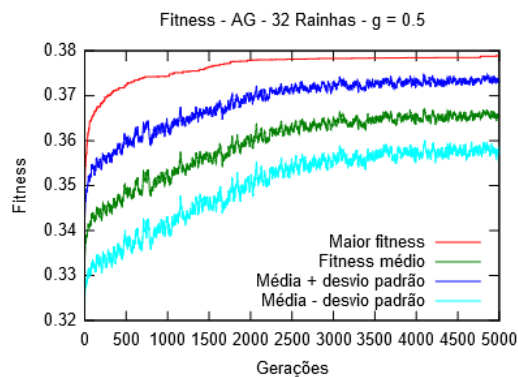


Figura 20. Evolução do maior fitness, fitness médio, e fitness médio somado e subtraído pelo desvio padrão médio para o algoritmo genético aplicado ao tabuleiro 32x32, com $g = 0.5$.

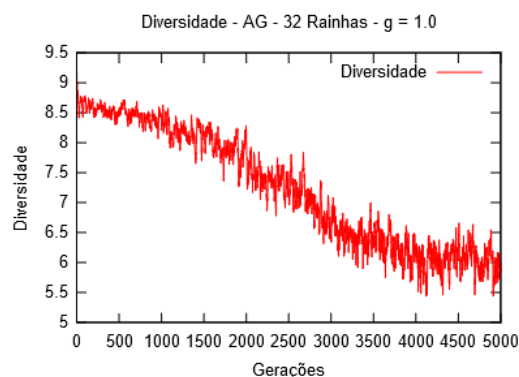


Figura 19. Evolução da diversidade genética da população para o algoritmo genético aplicado ao tabuleiro 32x32, com $g = 1.0$.

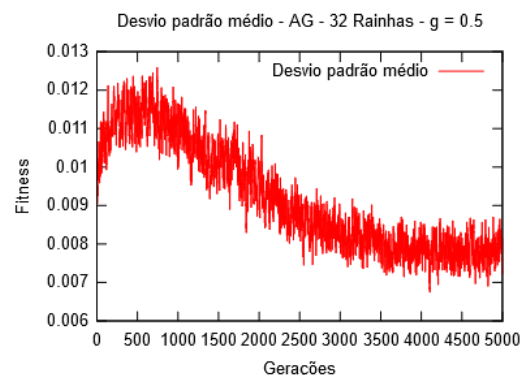


Figura 21. Evolução do desvio padrão da população para o algoritmo genético aplicado ao tabuleiro 32x32, com $g = 0.5$.

REFERÊNCIAS

- [1] Parpinelli. Rafael: "Computação Evolucionária: fundamentos teóricos de AG, DCC, UDESC-Joinville. Disponível em: http://www.joinville.udesc.br/portal/professores/parpinelli/materiais/CE_0403_2018.pdf
- [2] Renkavieski, Christopher. "Desenvolvimento de um otimizador com base em algoritmos genéticos". Disponível em: <https://github.com/ChrisRenka/OCEV/>
- [3] Silva, Renan S. "Medida de diversidade do Renan". Disponível em : http://www.joinville.udesc.br/portal/professores/parpinelli/materiais/medida_de_diversidade__2_.pdf
- [4] Morrison R.W., De Jong K.A. (2002) Measurement of Population Diversity. In: Collet P., Fonlupt C., Hao JK., Lutton E., Schoenauer M. (eds) Artificial Evolution. EA 2001. Lecture Notes in Computer Science, vol 2310. Springer, Berlin, Heidelberg

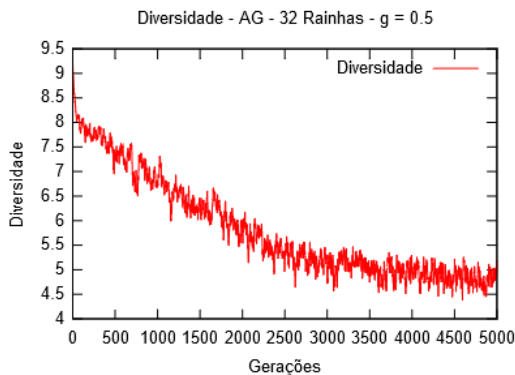


Figura 22. Evolução da diversidade genética da população para o algoritmo genético aplicado ao tabuleiro 32x32, com $g = 0.5$.