



Lucas Marsol Vieira

Otimização de equipes em League of Legends utilizando Algoritmos Genéticos Multiobjetivo

Recife

Maio de 2022

Lucas Marsol Vieira

Otimização de equipes em League of Legends utilizando Algoritmos Genéticos Multiobjetivo

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE
Departamento de Estatística e Informática
Curso de Bacharelado em Sistemas de Informação

Orientador: Cícero Garrozi

Recife
Maio de 2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- V658o Vieira, Lucas Marsol
Otimização de equipes em League of Legends utilizando Algoritmos Genéticos Multiobjetivo / Lucas Marsol Vieira.
- 2022.
20 f. : il.
- Orientador: Cicero Garrozi.
Inclui referências.
- Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em
Sistemas da Informação, Recife, 2022.
1. League of Legends. 2. Algoritmos Genéticos. 3. Algoritmos de Otimização Multiobjetivo. I. Garrozi, Cicero,
orient. II. Título

Lucas Marsol Vieira

Otimização de equipes em League of Legends utilizando Algoritmos Genéticos Multiobjetivo

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 03 de Junho de 2022.

BANCA EXAMINADORA

Cícero Garrozi
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Hansenclever de França Bassani
Centro de informática
Universidade Federal de Pernambuco

Otimização de equipes em League of Legends utilizando Algoritmos Genéticos Multiobjetivo

Lucas Marsol Vieira ¹, Cícero Garrozi ²

¹Departamento de Estatística e Informática – Universidade Federal Rural de Pernambuco
Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

²Department of Statistics and Informatics - Federal Rural University of Pernambuco
Recife, Brazil.

[lucasmarsolv@hotmail.com, cicero.garrozi@ufrpe.br]

Resumo. *O League of Legends, jogo da categoria Multiplayer Online Battle Arena ou MOBA (como é conhecido popularmente), segue sendo um dos jogos eletrônicos que mais pagam em premiação no mundo. Esta categoria se baseia em dois times que se enfrentam em um mapa simétrico com o objetivo de destruir a base adversária. Um dos principais pontos nesse estilo de jogo e mais especificamente no League of Legends é a etapa de seleção de personagens (também conhecidos como campeões), visto que irá guiar a estratégia de cada equipe. Nesta etapa, os jogadores selecionam quais personagens irão utilizar dentro da partida, onde cada personagem possui características e habilidades distintas dos demais. Por envolver diversos fatores no processo de seleção, é considerado um problema complexo que pode ser resolvido com técnicas de busca e inteligência artificial para encontrar as melhores soluções. Neste projeto uma nova abordagem através de Algoritmos Evolucionários Multiobjetivo (MOEA) é apresentada para geração de equipes no jogo. Com o objetivo de estimar a qualidade das equipes geradas, foi conduzida uma pesquisa com um grupo de jogadores. Foram atingidos resultados significantes com essa abordagem, obtendo uma avaliação média de 4.5 para um total de 5 pontos.*

Abstract. *League of Legends, a game of the Multiplayer Online Battle Arena or MOBA category (as it is popularly known), continues to be one of the highest paying electronic games in the world. This category is based on two teams that face each other on a symmetrical map with the objective of destroying the opposing base. One of the main points in this style of play and more specifically in League of Legends is the character selection stage (also known as champions), as it will guide the strategy of each team. In this step, players select which characters they will use within the game, where each character has characteristics and abilities that are different from the others. As it involves several factors in the selection process, it is considered a complex problem that can be solved with search techniques and artificial intelligence to find the best solutions. In this project, a new approach through Multi-Objective Evolutionary Algorithms (MOEA) is presented to generate teams in the game. In order to estimate the quality of the generated teams, a survey was conducted with a group of players. Significant results were achieved with this approach, obtaining an average rating of 4.5 for a total of 5 points.*

1. Introdução

O cenário de jogos eletrônicos vem crescendo em números expressivos nos últimos anos, mostrando uma grande adesão tanto de jogadores e entusiastas de jogos como também de empresas interessadas em divulgar suas marcas e fazer parte de uma indústria transformadora.

Nos anos 2000, uma nova categoria de jogos começou a se formar, de forma inusitada e bastante curiosa, sendo hoje uma das categorias mais populares no mundo, o Multiplayer Online Battle Arena, também conhecido popularmente como MOBA. Diferentemente de outros gêneros, o estilo MOBA conta com dois times contendo cinco jogadores cada, que se enfrentam em um mapa simétrico até um dos times conseguir destruir a base adversária. Conhecida por jogos como Warcraft, Dota e League of Legends (alvo de estudo neste artigo), consolidados há anos, com milhares de seguidores e com premiações para o cenário competitivo que alcançam valores impressionantes.



Figura 1. Mapa do Dota

A popularidade dos jogos MOBA é de grande importância para a consolidação e divulgação dos eSports como categoria. De acordo com o site eSports earnings [6], no ano de 2021, o Dota foi o jogo com mais premiações, totalizando 206 milhões de dólares em 4340 torneios. Ficando um pouco para trás, mas também com números impressionantes, o League of Legends ficou em sétimo lugar, com quase 8 milhões de dólares em premiação. No Brasil, com o notório sucesso e visibilidade dessas modalidades, canais de televisão começaram a transmitir grandes campeonatos e clubes de futebol adentraram no cenário de eports (inclusive em jogos MOBA), como o Cruzeiro, Santos, Flamengo e outros.

Os jogos da categoria MOBA oferecem vários personagens (também conhecido como campeões) com habilidades diferentes e que podem impactar a partida de diversas formas. Além disso, a maioria dos jogos oferece uma dinâmica de compra de itens (comprados com as finanças levantadas durante a partida) e aprimoramento das habilidades do personagem.

Como dito anteriormente, os times se enfrentam em um modelo de cinco contra cinco, onde as partidas podem levar de 15 minutos, normalmente quando uma das equipes comete muitos erros, até mais de 1 hora em partidas equilibradas (salvo raras excessões). Nesse processo, o jogo passa por várias etapas: O early game (início do jogo onde os personagens estão fracos e acumulando recursos), mid game (momento intermediário da

partida) e late game (momento onde os personagens atingem seu potencial máximo de itens e habilidades).

Ao contrário do que se pode esperar em um primeiro momento, existem outros elementos que complementam a partida: robôs automatizados que lutam ao lado dos personagens e estruturas de defesa que funcionam como obstáculos para o avanço de ambas as equipes. Após abater essas estruturas de defesa, os times conseguem avançar para a base inimiga, contendo a principal defesa e peça principal do jogo (como um rei no jogo de xadrez). A equipe que primeiro destruiu essa estrutura vence a partida.

No geral, uma partida pode ser definida com base em diversos fatores, assim como a maioria dos esportes tradicionais: estratégia, habilidade individual dos jogadores e especificamente nos jogos MOBA, os personagens escolhidos por esses jogadores. Essa última etapa é realizada antes do jogo começar, definindo quais campeões cada time irá utilizar, guiando a forma de jogar das equipes durante a partida.

A etapa de seleção de campeões pode ser vista como um problema complexo, visto que os jogadores devem levar em conta os pontos fracos e fortes de cada campeão, as escolhas do time adversário e a estratégia definida pelo time. Uma boa montagem de times requer a combinação de campeões de tal forma que eles maximizem determinados atributos para uma estratégia e não possuam muitas fraquezas.

O artigo em questão trata a seleção de campeões como um problema de otimização. Um problema de otimização por sua vez, é um problema de encontrar a melhor solução de todas as soluções viáveis. A maioria dos problemas de otimização envolve grandes quantidades de soluções e restrições complexas que tornam uma atividade impossível de ser realizada manualmente, mas podem ser tratadas usando técnicas baseadas em busca.

Atualmente o League of Legends (alvo de estudo neste trabalho) possui 158 personagens disponíveis. Analisar todas as composições possíveis torna-se um processo computacionalmente intratável. Ao todo seriam aproximadamente 769 milhões de possibilidades possíveis. Adiciona-se ainda o fato que novos campeões são adicionados no jogo periodicamente, elevando a complexidade do problema de forma significativa.

Este artigo apresenta uma nova abordagem automatizada para composição de equipes em League of Legends, combinando atributos dos campeões e estratégias de jogo por meio de uma técnica meta-heurística. Essa técnica é conhecida como Algoritmo Genético (AG) e é guiada por uma função objetivo para avaliar as soluções candidatas.

2. Trabalhos Relacionados

Desenvolver algoritmos inteligentes para geração de composições é algo bastante inovador e de grande impacto para a indústria de jogos eletrônicos. Uma série de trabalhos relacionados à indústria de jogos eletrônicos, mais especificamente a jogos do tipo Multiplayer Online Battle Arena podem ser encontrados, porém, apenas uma pequena parcela refere-se a geração de composições nesse tipo de jogos.

Um trabalho semelhante e que serviu de inspiração para este projeto [2], criado por alunos da Universidade Federal do Paraná, atingiu resultados expressivos na geração de composições, com valores de aptidão dos modelos gerados acima de 90%. As composições foram geradas através de técnicas de busca com Algoritmos Genéticos,

avaliando-as de acordo com as 3 estratégias comumente utilizadas no jogo: Poke, Team Fight e Hard Engage. Essas estratégias serão explicadas com mais detalhes na seção 3.1.1.

Para validar a efetividade das composições geradas, foram realizados testes utilizando 3 configurações de parâmetros no Algoritmo Genético, para as 3 estratégias descritas. A configuração que resultou em melhores resultados de aptidão gerou as composições com 30 soluções candidatas, probabilidade de mutação de 0.7 e um total de 1000 gerações. Os resultados obtidos para as estratégias de Poke, Team Fight e Hard Engage foram 94,62%, 95,10% e 97,48%, respectivamente.

O contexto deste trabalho introduz uma nova abordagem para a geração de composições, utilizando Algoritmos Genéticos Multiobjetivo. Com isso, os atributos selecionados para cada estratégia serão analisados e avaliados individualmente. As soluções encontradas, também conhecidas por soluções ótimas de Pareto (descritas na seção 3.2.2), descrevem os melhores trade-offs (não pode ser melhorada em relação a um objetivo sem a piora de ao menos um outro objetivo) dos atributos selecionados para as estratégias.

3. Materiais e Métodos

3.1. League of Legends

League of Legends é um jogo do estilo multiplayer online battle arena gratuito, também conhecido como MOBA, que é jogado no PC. Desde seu lançamento em 2009, o League of Legends cresceu e se tornou um dos jogos mais populares do mundo. A desenvolvedora, Riot Games, estima que o jogo atraia cerca de 8 milhões de usuários simultâneos todos os dias, com mais de 100 milhões de jogadores ativos a cada mês. O League of Legends tem um cenário competitivo global, com o jogo desempenhando um grande papel no desenvolvimento do e-sport em um fenômeno mundial.

Um dos motivos para a escolha do League of Legends como alvo de estudo neste artigo, além das possibilidades de aplicações, é a disponibilidade de dados oferecida pelo jogo. Utilizando serviços oferecidos pelo jogo é possível ter dados sobre jogadores, partidas ou campeões.

3.1.1. Funcionamento do jogo

League of Legends possui mais de um modo de jogo. O principal deles, mais utilizado e alvo de estudo deste trabalho é o Summoner's Rift. Nesta modalidade, 2 (duas) equipes formadas de por 5 (cinco) jogadores cada disputam uma partida que dura, em média, entre 20 (vinte) e 40 (quarenta) minutos por jogo. O tempo de uma partida não é fixo porque ela só acaba quando uma equipe (vencedora) destrói o Nexus da equipe inimiga. Até chegar ao Nexus, o time precisa destruir diversas outras estruturas dispostas no território inimigo. A equipe precisa destruir várias outras estruturas colocadas em território inimigo.

No League of Legends, os 5 jogadores são divididos em 4 rotas (figura 2), onde cada um deles exerce uma função diferente. Cada função e sua respectiva rota é descrita da seguinte forma [10]:

- **Topo:** O jogador dessa rota fica na parte superior do mapa, o próprio Topo, e geralmente optam por bonecos com muita vida, que são capazes de resistir por mais tempo nas lutas.

- Caçador: O jogador ficará na selva e tem como dever garantir objetivos, que abordaremos adiante, e ajudar os outros jogadores a vencerem em suas rotas.
- Meio – rota geralmente destinada para que um dos jogadores seja capaz de carregar a sua equipe ao “explodir” os adversários com bastante dano. Como o próprio nome função denuncia, o jogador ficará na rota do meio.
- ADC ou atirador: Jogador que geralmente será o responsável por causar mais dano por segundo nas lutas. Ele ficará no Bot;
- Suporte: Jogador que deverá proteger os seus aliados ou até mesmo atuar como um dos criadores de jogadas. Durante a fase de rotas, apesar de se movimentar pelo mapa, o jogador começa no Bot.

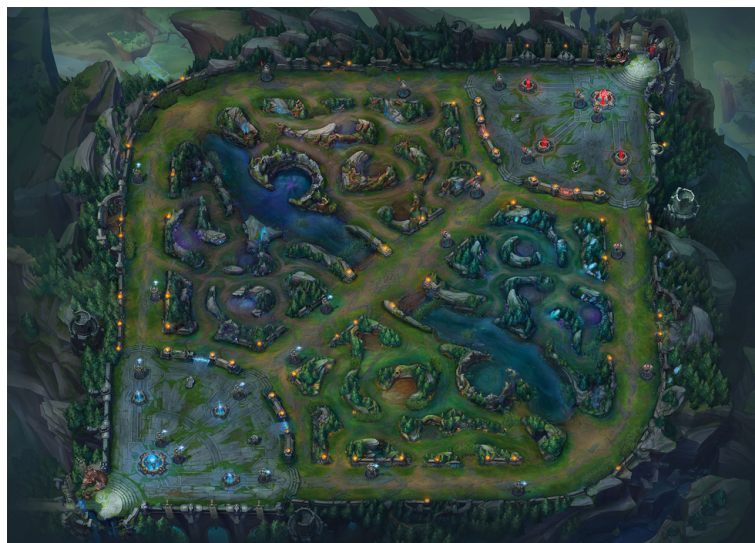


Figura 2. Summoner's Rift: Mapa do League of Legends

Antes do jogo iniciar, cada jogador deve selecionar um personagem, sendo uma das etapas mais importantes da partida. Esses personagens, por sua vez, são conhecidos e chamados de campeões dentro do jogo. Atualmente o jogo possui pouco mais de 158 campeões disponíveis para serem escolhidos. Cada campeão possui características e habilidades específicas, garantindo pontos fracos e fortes, dado que o jogo oferece um balanceamento dos campeões durante o tempo, evitando que campeões tenham muitos pontos fortes e poucos pontos fracos.

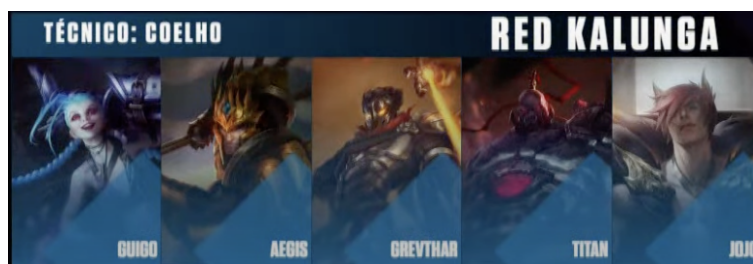


Figura 3. Seleção de campeões

A colaboração é um fator essencial em League of Legends. Existem campeões que possuem grande sinergia entre si e a combinação deles faz com que suas habilidades

e características sejam amplificadas. A utilização de composições onde os campeões possuem suas características convergentes faz com que o time possua uma força singular para vencer a partida. Dentro do League of Legends podemos identificar algumas composições mais consolidadas e que são bastante vistas no cenário competitivo e comum:

- **Poke:** Como o próprio nome diz, uma composição de Poke se baseia em debilitar os inimigos até que a luta se inicie de fato. Geralmente esse tipo de composição evita lutar diretamente contra o time inimigo, pois possuem campeões frágeis e outros fatores que influenciam uma luta. É caracterizada por campeões com alto dano e alcance, mantendo-se longe do raio de ação inimigo.
- **Team Fight:** As composições de Team Fight possuem um objetivo claro: selecionar campeões que se destaquem em lutas de equipe, levar o jogo até determinado momento e focar em lutas. Os campeões que possuem alto dano em área, vida e controle de grupo são adequados para este tipo de composição.
- **Hard Engage:** Já para a estratégia de Hard Engage, é necessário campeões que conseguem iniciar uma luta com uma certa facilidade. Se caracterizam por uma luta rápida e explosiva, abatendo o time inimigo antes que ele consiga fugir ou permanecer muito tempo resistindo.

Dito isso, observa-se que a importância desse procedimento é altamente elevada, pois cada personagem possui características próprias e atributos que se sobressaem ou que são ofuscados quando confrontados por outros personagens. Soma-se a esses fatores, a habilidade pessoal de cada jogador, já que existem atletas profissionais que são mundialmente conhecidos por possuírem extrema habilidade com heróis específicos.

3.2. Computação evolucionária

Computação Evolucionária ou computação evolutiva (CE) é um ramo da inteligência artificial e mais especificamente da inteligência computacional, possuindo um conjunto de algoritmos inspirados na seleção natural (Darwin) [3].

Inspirados na evolução natural das espécies, os algoritmos de CE buscam resolver os problemas criando uma população de indivíduos que vão reproduzir e competir pela sobrevivência. Desta forma, os melhores indivíduos irão sobreviver e repassar suas características para as próximas gerações. Atualmente, os tipos de algoritmos utilizados dentro do mundo de Computação Evolucionária são: Programação Evolucionária, Estratégias Evolucionárias, Algoritmos Genéticos e Programação Genética. A seleção de qual método utilizar vai de acordo com o problema em questão, visto que cada técnica é adequada para diferentes tipos de representação [1].

3.2.1. Algoritmos Genéticos

Algoritmos Genéticos (AGs) se caracterizam como uma forma de busca estocástica que imita a evolução natural das espécies Darwiniana, compreendendo processos de evolução, adaptação e sobrevivência dos indivíduos. A seleção de indivíduos privilegia aqueles mais aptos com maior longevidade, aumentando a possibilidade de reprodução. Esses indivíduos terão maior probabilidade de passar seu material genético para as próximas gerações, constituindo a identidade dos novos indivíduos. Estes princípios são usados para a construção dos algoritmos computacionais que buscam a melhor solução

para um determinado problema, através da evolução da população representada por indivíduos artificiais. Esses indivíduos por sua vez, são representados por estruturas de dados submetidas a processos evolucionários, como:

- **Avaliação:** A avaliação dos indivíduos é feita através de uma função que melhor representa o problema em questão, guiando o espaço de busca do algoritmo. A função da avaliação tem como objetivo fornecer uma medida de aptidão para cada indivíduo da população atual. Para exemplificar, iremos considerar a seguinte função 1 para medir a aptidão dos indivíduos de uma população:

$$f(x) = 2x + 1 \tag{1}$$

Na tabela 1 podemos verificar que o indivíduo 1 possui maior aptidão que os demais indivíduos (representados por binários):

Id	Indivíduo	Aptidão
1	0 1 0	5
2	0 0 1	3
3	0 0 0	0

Tabela 1. Exemplo de aptidão de indivíduos

- **Mutação:** O operador de mutação atua nos indivíduos, alterando seu material genético de acordo com uma probabilidade pm. Este processo é de caráter exploratório e tem como objetivo aumentar a diversidade da população.
- **Recombinação ou cruzamento:** Os indivíduos selecionados são introduzidos a um processo de recombinação, trocando seu material genético e gerando novos indivíduos. Nos AGs, isso ocorre de acordo com uma probabilidade pc. Os novos indivíduos são diferentes de seus pais mas com características provenientes de ambos.
- **Seleção:** A etapa de seleção se caracteriza pela escolha dos indivíduos que irão participar da reprodução. Essa escolha por sua vez, é guiada pela aptidão dos indivíduos: indivíduos mais aptos possuem maior probabilidade de serem escolhidos. Geralmente este processo ocorre através de um sorteio com roleta (figura 4), onde cada indivíduo possui um percentual relativo a sua a aptidão. Existem outros métodos de seleção identificados pela literatura: ranking, torneio, proporcional, truncamento, normalização linear e normalização exponencial.

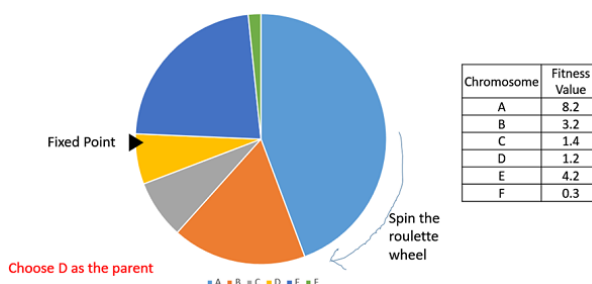


Figura 4. Seleção através da roleta

Todos esses operadores estão presentes no processo de evolução dos indivíduos e juntos são responsáveis por encontrar as melhores soluções para o problema. Atualmente já existem diversas implementações para os operadores de mutação, recombinação e seleção, restando analisar quais são compatíveis com o problema a ser resolvido. Tendo isso em vista, os AGs são considerados uma das principais e mais difundidas técnicas de Computação Evolucionária. Na figura 5 é possível visualizar os passos seguidos pelo algoritmo genético em seu processo evolucionário.

```

t = 0;
initialize(P(t=0));
evaluate(P(t=0));
while isNotTerminated() do
    Pp(t) = P(t).selectParents();
    Pc(t) = reproduction(Pp);
    mutate(Pc(t));
    evaluate(Pc(t));
    P(t+1) = buildNextGenerationFrom(Pc(t), P(t));
    t = t + 1;
end

```

Figura 5. Algoritmo Genético

Hoje os AGs são bastante conhecidos e aplicados em problemas de otimização e alta complexidade, que não são solucionáveis através de técnicas convencionais de programação. A complexidade por sua vez está diretamente ligada com o espaço de busca do problema, ou seja, com a quantidade de soluções candidatas a serem analisadas.

3.2.2. Algoritmos Genéticos Multiobjetivo

Os algoritmos genéticos são utilizados principalmente para problemas com um único objetivo. Porém, existem problemas que possuem várias funções objetivo, muitas vezes conflitantes entre si. Para resolver este tipo de problema com um algoritmo genético mono-objetivo, seria necessário combinar os objetivos em uma função de aptidão escalar.

Algoritmos Genéticos Multiobjetivo (também conhecidos por MOEAs) foram introduzidos pela primeira vez em 1993. Utiliza o conceito de dominância para encontrar soluções ótimas de Pareto [4]. Os MOEAs divergem de um AG comum na forma de atribuição da aptidão aos indivíduos. Os demais operadores (seleção, mutação e recombinação) são aplicados igualmente.

O objetivo dos problemas de otimização multiobjetivo é encontrar todas as possíveis compensações entre as múltiplas funções objetivo que geralmente são conflitantes. Como é difícil escolher uma única solução para um problema de otimização multiobjetivo sem interação iterativa com o tomador de decisão, uma abordagem geral é mostrar o conjunto de soluções ótimas de Pareto para o tomador de decisão. Então, uma das soluções ótimas de Pareto pode ser escolhida dependendo da preferência [8] (Figura 6).

O primeiro Algoritmo Genético Multiobjetivo foi proposto por Schaffer [9]. Em seguida, diversos outros algoritmos evolutivos multiobjetivo foram introduzidos. Dentre

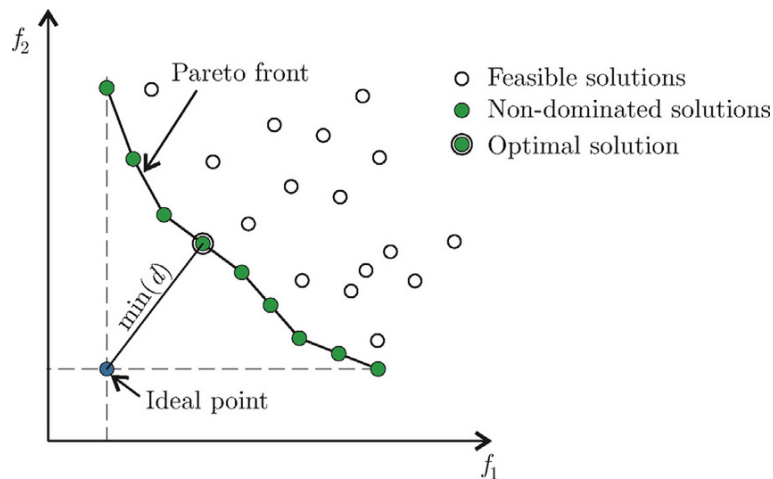


Figura 6. Soluções ótimas de Pareto

eles, o Algoritmo Genético de Classificação Não Dominada II (NSGA-II) [4], um algoritmo multi-objetivo sólido, amplamente utilizado em muitas aplicações do mundo real.

Já o Algoritmo Genético de Classificação Não Dominado III ou NSGA-III [7], utilizado no contexto deste trabalho, surgiu a partir da evolução do algoritmo NSGA-II, com o objetivo de resolver problemas de otimização com mais de 3 objetivos, utilizando conceitos de ponto de referência para realizar a seleção das soluções [5].

3.3. Metodologia

A abordagem introduzida por este trabalho busca gerar composições no League of Legends através de algoritmos de busca, no caso os Algoritmos Genéticos. Para a montagem e avaliação das equipes, atributos e características dos campeões serão utilizados. Devido a isso, dados disponibilizados pelo próprio jogo foram consumidos através de API's (também conhecido como Interface de programação de aplicações), com informações de todos os campeões. Os dados de cada campeão são representados através de um arquivo json ou JavaScript Object Notation, comumente utilizado para troca de dados entre sistemas e serviços. Cada um dos campeões possuem atributos como: dano de ataque, alcance, vida, velocidade de movimento, entre outros. Alguns desses atributos foram selecionados para avaliar a qualidade de uma composição, guiando a etapa de busca do algoritmo.

Dando início à etapa de busca, uma população inicial é gerada, formada por indivíduos criados aleatoriamente. Logo em seguida, toda a população de indivíduos é avaliada. Através do operador de seleção, são escolhidos os indivíduos que servirão de base para criação de um novo conjunto de possíveis soluções, chamado de nova "geração". Esta nova geração é obtida aplicando-se sobre os indivíduos selecionados operações que combinem suas características, através dos operadores de recombinação e mutação. Estes passos são repetidos até que uma condição de parada seja atingida, representada por um número máximo de gerações. O processo descrito pode ser visualizado na figura 7.

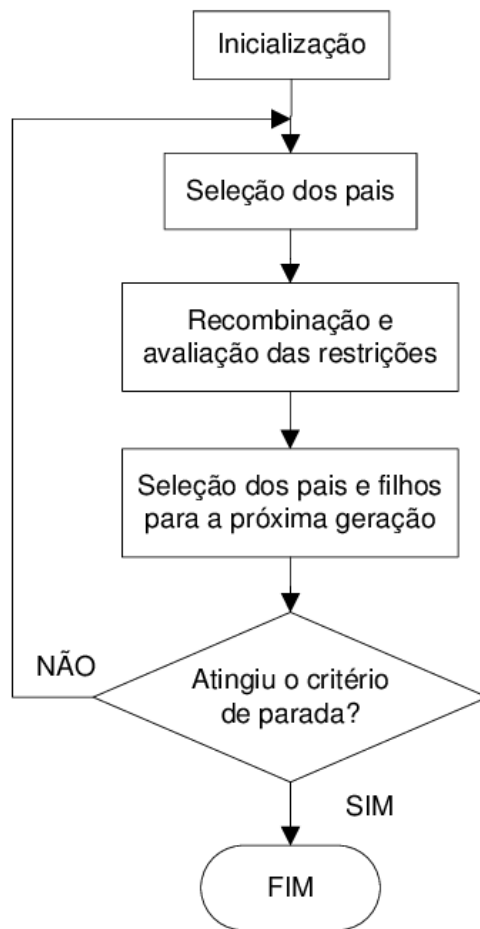


Figura 7. Passo a passo algoritmo genético

3.3.1. Representação cromossômica

Nos Algoritmos Genéticos a representação cromossômica é necessária para descrever os indivíduos da população. Essa representação determina como o problema será estruturado e impacta diretamente no tempo computacional gasto para buscar as soluções.

Para representação dos indivíduos neste trabalho, foi utilizado um vetor (cromossomo) com uma sequência de números inteiros, onde cada número identifica unicamente um campeão da base de dados. No total serão necessários 5 números inteiros para representar um indivíduo, visto que uma equipe possui 5 campeões.

3.3.2. Seleção de pais

Na etapa de seleção de pais, pares de indivíduos foram escolhidos aleatoriamente. Os pares de indivíduos gerados são recombinados de acordo com uma probabilidade de cruzamento. Essa probabilidade interfere na quantidade de novos indivíduos formados para a próxima geração.

3.3.3. Seleção de sobreviventes

Essa etapa é responsável por selecionar os indivíduos que irão permanecer nas gerações seguintes. A seleção de sobreviventes ocorre em duas etapas. Primeiramente, são selecionados os indivíduos seguindo o critério de dominância. Caso o número de indivíduos para a população final não seja atendido, o NSGA-III utiliza o conceito de pontos de dominância para selecionar as demais soluções [5].

3.3.4. Recombinação

Os indivíduos selecionados são recombinados sexualmente através do operador de cruzamento, sendo uma das principais etapas no processo de evolução. Os pais são selecionados de forma aleatória, com base na aptidão, gerando novos indivíduos. Os descendentes serão diferentes dos seus pais, mas irão possuir características de ambos.

Um exemplo é o cruzamento de um ponto, onde os pais são cortados em um ponto aleatório e combinados, gerando dois novos possíveis indivíduos.

Para garantir que cada composição possua campeões distintos, foi necessário adicionar uma validação no operador de cruzamento. Desta forma, um campeão só será trocado de um indivíduo “a” para um indivíduo “b”, caso o indivíduo “b” não possua o campeão na equipe e vice-versa.

3.3.5. Mutação

Os novos indivíduos gerados a partir da recombinação são submetidos ao processo de mutação. O operador de mutação tem o propósito de prover diversidade à população.

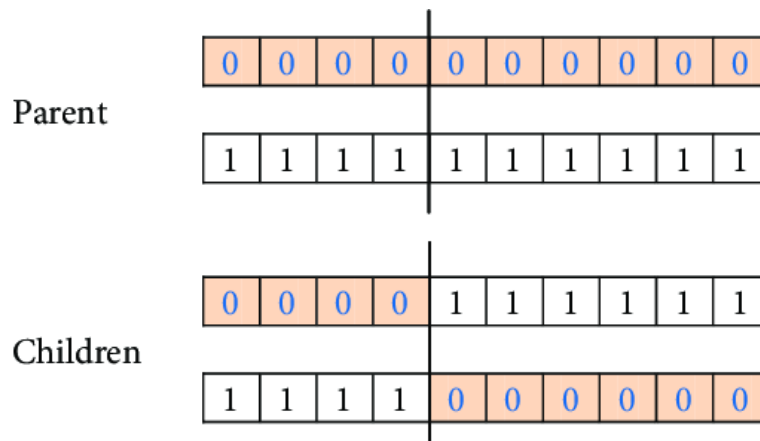


Figura 8. Cruzamento de um ponto

Funciona através da troca de uma ou mais características dos indivíduos. O operador de mutação é aplicado aos indivíduos através de uma taxa de mutação.

O operador de mutação desenvolvido atua na troca de 1 campeão da equipe (escolhido aleatoriamente), por um novo campeão sorteado da base de dados. São removidos do sorteio, para evitar repetições, os personagens já presentes na equipe. A figura 9 exemplifica a mutação de um indivíduo.

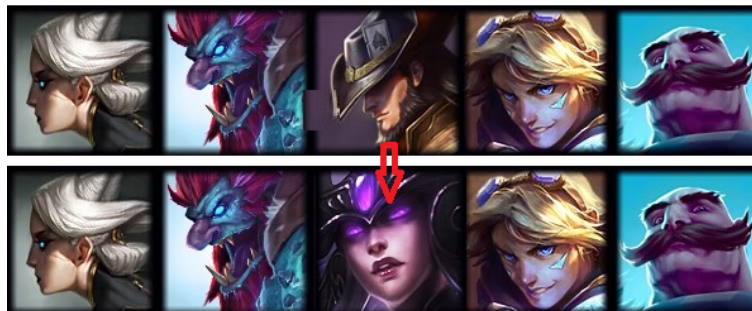


Figura 9. Mutação de um indivíduo

3.3.6. Função de avaliação

A função de aptidão é a chave para encontrar as soluções do problema. Seu objetivo é fornecer uma medida de aptidão para os indivíduos da população, que irá guiar o processo de busca. Em algoritmos multiobjetivo a aptidão é designada para cada objetivo.

No League of Legends existem muitas composições e estratégias capazes de levar um time a vitória. Como qualquer jogo, essas estratégias possuem pontos fracos e fortes. Como descrito anteriormente, uma composição de Poke possui o objetivo de abater campeões e estruturas através do alcance e dano elevado dos campeões, porém, possui baixa resistência e defesa nos campeões. Baseado nisso, foram utilizadas 3 funções de aptidão para geração de composições, baseadas nas 3 estratégias descritas anteriormente: Poke, Team Fight e Hard Engage.

Foram selecionados diferentes atributos dos campeões de acordo com cada estratégia descrita. Os atributos utilizados para cada tipo de estratégia podem ser visualizados na tabela 2:

Estratégia	Atributos
Poke	Alcance, Dano de ataque e Dano por nível
Team Fight	Dano de ataque, Vida e Dano por nível
Hard Engage	Dano de ataque e Velocidade de movimento

Tabela 2. Estratégias e atributos utilizados

Os atributos selecionados para cada estratégia serão utilizados para medir a aptidão dos indivíduos. Essa aptidão é representada por uma lista de valores, referentes a cada objetivo ou atributo a ser otimizado.

A função que representa o cálculo da aptidão dos indivíduos para a estratégia de Poke pode ser descrita através da Equação 2.

$$\text{Maximizar}\{f_1, f_2, f_3\} \quad (2)$$

onde

$$f_1 = \text{danoAtaqueCampeao} / \text{Max}(\text{danoAtaqueDatabase})$$

$$f_2 = \text{danoAtaqueNivelCampeao} / \text{Max}(\text{danoAtaqueNivelDatabase})$$

$$f_3 = \text{alcanceCampeao} / \text{Max}(\text{alcanceDatabase})$$

O valor calculado para cada função objetivo utiliza como base os valores máximos de dano, alcance e dano por nível, obtidos a partir da base de dados. Esses valores são pré-calculados e armazenados para serem utilizados durante a execução do algoritmo. Os valores máximos dos atributos utilizados nas estratégias de Poke, Team Fight e Hard Engage podem ser encontrados na tabela 3.

Atributos	Valor máximo
Dano de ataque	72
Dano por nível	5
Velocidade de movimento	355
Vida	650

Tabela 3. Valores máximos dos atributos

Para a estratégia de Team Fight, os seguintes atributos foram utilizados: dano de ataque, dano de ataque por nível e vida (Equação 3).

$$\text{Maximizar}\{f_4, f_5, f_6\} \quad (3)$$

onde:

$$\begin{aligned}f_4 &= \text{danoAtaqueCampeao} / \text{Max}(\text{danoAtaqueDatabase}) \\f_5 &= \text{danoAtaqueNivelCampeao} / \text{Max}(\text{danoAtaqueNivelDatabase}) \\f_6 &= \text{vidaCampeao} / \text{Max}(\text{vidaDatabase})\end{aligned}$$

A função de aptidão descrita na Equação 4 refere-se à estratégia de Hard Engage. Nessa montagem, o ataque e vida dos campeões são utilizados para medir a aptidão dos indivíduos.

$$\text{Maximizar}\{f_7, f_8\} \quad (4)$$

onde

$$\begin{aligned}f_7 &= \text{danoAtaqueCampeao} / \text{Max}(\text{danoAtaqueDatabase}) \\f_8 &= \text{velocidadeMovimentoCampeao} / \text{Max}(\text{velocidadeMovimentoDatabase})\end{aligned}$$

4. Resultados (Estudo de Caso)

Para melhor visualização dos dados gerados, foi desenvolvido um aplicativo móvel que irá consumir os resultados produzidos pelo algoritmo. O aplicativo contém funcionalidades básicas que permitem os usuários gerarem composições de acordo com a estratégia selecionada, filtrar as composições geradas e visualizar informações dos campeões alocados. As funcionalidades descritas estão ilustradas na figura 10.

Os dados disponibilizados para os usuários foram gerados através de execuções do algoritmo e armazenados na nuvem, sendo apenas consumidos posteriormente. Com isso, o aplicativo apenas consome a base de soluções geradas e o tempo de resposta para usuários do aplicativo estará apenas associado ao processo de obtenção dos dados, não necessitando de execuções em tempo real para geração das composições. Os tempos de execução do algoritmo para as 3 estratégias podem ser visualizados na tabela 4. Os tempos foram obtidos através de 10 execuções do algoritmo, obtendo o valor médio dessas execuções.

Estratégia	Tempo de execução (em segundos)
Poke	21.7911
Team Fight	20.5534
Hard Engage	1.7386

Tabela 4. Tempo de execução para cada estratégia

Na estratégia de Hard Engage o número de objetivos utilizados é menor, impactando significativamente no tempo de execução. As estratégias de Poke e Team Fight possuem o mesmo número de objetivos, obtendo tempos de execução similares.

Com o objetivo de avaliar a qualidade das equipes geradas para as 3 estratégias descritas neste trabalho (Poke, Team Fight e Hard Engage), foi realizada uma pesquisa

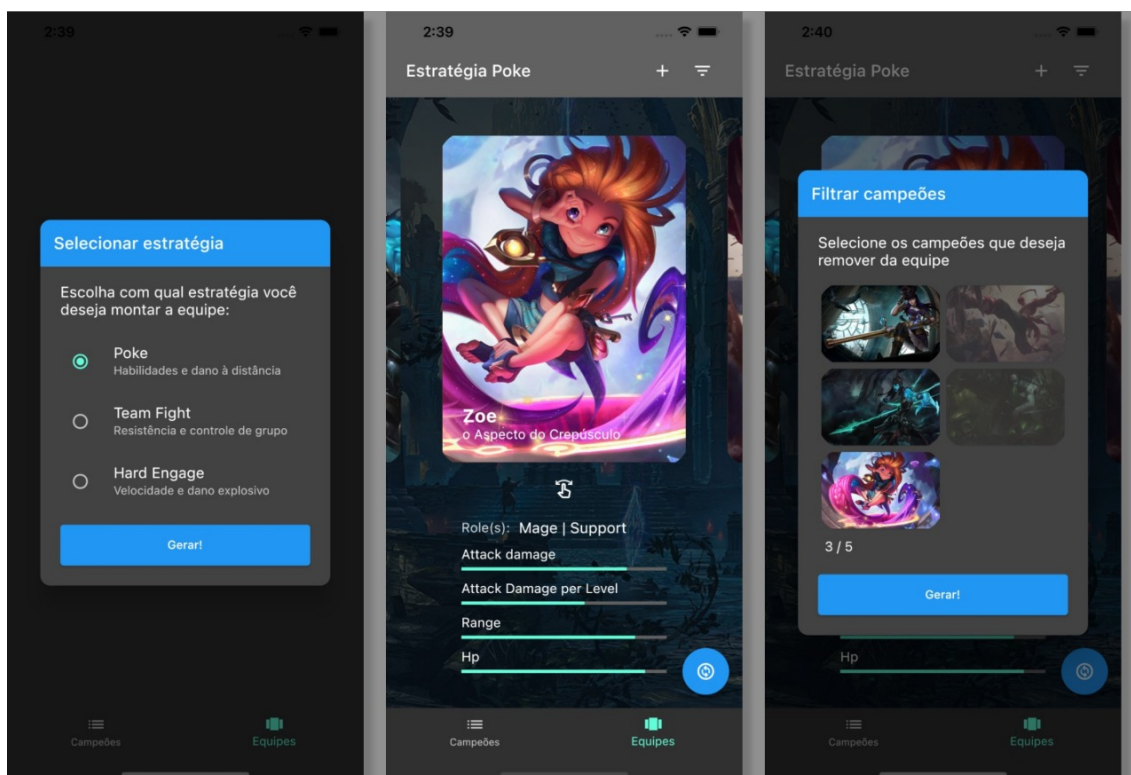


Figura 10. Telas do aplicativo. Esquerda: Selecionando estratégia. Meio: Visualizando equipes geradas. Direita: Filtrando equipes

conduzida por um parcela de usuários do aplicativo. Os primeiros usuários foram convidados a participar através de aplicativo de mensagens e redes sociais, sendo pessoas conhecidas de dentro e até mesmo fora do jogo. No total, 22 usuários participaram da pesquisa. Foram selecionados participantes que possuem diferentes níveis de conhecimento a respeito do jogo, com o intuito de encontrar quais possuem mais interesse na plataforma. Para atingir esses objetivos, os seguintes questionamentos foram feitos:

1. Como você avalia a qualidade das composições geradas para estratégia Poke?
2. Como você avalia a qualidade das composições geradas para estratégia Team Fight?
3. Como você avalia a qualidade das composições geradas para estratégia Hard Engage?
4. Você utilizaria esta ferramenta para auxiliar na seleção de equipes?

Nas respostas relacionadas à estratégia de Poke (item 1), 54,5% dos usuários responderam com nota 5, 31,8% com nota 4 e 13,6% com nota 3. As respostas podem ser visualizadas na figura 11. Uma avaliação média de aproximadamente 4.4 foi obtida para essa estratégia.

Para Team Fight (item 2), também foram obtidas notas iguais ou maiores 3, onde 63,6% das avaliações foram notas 5, 31,8% notas 4 e 4,5% de notas 3 (figura 12). Concluindo com a estratégia de Hard Engage (item 3), representada pelo item 3, foram obtidos os seguintes resultados: 59,1% de notas 5, 31,8% notas 4 e 9,1% de notas 3 (figura 13). As avaliações médias para Team Fight e Hard Engage foram 4,6 e 4,5, respectivamente.

Como você avalia a qualidade das composições geradas para estratégia Poke?

[Copiar](#)

22 respostas

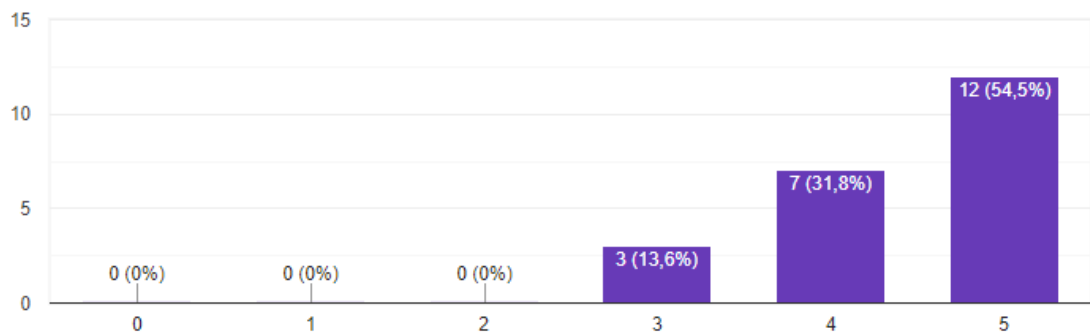


Figura 11. Respostas do questionário para o item 1

Como você avalia a qualidade das composições geradas para estratégia Team Fight?

[Copiar](#)

22 respostas

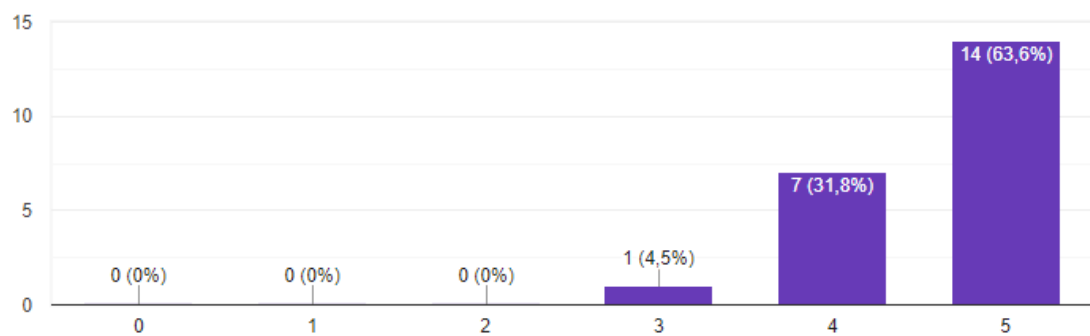


Figura 12. Respostas do questionário para o item 2

Como você avalia a qualidade das composições geradas para estratégia Hard Engage?

 Copiar

22 respostas

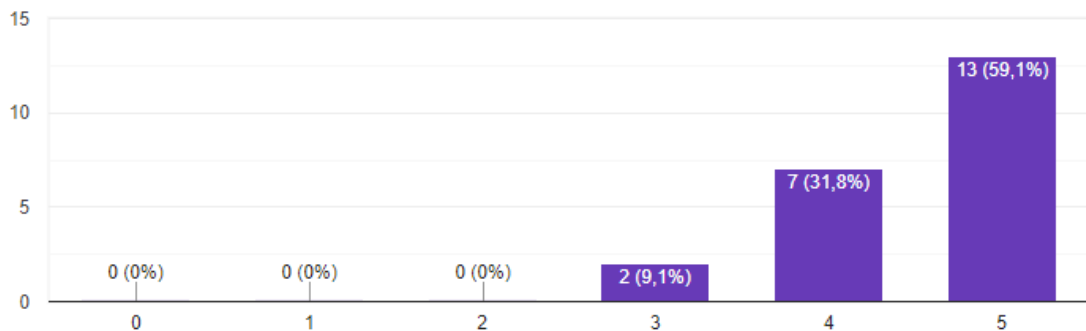


Figura 13. Respostas do questionário para o item 3

Os resultados obtidos a partir das avaliações dos usuários foram bastante expressivos, principalmente para Team Fight e Hard Engage, indicando que as equipes geradas estão de acordo com a proposta das estratégias. Já para a composição Poke, a mudança dos atributos utilizados para avaliar os indivíduos e gerar as composições podem ser modificados para obter resultados mais coerentes.

Nas respostas relacionadas ao item 4 (figura 14), percebe-se um interesse dos usuários em continuar utilizando o aplicativo, principalmente por jogadores iniciantes, interessados em aprender mais sobre a geração de composições dentro do jogo. Dos usuários que responderam este item, 72,7% indicaram que possuem interesse em continuar utilizando a ferramenta, 13,6% responderam que utilizariam mediante melhorias e os demais responderam que não possuem interesse em utilizar.

Você utilizaria esta ferramenta para auxiliar na seleção de equipes?

 Copiar

22 respostas

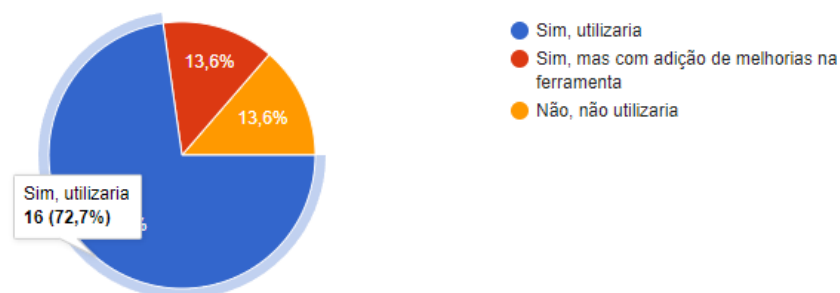


Figura 14. Respostas do questionário para o item 4

A figura 15 descreve uma das equipes geradas para estratégia de Poke. Todos

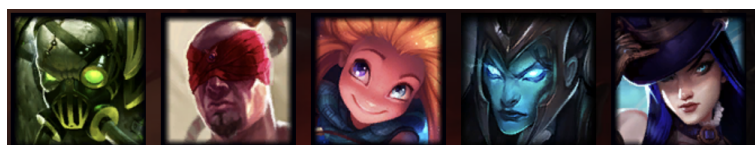


Figura 15. Composição gerada para Poke

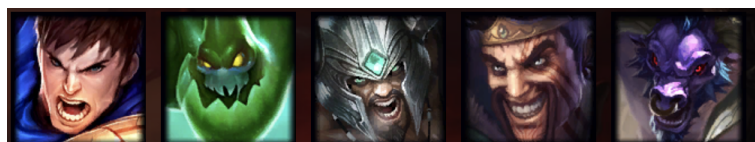


Figura 16. Composição gerada para Team Fight

os campeões selecionados possuem alto alcance e dano, objetivos selecionados para essa estratégia. Já as figuras 16 e 17 representam composições geradas para Team Fight e Hard Engage, respectivamente.

Os campeões selecionados para a estratégia de Team Fight 16 possuem alta resistência e controle de grupo, sendo boas escolhas para lutas em equipe. Já os campeões da estratégia de Hard Engage 17 possuem bons mecanismos para iniciar lutas, alta velocidade de movimento e dano, características importantes para este tipo de composição.

No geral, técnicas baseadas em busca combinadas com uma boa função de aptidão se mostraram um bom caminho para montagem de equipes, visto que o algoritmo pode se adaptar para qualquer estratégia dentro do jogo. Isso reforça o que o fornecimento de uma ferramenta como esta pode auxiliar jogadores e treinadores, parcialmente ou até mesmo totalmente, na montagem de equipes.

Utilizando poucos atributos dos campeões foi possível atingir resultados promissores na geração de composições. Com a adição de novas restrições e atributos, é possível personalizar e aprimorar a qualidade das equipes geradas. Adicionar novos atributos incrementa a complexidade do problema, necessitando de mais tempo para gerações das soluções candidatas. Entretanto, pelo processamento dos dados ser realizado offline e armazenados em nuvem, o tempo de resposta dos usuários através do aplicativo se mantém.

5. Discussões e Trabalhos futuros

Este trabalho introduz uma abordagem através de Algoritmos Genéticos Multiobjetivo para gerar composições no League of Legends. Através de estratégias comumente utilizadas no jogo, foram construídas diferentes funções de aptidão para avaliar os indivíduos. O algoritmo utiliza dados disponibilizados pelo jogo, com informações de todos os campeões disponíveis.

Os resultados encontrados são promissores na geração de composições, sendo um pontapé inicial no auxílio à tomada de decisão de jogadores e times. Um aplicativo móvel foi desenvolvido para a visualização e interação com os resultados obtidos, possibilitando ao usuário a geração, filtragem e análise de composições.

Avaliamos a abordagem conduzida por este trabalho por meio de uma pesquisa, como forma de analisar a adequação das equipes geradas a partir do algoritmo para cada estratégia, ou seja, Poke, Team Fight e Hard Engage. Analisando as respostas, pode-se

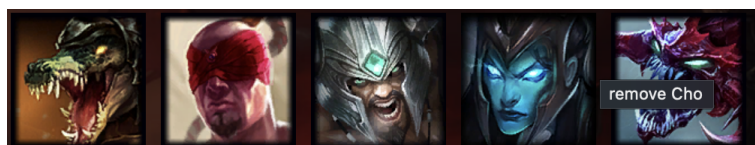


Figura 17. Composição gerada para Hard Engage

notar que para todas as estratégias a qualidade das equipes foram avaliadas como “boa” ou superior. Somando os resultados, foi obtido uma média de 4.5 pontos de um total de 5. O resultado final foi bastante positivo, indicando a relevância da ferramenta desenvolvida para jogadores iniciantes e avançados.

Para a aplicação e adesão da ferramenta por equipes em jogos competitivos, outros fatores podem ser levados em consideração, visto que múltiplas variáveis estão envolvidas no processo de montagem das composições. A colaboração com jogadores e técnicos pode ser de grande impacto para a evolução da ferramenta, de modo a selecionar os melhores atributos e restrições para o problema. As equipes geradas para as 3 estratégias seguidas neste trabalho possuem grande semelhança com os demais trabalhos utilizados como base de comparação.

Ainda há uma série de avanços que podem ser considerados na evolução do trabalho em questão, contribuindo para geração de equipes mais otimizadas e consistentes. Uma delas seria levar em consideração, para a escolha dos campeões, as seleções realizadas pelo time adversário, técnica conhecida como counterpicking e utilizada constantemente pelos times profissionais no cenário competitivo. Muitas vezes as estratégias são alteradas durante o processo de seleção de campeões, visto que os personagens são selecionados de forma alternada e as escolhas do time inimigo podem sugerir alguma modificação da tática inicial.

Outro tópico a ser trabalhado no futuro, como forma de validar a qualidade das equipes geradas, trata-se de utilizar os dados históricos de partidas. Com esse histórico, seria possível analisar os campeões selecionados por cada time e a taxa de sucesso das equipes. Essas informações serão úteis para entender a efetividade das equipes em jogos passados, priorizando equipes com maiores taxa de sucesso. Esses dados podem ser consumidos através de APIs disponibilizadas pelo League of Legends.

A utilização de técnicas inteligentes como Inteligência Artificial e Algoritmos Genéticos para geração de composições em jogos MOBA é bastante recente e poucos trabalhos relacionados podem ser encontrados a respeito do assunto. Portanto, o presente trabalho pode incentivar ao desenvolvimento de abordagens mais robustas para auxiliar no processo de seleção de equipes, seja em League of Legends ou outros jogos da categoria. A aplicação de ferramentas como esta pode ser de grande impacto para times profissionais, trazendo vantagens competitivas em análise e geração de composições.

Referências

- [1] *Computação Evolutiva*. URL: <https://www.inf.ufpr.br/aurora/disciplinas/datamining/Ceapostila.pdf>. (accessed: 19.05.2022).
- [2] L. Costa, A. Souza e F. M. Souza. “An Approach for Team Composition in League of Legends using Genetic Algorithm”. Em: (out. de 2019), pp. 52–

61. DOI: 10.1109/SBGames.2019.00018. URL: <https://doi.ieeeecomputersociety.org/10.1109/SBGames.2019.00018>.
- [3] *Darwin, evolution, & natural selection*. URL: <https://www.khanacademy.org/science/ap-biology/natural-selection/natural-selection-ap/a/darwin-evolution-natural-selection#:~:text=Charles%20Darwin%20was%20a%20British,and%20share%20a%20common%20ancestor..> (accessed: 19.05.2022).
- [4] K. Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. Em: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197. DOI: 10.1109/4235.996017.
- [5] Kalyanmoy Deb e Himanshu Jain. “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints”. Em: *IEEE Transactions on Evolutionary Computation* 18.4 (2014), pp. 577–601. DOI: 10.1109/TEVC.2013.2281535.
- [6] *Esport Earning*. URL: <https://www.esportsearnings.com/>. (accessed: 19.05.2022).
- [7] Hisao Ishibuchi, Noritaka Tsukamoto e Yusuke Nojima. “Evolutionary many-objective optimization: A short review”. Em: (2008), pp. 2419–2426. DOI: 10.1109/CEC.2008.4631121.
- [8] T. Murata e H. Ishibuchi. “MOGA: Multi-Objective Genetic Algorithms”. Em: *IEEE* (1995), pp. 290–291. DOI: 10.1109/ICEC.1995.489161.
- [9] J. David Schaffer. “Multiple Objective Optimization with Vector Evaluated Genetic Algorithms”. Em: *ICGA*. 1985.
- [10] Alvaro Scola. *League of Legends: guia para iniciantes*. URL: <https://tecmasters.com.br/league-of-legends-guia-para-iniciantes/>. (accessed: 23.05.2022).