

# Algoritmo Genético como Mecanismo de Aprendizagem do Agente na Resolução do Mundo de Wumpus

Natália Freitas Araújo<sup>1</sup>, Adriano A. Addario dos Santos<sup>2</sup>, Otávio Noura Teixeira<sup>1</sup>

<sup>1</sup> Universidade Federal do Pará, Tucuruí PA 68455-747, BRA

<sup>2</sup> Faculdade Uninorte, Tucuruí PA 68455-134, BRA

{taiaraujo20, adriano.addario, onoura}@gmail.com

**Resumo.** O presente artigo faz uma abordagem sobre conceitos relacionados à Inteligência Artificial e sua aplicação para o problema do mundo de Wumpus com a utilização da técnica de Algoritmo Genético (AG). O objetivo é aplicar a técnica de AG como Mecanismo de Aprendizagem de um Agente Inteligente. O modelo de ambiente utilizado foi devidamente formalizado, de acordo com as recomendações feitas por Stuart Russel e Peter Norvig, além do ambiente, foi realizada uma categorização do Algoritmo Genético desenvolvido conforme a teoria e definição de agentes inteligentes. Também foi desenvolvida uma tabela de pontuação e uma equação *fitness* para o processo de avaliação de cada indivíduo gerado. Além do referencial teórico, apresenta-se todo o processo de execução, os métodos utilizados e os resultados obtidos. O projeto conta com um robusto conjunto de dados resultantes, com a realização de 4.200 execuções obteve-se o total de 12.600 arquivos, os quais contêm informações como – pontuação, tempo de execução, melhor cromossomo de cada uma das gerações executadas. Após as discussões dos resultados apresenta-se que os indivíduos gerados tiveram um melhor desempenho em ambientes com menores escalas, destaca-se que o algoritmo teve 97,7% de vitórias no ambiente de dimensão 5x5, seguido de uma drástica queda, como 10,5% de vitórias para o ambiente de dimensão 10x10 e a ineficiência aos demais ambientes testados.

**Palavras-chave:** Algoritmo Genético, Agentes Inteligentes, Mundo de Wumpus, Ambiente.

## 1 Introdução

A Inteligência Artificial trata-se de uma área com diversas vertentes, entretanto, com um objetivo central a longo prazo – o estudo e a criação de inteligência em sistemas computacionais [9]. Segundo Russell, Dewey e Tegmark [10] as pesquisas e desenvolvimento na área, nos últimos 20 anos, têm-se concentrado na construção de agentes, pois esta vertente está diretamente ligada ao estudo de desenvolvimento de uma máquina que tenha a capacidade de tomar boas decisões, planos ou inferências.

Por definição, agente é toda a entidade capaz de perceber e atuar no ambiente em que está inserido [4], [6], [7], [8] e [11]. Assim como, agentes inteligentes são sistemas

que englobam a mesma característica de um agente, com o acréscimo de um fator primordial, a busca por racionalidade [6] e [11].

Outra técnica que, por meio de sua busca pela melhor solução, simula um comportamento inteligente é o Algoritmo Genético, baseado na teoria darwiniana, destaca-se pelo seu alto desempenho em resoluções de problemas de otimização [12], tendo como objetivo maximizar o retorno das soluções aspirantes versus uma função custo do domínio do problema [5].

Por sua característica de adaptação e evolução, os AGs são aplicados, principalmente, em problemas que não possuem soluções por meio de métodos algébricos ou estatísticos, por exemplo, problemas de busca. Um fato importante a ser ressaltado é que o algoritmo busca a melhor solução possível, porém essa solução nem sempre será a melhor solução do problema [14].

Dentro do contexto apresentado, a definição do ambiente, assim como a compreensão de suas características são de extrema importância para a construção de um agente inteligente [6], pois estas características afetam diretamente na escolha, avaliação e evolução do agente [1]. De acordo com a definição de Weyns, Omicini e Odell [15] ambiente é uma abstração primária que fornece as condições necessárias para a existência de um agente, intermediando a interação entre agente e o acesso aos recursos. Neste trabalho, o Algoritmo Genético desenvolvido será o agente responsável por buscar a melhor solução possível para o problema do Mundo de Wumpus.

## 2 O Mundo de Wumpus

Em 1972 Gregory Yob [16] desenvolveu um jogo de computador para *People's Computing Company* (PCC) com o intuito de quebrar o paradigma da construção de jogos da época. O seu projeto foi tão bem-sucedido que o jogo, originalmente nomeado de *Hunt the Wumpus*, ganhou diversas adaptações, sendo inserido no cenário de IA como ambiente de testes [13].

O modelo de referência para o desenvolvimento deste trabalho foi apresentado por Russell e Norvig [11], o qual consiste em uma matriz bidimensional, segmentada em salas contendo um número determinado de objetos (Fossos, Wumpus, Ouro e Agente) [13].

Descrevendo de maneira formal o ambiente desenvolvido, em conformidade com as Propriedades de Ambientes de Tarefas [11], apresenta-se um modelo: *parcialmente observável, agente único, determinístico, sequencial, dinâmico, discreto e desconhecido*. E de acordo com o quadro de especificidades PEAS (*Performance, Environment, Actuator, Sensors* – desempenho, ambiente, atuadores e sensores) [11] têm-se os dados presentes na Tabela 1.

Tabela 1. Quadro de especificidades PEAS

Tipo de Agente	Medida de Desempenho	Ambiente	Atuadores	Sensores
Algoritmo Genético	Pontuação (fitness)	Mundo de Wumpus (descrito na seção 2)	Ações do agente (Andar, Atirar e Pegar)	Sentidos do Agente (Visão – vê brilho do ouro e Audição – Escuta grito do Wumpus)

Para a realização das simulações foi desenvolvido um gerador de ambiente aleatório. A variável de entrada do gerador é o tamanho do ambiente, de acordo com o tamanho os objetos são dispostos. Obrigatoriamente, terá um ouro e um Wumpus para qualquer cenário e o número de fossos será proporcional ao tamanho de ambiente requerido, considerando que em um ambiente de 16 salas (4x4 – a menor escala do modelo) têm-se 3 fossos. A saída do gerador é uma matriz bidimensional quadrada.

De acordo com Russell e Norvig [11], sem uma formulação de regras para a criação mundo de Wumpus cerca de 21% dos ambientes gerados não seriam solucionáveis, portanto, a geração do ambiente obedece às seguintes regras de criação:

- A sala inicial (0,0) não terá nenhum objeto
- O ouro não estará na mesma sala de um Fosso ou Wumpus
- Os Fossos são distribuídos em salas distintas
- O Wumpus não estará na mesma sala de um Fosso

Após a geração, o ambiente é salvo de duas maneiras, em forma gráfica (formato *png*) e em formato *csv* contendo os dados do ambiente. Na Fig. 1 apresentam-se os 7 ambientes utilizados para os testes de simulações deste artigo, nomeados de *A a G*, com dimensões de 4x4, 5x5, 10x10, 25x25, 50x50, 100x100 e 200x200, respectivamente.

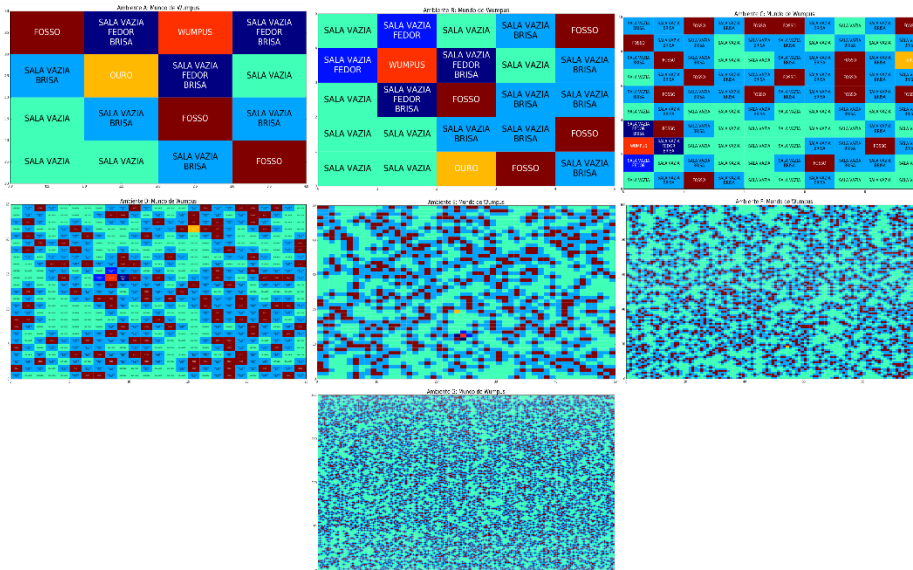


Fig. 1. Ambientes do mundo de Wumpus gerados e utilizados para as simulações.

Apresenta-se uma escala de cores nos gráficos da Fig. 1, cada cor representa um objeto na matriz, são eles: verdes – salas vazias, marrons – salas com fossos, laranja – sala com Wumpus, amarela – sala com ouro. As cores azul-claros representam sala adjacente ao fosso e as azul-escuros sala adjacente ao Wumpus.

### 3 Estrutura do Algoritmo Genético

O algoritmo genético, ambientado no mundo de Wumpus, busca o agente com a maior pontuação (*fitness*) obtida durante o jogo, essa pontuação representa as consequências do conjunto de ações realizadas pelo agente. Atribuir uma nota a cada ação de um agente ditará se este teve ou não um bom desempenho na busca do seu objetivo. O resultado dessa pontuação será o guia para a evolução das populações ao longo de suas gerações [5].

O AG desenvolvido pode ser categorizado como um agente racional, pois age para maximizar o valor esperado da medida de desempenho, e ao analisar suas características, é possível enquadrar a sua estrutura de programa como um Agente Baseado em Objetivo [11], sendo o objetivo do agente – entrar na caverna, pegar o ouro e sair no menor caminho possível. Mas, para tanto, é necessário evitar os Fossos e o Wumpus, o agente ainda possui uma oportunidade de matar o Wumpus, para a sua defesa, caso o agente consiga terá uma bonificação em seu *fitness*.

Russel e Norvig [11] propõem uma tabela de pontuação para o desenvolvimento do mundo de Wumpus, porém nesta tabela apresentam-se alguns valores negativos para as punições do agente, o que torna-se um problema quando adaptado para técnica de AG, pois, durante testes, quando o *fitness* apresentou valores negativos a evolução das

gerações apresentou uma curva decrescente e um decaimento na diversidade da população, não gerando nenhuma proposta de solução. Com intuito de sanar este problema foi desenvolvido, por meio de experimentações, uma tabela de pontuação a Tabela 2 e uma equação de fitness a Equação 1.

Para o desenvolvimento da Tabela 2 foi realizado um mapeamento das ações possíveis do agente realizar e suas implicações, após análise, os dados foram segmentados em três fatores: o custo, o acerto e o erro. O *custo* recebe a pontuação atribuída de acordo com a ação tomada pelo agente, caso a implicação da ação tomada seja positiva, o fator *acerto* recebe uma nota equivalente a consequência da ação, caso contrário, a nota é atribuída ao fator *erro*. As notas empregadas para os fatores variam de zero até dez, onde valores muito próximos de zero representam medidas de punição e os valores mais próximos de dez representam as bonificações.

**Tabela 2.** Tabela de pontuação de custo, bonificação e punição gerada por cada ação do agente<sup>1</sup>.

AÇÃO		IMPLICAÇÃO			
Custo		Acerto		Erro	
Andar	0.25	AV	1.5	ES	0.2
		AO	5	EF	1
		AF	10	ED	0.01
Atirar	0.2	AW	10	EW	0.1
Pegar	0.3	AC	10	EC	0.1
				EC2	0.05

A Equação 1 *fitness* é a normalização do conjunto de somatórios dos fatores apresentados na Tabela 2 ponderados por coeficientes, onde estes coeficientes assumem os valores de  $\alpha=2$ ,  $\beta=0,5$  e  $\gamma=0,25$ . Esses valores foram atribuídos com o intuito de maximizar os acertos e minimizar os erros e os custos, sendo que  $\alpha$  e  $\beta$  são valores inversamente proporcionais e  $\gamma$  tende a ser um valor mínimo exercendo pouca influência sobre os resultados.

$$fitness = \frac{1}{N} \left\{ \alpha \sum_{i \in A} acerto_i + \beta \sum_{j \in E} erro_j + \gamma \sum_{k \in N} custo_k \right\} \quad (1)$$

<sup>1</sup> Na Tabela 2 foram utilizadas algumas siglas para auxiliar em sua construção e visualização, onde seus significados são – AV: acerto ao andar em casa vazia, enquanto explora-se o ambiente. AO: acerto ao andar e encontrar ouro. AF: acerto ao andar e chegar na casa (0,0) tendo em posse o ouro. AW: acerto ao atirar e acertar o Wumpus. AC: acerto ao pegar o ouro. ES: erro ao andar e tentar sair do ambiente. EF: erro ao andar e entrar na sala do Wumpus ou de um Fosso. ED: erro ao andar depois de cumprir o objetivo do jogo. EW: erro ao atirar e não acertar o Wumpus. EC: erro ao tentar pegar em sala onde não está o ouro. EC2: erro ao tentar pegar quando já possui o ouro.

O método de seleção proporcional utilizado foi o Torneio (*Tournament*) [3]. E a implementação do elitismo deu-se da seguinte maneira: após as etapas de cruzamento e mutação os filhos gerados são avaliados e inseridos na população, por seguinte, a população é *ranqueada* e os indivíduos de menor *fitness* são excluídos. Esse método causa um “efeito sanfona” no tamanho da população, no entanto, as gerações mantêm um número fixo de indivíduos, desde a população inicial até a última população gerada, sempre mantendo os melhores indivíduos encontrados [2].

## 4 Metodologia

Para a realização dos testes foram utilizados 84 conjuntos de configurações distintos de simulações com 50 execuções para cada configuração, resultando em 4200 execuções. Nas quais foram utilizados os 7 ambientes descritos na seção 2 (ver Fig. 1) e 5 populações iniciais geradas aleatoriamente, nomeadas de *P1 A P5* que possuem, respectivamente, o número de 50, 100, 200, 500 e 1000 indivíduos.

Outro parâmetro variado foi o número de gerações utilizadas na evolução: 1000, 2000 e 5000 gerações. Entre os ambientes *A e D* têm-se 15 configurações de simulação, e a partir do ambiente *E* há uma baixa no número de configurações, essa escolha deu-se pelo alto custo computacional apresentado combinado pelos resultados obtidos desde o ambiente *D*.

A taxa de cruzamento (*crossover*) utilizada foi de 90% e a de mutação 5% permanecendo as mesmas para todas as execuções. Assim como, a tabela de pagamento (ver Tabela 2) e a equação *fitness* (ver Equação 1). Todas as disposições de configuração foram registradas em uma tabela própria que pode ser acessada através do link<sup>2</sup>.

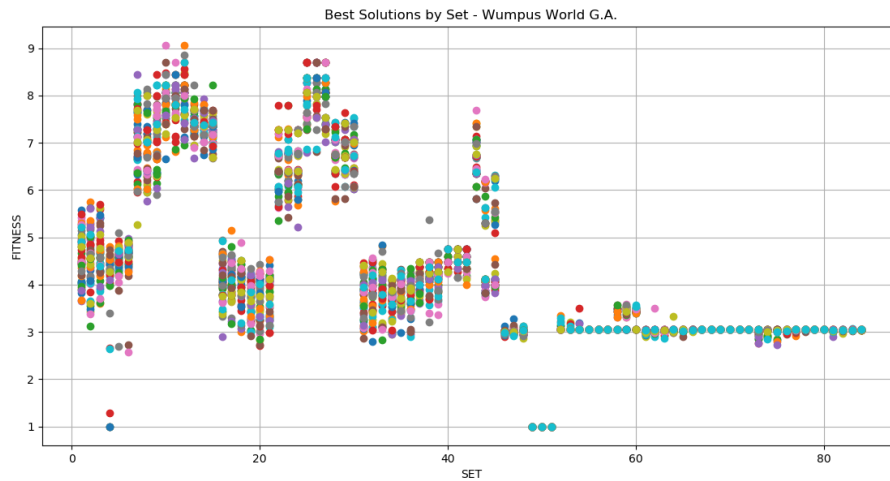
A partir das execuções descritas acima, foram salvos três arquivos para cada execução. O primeiro arquivo (formato *png*) é um gráfico com as melhores, piores e a média das notas de todas as gerações. O segundo arquivo (formato *csv*) apresenta se foi ou não gerado um indivíduo que pegou o ouro e chegou na sala inicial (0,0), as melhores, as piores e as médias das notas de cada geração. E o terceiro arquivo (formato *csv*) apresenta os dados do melhor indivíduo gerado: o seu tempo de execução, a geração que foi criado, sua nota, se pegou ou não o ouro, se matou ou não o Wumpus, se tentou ou não matar o Wumpus, se pegou o ouro e chegou na sala inicial, e o seu cromossomo. Todos esses dados estão disponíveis no link<sup>3</sup>.

## 5 Resultados

Obteve-se um expressivo conjunto de dados que será discutido por seguinte, iniciando com a Fig. 2 que apresenta a nota do melhor indivíduo de cada uma das 4200 execuções, organizadas por configuração.

<sup>2</sup> Acesse a tabela de configuração das simulações em: <https://1drv.ms/x/s!AiigAtsZamvmibB-pxzd89Ix9iZz5wQ?e=UHhG9D>

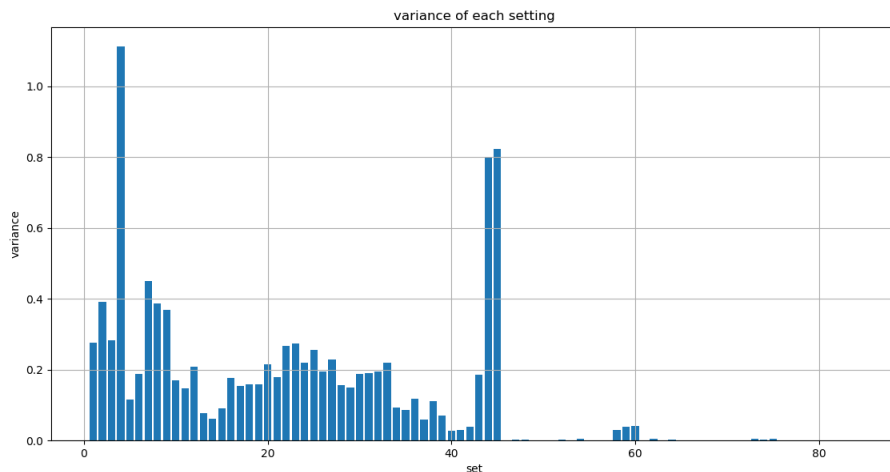
<sup>3</sup> Acesse os arquivos descritos em: <https://1drv.ms/u/s!AiigAtsZamvmiNME1kn2d835Vwx37w?e=Pe3rhw>



**Fig. 2.** A melhor nota de cada execução das 84 configurações de simulações realizadas.

Percebe-se que os pontos apresentam um certo padrão em sua disposição, ressaltando uma maior consistência, para os primeiros trinta conjuntos mantendo um modelo de desempenho, o qual demonstra que as populações iniciais, em dois ambientes distintos, tendem a um mesmo padrão de comportamento, esse intervalo do gráfico apresenta os *fitness* para os ambientes *A* e *B* os quais possuem as maiores notas dentre todas as execuções. Por outro lado, as configurações seguintes possuem comportamentos distintos, e notas plotadas no gráfico se concentram muitas próximas umas das outras e em alguns conjuntos no mesmo ponto.

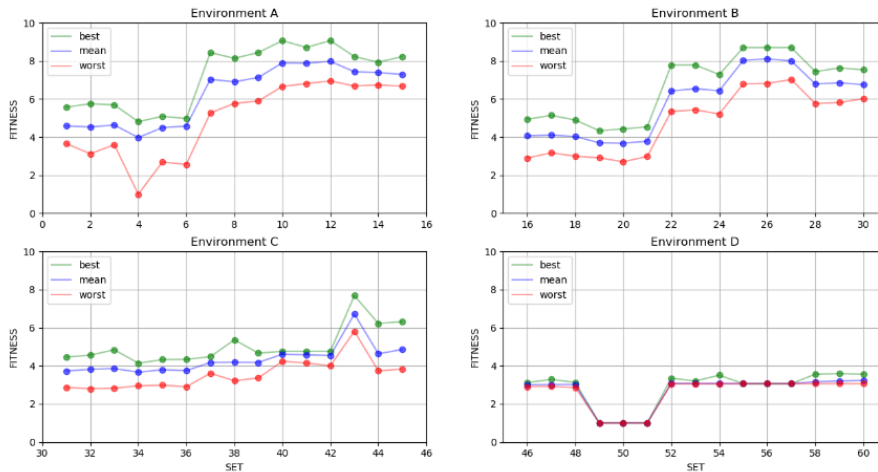
Em busca de outra perspectiva dos mesmos dados, foi calculado e plotado na Fig. 3 a variância de cada conjunto de simulação



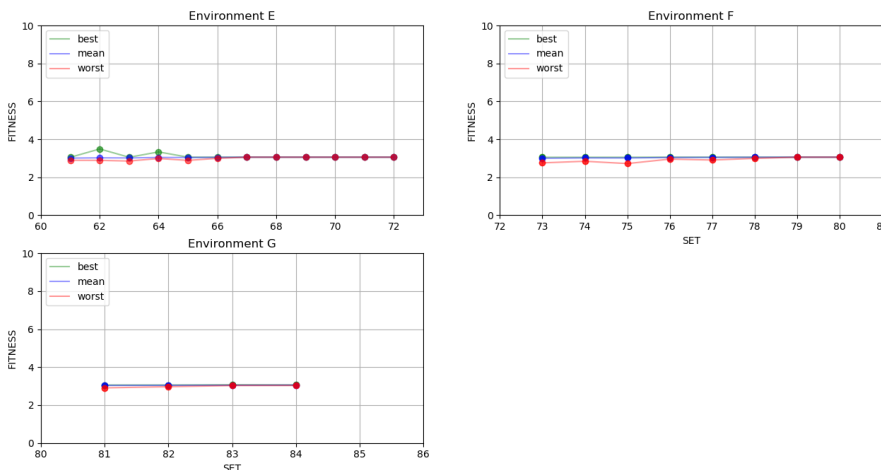
**Fig. 3.** Variância de cada conjunto de simulação

Nota-se que as primeiras configurações têm-se grandes variâncias no gráfico, destacando-se o conjunto 4, e essa variância tem um decaimento ao logo das configurações, chegando ao estado da variância se igualar a zero em algumas configurações. Essa falta de variância demonstra uma estagnação no algoritmo genético, ou seja, apesar das gerações, os parâmetros fornecidos não são suficientes para diversificar a população, desta forma não fornecendo diferentes candidatos para melhor indivíduo. Entretanto, apenas o valor da variância, não garante que foi encontrada a resolução do problema, ela pode ser usada como um dos parâmetros de análise.

Nas Fig. 4 e Fig. 5 têm-se gráficos que apresentam as melhores, piores e as médias das notas das execuções separadas por ambientes, no título de cada gráfico tem a nomenclatura do ambiente ao qual se refere.



**Fig. 4.** Melhores, piores e as médias das notas de cada configuração dos ambientes *A*, *B*, *C* e *D*.



**Fig. 5.** Melhores, piores e as médias das notas de cada configuração dos ambientes *E*, *F* e *G*.



Por conta da normalização apresentada na Equação 1 e das pontuações atribuídas na Tabela 2, os *fitness* resultantes variam entre 0 a 10. Neste cenário é possível determinar uma faixa de conceitos de desempenho para os resultados, tendo desta forma o conceito (I) *Insatisfatório* para notas abaixo de 5, (R) *Regular* para notas entre 5 e 6.9, (B) *Bom* para notas entre 7 e 8.9 e (E) *Excelente* para notas acima de 8.9. A Tabela 3 apresenta a confiabilidade do algoritmo em duas camadas: a primeira com os percentuais de vitórias e a segunda com os percentuais de desempenhos dos agentes, ambas segmentadas por ambientes.

**Tabela 3.** Descrição de vitórias e o desempenho do AG organizados por ambiente.

FAIXA DE TESTES		VITÓRIAS		DESEMPENHO (%)			
Ambiente	Con-junto	Número	%	I	R	B	E
A	1 a 15	608	81%	35,1%	16,3%	48,4%	0,2 %
B	16 a 30	733	97,7%	39,9%	28,8%	31,3%	0%
C	31 a 45	79	10,5%	87,7%	10,1%	2,1%	0%
D	46 a 60	0	0%	100%	0%	0%	0%
E	61 a 72	0	0%	100%	0%	0%	0%
F	73 a 80	0	0%	100%	0%	0%	0%
G	81 a 84	0	0%	100%	0%	0%	0%

Tendo essa faixa de conceitos estabelecida, nota-se que o ambiente A é o único que possui agentes com conceito *Excelente*, apesar de ser uma porcentagem muito pequena. Os ambientes B e C possuem *fitness* até o conceito *Bom*, o que significa que os agentes gerados conseguiram alcançar o objetivo e obter uma boa pontuação. Enquanto os ambientes D, E, F e G possuem, todos, conceitos *Insatisfatórios*, ou seja, os agentes não atingem o objetivo para esses ambientes. Percebe-se também, que a porcentagem de acertos inicia bastante elevada e com as mudanças de ambientes essa porcentagem sofre uma drástica queda, chegando a não marcar um percentual de vitórias.

## 6 Conclusões

A técnica de Algoritmo Genético tem-se mostrado eficiente em diferentes problemas e cenários. Aplicado ao problema do Mundo de Wumpus obteve-se resultados animadores para ambientes com pequenas dimensões, na primeira camada de confiabilidade obteve um percentual de até 97,7% de vitórias, o que significa que a equação desenvolvida, juntamente, com a tabela de pontuação apresenta valores coerentes para os primeiros ambientes testados. Por outro lado, a tabela de pontuação e a equação mostraram-se ineficientes para os demais ambientes – maiores dimensões –, com uma taxa de 10,5% de vitórias para o ambiente C, e nenhuma vitória para os demais (D a G).

Ao analisar a segunda camada de confiabilidade têm-se, para os ambientes que os agentes encontraram um solução, um alto percentual de agentes com desempenho *Insatisfatório* e um percentual nulo de agentes classificados com desempenho *Excelente*, uma vertente para a continuação deste trabalho é o estudo para maximizar a camada de desempenho do agente com objetivo de diminuir os indivíduos com desempenho *Insatisfatório* e aumentar indivíduos com desempenhos *Bom e Excelente*.

É importante ressaltar para essa análise que todos os elementos envolvidos são determinantes para a obtenção do resultado. Ou seja, caso haja alguma mudança em um dos parâmetros – o ambiente de entrada, a população inicial, a taxa de cruzamento, a taxa de mutação, a tabela de pontuação e/ou a equação de fitness – os resultados também sofrerão mudanças, como realmente ocorre. Outro ponto a ser ressaltado é que as populações, apesar de apresentarem os mesmos indivíduos, vão se comportar de maneiras distintas a cada execução.

Os ambientes *D, E, F e G* são mais extensos e possuem uma quantidade maior de fossos, essas características aumentam o nível de dificuldade de resolução. Outra vertente para trabalhos futuros é a análise mais profunda do comportamento que os agentes possuem nestes cenários, a verificação e adequação dos parâmetros da tabela de pontuação e da equação.

Existem dois fatos interessantes na classificação de desempenho aplicada neste trabalho. Alguns agentes que têm o seu desempenho classificado como *Insatisfatório* conseguem pegar o ouro e chegar na sala inicial, porém, esses agentes possuem um cromossomo muito maior que o necessário e cometem diversos erros para a obtenção desse resultado, portanto, a classificação empregada é um ponto positivo tendo em vista que o objetivo do agente é também executar no menor caminho possível. O outro fato, é que foram detectados alguns agentes “trapaceiros” classificados com desempenho *Regular*, estes agentes não atingem o objetivo, contudo descobrem que determinadas ações, como andar na sala do ouro sem pegá-lo, gera um aumento na sua pontuação, contudo, esses *agentes trapaceiros* não conseguiram obter uma classificação de desempenho *Bom ou Excelente*.

Os testes realizados aqui não levam em consideração algumas sensações que são perceptíveis aos agentes, o mal odor do Wumpus e a brisa dos fossos, um próximo passo para a continuação do trabalho é a inserção dessas variáveis na tabela de pontuação e o estudo de como esses novos fatores afetaram na equação fitness.

Outra abordagem possível é transformar o modelo em um ambiente multiagente, onde teríamos jogadores cooperando ou traindo uns aos outros, além de transformar o próprio Wumpus em um agente. Neste cenário teríamos uma gama de discussões sobre o comportamento de agentes e as estratégias escolhidas.

## Referências

1. AL-ZANITI, M.; ARAUJO, F.; KUIPER, D.; VALENTE, J.; WENKSTERM, R.Z. DIVAs 4.0: A Multi-Agent Based Simulation Framework. 17th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications. 2013. Disponível em: <<https://dl.acm.org/citation.cfm?id=2570883>>. Acessado em: 27 Mai. 2019.

2. BENTO, E. P.; KAGAN, N. Algoritmos genéticos e variantes na solução de problemas de configuração de redes de distribuição. **Sba: Controle & Automação Sociedade Brasileira de Automatica**, [s.l.], v. 19, n. 3, p.302-315, set. 2008. FapUNIFESP (SciELO).
3. BICKLE, T.; LOTHAR, T. A mathematical analysis of Tournament selection, Proceedings of the Sixth International Conference on Genetic Algorithms, In Morgan Kaufmann Publishers (Ed.), San Francisco, CA, pp. 9-16. 1995.
4. BRENNER, W.; ZARNEKOW, R.; WITTIG, H. Intelligent Software Agents. Springer, 1998.
5. BROWNLEE, Jason. Clever Algorithms: Nature-Inspired Programming Recipes. Creative Commons. 2012.
6. GARRO, Alfredo et al. Intelligent Agents and Environment. Encyclopedia Of Bioinformatics And Computational Biology, [s.l.], p.309-314, 2019. Elsevier.
7. MÜLLER, J.P. The design of intelligent agents: a layered approach. Lecture Notes in Computer Science, Springer-Verlag, Heidelberg, vol. 1177, 1996.
8. REZENDE, S. O. Sistemas inteligentes: fundamentos e aplicações. Barueri, SP: Manole. 2005.
9. RUSSELL, Stuart. Rationality and Intelligence: A Brief Update. Fundamental Issues Of Artificial Intelligence, [s.l.], p.7-28, 2016. Springer International Publishing.
10. RUSSELL, Stuart; DEWEY, Daniel; TEGMARK, Max. Research Priorities for Robust and Beneficial Artificial Intelligence. Ai Magazine, [s.l.], v. 36, n. 4, p.105-114, 31 dez. 2015. Association for the Advancement of Artificial Intelligence (AAAI).
11. RUSSEL, S.; NORVIG, P. Inteligência Artificial: Uma abordagem moderna. 2a. ed. Rio de Janeiro: Elsevier Editora, 2013.
12. SILVEIRA, Sidnei Renato. Formação de Grupos Colaborativos em um Ambiente Multiagente Interativo de Aprendizagem na Internet: um estudo de caso utilizando sistemas multiagentes e algoritmos genéticos. Tese de Doutorado (Programa de Pós-graduação em Computação) – Universidade Federal do Rio Grande do Sul. 2006.
13. STANGE, R L; CEREDA, P R M; NETO, J José. Agentes adaptativos reativos: formalização e estudo de caso. p. 63–71, 2017.
14. TEIXEIRA, O. N. ALGORITMO GENÉTICO COM INTERAÇÃO SOCIAL NEBULOSA. Tese de Doutorado (Programa de Pós-graduação em Engenharia Elétrica – Computação Aplicada) - Universidade Federal do Pará, Belém, 2012.
15. WEYNS, D.; OMICINI, A.; ODELL, J. Environment as a first class abstraction in multiagent systems. Autonomous Agents and Multi-Agent Systems 14, 5–30. 2007.
16. YOB, Gregory. Hunt the Wumpus. The Best Of Creative Computing. Vol. 1. P. 247-250. 1976. Disponível em: < <https://www.atariarchives.org/bcc1/showpage.php?page=247>>. Acesso em: 25 Mai. 2019.