

Proposal of an automatic tool for evaluating the quality of decision-making on Checkers player agents

Matheus Prado Prandini Faria¹, Rita Maria Silva Julia¹, Lídia Bononi Paiva Tomaz^{1,2}

¹Departamento de Ciência da Computação – Universidade Federal de Uberlândia (UFU)
Uberlândia, MG – Brasil

²Instituto Federal do Triângulo Mineiro (IFTM) – Uberaba, MG – Brasil
{matheusprandini.96, ritasilvajulia, ldbononi}@gmail.com

Abstract. *Checkers player agents represent an appropriate case study for the best unsupervised methods of Machine Learning. This work presents a tool to measure the performance of these methods based on the quality of the decision making of these agents. The proposed tool, based on the data of movements performed in real games by the agents under evaluation, provides a statistical way of automatically comparing the coincidence rates between the decision making of the evaluated agents with those that the remarkable player agent Cake would do in the same situations. The tool was validated through tournaments between agents comparing their respective coincidence rates and their performance.*

Resumo. *Agentes para Damas representam um estudo de caso apropriado para avaliar métodos não supervisionados de Aprendizagem de Máquina. Este trabalho propõe uma ferramenta para mensurar o desempenho desses métodos com base na qualidade das tomadas de decisão desses agentes. A ferramenta proposta, a partir de dados de movimentos efetuados em jogos reais pelos agentes sob avaliação, provê uma maneira estatística de comparar, automaticamente, as taxas de coincidência entre as tomadas de decisão dos agentes avaliados com aquelas que o consagrado agente Cake faria nas mesmas situações de jogo. Validou-se a ferramenta por meio de torneios entre os agentes em que se comparam suas respectivas taxas de coincidência e seus desempenhos.*

1. Introdução

A Teoria dos Jogos tornou-se um importante segmento da matemática que foi aplicada de forma mais intensa durante a metade do século passado, especialmente após a publicação do clássico livro “*The Theory of Games and Economic Behavior*” [Neumann e Morgenstern 1953]. Desde então, tem despertado interesse crescente na área da Ciência da Computação, com ênfase especial nos desafios associados à Aprendizagem de Máquina (AM) na Inteligência Artificial (IA). Resumidamente, a AM consiste em dotar os agentes da IA da habilidade de adquirir conhecimentos que os tornem aptos a tratar os problemas para os quais foram projetados com a máxima autonomia possível [Russell e Norvig 2004]. Assim sendo, as dificuldades inerentes a diversos jogos os tornam um notável desafio a ser enfrentado no domínio da AM. Neste caso, a aplicação da AM tem como meta o desenvolvimento de agentes que desenvolvam progressivamente suas habilidades de modo a encontrar estratégias apropriadas para atingir um bom nível de jogo. Para tanto, tais agentes devem ser capazes de:

- Aprender a se comportar em um ambiente onde o conhecimento adquirido é armazenado em uma função de avaliação;
- Escolher um mínimo de atributos possíveis que melhor caracterizem o domínio e que sirvam como um meio pelo qual a função de avaliação adquirirá novos conhecimentos;
- Selecionar a melhor ação para um determinado estado ou configuração do ambiente onde o agente está interagindo (problema de otimização), levando em conta o fato de que, ao executar tal seleção, ele estará alterando o leque de opções de ações possíveis para os demais agentes envolvidos no ambiente.

No caso particular dos jogos de tabuleiro, como os relacionados na Tabela 1, o agente jogador deve também lidar com dois elementos que dificultam a sua tomada de decisão, a saber: a complexidade do espaço de estados e a complexidade da árvore de jogo. A complexidade do espaço de estados é definida como um conjunto de todos os possíveis estados de jogo legais que podem ser atingidos a partir do estado inicial. A complexidade da árvore de jogo é definida pelo fator de ramificação da árvore de busca [van den Herik et al. 2002].

Tabela 1. Complexidade do espaço de estados e fator de ramificação de alguns jogos [Campos e Langlois 2003]

Jogo	Fator de Ramificação	Espaço de Estados
Xadrez	30 - 40	10^{50}
Damas	8 - 10	10^{17}
Gamão	+ - 420	10^{20}
Othello	+ - 5	$< 10^{30}$
Go 19x19	+ - 360	10^{160}
Abalone	+ - 80	$< 3^{61}$

Tais dados retratados na Tabela 1 reforçam os argumentos que tornam os jogos de tabuleiro - dentre eles, destacadamente o jogo de Damas - um domínio extremamente rico a ser explorado em AM. De fato, vários agentes jogadores de Damas vêm sendo propostos pela comunidade científica internacional, tanto no contexto da aprendizagem supervisionada (quando o processo de aprendizagem é fortemente norteado por especialistas humanos ou por bases de dados), quanto no contexto da não-supervisionada (quando o agente aprende baseado essencialmente em suas próprias experiências) [Russell e Norvig 2004]. Exemplos de agentes supervisionados para Damas são encontrados em [Schaeffer et al. 1996] e [Fierz 2010], ao passo que propostas não supervisionadas são apresentadas, por exemplo em [Fogel e Chellapilla 2002], [Caixeta e Julia 2008], [Tomaz et al. 2013], [Neto et al. 2014], [Duarte et al. 2015], [Tomaz et al. 2017]. Salienta-se que os agentes supervisionados acima referenciados, denominados *Chinook* [Schaeffer et al. 1996] e *Cake* [Fierz 2010], são jogadores de topo no cenário mundial de competições de Damas. O *Chinook* é o atual campeão mundial em torneios homem-máquina. Em 2007, os autores do *Chinook* provaram que, com ele, o jogo de Damas estava fracamente resolvido, isto é, tal agente é invencível em jogos que partem do tabuleiro inicial padrão do jogo de Damas (no pior dos casos, há empate) [Schaeffer et al. 2007]. O *Cake* é um jogador automático supervisionado que possui grande renome por já ter vencido uma versão preliminar do *Chinook* - versão

esta famosa por ter derrotado o campeão mundial de Damas da época, *Marion Tinsley* [Schaeffer et al. 1996]. É importante ressaltar que o *Cake* está disponível na plataforma *CheckerBoard*, que é a mais completa interface livre para o jogo de Damas [Fierz 2008]. Além disso, esta plataforma dispõe de uma funcionalidade que permite comparar um movimento realizado por meio dela com o movimento que o *Cake* efetuará a partir do mesmo estado de jogo.

Apesar das indiscutíveis contribuições científicas providas pela construção dos jogadores automáticos *Chinook* e *Cake*, eles foram concebidos sob a ótica de técnicas de aprendizagem supervisionadas, fato que faz com que Damas continue a ser um problema muito interessante a ser estudado no contexto da aprendizagem não supervisionada.

No cenário das pesquisas relacionadas a agentes jogadores de Damas, é um desafio avaliar o quão positivo tem sido o emprego de determinadas técnicas de IA, mais especificamente, de AM. Diversos agentes jogadores automáticos propostos na literatura tem o desempenho avaliado a partir de parâmetros como: o número de vitórias obtidas em torneios disputados, o tempo de tomada de decisão (ou escolha de movimento) e o tempo de treinamento do agente [Caixeta e Julia 2008], [Tomaz et al. 2013], [Tomaz et al. 2017]. Trabalhos mais recentes tem realizado a avaliação de desempenho por meio da comparação manual entre as tomadas de decisão dos agentes avaliados e as tomadas de decisão que o *Cake* faria nas mesmas condições [Neto et al. 2014] e [Duarte et al. 2015]. No entanto, tal método de avaliação manual é muito trabalhoso e oneroso em termos de tempo, fato que inviabiliza realizar essa avaliação em uma elevada quantidade de jogos (o que representaria a situação ideal). Por exemplo, em [Neto et al. 2014] esta avaliação foi realizada em 14 jogos em que foram considerados apenas os 15 primeiros movimentos. Em [Duarte et al. 2015], apesar de terem sido considerados jogos completos, a análise ficou restrita ao número de 6 jogos.

Desta forma, a principal contribuição do presente trabalho consiste na implementação de uma ferramenta para avaliação automática da qualidade da tomada de decisão de agentes jogadores de Damas a partir da plataforma *CheckerBoard*. Para tanto, o algoritmo de busca do agente a ser avaliado - cujo papel é definir o melhor movimento que ele deve executar a partir do tabuleiro corrente - deve gerar um arquivo no formato PDN (*Portable Draughts Notation*). Tal arquivo contém todas as decisões tomadas pelo agente ao longo de jogos reais nos quais tenha atuado. A partir deste arquivo, a ferramenta proposta efetua uma conexão com a plataforma *CheckerBoard* de forma a comparar estatisticamente a taxa de coincidência entre tais tomadas de decisão e as escolhas de movimento que seriam efetuadas, nas mesmas situações, pelo *Cake*.

A fim de validar a ferramenta proposta, este trabalho realizou um estudo de caso a partir de dois agentes jogadores automáticos não-supervisionados de Damas: o *VisionDraughts* [Caixeta e Julia 2008] e sua versão aprimorada *D-VisionDraughts* [Tomaz et al. 2013]. Para tanto, foi realizado um torneio entre estes jogadores de modo a gerar os arquivos PDN relativos às partidas realizadas. Após a execução da ferramenta, foi possível notar que a taxa de coincidência dos movimentos escolhidos pelo *D-VisionDraughts* comparada aos do *Cake* foi superior àquela calculada para o *VisionDraughts*. Tal resultado ratifica o acerto do cálculo efetuado pela ferramenta, uma vez que trabalhos anteriormente publicados já haviam comprovado, por meio de número de vitórias e de tempo de treinamento, a superioridade do *D-VisionDraughts* em relação a seu

oponente [Tomaz et al. 2013]. Desta forma, esta ferramenta revela ser um complemento no processo avaliativo de desempenho de novas arquiteturas automáticas para jogadores de Damas. Além disso, conforme apresentado nos resultados, o tempo de execução requerido pela ferramenta para conclusão da avaliação é bem reduzido, fato que permite que sejam analisados diversos jogos completos (que iniciam no estado padrão do tabuleiro) e/ou parciais (com estados de tabuleiro já iniciados) em um curto espaço de tempo.

As seções contidas nesse trabalho estão organizadas da seguinte maneira: a Seção 2 apresenta os fundamentos técnicos e teóricos; a Seção 3 apresenta os trabalhos relacionados; a Seção 4 descreve a arquitetura da ferramenta de automatização da análise estatística de movimentos de agentes com relação ao *Cake*; a Seção 5 apresenta os experimentos e resultados; e, finalmente, a Seção 6 descreve as conclusões e propostas de trabalhos futuros.

2. Fundamentos Técnicos e Teóricos

Esta seção apresenta conceitos técnicos e teóricos importantes para a compreensão deste trabalho.

2.1. Arquivo PDN

O formato de arquivo padrão de jogos de Damas é o PDN (*Portable Draughts Notation*). Independente da plataforma e/ou linguagem de programação que o agente jogador foi desenvolvido, o arquivo PDN segue a mesma estrutura. Desta forma, um arquivo PDN deve conter os seguintes atributos:

- *Event*: nome da partida;
- *Date*: data de realização da partida;
- *Black*: identificação do jogador preto;
- *White*: identificação do jogador branco;
- *Result*: resultado da partida;
- Registro de todos os movimentos realizados ordenados do início ao fim de partida.

O objetivo do arquivo PDN é criar um meio simples para armazenar todos os passos existentes em uma partida de tal modo que seja possível compartilhar um jogo completo. Neste contexto, existem diversas coleções de arquivos PDN com a finalidade de estudo e utilização no aprimoramento das habilidades de jogadores automáticos.

É importante ressaltar que várias partidas podem ser armazenadas em um único arquivo PDN, porém, neste trabalho, cada arquivo PDN representa somente uma partida.

2.2. CheckerBoard

A *CheckerBoard* [Fierz 2008] é uma plataforma internacional livre para jogos de Damas e suporta a notação PDN. Esta plataforma disponibiliza algumas das *engines* (agentes automáticos) de maior sucesso para Damas como por exemplo *Cake* e *Kingsrow* [Fierz 2010], [Gilbert 2000] e [Fierz 2008]. A interface desta plataforma permite que jogadores humanos e automáticos possam jogar contra as *engines* disponíveis nela. Além disso, a *CheckerBoard* disponibiliza base de dados contendo milhares de jogos de competições entre os grandes jogadores de Damas, incluindo jogos entre *Marion Tinsley* e *Chinook* [Schaeffer et al. 1996].

Destaca-se que uma das principais funcionalidades desta plataforma é a análise completa de jogos executada pelo agente *Cake*. Neste caso, tal agente percorre todos os movimentos realizados em uma partida armazenada em um arquivo PDN de modo a indicar quais seriam as melhores jogadas para cada estado de tabuleiro atingido. Desta forma, é possível comparar cada tomada de decisão em uma partida com a ação que o *Cake* executaria na mesma situação de jogo.

Neste trabalho, a *CheckerBoard* foi utilizada como meio para reproduzir as partidas dos agentes jogadores a serem avaliados de modo a permitir a comparação das tomadas de decisão de tais agentes com o *Cake*.

2.3. Algoritmo de Busca Alpha-Beta

Um jogo de Damas pode ser visualizado como uma árvore de estados de tabuleiro. Cada nó da árvore representa um estado e cada ramo representa um movimento. Para lidar com esse tipo de jogo, o algoritmo de busca Alpha-Beta é muito utilizado. Ele consiste de uma otimização do algoritmo de busca Minimax.

O Minimax pode ser simplificado como um algoritmo recursivo que busca o melhor movimento a partir do estado de tabuleiro atual através de uma busca em profundidade da esquerda para a direita em árvores de jogo [Caixeta e Julia 2008] [Russell e Norvig 2004]. Neste caso, a raiz da árvore corresponde ao estado do tabuleiro e o melhor movimento é escolhido de acordo com a avaliação calculada para os nós folhas. Por exemplo, na Figura 1 (a) o Minimax indica que o agente deve escolher o *Movimento A* a partir do estado corrente (nó raiz).

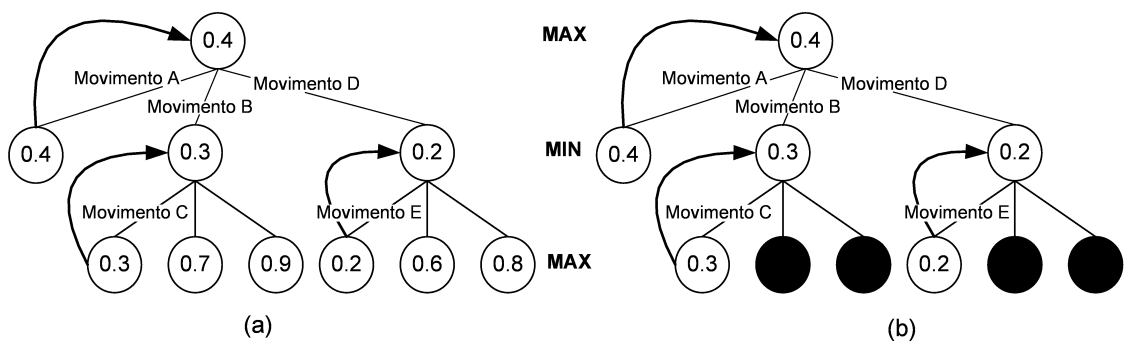


Figura 1. Exemplo de árvore explorada pelos algoritmos (a) Minimax e (b) poda Alpha-Beta

A desvantagem do Minimax é que ele examina mais nós do que o necessário. A fim de resolver este problema, o algoritmo de busca poda Alpha-Beta o otimiza ao eliminar partes da árvore que de fato não possuem o melhor movimento [Caixeta e Julia 2008]. Este algoritmo recebe como parâmetro o estado de tabuleiro atual, bem como dois valores: alfa e beta. Alfa e beta delimitam o intervalo para o valor de avaliação correspondente ao melhor movimento a ser executado a partir do estado corrente do tabuleiro. A avaliação de um nó minimizador pode ser interrompida no momento em que o valor de avaliação calculado é menor que alfa (conhecido como poda alfa). Analogamente, a avaliação de um nó maximizador pode ser interrompida quando o valor de avaliação calculado para um

filho é maior que beta (conhecido como poda beta). Por exemplo, a Figura 1 (b) apresenta a mesma solução da Figura 1 (a). Todavia, o algoritmo Alpha-Beta detecta que não é necessário avaliar os nós destacados. Desta forma, o melhor movimento (Movimento A) é encontrado mais rapidamente.

O algoritmo Alpha-Beta tem sido muito empregado na solução de jogos como Damas, todavia, nestes jogos há uma forte correlação entre a profundidade de busca que um jogador consegue atingir (visão futura da partida) e a sua força em partidas. Desta forma, devido à complexidade do problema (veja tabela 1), diversas propostas de aprimoramento do Alpha-Beta tem sido propostas na literatura com destaque à paralelização deste algoritmo. Dentre as propostas de paralelização do Alpha-Beta encontra-se o *Young Brothers Wait Concept* (YBWC) [Feldmann et al. 1990].

3. Trabalhos Relacionados

Esta seção apresenta alguns dos principais agentes jogadores automáticos para Damas existentes na literatura enfocando aqueles relacionados a este trabalho.

O *Chinook* é o atual campeão mundial homem-máquina para Damas. Ele foi concebido de modo fortemente supervisionado, uma vez que conta com funções de avaliação que foram ajustadas manualmente ao longo de muitos anos. Em 2007, sua equipe executora provou que o jogo de Damas estava fracamente resolvido, isto é, partindo-se do tabuleiro inicial padrão do jogo de Damas, existe uma prova computacional de que o *Chinook*, no pior caso, obtém um empate no jogo [Schaeffer et al. 2007].

O *Cake* é um jogador concebido de modo supervisionado [Fierz 2010]. Ele conta com uma base de abertura de jogos, que possui cerca de 2 milhões de movimentos por padrão e uma base de final de jogo que permite com que esse jogador tenha perfeito conhecimento do valor de qualquer estado de tabuleiro com até 8 peças. Para atuar no jogo entre as fases de início e fim de jogo (fases em que o jogador não possui acesso às bases de dados) o *Cake* utiliza o algoritmo de busca *MTD(f)*. Tal algoritmo tem capacidade de processar em média 2 milhões de posições por segundo [Fierz 2010]. O *Cake* é conhecido por ter vencido uma versão preliminar do *Chinook* - versão esta famosa por ter derrotado o campeão mundial da época *Marion Tinsley*. Este jogador está disponível por meio da plataforma *CheckerBoard* [Fierz 2008].

O *VisionDraughts* é um jogador que consiste de uma Rede Neural Multicamadas (MLP - *Multilayer Perceptron*) que aprende sem supervisão humana através dos Métodos das Diferenças Temporais $TD(\lambda)$ aliado à estratégia de treino por *self-play* com clonagem. A ideia básica desta estratégia de treino é realizar diversos jogos contra uma cópia de si próprio de modo a ajustar os parâmetros do agente que está treinando. À medida tal agente melhora sua performance, um novo clone é produzido e o treino continua com o agente ajustado e seu clone. A representação do tabuleiro é feita através de um conjunto de funções que descrevem características inerentes ao próprio jogo de Damas. Tal representação é denominada NET-FEATUREMAP. A escolha da melhor ação a ser executada pelo jogador é conduzida pelo algoritmo de busca Alpha-Beta aliado com Tabelas de Transposição (TT), que é um repositório para armazenar estados de tabuleiro já avaliados.

O *D-VisionDraughts* [Tomaz et al. 2013] estende a arquitetura do *VisionDraughts* aprimorando o processo de escolha da melhor ação. Para isso, o algoritmo de

busca serial Alpha-Beta é substituído pela versão distribuída do Alpha-Beta YBWC. Em [Tomaz et al. 2013], os autores realizaram uma análise comparativa entre o *D-VisionDraughts* e o *VisionDraughts* em termos de tempo de treinamento da rede jogadora e da performance dos jogadores no que se refere ao número de vitórias obtidas em torneios. Os resultados mostraram que o *D-VisionDraughts* teve tempo de treinamento 43% menor e, nos torneios, o *D-VisionDraughts* apresentou uma performance 17.5% superior ao seu oponente *VisionDraughts*. Desta forma, os resultados da literatura apontam que a utilização do algoritmo de busca distribuído fez com que o *D-VisionDraughts* se mostrasse superior ao seu predecessor *VisionDraughts*.

Este trabalho utiliza os jogadores *VisionDraughts* e *D-VisionDraughts* como estudo de caso a fim de validar a ferramenta aqui proposta comparando os resultados obtidos com os resultados da literatura.

4. Método Automatizado para Avaliação de Qualidade de Movimentos de Agentes Jogadores de Damas

Esta seção apresenta a arquitetura da ferramenta proposta neste trabalho. Basicamente, ela consiste de uma integração entre dois módulos: Ambiente de Atuação dos Agentes Avaliados (AAAA) e Ferramenta de Avaliação de Qualidade de Movimentos (FAQM), conforme ilustrado na Figura 2 e detalhado na sequência.

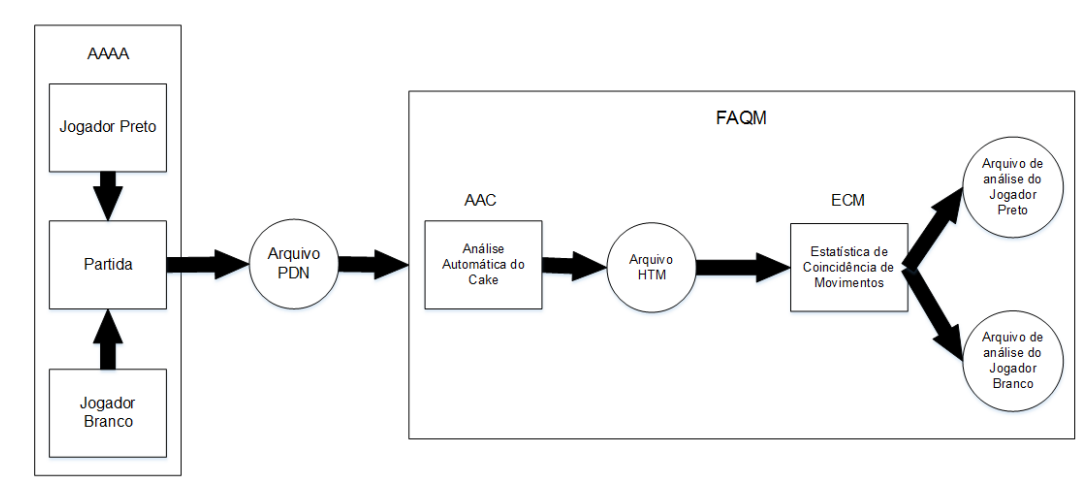


Figura 2. Arquitetura da ferramenta de automatização da análise estatística de movimentos

Módulo AAAA: Este componente tem como objetivo gerar os arquivos PDN que armazenam os conjuntos de movimentos executados pelos agentes a serem avaliados ao longo de um torneio composto por diversos jogos-teste. Cada um desses conjuntos se refere a um jogo-teste completo (a partir do qual será gerado um dos arquivos PDN). Contudo, os tabuleiros a partir do qual cada um desses jogos tem início varia de um conjunto para outro - sendo que um deles corresponde, necessariamente, ao tabuleiro inicial padrão de Damas, ao passo que cada um dos demais contém um número de peças pré-estabelecido, respeitando-se a restrição de que, tanto a quantidade, quanto a natureza das peças pretas e brancas têm de ser

a mesma para ambos os jogadores envolvidos. Convém salientar que tal estratégia é fundamental para a acuidade do cálculo da qualidade de movimentos, uma vez que em Damas (como em vários outros problemas práticos) o desempenho do agente é extremamente dependente do estado inicial do qual partiu. Tal fator é tão significativo que, tal como citado na seção 1, a garantia de invencibilidade do *Chinook* somente ocorre em jogos que partem do tabuleiro inicial padrão do jogo de Damas [Schaeffer et al. 2007]. Ainda com relação aos jogos-teste, é importante ressaltar que, neles, o agente pode enfrentar qualquer adversário, inclusive ele mesmo (situação a que se denomina jogos por *self-play*). Assim sendo, caso os jogos-teste envolvam dois adversários A (jogador preto) e B (jogador branco) distintos, os arquivos PDN gerados pelo módulo armazenarão, tanto os conjuntos de movimentos executados pelos jogador A (jogando com as peças pretas), quanto os conjuntos executados pelo competidor B (jogando com as peças brancas). Por outro lado, caso os jogos-teste ocorram por *self-play*, os conjuntos armazenados nos arquivos PDN se referirão aos movimentos executados pelo único jogador envolvido na competição jogando, ora com as peças pretas, ora com as peças brancas. Uma vez gerados todos os arquivos PDN correspondentes a todos os jogos-teste do torneio, o módulo AAAA produz uma pasta contendo todos eles.

Módulo FAQM: Este componente tem duas finalidades básicas: 1) Gerar um conjunto de informações relacionadas ao comportamento que o agente *Cake* teria tido diante das várias situações (estados de tabuleiro) enfrentadas pelos agentes durante os jogos-teste; 2) Gerar as taxas de coincidência entre os movimentos executados pelos agentes nos jogos-teste (armazenados no arquivo PDN) e os movimentos que teriam sido executados pelo *Cake* nas mesmas situações. A primeira finalidade é executada no sub-módulo do FAQM denominado Análise Automática do *Cake* (AAC), ao passo que a segunda é cumprida no sub-módulo denominado Estatística de Coincidência de Movimentos (ECM). Tais sub-módulos serão descritos com mais detalhes nas seções 4.1 e 4.2, respectivamente.

4.1. Sub-Módulo AAC: Análise Automática do *Cake*

O objetivo deste sub-módulo do FAQM é gerar um conjunto de informações referentes à maneira como o agente *Cake* se comportaria em situações de tabuleiro análogas às enfrentadas ao longo dos jogos-teste efetuados em um torneio pelos agentes a serem analisados. Tais informações são armazenadas em arquivos com extensão *htm*, referenciados a partir daqui como HTM. Cada arquivo HTM corresponde a um dado arquivo PDN e é produzido da seguinte forma: o módulo AAC reproduz no ambiente da plataforma *CheckerBoard* a partida armazenada no arquivo PDN que lhe corresponde (conforme seção 2.2). Para cada movimento obtido a partir dessa partida reproduzida, o sub-módulo recupera um conjunto contendo as seguintes informações disponibilizadas pela plataforma relativas àquele movimento (sob a ótica do *Cake*):

- *Movimento-agente-avaliado*: movimento executado pelo agente avaliado (informação recuperada a partir do arquivo PDN correspondente);
- *Lado*: cor (branco ou preto) com que o agente avaliado executou o movimento acima;
- *Movimento-Cake*: movimento que o *Cake* indicaria na mesma situação de tabuleiro e de cor com que foi executado o movimento do agente avaliado.

Conforme pode-se observar, as informações consideradas incluem o parâmetro *lado* representando a cor das peças com que o movimento foi executado. Isso representa um cuidado essencial, pois a avaliação de um dado movimento sofre profundo impacto em função do lado que o executou.

A título de aumentar a eficiência avaliativa da ferramenta aqui proposta, o sub-módulo AAC foi implementado de modo a estar apto a gerar diversos arquivos HTM simultaneamente. Para tanto, ele conta com dois parâmetros: *torneio* e *nro-jogos*. O primeiro, indica o caminho para a pasta de arquivos PDN produzida pelo módulo AAAA. O segundo, indica a quantidade de arquivos PDN armazenados na referida pasta. Tal habilidade permite que as informações referentes à totalidade dos jogos-teste sejam recuperadas em um único processamento do AAC, o que é um fator determinante para a redução do tempo necessário para se analisar uma quantidade significativa de arquivos PDN.

Uma vez que os arquivos HTM foram gerados, o AAC produz uma pasta contendo os mesmos a ser utilizada pelo sub-módulo ECM exposto na próxima seção.

4.2. Sub-Módulo ECM: Estatística de Coincidência de Movimentos

A finalidade deste sub-módulo é gerar as taxas de coincidência entre os movimentos executados pelos agentes nos jogos-teste (armazenados na pasta de PDNs) e os movimentos que teriam sido executados pelo *Cake* nas mesmas situações (armazenados na pasta de HTMs). Para tanto, o ECM, a partir das informações produzidas pelo sub-módulo AAC relativas aos movimentos executados pelos agentes avaliados no torneio (ou seja, *Movimento-agente-avaliado*, *Lado* e *Movimento-Cake*), calcula a estatística de coincidências entre todos os movimentos executados por eles ao longo do torneio e os movimentos que o *Cake* teria executado nas mesmas condições de jogo (tabuleiro e cor).

Ao final da análise, o ECM gera um arquivo para cada jogador avaliado com as seguintes informações estatísticas:

- Quantidade total de movimentos realizados pelo jogador;
- Quantidade total de movimentos realizados pelo agente coincidentes com o *Cake*;
- Média da porcentagem dos movimentos do agente coincidentes com o *Cake*;

Tais estatísticas são calculadas, independentemente, para ambos os lados em que o jogador automático pode atuar. Assim sendo, o referido arquivo é composto por três conjuntos de informações estatísticas distintos: os dois primeiros, onde as informações se referem exclusivamente às estatísticas envolvendo os movimentos executados com peças brancas ou pretas, respectivamente, e, o terceiro, apresentando as informações globais consolidadas relativas aos jogos completos (jogadas com peças brancas e pretas). Tal estratégia é bastante pertinente, uma vez que permite avaliar as taxas de coincidências de movimentos dos agentes avaliados em relação ao *Cake* nos cenários em que jogam como jogador branco e como jogador preto, além de prover uma visão global do desempenho dos mesmos jogando em ambos os lados.

5. Experimentos e Resultados

Esta seção apresenta os testes realizados com o objetivo de validar a presente proposta. Os agentes avaliados como estudo de caso são o *VisionDraughts* e o *D-VisionDraughts*. Os

testes calculam a taxa média de coincidência entre os movimentos escolhidos pelos agentes *VisionDraughts* e *D-VisionDraughts* no curso de jogos-teste em que eles se enfrentam e os movimentos que, nas mesmas situações, seriam executados pelo agente *Cake*. As versões de tais agentes utilizadas nos experimentos aqui executados são as mesmas empregadas nos testes realizados em [Tomaz et al. 2013], portanto, são agentes já treinados por *self-play* com clonagem conforme apresentado na seção 3.

Dessa forma, foram realizadas 40 partidas entre os dois agentes, sendo que 20 delas representam a situação em que o *VisionDraughts* atua com peças brancas e o *D-VisionDraughts* com peças pretas, enquanto que, nas outras 20 partidas, o *VisionDraughts* atua com peças pretas e o *D-VisionDraughts* com peças brancas. Além disso, conforme previsto na seção 4, tais jogos partem de estados iniciais pré-definidos e diversos, respeitando-se a restrição de que, tanto a quantidade, quanto a natureza das peças, são as mesmas para ambos os agentes. Tal fato impede que um agente esteja em posição mais vantajosa do que o outro, o que valida a consistência dos resultados, além de permitir que ambos sejam avaliados em situações variadas de jogo (uma vez que, conforme apresentado na seção 4, o tabuleiro inicial tem forte impacto no desempenho do agente).

Assim sendo, são apresentados três cenários de teste (todos executados em uma máquina com processador *Intel Core i5-4440S CPU @ 2.80GHz com 4 núcleos e 8 GB de RAM*): o primeiro cenário (I) avalia a qualidade de movimentos dos agentes *VisionDraughts* e *D-VisionDraughts* com relação ao *Cake*, em situação em que os agentes atuam com peças brancas, ao passo que o segundo cenário (II) efetua a mesma avaliação na situação oposta, ou seja, em jogos nos quais os agentes atuam com peças pretas. Por fim, o terceiro cenário (III) avalia a qualidade dos movimentos de ambos agentes segundo uma ótica global, isto é, atuando como jogadores brancos e pretos. Para tanto, o cálculo deste terceiro cenário é efetuado a partir dos 40 jogos realizados entre os referidos agentes. Os resultados obtidos pelos três cenários de teste são apresentados na Tabela 2.

Tabela 2. Taxa de Coincidência de Movimentos em Relação ao *Cake*

Cenários	VisionDraughts	D-VisionDraughts
Cenário I - Atuação com peças brancas (Média de coincidência com o <i>Cake</i> sobre 20 partidas)	31,52%	33,05%
Cenário II - Atuação com peças pretas (Média de coincidência com o <i>Cake</i> sobre 20 partidas)	35,01%	43,25%
Cenário III - Média dos cenários anteriores (Média de coincidência com o <i>Cake</i> sobre 40 partidas)	33,20%	38,39%

Analisando os resultados dos três cenários mostrados na Tabela 2, observa-se a superioridade geral (como jogador preto ou branco) da qualidade dos movimentos executados pelo *D-VisionDraughts* com relação a seu oponente. De fato, como jogador preto, o *D-VisionDraughts* obteve uma taxa de coincidência de movimentos com relação ao *Cake* de 43,25%, contra 35,01% obtida pelo *VisionDraughts*. Além disso, como jogador branco, essas mesmas porcentagens para o *D-VisionDraughts* e seu oponente foram de 33,05% e de 31,52%, respectivamente. Esse panorama é sintetizado nos valores mostrados no cenário III, em que o agente *D-VisionDraughts* apresentou uma taxa média de coincidência de movimentos com relação ao *Cake* de 38,39%, contra 33,20% apresentada pelo *VisionDraughts*.

É importante destacar que, quando um agente apresenta uma taxa de coincidência maior em relação ao *Cake*, ele mostra que suas tomadas de decisão estão mais condizentes daquelas que seriam realizadas por um agente com alto nível de desempenho em Damas. Logo, considera-se que suas ações foram mais assertivas rumo à vitória. De fato, o parâmetro *taxa de vitórias* está diretamente associado à qualidade dos movimentos selecionados pelo agente jogador. Nesta direção, é possível atestar a validade da ferramenta aqui proposta relacionando os resultados apresentados na Tabela 2 aos resultados reportados em [Tomaz et al. 2013] em que os autores provaram que o *D-VisionDraughts* tem um desempenho geral em termos de taxas de vitórias em torneios 17.5% superior ao *VisionDraughts*. Desta forma, é possível verificar a coerência entre a maior taxa de coincidência que o *D-VisionDraughts* apresentou em relação ao *Cake* e a sua superioridade nas vitórias nos torneios apresentados em [Tomaz et al. 2013].

Um outro aspecto relevante a ser destacado é a agilidade que a ferramenta automática aqui proposta proporciona ao processo avaliativo baseado em comparações com os movimentos indicados pelo *Cake*. De fato, o cálculo estatístico automático relativo às 40 partidas envolvidas nos cenários citados foi efetuado pela ferramenta proposta em apenas 3 minutos e 10 segundos, aproximadamente. Obviamente, uma avaliação análoga feita manualmente (tal como efetuado em [Neto et al. 2014] e em [Duarte et al. 2015]) requereria um tempo substancialmente maior, além de ser passível de falhas, uma vez que precisaria tratar individualmente cada movimento, de cada partida, comparando-o, manualmente, com o resultado da busca de melhor movimento do *Cake*. Isso diminuiria, drasticamente, a quantidade de jogos-teste e de movimentos que poderiam ser analisados, limitação que desaparece com o uso da ferramenta aqui proposta.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma ferramenta automática para mensurar a qualidade da tomada de decisão para agentes de Damas com base na métrica comparativa entre a taxa de coincidência dos movimentos escolhidos por eles e os que teriam sido feitos, nas mesmas situações, pelo consagrado agente para Damas *Cake*. Ressalte-se que o elevadíssimo nível de jogo de tal agente - que o torna uma referência indiscutível para ser usado como lastro de comparação - foi atingido por meio de um processo fortemente supervisionado de aprendizagem. A presente proposta foi validada por meio da análise de coerência entre os resultados obtidos pelo uso da ferramenta aqui proposta em torneios envolvendo os agentes não-supervisionados conhecidos na literatura como *VisionDraughts* e *D-VisionDraughts* e aqueles divulgados por seus autores relativos a suas respectivas taxa de vitórias em competições [Tomaz et al. 2013]. É importante destacar que tal coerência provém do fato de que o parâmetro *taxa de vitórias* é diretamente associado ao parâmetro que estima a qualidade da tomada de decisão. Além de representar um critério bastante eficaz para avaliar o nível de aprendizagem de agentes jogadores, a ferramenta aqui proposta traz como contribuição adicional o fato de agilizar substancialmente o processo de avaliação da qualidade de movimentos escolhidos por eles. Tal contribuição elimina o gargalo imposto pelo processo manual que limita drasticamente a quantidade de jogos-teste efetuados no processo avaliativo. Como trabalho futuro, pretende-se utilizar a ferramenta proposta para avaliar o desempenho de novos agentes automáticos de Damas com arquiteturas baseadas em inéditas propostas de versões distribuídas do algoritmo de busca Alpha-Beta que vêm sendo desenvolvidas pelos autores do presente trabalho.

Referências

- Caixeta, G. S. e Julia, R. M. S. (2008). *A Draughts Learning System Based on Neural Networks and Temporal Differences: The Impact of an Efficient Tree-Search Algorithm*, pages 73–82. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Campos, P. e Langlois, T. (2003). Abalearn: Efficient self-play learning of the game abalone. In *INESC-ID, Neural Networks and Signal Processing Group*.
- Duarte, V. A. R., Julia, R. M. S., Albertini, M. K., e Neto, H. C. (2015). Mp-draughts: Un-supervised learning multi-agent system based on mlp and adaptive neural networks. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 920–927.
- Feldmann, R., Monien, B., Mysliwicz, P., e Vornberger, O. (1990). *Distributed Game Tree Search*, pages 66–101. Springer New York, New York, NY.
- Fierz, M. C. (2008). *CheckerBoard informations and API*.
- Fierz, M. C. (2010). *Cake informations. Technical report*.
- Fogel, D. B. e Chellapilla, K. (2002). Verifying anaconda’s expert rating by competing against chinook: experiments in co-evolving a neural checkers player. *Neurocomputing*, 42(1):69 – 86. Evolutionary neural systems.
- Gilbert, E. (2000). *KingsRow Technical report*.
- Neto, H. C., Julia, R. M. S., e Caexeta, G. S. (2014). Ls-visiondraughts: improving the performance of an agent for checkers by integrating computational intelligence, reinforcement learning and a powerful search method. *Appl. Intell.*, vol. 41, no. 2, pages 225–250.
- Neumann, J. e Morgenstern, O. (1953). *Theory of games and economic behavior*. Princeton Univ. Press, Princeton, NJ, 3. ed. edition.
- Russell, S. e Norvig, P. (2004). *Inteligência Artificial - Uma abordagem Moderna*. Campus, 2nd edition.
- Schaeffer, J., Burch, N., Bjornsson, Y., Kishimoto, A., Muller, M., Lake, R., Lu, P., e Sutphen, S. (2007). Checkers is solved. *Science*.
- Schaeffer, J., Robert, L., Paul, L., e Martin, B. (1996). CHINOOK: The world man-machine checkers champion. *The AI Magazine*, 16(1):21–29.
- Tomaz, L. B. P., Julia, R. M. S., e Barcelos, A. R. A. (2013). Improving the accomplishment of a neural network based agent for draughts that operates in a distributed learning environment. In *2013 IEEE 14th International Conference on Information Reuse Integration (IRI)*, pages 262–269.
- Tomaz, L. B. P., Julia, R. M. S., e Duarte, V. A. (2017). A multiagent player system composed by expert agents in specific game stages operating in high performance environment. *Applied Intelligence*, pages 1–22.
- van den Herik, H., Uiterwijk, J. W., e van Rijswijck, J. (2002). Games solved: Now and in the future. *Artificial Intelligence*, 134(1):277 – 311.