

Uma Proposta de Modelagem para uso do Aprendizado por Reforço na Definição de Táticas de Combate em Jogos de Estratégia em Tempo Real

Gutenberg Pessoa Botelho Neto Clairton de Albuquerque Siebra

Universidade Federal da Paraíba, Programa de Pós-Graduação em Informática (PPGI/UFPB)
João Pessoa – PB – Brasil

Abstract

Jogos de estratégia em tempo real estão sendo cada vez mais utilizados em pesquisas de inteligência artificial por contarem com características importantes para a área, como a presença de ambientes complexos, muitas vezes dinâmicos e com múltiplos agentes. Este trabalho propõe a utilização de SARSA, uma técnica de aprendizagem de máquina, aplicada a situações de combate em jogos do gênero. O domínio do combate é fundamental na busca da vitória nestes jogos e, com a utilização da aprendizagem de máquina, busca-se permitir ao computador avaliar de maneira autônoma e dinâmica as situações que encontrar nas partidas, aprendendo com elas e mudando seu comportamento ao longo do tempo. Assim, são evitados problemas comuns existentes na inteligência artificial de jogos comerciais, que além de serem geralmente estáticas, tornando-se previsíveis para o jogador após algum tempo, são desenvolvidas com comportamentos definidos manualmente, o que é demorado e propício a erros.

Keywords: inteligência artificial, aprendizado por reforço, RTS, SARSA

Authors' contact:

gutenberg@fuze.cc
clairton@di.ufpb.br

1. Introdução

Jogos de Estratégia em Tempo Real (*Real-Time Strategy Games* – RTS) representam um gênero dos jogos eletrônicos caracterizado pela disputa de duas ou mais facções controladas por humanos ou pelo computador. As responsabilidades dos jogadores nestes jogos incluem, geralmente: coletar recursos, adquirir unidades e estruturas, explorar e expandir o território, engajar em combate com inimigos, entre outras [Aha et al. 2005; Churchill and Buro 2011; Sailer et al. 2007].

Este gênero é considerado um dos mais promissores para pesquisas de inteligência artificial (IA) em jogos, pois engloba uma variedade de problemas fundamentais da IA: planejamento adversarial em tempo real, tomada de decisão com informação incompleta, colaboração de grupos, entre outros [Buro 2004; Buro and Furtak 2003].

Mesmo com o aumento das pesquisas na área, a dificuldade de desenvolver uma IA completa e desafiadora faz com que muitos jogos comerciais não utilizem uma IA “verdadeira”, permitindo que o computador trapaceie para oferecer um maior desafio ao oponente [Dimitriadis 2009; Sailer 2007; Sandberg 2011; Walther 2006]. Além disso, a IA é geralmente estática e não-adaptativa. Assim, o computador torna-se um jogador previsível, que utiliza sempre a mesma estratégia, e os jogadores que o enfrentam são capazes de criar uma estratégia ótima para derrotá-lo [Dimitriadis 2009; Kok 2008; Ponsen 2004].

Por conta destes fatores, muitas das pesquisas recentes da IA em jogos RTS são voltadas a técnicas de aprendizagem de máquina (*Machine Learning* - ML) aplicadas a áreas diversas do jogo, buscando permitir ao computador jogar partidas de maneira inteligente, com as mesmas informações dos jogadores humanos e com a possibilidade de adaptar-se ao adversário e explorar novas estratégias.

Este trabalho propõe uma técnica para a utilização de aprendizagem de máquina voltada a situações de combate em jogos RTS. O combate é um campo interessante de pesquisa, pois conta com características complexas como a necessidade de interação e cooperação entre unidades aliadas e a presença de unidades inimigas com objetivos conflitantes, além de não ter sido tão explorado como outras situações da IA de jogos RTS (e.g., planejamento de estratégia de alto nível, coleta de recursos).

A proposta utiliza SARSA, uma técnica de uma subárea da aprendizagem de máquina chamada de aprendizado por reforço (*Reinforcement Learning* – RL), e será implementada no ambiente BWAPI, que permite o desenvolvimento e validação da IA diretamente no jogo comercial *Starcraft: Brood War*.

2. Aprendizagem de Máquina

A aprendizagem de máquina é uma área da IA que busca fazer com que os computadores possam modificar seu comportamento através do conhecimento adquirido a partir de dados diversos.

O aprendizado pode ocorrer de forma *offline*, onde todo o treinamento é realizado *a priori* e o

conhecimento do sistema não é modificado durante a execução, ou de forma *online*, onde o sistema refina continuamente o seu conhecimento do mundo a partir dos dados coletados durante sua operação, adaptando seu comportamento de acordo com os novos dados. As aplicações *online* possuem diversas vantagens em relação às *offline* [Sutton and Whitehead 1993], sendo mais robustas por permitirem a correção de erros do conjunto de treinamento original a partir dos novos conhecimentos obtidos e também facilitando a geração de dados de treinamento confiáveis e em volume adequado.

O aprendizado por reforço é uma subárea da aprendizagem de máquina, caracterizada por sistemas que recebem apenas um conjunto de entrada e buscam obter informação sobre os possíveis valores de cada entrada através de uma função de recompensa e geralmente formalizada por um Processo de Decisão de Markov (*Markov Decision Process* – MDP) [Sutton 1999]. Nesta formalização, o ambiente é representado por um estado s_t , sendo t um momento discreto de tempo no qual há interação com o ambiente, onde o agente deve escolher uma ação a_t que mudará o ambiente estocasticamente para um novo estado s_{t+1} , resultando numa recompensa numérica r_t , indicando se aquela ação específica tomada naquele estado foi boa ou não. A partir da recompensa recebida por cada entrada e após seguidas execuções, o computador passa a melhorar seu conhecimento de mundo e deve ser capaz de formular uma política definindo quais ações ele considera as melhores para os possíveis estados do ambiente. O *framework* básico do aprendizado por reforço é ilustrado na Figura 1.

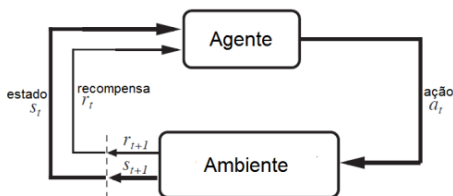


Figura 1: Framework básico do RL. Adaptado de [Sutton 1999].

Um ponto fundamental para a utilização com sucesso de técnicas de RL baseadas em MDP é a representação do estado. Como nesta formalização o espaço de busca cresce exponencialmente, é normalmente impossível representar com exatidão os estados do ambiente do problema. A representação do estado é uma abstração que deve simplificar o estado, diminuindo o espaço de busca, mas também mantendo uma dose adequada de fidelidade ao ambiente real para não prejudicar o aprendizado. Outra questão importante do RL é o dilema aproveitamento-exploração, referente à necessidade de balancear a preferência que deve ser dada a ações tomadas previamente com bons resultados pelo agente com a investigação de outras possíveis ações que possam permitir ao agente alcançar resultados melhores do que os atualmente conhecidos.

SARSA [Rummery and Niranjan 1994] é uma técnica do aprendizado por reforço, bastante popular por ser de fácil implementação e por garantir convergência para uma política ótima (i.e., que garanta a escolha da melhor ação para cada estado) sob condições apropriadas. A técnica já foi utilizada com sucesso em diversos problemas de RL e, portanto, é considerada uma escolha atraente para a investigação de problemas da área.

Nesta técnica, cada par de estado-ação (i.e., entrada e saída) é representado por um valor $Q(s,a)$, indicando a qualidade da ação a no estado s , levando em conta o estado atual e também as situações futuras. Com isto, garante-se que o sistema não terá uma visão imediatista, buscando apenas a maior recompensa atual, mas também avaliará possíveis melhorias que a ação possa trazer a longo prazo.

3. Ambiente

Uma forma de facilitar o desenvolvimento de uma IA para jogos RTS é a utilização de um ambiente apropriado para isto. O ambiente provê o jogo em si e todas as ferramentas necessárias para a modificação do comportamento do jogador durante uma partida. Isto permite que o pesquisador foque apenas na programação relacionada à IA, e não em outras áreas como o desenvolvimento do jogo propriamente dito. Para este trabalho, o ambiente escolhido foi o BWAPI (*Brood War Application Programming Interface*).

O BWAPI é um *framework* que permite a criação de módulos de IA a serem utilizados no jogo *Starcraft: Brood War*. Nele, o desenvolvedor tem acesso a todas as informações que o jogo oferece a qualquer jogador e pode ordenar que a IA execute qualquer ação que um jogador regular poderia executar durante uma partida.

A grande vantagem deste ambiente é permitir o desenvolvimento e validação da IA diretamente em um jogo RTS comercial de alta popularidade, ajudando a demonstrar a viabilidade prática dos trabalhos desenvolvidos no mesmo. Por conta disto, tornou-se a principal ferramenta utilizada em pesquisas de IA para jogos RTS.

4. Trabalhos Relacionados

A importância da pesquisa da IA em jogos RTS já era mencionada e detalhada em dois artigos de 2003 e 2004 [Buro 2004; Buro and Furtak 2003], citando o fato dos jogos do gênero envolverem muitos problemas fundamentais da IA.

O problema do combate em jogos RTS já foi tratado com técnicas variadas, como a utilização de árvores de jogos [Laursen and Nielsen 2005], que apesar de obter resultados satisfatórios teve dificuldades por conta da complexidade do problema;

com planejamento [Balla and Fern 2009], que obteve bons desempenhos em diversos cenários de testes; com campos potenciais [Sandberg 2011], que também utilizou o BWAPI e conseguiu derrotar a IA padrão do jogo; com autonomia orientada a objetivos [Jaidee et al. 2011], também com resultados interessantes em dois cenários diferentes e com melhoria de desempenho por conta de aprendizado ao longo do tempo.

5. Proposta

O trabalho proposto visa à aplicação de uma técnica *online* de ML para controlar unidades de um jogo RTS durante o combate com outra facção. Assim, espera-se que o computador seja capaz de aprimorar sua estratégia no decorrer das partidas, tornando-se cada vez mais difícil de ser derrotado. Outro resultado da utilização da ML deve ser a capacidade de adequar-se aos oponentes após disputar uma quantidade suficiente de partidas, evoluindo sua estratégia de forma a diminuir a sua previsibilidade e tentar combater o oponente de maneira mais eficiente.

Além do ambiente escolhido para o desenvolvimento, existem outros aspectos fundamentais da proposta que serão detalhados a seguir. São eles: representação do estado, possíveis ações dos agentes, detalhes da utilização da SARSA, além de uma ideia inicial de como será realizada a validação futura do trabalho.

5.1 Representação do Estado

Como dito na Seção 2, a representação do estado é a forma que o agente tem de perceber o mundo ao seu redor e é fundamental para o aprendizado, pois cada conhecimento obtido pelo agente é associado a um dos estados do espaço de busca. Assim, a representação deve permitir que o agente compreenda a situação atual da partida e deve oferecer informação suficiente para que as melhores ações sejam escolhidas.

Como não há abstração padrão de estados para a situação de combate em jogos RTS, uma das contribuições deste trabalho é a definição de uma representação para o agente nesta situação. A representação conta com seis variáveis possuindo três valores cada, totalizando 729 possíveis estados em que o agente pode estar situado. As variáveis usadas na representação, assim como seus possíveis valores, são:

- **Vida** atual do agente. Valores: baixa (menos de 1/3 restante), média (entre 1/3 e 2/3 restante), alta (mais de 2/3 restante);
- Quantidade de **inimigos que podem ser atacados** pelo agente a partir da sua posição atual, sem exigir movimentação extra. Valores: 0, 1, 2+;
- Quantidade de **inimigos que podem atacar** o agente em sua posição atual. Valores: 0, 1, 2+;

- Quantidade de **inimigos atacando** atualmente o agente. Valores: 0, 1, 2+;
- Quantidade de **aliados próximos** ao agente, que estão no raio de vizinhança do mesmo, definido pelo jogo de acordo com o alcance da arma da unidade. Valores: 0, 1, 2+;
- **Vida média** dos inimigos que estão próximos ao agente. Valores: baixa (menos de 1/3 restante), média (entre 1/3 e 2/3 restante), alta (mais de 2/3 restante).

Com esta representação, acredita-se que o agente será capaz de obter um sentido geral da sua situação na batalha, sem trazer prejuízos ao seu processo de decisão de ação a ser tomada.

5.2 Ações

Como as ações também influenciam no tamanho do espaço de busca, é necessário fazer uma abstração das possíveis ações assim como na representação de estado.

Para o trabalho atual, a proposta é a de ter ações que definam uma postura de alto nível para o agente, deixando a implementação de baixo nível de fora do aprendizado. Como sugestão inicial tem-se as seguintes posturas que podem ser tomadas:

- **Ataque:** postura ofensiva, buscando causar o máximo de dano possível ao oponente;
- **Defesa:** nesta postura o agente está mais preocupado em sobreviver, mas sem fugir da batalha corrente;
- **Recuo:** o agente assume uma postura de fuga para garantir sua sobrevivência, abdicando momentaneamente de ações ofensivas.

5.3 Técnica

O sistema foi pensado de maneira a utilizar conceitos conhecidos do RL já aplicados com sucesso a diversos ambientes. Também houve a preocupação de propor uma utilização simples, tendo em vista o fato de que o tempo de processamento destinado a IA para combate não será exclusivo, mas dividido com o tempo cedido aos outros aspectos da simulação do jogo e, portanto, exigirá execução rápida da computação necessária.

O trabalho utilizará uma tabela de valores Q única para todas as unidades de um mesmo tipo. Isto é útil por permitir que cada agente utilize o conhecimento adquirido pelos outros agentes semelhantes a ele, acelerando o processo de aprendizado. Outra vantagem é que, como normalmente cada jogador utiliza em um ataque diversas unidades de cada tipo, serão aliviadas as necessidades de memória do sistema na maioria dos casos.

Apesar de SARSA não ser uma técnica multi-agentes, é fundamental que o sistema tenha um sentido

de cooperação de agentes do mesmo time, de forma a permitir que estes trabalhem de maneira conjunta e sem ações conflitantes. Uma forma de obter isto é a partir da função de recompensa utilizada na técnica. A proposição de uma fórmula nestes termos segue como contribuição adicional do trabalho:

$$\left(\sum_{i \in I} h_{i,t} - \sum_{i \in I} h_{i,t+1} \right) + (100 * n_{dt}) - [(v_{t+1} - v_t) + (100 * m_{dt})]$$

Figura 2: Função de recompensa

onde I representa o conjunto de inimigos próximos do agente no tempo t , $h_{i,t+1}$ e $h_{i,t}$ representam a vida da unidade i nos tempos $t+1$ e t , n_{dt} representa a quantidade de inimigos mortos entre os tempos t e $t+1$, v_{t+1} e v_t representam a vida do agente nos tempos $t+1$ e t e m_{dt} vale 1 se o agente tiver morrido entre os tempos t e $t+1$ e vale 0, caso contrário. Não há indicação do agente causador de dano aos inimigos. Assim, valoriza-se a ação conjunta ofensiva dos aliados próximos, mas penaliza-se o agente pelo seu dano individual. Com esta abordagem híbrida, espera-se dar ao agente *feedback* suficiente sobre a sua ação tanto em relação a seu desempenho individual quanto ao de sua equipe.

5.4 Validação

A validação se dará com partidas disputadas entre a IA proposta e a IA padrão do jogo em mapas diversos, específicos para combate (i.e., sem base ou recursos a coletar), com configurações distintas de equipes. O objetivo será mostrar a capacidade de evoluir o comportamento do computador ao longo do tempo com a técnica de ML, de modo que possa eventualmente aprender o suficiente para derrotar consistentemente seu adversário nas situações avaliadas.

6. Conclusão e Trabalhos Futuros

Atualmente, o sistema proposto já se encontra em desenvolvimento, com o algoritmo da técnica SARSA implementado e integrado ao ambiente BWAPI, assim como a obtenção do estado atual por parte de cada agente e de acordo com a representação sugerida. O foco no futuro próximo é na definição e execução dos testes a serem utilizados para validação, de forma a poder confirmar a viabilidade da técnica descrita.

Com a aplicação de sucesso da proposta, espera-se comprovar a possibilidade de utilização de técnicas clássicas da IA e ML em jogos comerciais, que possam aumentar a qualidade do jogo e a percepção de valor dos jogadores. O fato do problema escolhido – combate em jogos RTS – possuir muitas das características mais complexas que os jogos eletrônicos podem apresentar para a IA serve para aumentar a importância da investigação. Outra contribuição esperada é a de mostrar o resultado da utilização de SARSA, já aplicada com sucesso em diversas situações, ao problema do combate em jogos RTS.

Referências Bibliográficas

- AHA, D. W., MOLINEAUX, M. AND PONSEN, M., 2005. Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game. In: *Proceedings of the Sixth International Conference on Case-Based Reasoning*, 5-20.
- BALLA, R-K. AND FERN, A., 2009. UCT for Tactical Assault Planning in Real-Time Strategy Games. In: *Proceedings of the International Joint Conference on Artificial Intelligence*, 40-45.
- BURO, M., 2004. Call for AI Research in RTS Games. *AI in Games Workshop, AAAI*
- BURO, M. AND FURTAK, T., 2003. RTS Games as Test-Bed for Real-Time Research. *Invited Paper at the Workshop on Game AI, JCIS*, 481-484.
- CHURCHILL, D. AND BURO, M., 2011. Build Order Optimization in Starcraft. *AI and Interactive Digital Entertainment Conference (AIIDE)*.
- DIMITRIADIS, V. K., 2009. Reinforcement Learning in Real Time Strategy Games Case Study on the Free Software Game Glest. *Technical University of Crete*.
- JAIDEE, U., MUÑOZ-AVILA, H. AND AHA, D. W., 2011. Case-Based Learning in Goal-Driven Autonomy Agents for Real-Time Strategy Combat Tasks. *Case-Based Reasoning in Computer Games: Papers from the ICCBR Workshop*.
- KOK, E., 2008. *Adaptive Reinforcement Learning Agents in RTS Games*. M.Sc. Thesis, University Utrecht.
- LAURSEN, R. AND NIELSEN, D., 2005. *Investigating Small Scale Combat Situations in Real Time Strategy Computer Games*. M.Sc. Thesis, University of Aarhus.
- PONSEN, M., 2004. *Improving Adaptive Game AI with Evolutionary Learning*. M.Sc. Thesis, Delft University of Technology.
- RUMMERY, G. A. AND NIRANJAN, M., 1994. *On-Line Q-Learning Using Connectionist Systems*. Technical Report no. 166, Engineering Department, University of Cambridge.
- SAILER, F., BURO, M. AND LANCTOT, M., 2007. Adversarial Planning Through Strategy Simulation. *IEEE Symposium on Computational Intelligence and Games*, 80-87.
- SANDBERG, T. W., 2011. *Evolutionary Multi-Agent Potential Field Based AI Approach for SSC Scenarios in RTS Games*. M.Sc. Thesis, IT University.
- SUTTON, R. S., 1999. *Reinforcement Learning*. The MIT Encyclopedia of Cognitive Sciences, MIT Press.
- SUTTON, R. S. AND WHITEHEAD, S. D., 1993. Online Learning with Random Representations. In: *Proceedings of the International Conference on Machine Learning*, 314-321.
- WALTHER, A., 2006. *AI for Real-Time Strategy Games*. M.Sc. Thesis, IT-University of Copenhagen.