

Otimização de Partidas em Dota 2: Balanceamento e Maximização de Rankings

João Pedro Lobato de Pinho¹, Vinícius Henrique Giovanini¹

¹ICEI - Pontifícia Universidade Católica de Minas Gerais (PUC-MG)
Belo Horizonte, Minas Gerais - Brasil

joao.pinho.1275438@sga.pucminas.br, vgiovanin@sga.pucminas.br

Abstract. Matchmaking in competitive games like Dota 2 is a significant challenge affecting player experience. This study tackles the problem using a public dataset of player statistics and applies three optimization methods: Genetic Algorithm, Branch and Bound (B&B), and Simulated Annealing. The approaches aim to form balanced teams by minimizing the average skill difference (TrueSkill_{mu}) while accounting for uncertainty (TrueSkill_{sigma}). Results indicate that B&B achieves the most accurate solutions but at a high computational cost, whereas Genetic Algorithm and Simulated Annealing offer faster solutions with slightly lower quality. This research highlights the strengths and limitations of each method, providing valuable insights for matchmaking systems.

Resumo. O balanceamento de partidas (matchmaking) em jogos competitivos, como Dota 2, é um desafio que influencia diretamente a experiência dos jogadores. Este trabalho aborda esse problema utilizando um dataset público de jogadores e aplicando três métodos de otimização: Algoritmo Genético, Branch and Bound (B&B) e Simulated Annealing. As abordagens focam em formar equipes equilibradas minimizando a diferença de habilidade média (TrueSkill_{mu}) e considerando a incerteza (TrueSkill_{sigma}). Os resultados mostraram que o B&B apresenta soluções mais precisas, mas com alto custo computacional, enquanto o Algoritmo Genético e o Simulated Annealing oferecem soluções rápidas, mas com qualidade inferior. Esse estudo destaca as vantagens e limitações de cada método, fornecendo insights para aplicações em sistemas de matchmaking.

1. Introdução

O crescente interesse por jogos online competitivos, como Dota 2, tem gerado desafios relacionados ao balanceamento de partidas, conhecido como matchmaking. Esse processo é fundamental para garantir que jogadores enfrentem adversários de habilidade semelhante, proporcionando uma experiência justa e divertida. Para abordar esse problema, utilizou-se um dataset público que reúne informações detalhadas sobre o desempenho de jogadores de Dota 2, incluindo métricas como TrueSkill_{mu} (habilidade média) e TrueSkill_{sigma} (incerteza da habilidade).

Neste trabalho, exploraram-se técnicas avançadas de otimização para resolver o problema de matchmaking, analisando e comparando três métodos distintos: Branch and Bound, Algoritmo Genético e Simulated Annealing. Cada um desses métodos apresenta características únicas em termos de eficiência computacional, qualidade das

soluções e aplicabilidade ao balanceamento de equipes. Essa abordagem busca minimizar a diferença de habilidade média entre as equipes e equilibrar as incertezas associadas, criando partidas mais competitivas e alinhadas com as expectativas dos jogadores.

2. Revisão de literatura

Diversas técnicas de otimização têm sido amplamente empregadas para resolver problemas complexos em diferentes domínios. O Branch and Bound (B&B) é um método exato que utiliza uma abordagem sistemática de divisão e limitação para encontrar soluções ótimas, explorando apenas as sub-regiões do espaço de busca que possuem potencial para conter a melhor solução. Em contraste, o Simulated Annealing (SA) é um método estocástico inspirado no processo físico de recozimento de metais, permitindo a aceitação de soluções temporariamente piores para escapar de mínimos locais e buscar um ótimo global. Já os Algoritmos Genéticos (AG), baseados nos princípios da evolução natural, utilizam operadores como seleção, cruzamento e mutação para explorar populações de soluções, combinando a exploração global e local de maneira eficiente.

Esses métodos são amplamente utilizados para abordar o matchmaking em jogos online competitivos, um tema explorado em diversos estudos recentes. O artigo [Sapienza et al. 2019], propõe um sistema de recomendação de partidas de Dota 2 que utiliza dados fornecidos pelo Opendota, uma API pública que disponibiliza registros de partidas e jogadores. Esse sistema aborda diversos problemas, como a avaliação da influência dos companheiros de equipe, o desenvolvimento de um sistema de recomendação de partidas ranqueadas e a previsão de transferência de habilidade. Tais objetivos são alcançados utilizando redes neurais profundas com autoencoders, implementados especificamente para trabalhar com redes direcionadas de co-jogos.

Já o trabalho de [Cristian et al. 2021] apresenta o conceito de Next Level Matchmaking, um sistema voltado para jogos do gênero MOBA, como Dota 2, Mobile Legends e League of Legends. Este sistema busca superar limitações dos algoritmos de matchmaking tradicionais, que se baseiam apenas no rank dos jogadores, incorporando critérios como desempenho individual em métricas de jogo (ex.: dano por minuto, ouro por minuto, mortes, assistências) e comportamento em partidas. O trabalho utiliza o algoritmo K-Nearest Neighbors (KNN) para classificar jogadores e identificar padrões anômalos, como jogadores experientes em contas novas. O sistema é projetado para equilibrar as equipes com base em características complementares, minimizando desbalanceamentos em tempo real.

3. Metodologia

O trabalho de [Sapienza et al. 2019] utilizou um dataset do OpenDota contendo partidas registradas entre janeiro e dezembro de 2015, com aproximadamente 3,5 milhões de jogos. Contudo, o OpenDota expandiu sua base de dados ao longo dos anos. Em 2016, foi disponibilizado um novo dump que abrange partidas realizadas no período de março de 2011 a março de 2016, totalizando cerca de 1,2 bilhões de partidas. Através dessa atualização o dataset de 2015 foi preterido. Neste trabalho utilizou-se o dataset de 2015 encontrado no kaggle, porém em uma dimensão menor, com 800.000 registros. Para diminuir o tempo de execução foi selecionado jogadores com mais de 5 partidas diminuindo para 250.000 registros.

Neste trabalho, a função objetivo a ser maximizada é definida como na fórmula 1, onde *TrueSkill_Mu* representa a estimativa média de habilidade dos jogadores, enquanto *TrueSkill_Sigma* reflete a incerteza associada a essa habilidade. Essa função foi escolhida para balancear as equipes de forma a maximizar a habilidade média dos jogadores ao mesmo tempo em que penaliza equipes com alta incerteza, promovendo uma maior confiabilidade na avaliação do desempenho. As restrições impostas incluem a seleção de jogadores que participaram de pelo menos 5 partidas e o uso de no máximo 120 mil registros, de forma a otimizar o tempo de processamento e a viabilidade do algoritmo em cenários reais.

O algoritmo genético foi implementado em Python, sem o uso de bibliotecas externas, o que permitiu a criação e personalização de métodos específicos para o problema de matchmaking. O método de mutação foi adaptado, pois, tradicionalmente, ele altera aleatoriamente o vetor de cromossomos. No entanto, no contexto deste problema, o vetor de cromossomos deve indicar exatamente 10 posições como verdadeiras, o que corresponde aos jogadores selecionados para uma partida de Dota 2 (5 contra 5). Para resolver isso, foi desenvolvido um método de mutação que modifica o vetor, garantindo que ao final apenas 10 posições sejam selecionadas.

O processo de avaliação utiliza a fórmula abaixo para calcular o score da partida. O score é obtido pela média de *TrueSkill_mu*, subtraída pela média de *TrueSkill_sigma* multiplicada por 0,5 valor utilizado para penalizar o score final. Isso ocorre porque o *TrueSkill_sigma* reflete a incerteza na habilidade do jogador, e sua penalização é importante para garantir que o desempenho tenha maior peso na avaliação final.

$$\text{score} = \left(\frac{\sum_{i=1}^n \text{TrueSkill_mu}}{n} \right) - 0.5 \times \left(\frac{\sum_{i=1}^n \text{TrueSkill_sigma}}{n} \right) \quad (1)$$

Outra abordagem utilizada neste trabalho foi o algoritmo Branch and Bound (B&B), aplicado para selecionar as melhores combinações de jogadores. Este método divide o problema em subproblemas menores e utiliza critérios de poda para descartar rapidamente combinações que não podem produzir resultados melhores do que os já encontrados.

O B&B começa ordenando os jogadores de acordo com o score individual, calculado conforme a fórmula 1, para priorizar os mais habilidosos. Em seguida, ele explora combinações de jogadores, avaliando quais delas formam os times mais balanceados. Para cada nó na árvore de busca (representando uma combinação parcial de jogadores), o algoritmo verifica se há potencial para encontrar uma solução melhor que a atual, com base nos scores acumulados. Caso não haja, o ramo é descartado (poda), economizando tempo computacional. A poda é realizada comparando a soma parcial dos scores dos jogadores já selecionados com a soma possível dos jogadores restantes. Se a diferença entre esses valores exceder a melhor solução encontrada até o momento, aquele ramo é abandonado.

A última abordagem implementada foi o algoritmo de Simulated Annealing. Nesse método, a cada iteração, o algoritmo realiza uma pequena modificação na composição do time, substituindo aleatoriamente um jogador, e calcula o score da nova

formação. Sua condição de parada foi determinada quando a temperatura chegar a 0.1. Caso o novo time apresente um score melhor, ele é aceito como a nova solução. No entanto, esse algoritmo também permite explorar soluções com scores ligeiramente piores, utilizando uma estratégia probabilística baseada na fórmula 2. Essa fórmula calcula o expoente da diferença entre os scores dividido pela temperatura da iteração atual. Se o valor resultante for maior que um número aleatório entre 0 e 1, o algoritmo opta por explorar essa solução, mesmo que ela seja inferior. Essa característica torna-se mais frequente em temperaturas altas, aumentando a probabilidade de explorar soluções sub-ótimas. Por outro lado, à medida que a temperatura diminui, a probabilidade de aceitar soluções piores reduz significativamente, refletindo o conceito de resfriamento gradual. Essa dinâmica balanceia exploração e refinamento, permitindo ao algoritmo escapar de mínimos locais e buscar uma solução globalmente ótima.

$$P(\text{aceitar}) = e^{\frac{\Delta \text{score}}{\text{temp}}} \quad (2)$$

$$P(\text{aceitar}) > \text{random}(0, 1)$$

A função de avaliação (score) considera a média dos valores conforme descrito na fórmula 1. A temperatura inicial e a taxa de resfriamento são ajustadas para controlar a exploração e a convergência do algoritmo. O algoritmo realiza várias iterações, onde a cada iteração, a solução mais próxima de um ótimo global é procurada, e as soluções inferiores podem ser aceitas dependendo da temperatura, permitindo uma busca mais abrangente pelo espaço de soluções.

4. Estudo experimental

Diversos testes foram realizados para ajustar os parâmetros dos algoritmos utilizados, a fim de avaliar a eficiência e a qualidade das soluções. Para o algoritmo genético, na qual demorou 5.73 minutos, foi definido um tamanho de população de 20, representando o número de possibilidades de partidas geradas, e 100 gerações, com uma taxa de mutação de 0.1. A escolha de uma população reduzida foi motivada pela alta demora na execução do algoritmo, uma vez que o processo de avaliação e evolução de soluções em um espaço de busca grande exige tempo significativo de computação. O score obtido com o algoritmo genético foi 27.15, como ilustrado na figura 1.

Em paralelo, o algoritmo de Simulated Annealing demorou 2.43 minutos e foi utilizado com um parâmetro de temperatura inicial de 1000 e um cooling rate de 0.95. A temperatura inicial de 1000 permite que o algoritmo explore mais amplamente o espaço de soluções, aceitando soluções piores nas primeiras iterações, o que pode possibilitar a descoberta de soluções mais promissoras em estágios posteriores. À medida que a temperatura diminui, o algoritmo se torna mais seletivo, favorecendo soluções melhores e mais estáveis à medida que avança no processo de otimização. O score obtido com o Simulated Annealing foi 32.82.

A execução do Branch and Bound, por sua vez, durou 147 minutos. Esse algoritmo se baseia em uma estratégia de busca exaustiva, mas com a vantagem de poda ramos do espaço de soluções que não podem melhorar a melhor solução já encontrada, o que permite uma busca mais eficiente. Embora o Branch and Bound seja teoricamente capaz de

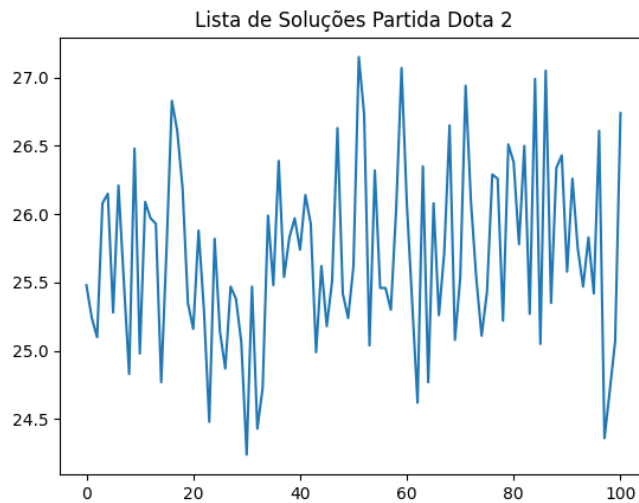


Figura 1. Soluções durante 100 gerações - Algoritmo Genético

encontrar a solução ótima global, seu desempenho depende da estrutura do problema e da qualidade das funções de poda. No caso deste estudo, a grande quantidade de jogadores e a necessidade de explorar múltiplas combinações de forma precisa resultaram em um tempo de execução relativamente longo, evidenciando o custo computacional dessa abordagem. Contudo, a vantagem do Branch and Bound é que ele garante, ao final, a ótima solução local para o problema, já que explora de forma sistemática as possíveis soluções sem recorrer a aproximações ou heurísticas. O score obtido com o Branch and Bound foi 38.23.

5. Conclusão

Conforme os testes realizados, foi possível comparar os três algoritmos utilizados: Genético, Branch and Bound (B&B) e Simulated Annealing. O algoritmo B&B, apresentou um tempo de execução significativamente maior, levando cerca de 147 minutos para processar o dataset, enquanto os outros algoritmos, com limites predefinidos de iterações ou gerações, executaram em menos de 10 minutos. Esse maior tempo de execução do B&B resulta de sua abordagem exaustiva, garantindo a busca pela solução ótima.

O algoritmo genético, por outro lado, se mostrou mais rápido, mas menos eficiente em encontrar a solução ótima. Isso ocorre devido à limitação imposta pelo tamanho da população (número de soluções avaliadas simultaneamente) e ao processo de mutação, que utiliza probabilidades para alterar cromossomos, introduzindo aleatoriedade sem garantir a busca direcionada pela melhor solução.

Já Simulated Annealing demonstrou ser uma alternativa interessante para problemas onde o equilíbrio entre qualidade da solução e tempo de execução é essencial, na qual explora soluções suavemente piores em busca da solução ótima, através do método probabilístico adaptado.

Referências

- Cristian, J., Nabel, F. M., Wisnu, C. A., and Wulandari, A. (2021). Next level matchmaking. In *Proceedings of the International Joint Conference on Science and Engineering 2021 (IJCSE 2021)*, pages 655–659. Atlantis Press.
- Sapienza, A., Goyal, P., and Ferrara, E. (2019). Deep neural networks for optimal team composition. *Frontiers in Big Data*, 2.