

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
MINAS GERAIS - *CAMPUS* IBIRITÉ
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Pedro Lucas da Silva

**ANÁLISE DA VARIAÇÃO DE OPERADORES DE MUTAÇÃO EM
ALGORITMOS DIFERENCIAIS EVOLUTIVOS PARA O
PLANEJAMENTO DE CAMINHO DE ROBÔS FUTEBOLISTAS**

Ibirité - MG

2025

PEDRO LUCAS DA SILVA

**ANÁLISE DA VARIAÇÃO DE OPERADORES DE MUTAÇÃO EM
ALGORITMOS DIFERENCIAIS EVOLUTIVOS PARA O
PLANEJAMENTO DE CAMINHO DE ROBÔS FUTEBOLISTAS**

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus Ibirité* para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Diogo Sampaio Cesar Souza

Ibirité - MG
2025

S586a
2025

Silva, Pedro Lucas da.

Análise da variação de operadores de mutação em algoritmos diferenciais evolutivos para o planejamento de caminho de robôs futebolistas.

[manuscrito] / Pedro Lucas da Silva. – Ibirité: Instituto Federal de Minas Gerais, 2025.

88 f.; il.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de controle e automação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – Campus Ibirité, 2025.

Orientador: Prof. Dr. Diogo Sampaio César Souza.

1. Robôs móveis. 2. Algoritmos. 3. Futebol. 4. Inteligência artificial. I. Souza Diogo Sampaio César. II. Instituto Federal de Minas Gerais – *Campus* Ibirité. III. Título


CDU 004.89

Catálogo: Luciana Batista Neves - CRB-6/2000


Pedro Lucas da Silva

ANÁLISE DA VARIAÇÃO DE OPERADORES DE MUTAÇÃO EM ALGORITMOS DIFERENCIAIS EVOLUTIVOS PARA O PLANEJAMENTO DE CAMINHO DE ROBÔS FUTEBOLISTAS


Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus* Ibirité para a obtenção do título de Engenheiro de Controle e Automação.

Documento assinado digitalmente
 **DIOGO SAMPAIO CESAR SOUZA**
Data: 10/03/2025 18:33:36-0300
Verifique em <https://validar.iti.gov.br>


Prof. Dr. Diogo Sampaio Cesar Souza - IFMG (Orientador)

Documento assinado digitalmente
 **PEDRO HENRIQUE FERREIRA MACHADO**
Data: 10/03/2025 18:42:18-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Pedro Henrique Ferreira Machado - IFSP

Documento assinado digitalmente
 **TAMIRES MARTINS REZENDE**
Data: 11/03/2025 13:20:13-0300
Verifique em <https://validar.iti.gov.br>

Profa. Dra. Tamires Martins Rezende - IFMG

Documento assinado digitalmente
 **CARLOS DIAS DA SILVA JUNIOR**
Data: 10/03/2025 18:58:02-0300
Verifique em <https://validar.iti.gov.br>

Prof. Me. Carlos Dias da Silva Júnior - IFMG

Dedico esta monografia aos meus amados pais, maiores incentivadores e fontes inesgotáveis de apoio, amor e compreensão, por cada palavra de incentivo, cada gesto de carinho e cada sonho compartilhado.

AGRADECIMENTOS

Gostaria de expressar minha profunda gratidão aos meus pais, Roselis Pereira da Silva e Arnaldo Conceição da Silva, pelo apoio incondicional e por sempre acreditarem em mim.

Agradeço também à minha família e amigos, que sempre me apoiaram e demonstraram carinho ao longo desta jornada. Em especial, agradeço à minha namorada, Rayandra Kethlyn Souza Teixeira, que sempre me apoiou e esteve ao meu lado nos momentos mais difíceis.

Agradeço ao Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais por proporcionar um ambiente de aprendizado e crescimento. Um agradecimento especial ao meu orientador, Diogo Sampaio Cesar Souza, por sua orientação e amizade, e ao professor Elias José de Rezende Freitas, que, além de sua orientação em projetos de pesquisa, me ofereceu conselhos valiosos que ajudaram no meu crescimento pessoal.

Sem a contribuição de vocês, este trabalho não teria sido possível.

*O pássaro que não desanima vai enxergar o arco-íris e
voltar para contar a melhor parte de voar.*

Ana Calline

RESUMO

O Planejamento de Caminho para Robôs Móveis é um tema relevante na robótica autônoma, sendo abordado em várias áreas, como no futebol de robôs. Este trabalho analisa o desempenho de Algoritmos Evolucionários, especificamente os Diferenciais Evolutivos, no planejamento de caminho, com foco na influência de diferentes operadores de mutação. Foram avaliados os algoritmos DE, SaDE, JADE, SHADE e L-SHADE, considerando operadores de mutação: "rand/1", "rand/2", "best/1", "best/2", "either-or", "current-to-rand/1", "current-to-best/1" e "current-to-pbest/1". Inicialmente, a estratégia de comparação foi testada em funções do *benchmark* CEC2013 e, posteriormente, aplicados ao planejamento de caminhos utilizando curvas NURBS. Os experimentos indicam que o algoritmo L-SHADE, combinado com os operadores "current-to-rand/1" e "rand/2", apresentaram um desempenho consistente na geração de caminhos suaves e factíveis para os robôs futebolistas. Os resultados sugerem que a escolha do operador de mutação pode influenciar na eficiência de determinados algoritmos na otimização dos caminhos.

Palavras-chave: Robôs móveis. Algoritmos. Futebol. Inteligência artificial.

ABSTRACT

Path Planning for Mobile Robots is a relevant topic in autonomous robotics, being addressed in various areas, such as robot soccer. This work analyzes the performance of Evolutionary Algorithms, specifically Differential Evolution algorithms, in path planning, with a focus on the influence of different mutation operators. The algorithms DE, SaDE, JADE, SHADE, and L-SHADE were evaluated, considering the following mutation operators: "rand/1", "rand/2", "best/1", "best/2", "either-or", "current-to-rand/1", "current-to-best/1", and "current-to-pbest/1". Initially, the comparison strategy was tested on benchmark functions from CEC2013 and subsequently applied to path planning using NURBS curves. The experiments indicate that the L-SHADE algorithm, combined with the "current-to-rand/1" and "rand/2" operators, showed consistent performance in generating smooth and feasible paths for the soccer robots. The results suggest that the choice of mutation operator can influence the efficiency of certain algorithms in optimizing paths.

Keywords: Mobile robots. Algorithms. Soccer. Artificial intelligence.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ambiente de uma partida da categoria VSSS.	14
Figura 2 – Partida disputada na categoria VSSS.	15
Figura 3 – Pontos x Curva Parametrizada.	16
Figura 4 – Classificação dos RMs por ambiente e método de interação.	20
Figura 5 – Exemplo de tipos de RMs terrestres com rodas.	21
Figura 6 – Robôs Futebolistas.	22
Figura 7 – Esquema do Robô Diferencial.	22
Figura 8 – Exemplo de um campo vetorial Atrativo aplicado ao VSSS.	24
Figura 9 – Exemplo de um campo vetorial atrativo e repulsivo aplicado ao VSSS.	25
Figura 10 – Grafo de conectividade do método PRM.	26
Figura 11 – Esquema Geral dos AEs.	27
Figura 12 – Exemplo gráfico do operador de mutação "rand/1".	31
Figura 13 – Função 9 do <i>benchmark</i> CEC 2013.	34
Figura 14 – Função 11 do <i>benchmark</i> CEC2013.	34
Figura 15 – Parâmetros de controle utilizado para uma curva NURBS.	37
Figura 16 – Pontos de controle colineares e otimizados pelos algoritmos DEs.	38
Figura 17 – Diagramas de caixa das funções CEC2013 para $D = 2$	43
Figura 18 – Diagramas de caixa das funções CEC2013 para $D = 10$	44
Figura 19 – Diagramas de caixa das funções CEC2013 para $D = 30$	45
Figura 20 – Diagramas de caixa das funções CEC2013 para $D = 50$	46
Figura 21 – Diagramas de caixa (média e desvio padrão) para cada dimensão.	47
Figura 22 – Cenários do futebol de robôs utilizados nos testes.	49
Figura 23 – Diagramas de caixas dos mutadores com o L-SHADE.	52
Figura 24 – Caminhos obtidos com L-SHADE e mutadores para o Cenário 1.	53
Figura 25 – Caminhos obtidos com L-SHADE e mutadores para o Cenário 2.	54
Figura 26 – Caminhos obtidos com L-SHADE e mutadores para o Cenário 3.	56
Figura 27 – Caminhos obtidos com L-SHADE e mutadores para o Cenário 4.	57
Figura 28 – Caminhos obtidos com L-SHADE e mutadores para o Cenário 5.	58
Figura 29 – Curvatura dos melhores caminhos com o L-SHADE para cada cenário.	59
Figura 30 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 1.	65
Figura 31 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 2.	66
Figura 32 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 3.	68
Figura 33 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 4.	69
Figura 34 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 5.	70

LISTA DE TABELAS

Tabela 1	– Desempenho do algoritmo L-SHADE e mutadores aplicado ao DE-NURBS.	50
Tabela 2	– Desempenho dos algoritmos DEs e mutadores aplicado ao DE-NURBS.	61
Tabela 3	– <i>Ranking</i> da combinação algoritmo e mutador com menores médias de fitness.	63
Tabela 4	– Caminhos factíveis por combinação de algoritmo e mutador.	63
Tabela 5	– Média e desvio padrão do valor de <i>fitness</i> para D=2.	78
Tabela 6	– Média e desvio padrão do valor de <i>fitness</i> para D=10.	79
Tabela 7	– Média e desvio padrão do valor de <i>fitness</i> para D=30.	80
Tabela 8	– Média e desvio padrão do valor de <i>fitness</i> para D=50.	81

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
NURBS	<i>Non-Uniform Rational B-Spline</i>
VSSS	<i>Very Small Size Soccer</i>
AEs	Algoritmos Evolucionários
CEC	<i>Congress on Evolutionary Computation</i>
RMs	Robôs Móveis
PRM	<i>Probabilistic Roadmaps</i>
DEs	Diferenciais Evolutivos
DE	<i>Differential Evolution</i>
SaDE	<i>Self-Adaptive Differential Evolution</i>
JADE	<i>Adaptive Differential Evolution with Optional External Archive</i>
SHADE	<i>Success History based Adaptive Differential Evolution</i>
L-SHADE	<i>Linear Success History based Adaptive Differential Evolution</i>

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Objetivos	18
<i>1.1.1</i>	<i>Objetivo geral</i>	<i>18</i>
<i>1.1.2</i>	<i>Objetivos específicos</i>	<i>18</i>
1.2	Organização do Texto	18
2	REFERENCIAL TEÓRICO	19
2.1	Robôs móveis	19
<i>2.1.1</i>	<i>Modelo cinemático do robô</i>	<i>21</i>
2.2	Planejamento de caminho	23
2.3	Algoritmos Evolucionários	26
<i>2.3.1</i>	<i>Algoritmos Diferenciais Evolutivos</i>	<i>27</i>
2.4	Operadores de Mutação em algoritmos Diferenciais Evolutivos	30
3	MÉTODOS	33
3.1	Exploração dos mutadores em <i>benchmark</i> genérica	33
3.2	Avaliação dos algoritmos Diferenciais Evolutivos para planejamento de caminho	35
<i>3.2.1</i>	<i>Definição do problema</i>	<i>35</i>
<i>3.2.2</i>	<i>Representação do caminho</i>	<i>35</i>
<i>3.2.3</i>	<i>Problema de otimização</i>	<i>38</i>
4	RESULTADOS E DISCUSSÃO	40
4.1	Configuração dos experimentos	40
<i>4.1.1</i>	<i>Problemas genéricos de otimização</i>	<i>40</i>
<i>4.1.2</i>	<i>Planejamento de caminho</i>	<i>41</i>
4.2	Comparação dos operadores de mutação em problemas genéricos de otimização	41
4.3	Comparação dos operadores de mutação aplicados ao L-SHADE para o problema de planejamento de caminho	48

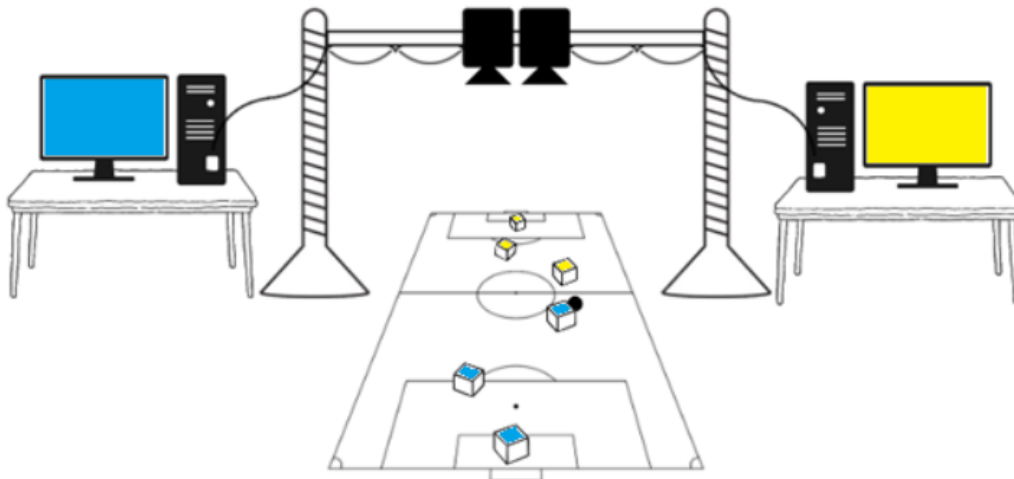
4.4	Comparação dos operadores de mutação e algoritmos diferenciais evolutivos aplicados ao problema de planejamento de caminho	60
5	CONCLUSÃO E TRABALHOS FUTUROS	71
5.1	Trabalhos Futuros	71
5.2	Publicações, participação em projetos de pesquisa e premiações	72
	REFERÊNCIAS	74
	APÊNDICE A – RESULTADOS PARA OS PROBLEMAS IRRESTRITOS DO <i>BENCHMARK</i> CEC2013	78
	ANEXO A – FUNÇÕES DO <i>BENCHMARK</i> CEC2013.	82

1 INTRODUÇÃO

O planejamento de caminho consiste em determinar uma sequência de pontos ou uma curva no espaço que permita ao robô se deslocar de sua posição atual até o destino desejado, enquanto evita possíveis obstáculos no percurso (Lynch; Park, 2017). Nesse contexto, a visão computacional, que é uma das áreas da Inteligência Artificial (IA), desempenha um papel essencial ao fornecer dados sobre o ambiente. Essa área capacita os robôs a interpretar imagens capturadas por câmeras de vídeo, transformando essas informações visuais em dados utilizáveis (Milano; Honorato, 2010). Um sistema de visão computacional pode permitir o mapeamento de um ambiente em tempo real, identificando obstáculos e localizando o robô em relação ao seu objetivo. Essa tecnologia auxilia na criação de caminhos, garantindo que o robô seja capaz de navegar de forma autônoma em ambientes complexos (Mendiburu, 2013).

Um exemplo prático que ilustra a aplicação de visão computacional no planejamento de caminho é o IEEE *Very Small Size Soccer* (VSSS), uma competição que demonstra os avanços da robótica e da visão computacional. Nessa modalidade, dois times, compostos por três ou cinco robôs cada, se enfrentam em partidas de futebol sem qualquer controle humano direto. Esses robôs, conhecidos como robôs futebolistas, são guiados por computadores que analisam as imagens captadas por uma câmera posicionada acima do campo e determinam as ações de cada robô do time (IEEE VSSS, 2024). Na Figura 1 é ilustrado o ambiente de uma partida, onde é possível visualizar as câmeras e os robôs de ambos os times no campo.

Figura 1 – Ambiente de uma partida da categoria VSSS.



Fonte: Ferreira *et al.*, 2023.

Na categoria VSSS, os robôs futebolistas possuem dimensões reduzidas, com diâmetro máximo de 7,5 cm e altura de até 7,5 cm, projetados para competir em um campo de apenas 1,5 m x 1,3 m. Na Figura 2, é possível visualizar o campo e esses robôs disputando uma partida que ocorreu na Competição Latino Americana e Brasileira de Robótica, realizada em Uberlândia no ano de 2015.

Figura 2 – Partida disputada na categoria VSSS durante a Competição Latino Americana e Brasileira de Robótica realizada na cidade de Uberlândia em 2015.



Fonte: <https://tinyurl.com/mpevdbx5>.

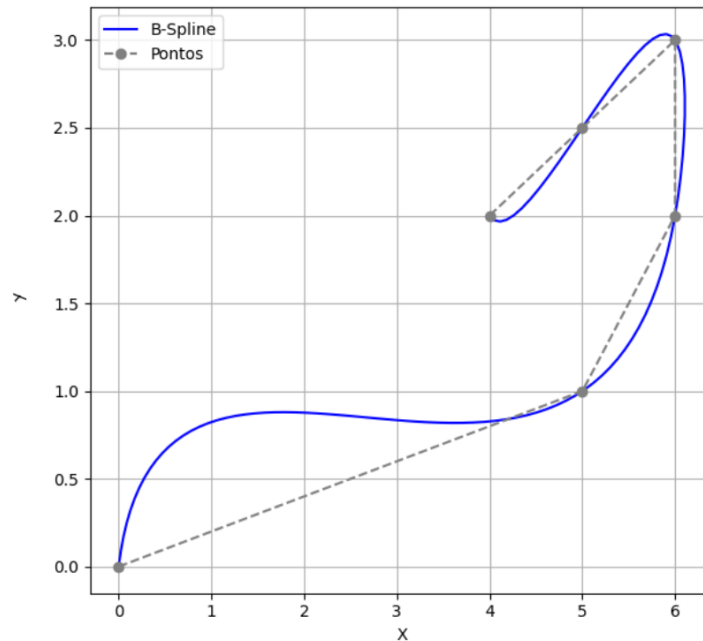
O planejamento de caminho para os robôs futebolistas apresenta desafios significativos, exigindo que eles se desloquem até a bola com a orientação correta voltada para o gol adversário. Durante esse percurso, é fundamental buscar o menor caminho possível, respeitando as restrições impostas pela curvatura máxima que o robô pode alcançar, sem ultrapassar os limites do campo. Além disso, o robô deve evitar colisões com outros robôs.

Na literatura, encontra-se duas principais representações de um caminho: (i) por meio de um conjunto de pontos, apresentado em (Ji *et al.*, 2023) e (Ganesan; Ramalingam; Mohan, 2024), e (ii) por meio de curvas paramétricas, apresentado em (Nguyen *et al.*, 2021) e (Riboli *et al.*, 2023). No primeiro caso são gerados caminhos discretos que podem ser úteis, mas que não garantem suavidade. Para superar essa limitação, são utilizadas as representações baseadas no segundo caso, como as curvas splines, B-splines e NURBS (*Non-Uniform Rational B-Spline*). Essas representações permitem a criação de caminhos suaves e contínuos, reduzindo, por exemplo, movimentos bruscos ao longo do percurso e, conseqüentemente, o desgaste dos atuadores. Por esse motivo, tais curvas são utilizadas com frequência (Ferreira; Carvalho, 2012).

Na Figura 3, é possível visualizar, em um exemplo genérico, a diferença entre uma

representação por pontos e uma representação por curva parametrizada. Note que, na representação por pontos, o caminho é marcado por coordenadas específicas que indicam locais de interesse ou destinos intermediários. Essa abordagem é simples e direta, mas não inclui informações sobre o comportamento entre os pontos, como suavidade ou continuidade. Por outro lado, a curva B-Spline garantiu uma transição gradual entre os pontos, apresentando um caminho suave e sem mudanças abruptas de direção.

Figura 3 – Pontos x Curva Parametrizada.



Fonte: Elaborada pelo autor, 2025.

Um dos métodos de navegação mais comuns para robôs móveis, devido ao seu baixo custo computacional, é baseado em campos vetoriais/potenciais, que utilizam vetores para indicar a direção e a magnitude do movimento em cada ponto do espaço, por meio da implementação de funções matemáticas, para atrair o robô ao objetivo e repelir de obstáculos, como apresentado em (Santos; Filho; Freitas, 2019), (Rosa; Rosa, 2021) e (Alam *et al.*, 2022). Além disso, existem estudos que utilizam outras técnicas, como espumas probabilísticas em (Nascimento *et al.*, 2021), que modelam o espaço de navegação como um conjunto de estados probabilísticos, onde cada região do espaço possui uma probabilidade associada de ser transitável ou de conter obstáculos. Já os *road-maps* em (Ayawli *et al.*, 2019) utilizam um Diagrama de Voronoi, no qual o planejador emprega o algoritmo A* para identificar o menor caminho. No entanto, há uma tendência crescente em explorar soluções baseadas em IA, que aplicam, por exemplo, algoritmos de aprendizado de máquina e redes neurais para adaptar e otimizar a

navegação em ambientes dinâmicos, como demonstrado em (Yan, 2020) e (Li *et al.*, 2024).

Ainda com relação às soluções baseadas em IA, os Algoritmos Evolucionários (AEs) também têm se destacado como uma técnica amplamente utilizada, devido à sua capacidade de encontrar soluções robustas em espaços de busca complexos (Vikhar, 2016). Exemplos de aplicação incluem o trabalho em (Freitas; Cohen; Guimarães, 2023), onde os AEs são utilizados para o planejamento de caminhos no futebol de robôs por meio da implementação de dois novos planejadores, DE-Spline e DE-NURBS, e em (Freitas *et al.*, 2024), onde essa abordagem é explorada na navegação autônoma de drones de asa fixa, através da implementação do planejador DE3D-NURBS, demonstrando a versatilidade e eficácia dos AEs em diferentes contextos robóticos.

Nos AEs, é importante destacar os operadores de mutação (mutadores), que desempenham um papel crucial. Eles são responsáveis por introduzir diversidade na população de soluções, ajudando a explorar novas regiões do espaço de busca e a evitar a convergência prematura (Trivedi *et al.*, 2017). Apesar da vasta aplicação de AEs na literatura, ainda há poucas análises sobre o impacto de diferentes operadores de mutação. Algumas comparações entre operadores de mutação já foram realizadas, como em (Opara; Arabas, 2018), onde os mutadores foram avaliados com base nas funções do *Congress on Evolutionary Computation* (CEC) 2013, que constituem um conjunto de *benchmarks* (testes padronizados para avaliar o desempenho de algoritmos) amplamente utilizado na área.

No entanto, falta na literatura um estudo mais específico do efeito dos operadores de mutação no contexto de planejamento de caminho. Em geral, diferentes operadores de mutação possuem características distintas e são mais adequados para diferentes tipos de problemas (Gong; Cai, 2013). Este trabalho justifica-se pela importância de analisar o desempenho de algoritmos evolucionários, especificamente, algoritmos diferenciais evolutivos, aplicados ao planejamento de caminho, com foco na avaliação de diferentes operadores de mutação. Assim, este trabalho pode contribuir na aplicação de técnicas de otimização evolucionária na robótica móvel.

Nesse contexto, este trabalho propõe adaptar e testar o desempenho de um planejador, com foco na comparação de diferentes operadores de mutação e sua influência no desempenho do planejamento de caminhos no contexto específico do futebol de robôs, formulado como um problema de otimização.

1.1 Objetivos

1.1.1 *Objetivo geral*

O objetivo geral do trabalho é analisar a influência dos operadores de mutação no desempenho de algoritmos diferenciais evolutivos, aplicado ao planejamento de caminho de robôs futebolistas.

1.1.2 *Objetivos específicos*

Como objetivos específicos, pode-se citar:

- Confirmar a influência de operadores de mutação aplicados a problemas de otimização irrestritos;
- Comparar o desempenho de operadores de mutação aplicados ao problema de planejamento de caminho;
- Comparar o desempenho de algoritmos diferenciais evolutivos aplicados ao planejamento de caminho;

1.2 Organização do Texto

Este trabalho está organizado da seguinte forma: O Capítulo 1 apresentou a introdução deste trabalho, destacando o contexto, a motivação e os objetivos gerais e específicos a serem alcançados. O Capítulo 2 apresenta o Referencial Teórico, onde são discutidos os conceitos fundamentais relacionados a robôs móveis, planejamento de caminho, AEs e operadores de mutação, servindo como base para o desenvolvimento do estudo. Em seguida, o Capítulo 3 descreve os métodos adotados para a análise do desempenho dos AEs aplicados ao planejamento de caminho. Já o Capítulo 4 apresenta os resultados obtidos, incluindo comparações entre algoritmos e operadores de mutação, além da avaliação das métricas de desempenho. Por fim, no Capítulo 5, são apresentadas as conclusões do trabalho, assim como sugestões para trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste capítulo é apresentado o referencial teórico necessário para o desenvolvimento deste trabalho. Na Seção 2.1, são abordados os conceitos fundamentais sobre robôs móveis, destacando sua importância e aplicações. Em seguida, a Subseção 2.1.1 detalha o modelo cinemático do robô, com foco nos robôs diferenciais utilizados em futebol de robôs. A Seção 2.2 explora o planejamento de caminho, discutindo técnicas como métodos vetoriais e probabilísticos. A Seção 2.3 discute o método dos algoritmos evolucionários e apresenta alguns dos mais comuns: DE, SaDE, JADE, SHADE, L-SHADE. Por fim, a Seção 2.4 discute os operadores de mutação em algoritmos diferenciais evolucionários, abordando suas características e aplicações no planejamento de caminho.

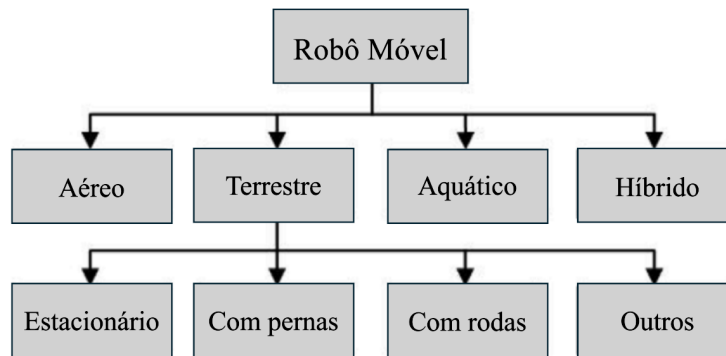
2.1 Robôs móveis

A robótica móvel é uma área de pesquisa que se dedica ao estudo e desenvolvimento de técnicas para os robôs que são capazes de se mover em diferentes tipos de ambientes, sejam eles terrestres, aquáticos ou aéreos (Santos; Filho; Freitas, 2019).

Com o avanço acelerado da economia comercial e da manufatura inteligente, a demanda por robôs móveis (RMs) flexíveis, adaptáveis e inteligentes está aumentando rapidamente (Xu *et al.*, 2024). No começo, os RMs eram restritos ao setor de manufatura, mas hoje, eles são amplamente utilizados em diversas áreas, incluindo entretenimento, medicina, mineração, exploração espacial, agricultura e outras (Patle *et al.*, 2019). Exemplos de aplicação em algumas dessas áreas incluem (Mireles *et al.*, 2022), (Gomez *et al.*, 2023) e (Šljajpah; Munih; Mihelj, 2023).

A Figura 4 apresenta uma classificação geral dos RMs com base no ambiente e no método de interação.

Figura 4 – Classificação dos RMs por ambiente e método de interação.



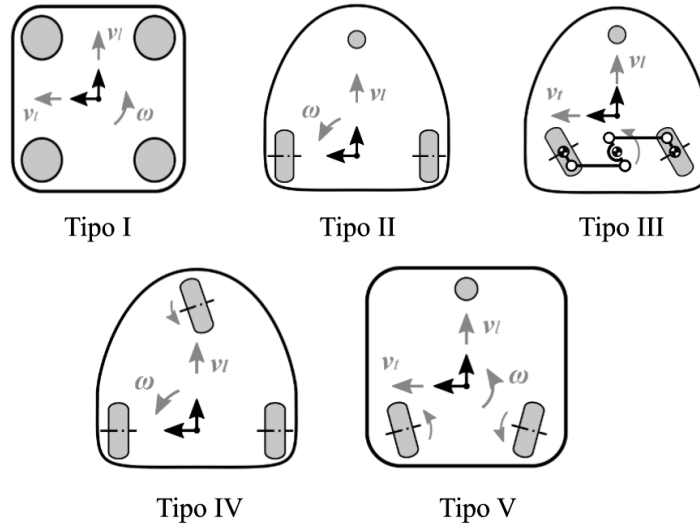
Fonte: Adaptado de Ullah *et al.*, 2024.

Quanto aos RMs terrestres com rodas, Tagliavini *et al.* (2022) resume os cinco tipos principais em que podem ser classificados, cada um com características distintas que influenciam sua mobilidade. São eles:

- Tipo I - Robôs omnidirecionais sem rodas direcionais: Esses robôs utilizam rodas especiais, como rodas esféricas, para alcançar mobilidade omnidirecional no plano de movimento, ou seja, eles podem se mover em qualquer direção sem precisar reorientar seu corpo, pois a direção do movimento é independente da orientação do robô;
- Tipo II - Robôs com rodas fixas em um eixo comum: Esses robôs não possuem rodas direcionais, mas têm duas ou mais rodas montadas em um eixo comum, com a possibilidade de controlar suas velocidades de forma independente. Um exemplo clássico é o robô diferencial;
- Tipo III - Robôs com rodas direcionais alinhadas: Esses robôs possuem pelo menos uma roda direcional que pode girar para ajustar o movimento. Se houver mais de uma roda direcional, todas apontam para a mesma direção, limitando a mobilidade a um plano 2D;
- Tipo IV - Robôs com rodas fixas e direcionais combinadas: Esses robôs possuem uma combinação de rodas fixas em um eixo comum e rodas direcionais separadas. Exemplos incluem robôs que utilizam o sistema Ackerman, como carros ou bicicletas;
- Tipo V - Robôs pseudo-omnidirecionais com rodas direcionais independentes: Esses robôs possuem pelo menos duas rodas direcionais que podem ser controladas de forma independente, permitindo reconfigurações para alcançar diferentes tipos de mobilidade.

Na Figura 5 são apresentados alguns exemplos dos cinco tipos de RMs terrestres com rodas, ilustrando graficamente as diferenças de configuração das rodas e os vetores de movimento correspondentes para cada tipo de robô.

Figura 5 – Exemplo de tipos de RMs terrestres com rodas.



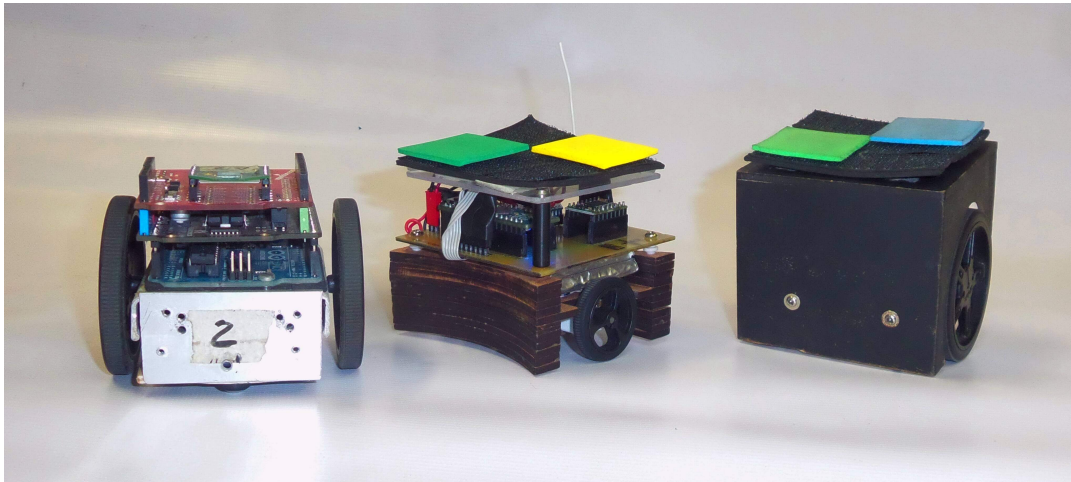
Fonte: Adaptada de Tagliavini *et al.*, 2022.

2.1.1 Modelo cinemático do robô

Os robôs futebolistas são do tipo robôs diferenciais (Tipo II - Robôs com rodas fixas em um eixo comum), que apresentam características de não-holonomia. Isso significa que não é possível que o robô execute movimentos perpendiculares à superfície de suas rodas, ou seja, eles não conseguem se mover lateralmente sem primeiro girar (Ramos *et al.*, 2021). Por exemplo, para se deslocar para a direita, o robô precisa primeiro girar para a direita e depois avançar, em vez de simplesmente deslizar para o lado.

Esses robôs utilizam duas rodas paralelas para locomoção e a diferença de velocidade entre elas permite controlar tanto a velocidade linear quanto a angular, além de ajustar sua orientação no campo. Por exemplo, quando as rodas giram com a mesma velocidade, mas em sentidos opostos, o robô realiza uma rotação em torno de seu próprio eixo, sem se deslocar. Já quando as rodas giram com velocidades e direções iguais, o robô se move em linha reta, conforme um modelo ideal (Siegwart; Nourbakhsh; Scaramuzza, 2011). Na Figura 6 são apresentados três modelos diferentes de robôs futebolistas da categoria VSSS, onde é possível observar as rodas paralelas que destacam as características de não-holonomia, típicas de robôs diferenciais.

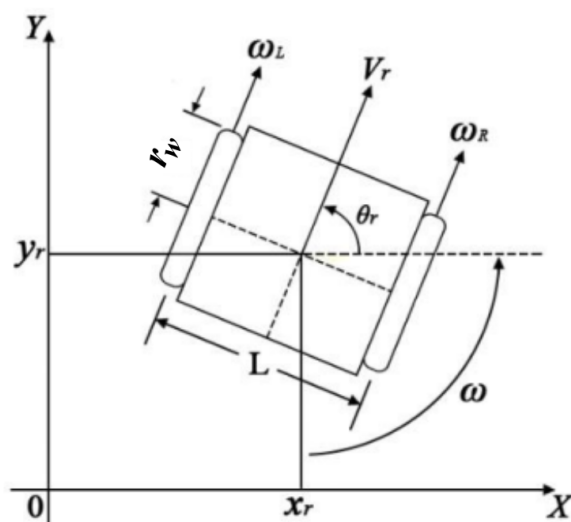
Figura 6 – Robôs Futebolistas.



Fonte: <https://sirlab.github.io/vss.html>.

Já na Figura 7, é possível compreender melhor as variáveis relacionadas à cinemática de um robô diferencial. Nessa representação, r_w corresponde ao raio da roda, L é a distância entre as rodas, x_r e y_r indicam a posição do centro do robô no plano do campo, θ_r representa a orientação angular do robô, V_r é a velocidade linear, e ω é a velocidade angular. Essas variáveis são manipuladas para determinar as velocidades angulares necessárias para as rodas esquerda (ω_L) e direita (ω_R), permitindo o controle preciso da direção e do movimento do robô (Oliveira *et al.*, 2022). O modelo cinemático do robô diferencial pode ser definido pela Equação 2.1.

Figura 7 – Esquema do Robô Diferencial.



Fonte: Adaptado de Oliveira *et al.*, 2022.

$$\begin{bmatrix} v_x \\ v_y \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos \theta_r & 0 \\ \sin \theta_r & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} V_r \\ \omega \end{bmatrix} \quad (2.1)$$

O raio mínimo de curvatura (p_{\min}) de um robô diferencial é determinado pela diferença máxima entre as velocidades das duas rodas motrizes do robô. A restrição de curvatura está diretamente ligada ao controle dessas velocidades e à capacidade do robô de realizar curvas. O significado prático é que o robô limita-se a realizar curvas com um raio mínimo específico quando está em movimento, o que afeta sua manobrabilidade em situações que exigem mudanças rápidas de direção, como no futebol de robôs. Para curvas mais fechadas, o robô precisa aumentar a diferença de velocidade entre as rodas, o que pode ser limitado pelas características dos motores e pela estrutura física do robô.

Para um robô diferencial, o caminho curvo é obtido através do controle da velocidade angular e da velocidade linear, conforme a Equação 2.2, onde R é o raio instantâneo do caminho.

$$R = \frac{V_r}{\omega} \quad (2.2)$$

A curvatura máxima (κ_{\max}) que o robô pode alcançar é expressa pela Equação 2.3, onde ω_{\max} representa a velocidade angular máxima, enquanto $V_{r_{\min}}$ indica a menor velocidade linear possível que o robô consegue manter.

$$\kappa_{\max} = \frac{1}{p_{\min}} = \frac{\omega_{\max}}{V_{r_{\min}}} \quad (2.3)$$

Esse limite de curvatura deve ser considerado no controle de robôs futebolistas da categoria VSSS, pois impacta diretamente na agilidade e na capacidade de realizar movimentos táticos.

2.2 Planejamento de caminho

Os robôs móveis também podem ser classificados pela autonomia com que se movem em: (i) teleoperados: totalmente dependentes dos humanos; (ii) semiautônomos: parcialmente dependentes, com o robô cuidando da segurança da navegação; (iii) autônomos: totalmente independentes, executando todas as tarefas sem intervenção humana (Romero *et al.*, 2017). No caso dos robôs autônomos, como os robôs futebolistas, são necessários três elementos essenciais para sua operação: percepção, planejamento de caminho e controle (Ghallabi, 2020).

Este trabalho tem como foco um desses elementos essenciais para a condução autônoma: o Planejamento de caminho. Como mencionado brevemente na Seção 1, existem diferentes técnicas utilizadas para que se obtenha um caminho factível, isto é, um caminho que atenda o seu objetivo com êxito. As técnicas mais comuns encontradas são as baseadas em: (i) campos vetoriais/potenciais; (ii) métodos probabilísticos; e (iii) AEs, que, devido à sua relevância para este trabalho, são detalhados na Seção 2.3.

1. Campos vetoriais/potenciais:

Essa técnica consiste em direcionar os robôs móveis (RMs) até seus objetivos enquanto os afasta dos obstáculos, utilizando princípios semelhantes aos do eletromagnetismo. Nessa abordagem, os obstáculos são considerados como tendo a mesma carga (campos repulsivos), enquanto os objetivos possuem carga oposta (campos atrativos). Dessa forma, os robôs evitam contato entre si e com os obstáculos, ao mesmo tempo em que se direcionam de maneira eficiente para os objetivos (Rosa; Rosa, 2021).

Na Figura 8 é apresentado um campo vetorial atrativo aplicado em um campo de futebol de robôs, onde é possível observar os vetores, representados pelas setas de cor azul, que atraem o robô para a direção e a orientação com que o deve se movimentar para seguir o campo de forma a direcionar a bola, representada pela cor laranja, direto para o gol, retângulo cinza.

Figura 8 – Exemplo de um campo vetorial Atrativo aplicado ao VSSS.

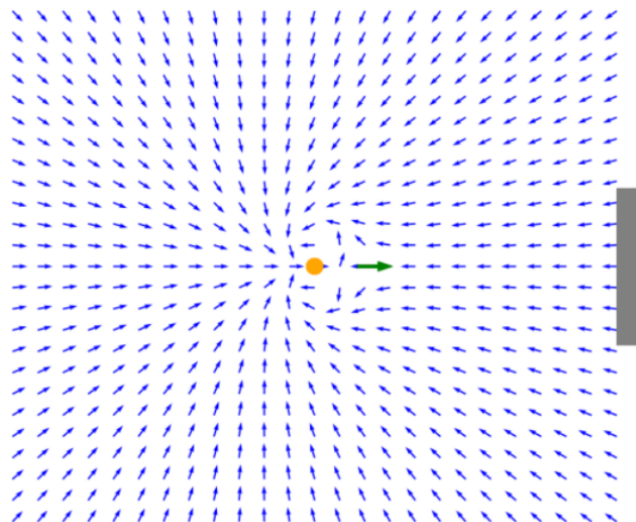
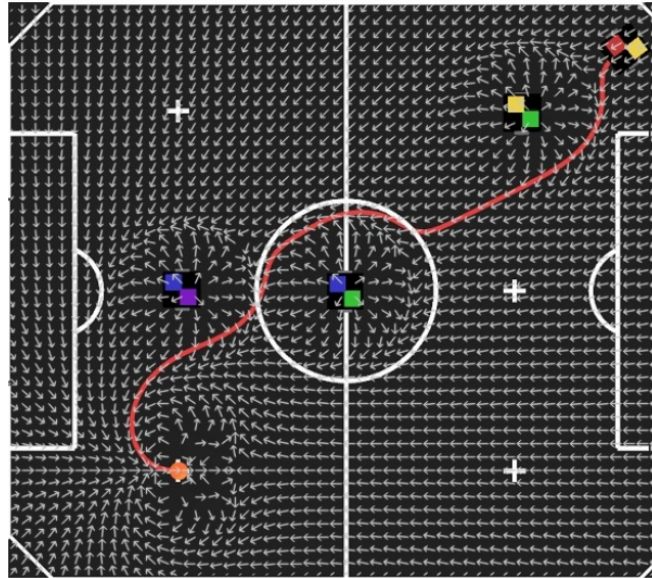


Figura 9 – Exemplo de um campo vetorial atrativo e repulsivo aplicado ao VSSS.



Fonte: Santos; Filho; Freitas, 2019.

Já na Figura 9 é ilustrado um conjunto de campos vetoriais atrativos e repulsivos também aplicado ao futebol de robôs. O campo vetorial repulsivo atua de forma oposta ao atrativo, evitando que o robô colida com obstáculos. Enquanto o campo vetorial atrativo "puxa" o robô em direção à bola, os campos repulsivos "empurram" o robô para longe dos obstáculos. Nessa figura, ainda é possível observar o robô seguindo o caminho vermelho em direção à bola, desviando dos outros robôs ao contornar os campos repulsivos.

2. Métodos probabilísticos:

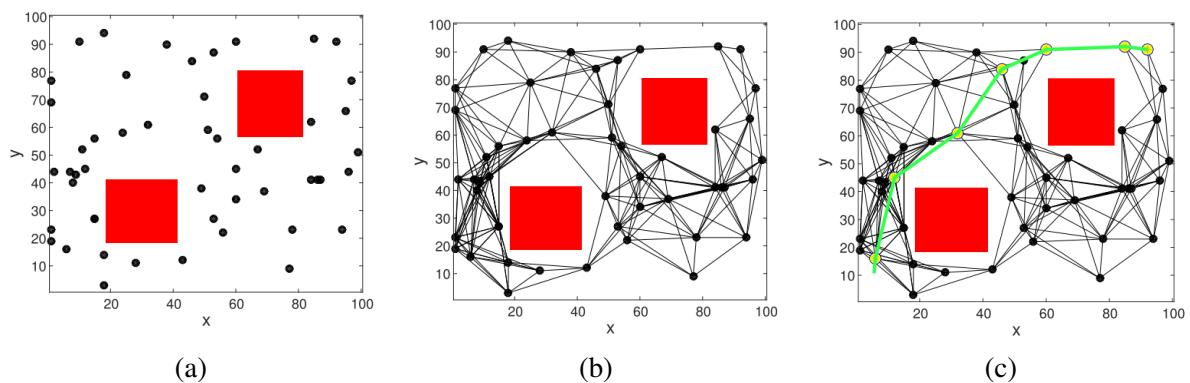
Os métodos probabilísticos para planejamento de caminho são amplamente utilizados em robótica devido à sua capacidade de lidar com a complexidade de ambientes de alta dimensionalidade. Uma característica central desses métodos é o uso de amostragem aleatória para explorar o espaço de configuração do robô, identificando soluções factíveis de forma eficiente. Essa abordagem é útil em cenários onde métodos determinísticos podem ser computacionalmente inviáveis.

Um exemplo de método utilizado é o *Probabilistic Roadmaps* (PRM), que se destaca pela sua capacidade de lidar com a complexidade de ambientes robóticos. O PRM, em particular, constrói um grafo de conectividade no espaço de configuração livre do robô por meio de amostragem aleatória.

Na fase de construção, o método distribui nós aleatoriamente no espaço de configuração livre do robô, ou seja, na região onde não há colisões, conforme Figura 10a. Esses nós são

conectados por planejadores locais para formar um grafo de conectividade, conhecido como mapa rodoviário, conforme ilustrado na Figura 10b. Esse grafo representa possíveis caminhos viáveis no espaço livre e é refinado ao longo da construção. Na fase de consulta, o PRM conecta os pontos inicial e final a esse mapa rodoviário, Figura 10c, buscando um caminho viável através dele por meio de algoritmos como A* e Dijkstra (Ravankar *et al.*, 2020).

Figura 10 – Grafo de conectividade do método PRM. (a) Distribuição aleatória dos nós; (b) Grafo de conectividade ; (c) Consulta de caminho viável.



Fonte: Ravankar *et al.*, 2020.

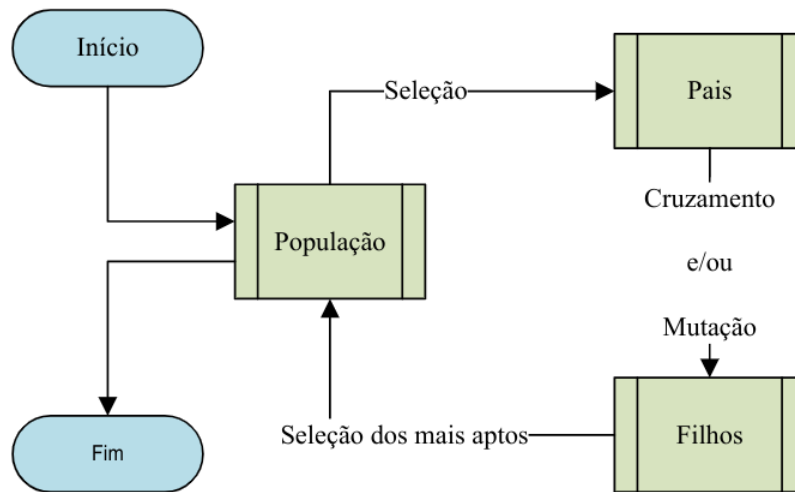
2.3 Algoritmos Evolucionários

Os AEs são uma classe de algoritmos meta-heurísticos amplamente utilizados para solucionar problemas complexos de otimização (NP-hard). Esses algoritmos são eficientes em encontrar soluções aproximadas em situações onde métodos tradicionais determinísticos se mostram ineficazes ou inviáveis (Vikhar, 2016). Baseados nos princípios de seleção natural e evolução biológica, os AEs replicam o processo evolutivo observado na natureza por meio de operações como seleção, cruzamento e mutação (Chen; Chiang, 2015).

Na Figura 11, é ilustrado o esquema geral dos AEs. O processo começa com uma população inicial formada por um conjunto de soluções candidatas geradas de maneira aleatória. A qualidade de cada solução é avaliada por meio de uma função de custo, que mede sua adequação (*fitness*) ao problema. Com base nessa avaliação, os melhores indivíduos (pais) são selecionados para criar novas gerações (filhos), repetindo o ciclo até que seja encontrada uma solução que satisfaça o critério de parada do algoritmo. Para gerar os filhos, utilizam-se ope-

radores como o cruzamento, que combina duas ou mais soluções existentes, e/ou a mutação, que modifica uma solução para produzir uma nova variante (Eiben; Smith, 2015).

Figura 11 – Esquema Geral dos AEs.



Fonte: Elaborada pelo autor.

2.3.1 Algoritmos Diferenciais Evolutivos

Os Algoritmos Diferenciais Evolutivos (DEs) são reconhecidos como uma das abordagens mais eficientes dentro dos AEs para otimização numérica global, sendo amplamente aplicados com sucesso em diversas áreas, como engenharia de *design*, processamento de imagens, mineração de dados, entre outras (Gong; Cai, 2013). Diferentemente de outros tipos de AEs, como os Algoritmos Genéticos (AGs), os DEs apresentam em geral maior velocidade de convergência, utilizam operadores simples, como soma e subtração de vetores, e requerem um número reduzido de hiperparâmetros (Silva; Souza; Freitas, 2024). Os principais hiperparâmetros desses algoritmos são o tamanho da população N , o fator de escala F e a taxa de cruzamento CR , sendo que as configurações ideais variam de acordo com o problema (Tanabe; Fukunaga, 2014).

Existem diversos algoritmos DEs, cada um com características e adaptações específicas. Alguns exemplos desses algoritmos são:

1. DE (*Differential Evolution*):

O algoritmo DE, proposto por Storn e Price (1997), é amplamente reconhecido como o precursor dos algoritmos DEs, servindo como a base para inúmeras variações e apri-

moramentos desenvolvidos posteriormente e se tornando uma referência no campo da otimização evolutiva.

De acordo com Mohamed, Hadi e Jambi (2019), no DE, a população inicial é gerada aleatoriamente dentro dos limites do espaço de busca. A cada geração, o DE evolui seus indivíduos (vetores alvo) usando três operadores principais: mutação, cruzamento e seleção. A mutação cria um vetor mutante a partir do vetor alvo, perturbando ele com vetores diferença entre outros indivíduos da população. O fator de escala F controla a amplitude dessas perturbações e o CR combina os hiperparâmetros do vetor mutante com o vetor alvo para gerar um vetor de teste. A seleção compara o vetor de teste com o vetor alvo original, mantendo na próxima geração o vetor com melhor valor de aptidão.

Mohamed, Hadi e Jambi (2019) também ressalta que o desempenho do DE é dependente da estratégia de mutação e do operador de cruzamento, bem como dos hiperparâmetros de controle. Esses hiperparâmetros afetam o equilíbrio entre a diversidade da população e a velocidade de convergência. Além disso, é enfatizado que o DE tem boa capacidade de exploração global, que consiste em percorrer amplamente o espaço de busca para encontrar regiões promissoras, mas pode apresentar lentidão na exploração, que corresponde ao refinamento das soluções dentro dessas regiões.

2. SaDE (*Self-Adaptive Differential Evolution*):

Conforme Qin e Suganthan (2005), o SaDE é uma variação avançada do algoritmo DE, que busca otimizar problemas numéricos sem a necessidade de pré-definir parâmetros críticos. Enquanto o DE tradicional depende muito do ajuste dos parâmetros F e CR , além de qual estratégia de aprendizagem aplicar, o SaDE supera isso usando uma abordagem de auto-adaptação. Ele incorpora múltiplas estratégias de aprendizagem e ajusta dinamicamente a probabilidade de cada estratégia ser utilizada, com base em seu desempenho histórico dentro da população em evolução.

Ainda segundo Qin e Suganthan (2005), o SaDE adapta os parâmetros F e CR ao longo das gerações, em vez de usar valores fixos. F é gerado aleatoriamente de uma distribuição normal e o CR é ajustado com base em um acúmulo dos valores de CR que tiveram sucesso anteriormente. Ao se ajustar continuamente ao longo da busca, o SaDE melhora a sua flexibilidade para encontrar soluções ótimas em uma ampla gama de problemas com diferentes características. O algoritmo SaDE se destaca pela capacidade de otimizar tanto a busca global quanto a busca local, sem o ajuste manual dos parâmetros, tornando-o um método mais eficiente e acessível para problemas complexos de otimização.

3. JADE (*Adaptive Differential Evolution with Optional External Archive*):

Em vez de usar apenas a melhor solução da população atual, como o DE, o JADE seleciona aleatoriamente uma solução entre as melhores, fornecendo uma abordagem mais diversificada e exploratória. Além disso, o JADE incorpora um arquivo externo opcional para armazenar soluções inferiores exploradas recentemente. Ao considerar a diferença entre essas soluções e a população atual durante a mutação, o JADE busca direcionar a busca de forma mais eficaz, ao mesmo tempo em que mantém a diversidade da população (Zhang; Sanderson, 2009).

Um componente crucial do JADE é o seu mecanismo de adaptação de hiperparâmetros. F é gerado a partir de uma distribuição Cauchy truncada e CR a partir de uma distribuição normal truncada. As médias dessas distribuições são atualizadas com base no sucesso das gerações anteriores, favorecendo valores que resultaram em melhorias na função objetivo. Essa adaptação automática evita que o usuário tenha que ajustar esses parâmetros manualmente e contribui para a robustez e o bom desempenho do algoritmo em diferentes problemas de otimização (Zhang; Sanderson, 2009).

4. SHADE (*Success History based Adaptive Differential Evolution*)

De acordo com Tanabe e Fukunaga (2013), o SHADE é uma evolução do algoritmo JADE para otimização numérica, que visa melhorar a adaptação de seus hiperparâmetros de controle. O SHADE emprega um histórico de valores de CR e F que obtiveram sucesso em iterações anteriores. Esse histórico, armazenado em memórias MCR e MF , direciona a geração de novos valores de CR e F , permitindo explorar o espaço de parâmetros de forma mais eficaz. A cada geração, os parâmetros usados por indivíduos bem-sucedidos são registrados, e o histórico é atualizado com esses valores. Um índice (k) é usado para controlar qual elemento do histórico é atualizado em cada geração, de forma que a influência de parâmetros mais antigos vá diminuindo ao longo do tempo.

Além disso, o SHADE incorpora uma melhoria na estratégia de mutação do JADE, introduzindo um parâmetro p ajustável. Ao invés de usar um valor fixo para p , o SHADE ajusta esse parâmetro individualmente para cada membro da população durante cada geração (Tanabe; Fukunaga, 2013).

5. L-SHADE (*Linear Success History based Adaptive Differential Evolution*)

O L-SHADE é uma melhoria do algoritmo SHADE, que incorpora uma redução linear do tamanho da população. O L-SHADE reduz continuamente o tamanho da população ao longo do processo de busca, seguindo uma função linear pré-definida. Essa redução gradual do tamanho da população visa acelerar a convergência do algoritmo e reduzir o risco de estagnação em ótimos locais. A cada geração, o L-SHADE calcula o novo

tamanho da população usando uma fórmula que considera o número atual de avaliações da função objetivo, o número máximo de avaliações permitidas e o tamanho mínimo da população (Tanabe; Fukunaga, 2014).

2.4 Operadores de Mutação em algoritmos Diferenciais Evolutivos

Nos algoritmos DEs, os operadores de mutação desempenham um papel importante para equilibrar a exploração do espaço de busca (procurar por novas regiões promissoras) e a exploração (refinar soluções já encontradas) (Silva; Souza; Freitas, 2024). A notação geral para identificar esses operadores é padronizada como "a/b". Aqui, "a" indica o vetor a ser mutado e "b" especifica o número de vetores de diferença utilizados (Gong; Cai, 2013).

Alguns dos operadores de mutação mais utilizados são descritos em (Opara; Arabas, 2018). São eles:

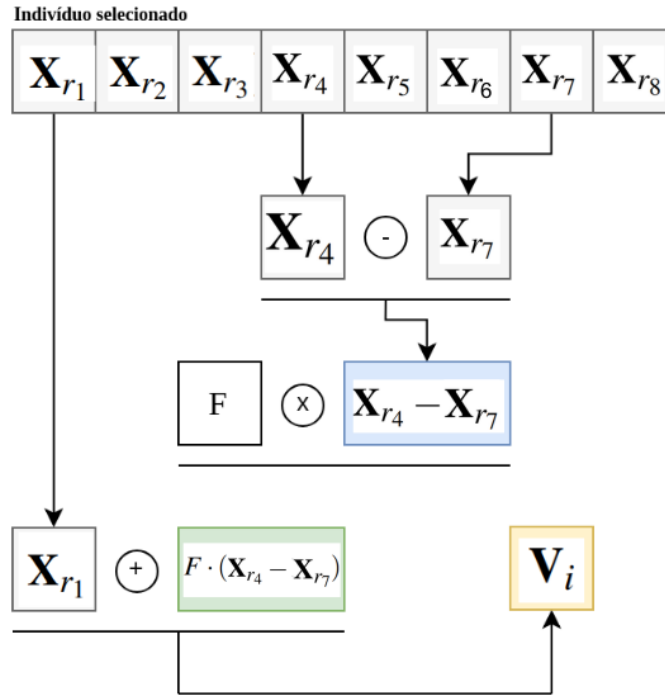
1. rand/1:

O método consiste em somar ao vetor base \mathbf{X}_{r_1} , escolhido aleatoriamente, um vetor de diferença escalado $F \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3})$, que é calculado a partir da diferença entre dois outros indivíduos \mathbf{X}_{r_2} e \mathbf{X}_{r_3} , também selecionados de forma aleatória. O parâmetro F é denominado fator de escala.

$$\mathbf{V}_i = \mathbf{X}_{r_1} + F \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}); \quad (2.4)$$

Na Figura 12 é mostrado de forma gráfica a Equação 2.4 a fim de melhor exemplificar. Nessa figura são selecionados três indivíduos aleatórios (X_{r_1} , X_{r_4} , e X_{r_7}) de uma população formada por oito indivíduos. Essa mesma lógica segue para os demais mutadores, cada um com suas respectivas especificidades.

Figura 12 – Exemplo gráfico do operador de mutação "rand/1". A população possui 8 indivíduos e três foram escolhidos aleatoriamente: 1, 4 e 7.



Fonte: Adaptado de <https://tinyurl.com/pj3m3azh>.

2. rand/2:

Segue o mesmo princípio do DE/rand/1, mas difere ao somar dois vetores de diferença escalados a um indivíduo selecionado aleatoriamente.

$$\mathbf{V}_i = \mathbf{X}_{r_1} + F_1 \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}) + F_2 \cdot (\mathbf{X}_{r_4} - \mathbf{X}_{r_5}); \quad (2.5)$$

3. best/1:

Nesse operador, o vetor base é definido como o indivíduo mais apto (X_{best}) da população na geração atual.

$$\mathbf{V}_i = \mathbf{X}_{best} + F \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}); \quad (2.6)$$

4. best/2:

Semelhante ao DE/best/1, mas soma dois vetores de diferença escalados ao X_{best} .

$$\mathbf{V}_i = \mathbf{X}_{best} + F_1 \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}) + F_2 \cdot (\mathbf{X}_{r_4} - \mathbf{X}_{r_5}); \quad (2.7)$$

5. either-or:

Esse operador funciona de duas maneiras. Com probabilidade p_f , os mutantes são gerados seguindo o método DE/rand/1. Caso contrário, um ponto aleatório X_{r_1} é modificado pela soma escalada de dois vetores de diferença, calculados a partir de X_{r_2} , X_{r_3} e X_{r_1} , onde $K = \frac{F+1}{2}$.

$$\mathbf{V}_i = \begin{cases} \mathbf{X}_{r_1} + F \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}), & \text{se } \text{rand}(0, 1) < p_f, \\ \mathbf{X}_{r_1} + K \cdot (\mathbf{X}_{r_2} + \mathbf{X}_{r_3} - 2\mathbf{X}_{r_1}), & \text{senão;} \end{cases} \quad (2.8)$$

6. current-to-rand/1:

Neste operador de mutação, o vetor base é igual ao vetor do pai (X_i). A ele são somados dois vetores de diferença: um que é gerado aleatoriamente e outro que é a diferença entre um indivíduo aleatório da população e X_i .

$$\mathbf{V}_i = \mathbf{X}_i + F_1 \cdot (\mathbf{X}_{r_1} - \mathbf{X}_i) + F_2 \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}); \quad (2.9)$$

7. current-to-best/1:

Nesse operador, o vetor base também é o (X_i). A ele são adicionados dois vetores de diferença: um selecionado de forma aleatória e outro que representa a diferença entre X_{best} e X_i .

$$\mathbf{V}_i = \mathbf{X}_i + F_1 \cdot (\mathbf{X}_{best} - \mathbf{X}_i) + F_2 \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}); \quad (2.10)$$

8. current-to-pbest/1:

Esse operador é semelhante ao DE/current-to-best/1. A ele são adicionados dois vetores de diferença: um selecionado de forma aleatória e outro que representa a diferença entre X_{pbest} e X_i . O vetor X_{pbest} é selecionado aleatoriamente entre uma porcentagem das melhores soluções da população.

$$\mathbf{V}_i = \mathbf{X}_i + F_1 \cdot (\mathbf{X}_{pbest} - \mathbf{X}_i) + F_2 \cdot (\mathbf{X}_{r_2} - \mathbf{X}_{r_3}). \quad (2.11)$$

3 MÉTODOS

Neste trabalho, como já mencionado anteriormente, o principal objetivo foi avaliar o desempenho de diferentes operadores de mutação aplicados a diferentes algoritmos DEs para o planejamento de caminhos no futebol de robôs.

Inicialmente, foi testada a abordagem em problemas de otimização irrestritos do *benchmark* CEC2013, como detalhado na Seção 3.1. Na Seção 3.2, é descrito a implementação dos algoritmos DEs e dos operadores de mutação utilizando a linguagem de programação Python e a biblioteca Mealpy aplicados ao problema de planejamento de caminho dos robôs futebolistas. Já a representação do caminho, é definida usando uma curva NURBS, conforme detalhado na Subseção 3.2.2. O problema de otimização, incluindo a função de custo e as restrições, é descrito na Subseção 3.2.3.

3.1 Exploração dos mutadores em *benchmark* genérica

Antes de explorar o uso de diversos operadores de mutação aplicados ao problema de planejamento de caminho, foi testado, inicialmente, a abordagem em problemas irrestritos do *benchmark* CEC2013. Esse conjunto de *benchmark* é formado por 28 funções, sendo dividido em: (i) Funções Unimodais, destinadas a avaliar a capacidade de exploração local, (ii) Funções Multimodais Básicas, projetadas para testar a habilidade de encontrar múltiplos ótimos, e (iii) Funções de Composição, que simulam cenários complexos e desafiadores (Liang *et al.*, 2013).

As funções presentes nesse conjunto podem ser visualizadas no Anexo A. As Equações 3.1 e 3.2 representam as Funções 9 e 11 desse *benchmark*, exemplificando um pouco esse conjunto de funções. Além disso, as Figuras 13 e 14 ilustra graficamente essas duas funções.

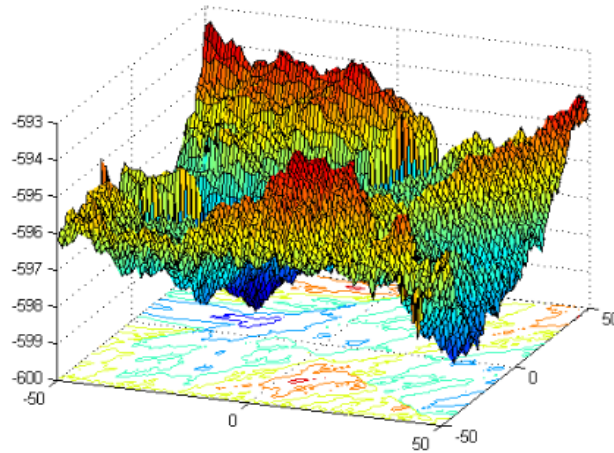
$$f_9(x) = \sum_{i=1}^D \left(\sum_{k=0}^{k_{\max}} \left[a^k \cos(2\pi b^k (z_i + 0,5)) \right] \right) - D \sum_{k=0}^{k_{\max}} \left[a^k \cos(2\pi b^k \cdot 0,5) \right] + b_9 \quad (3.1)$$

$$f_{11}(x) = \sum_{i=1}^D \left(z_i^2 - 10 \cos(2\pi z_i) + 10 \right) + b_{11} \quad (3.2)$$

Para realizar esse teste, foram utilizados três operadores de mutação apresentados na Seção 2.4. São eles: "rand/1" (Equação 2.4), "current-to-best/1" (Equação 2.10) e "current-to-pbest/1" (Equação 2.11). Esses mutadores foram integrados ao algoritmo L-SHADE, descrito na Subseção 2.3.1. Foram escolhidos esses três mutadores, pois são os mais comuns de serem

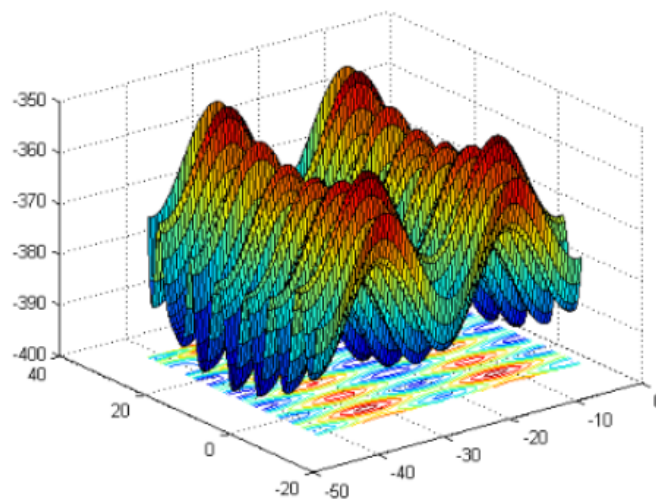
utilizados com o algoritmo L-SHADE. Além disso, os experimentos incluíram a otimização para diferentes dimensões, especificamente $D = 2, D = 10, D = 30$ e $D = 50$.

Figura 13 – Função 9 do *benchmark* CEC 2013.



Fonte: Liang *et al.*, 2013.

Figura 14 – Função 11 do *benchmark* CEC2013.



Fonte: Liang *et al.*, 2013.

Após o término das simulações e o armazenamento dos resultados, deu-se início a etapa de análise. Nesse processo, foram determinados, para cada problema, os valores médios e os desvios padrão dos melhores *fitness* encontrados, além de serem realizadas análises compara-

tivas percentuais entre essas médias. Adicionalmente, foram criados gráficos de diagramas de caixa que permitiram identificar, com base nos valores médios, quais operadores de mutação obtiveram o melhor desempenho no *benchmark* CEC2013.

3.2 Avaliação dos algoritmos Diferenciais Evolutivos para planejamento de caminho

3.2.1 Definição do problema

Conforme dito brevemente na Seção 1, os robôs futebolistas necessitam disputar as suas partidas de futebol, planejando os caminhos que devem percorrer de forma autônoma. Durante uma partida é necessário lidar com algumas restrições, como: (i) Alcançar a bola com a inclinação certa para o gol adversário, respeitando os limites de curvatura do robô diferencial por meio de curvas suaves; (ii) Desviar dos obstáculos, ou seja, dos demais robôs no campo; e (iii) Planejar um caminho que esteja dentro das limitações do campo.

Além disso, é importante destacar que o problema de planejamento de caminho com essas restrições, pode ser reformulado como o seguinte problema de minimização multiobjetivo sem restrições:

$$x^* = \arg \min f_T, \quad (3.3)$$

onde $x^* = [P_3, P_4, \dots, P_{N_p-4}, w_3, w_4, \dots, w_{N_p-4}, \lambda_1, \lambda_2]$ representa os parâmetros livres ótimos associados ao caminho, definidos com base na representação NURBS (*Non-Uniform Rational B-Spline*) e f_T é a função de custo, com ambos sendo apresentados com mais detalhes nas Subseções 3.2.2 e 3.2.3.

Para que esse problema do planejamento de caminho dos robôs futebolistas seja minimizado, foram utilizados os algoritmos DEs em conjunto com os operadores de mutação destacados anteriormente. O planejador DE-NURBS é utilizado a fim de verificar os parâmetros das curvas NURBS que foram otimizados e, por meio das visualizações dessas curvas, confirmar o cumprimento das restrições do problema.

3.2.2 Representação do caminho

Para representar um possível caminho $C(s)$, utilizado pelo robô futebolista para sair da sua localização e chegar até a bola com a inclinação para o gol, é utilizada uma curva

NURBS, definida como (Piegl; Tiller, 1996):

$$C(s) = \frac{\sum_{i=0}^{N_p-1} w_i \mathbf{P}_i B_{i,k}(s)}{\sum_{i=0}^{N_p-1} w_i B_{i,k}(s)}, \quad (3.4)$$

onde s é a variável paramétrica, $\mathbf{P}_i = [x, y, z]^T$ representa um ponto de controle na curva NURBS para o parâmetro $s \in [0, 1]$, $w_i \in \mathbb{R}_+^*$ é o peso atribuído a cada um dos $N_p \geq (k+1)$ pontos de controle pertencentes ao conjunto $\mathcal{P} = \{\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_{N_p-1}\}$. Além disso, $B_{i,k}(s)$ representa as funções base normalizadas de grau k , que são calculadas pelas Equações 3.5 e 3.6:

$$B_{i,k}(s) = \frac{s - v_i}{v_{i+k} - v_i} \cdot B_{i,k-1}(s) + \frac{v_{i+k+1} - s}{v_{i+k+1} - v_{i+1}} \cdot B_{i+1,k-1}(s), \quad (3.5)$$

$$B_{i,0}(s) = \begin{cases} 1, & \text{se } v_i \leq s < v_{i+1}, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.6)$$

O vetor de nós $\mathbf{v} = [v_0, v_1, \dots, v_M] \in \mathbb{R}^M$ é uma sequência de valores que não diminui. Ele contém $M = (k+1 + N_p)$ elementos, que variam entre 0 e 1. Esses nós dividem a curva em diferentes partes e, para que a curva comece e termine exatamente nos pontos de controle iniciais e finais, os primeiros $(k+1)$ elementos do vetor \mathbf{v} são definidos como 0. Da mesma forma, os últimos $(k+1)$ elementos do vetor são fixados como 1.

O caminho deve respeitar as restrições do robô, como a curvatura máxima (κ_{\max}). Como mencionado na subseção 2.1.1, a curvatura κ_{\max} está associada ao menor raio de curvatura p_{\min} que o robô consegue realizar em um plano bidimensional. A restrição pode ser analisada por meio da curvatura do caminho $\mathbf{C}(s)$, dada por:

$$\kappa(s) = \frac{\|\dot{\mathbf{C}}(s) \times \ddot{\mathbf{C}}(s)\|}{\|\dot{\mathbf{C}}(s)\|^3}, \quad (3.7)$$

onde $\dot{\square}$ e $\ddot{\square}$ representam, respectivamente, as derivadas de primeira e segunda ordem e $\|\square\|$ a norma vetorial. Essas derivadas são obtidas por meio da aproximação de primeira ordem de Euler. O número de pontos da curva, N_s , foi escolhido de modo a garantir uma precisão adequada na discretização, conforme utilizado em (Freitas *et al.*, 2024).

Já para garantir os ângulos de orientação inicial e final (γ_{init} , γ_{goal}), bem como uma curvatura zero nesses pontos, foram implementados pontos de controle também baseados em (Freitas *et al.*, 2024), porém adaptados para um modelo 2D, onde a coordenada z é sempre zero, como no caso do futebol de robôs. Esses pontos de controle são definidos como:

$$\mathbf{P}_1 = \mathbf{P}_0 + \frac{2}{3} \lambda_1 \begin{bmatrix} \cos \gamma_{\text{init}} \\ \sin \gamma_{\text{init}} \end{bmatrix}, \quad (3.8)$$

$$\mathbf{P}_2 = \mathbf{P}_1 + \frac{1}{3}\lambda_1 \begin{bmatrix} \cos \gamma_{\text{init}} \\ \sin \gamma_{\text{init}} \end{bmatrix}, \quad (3.9)$$

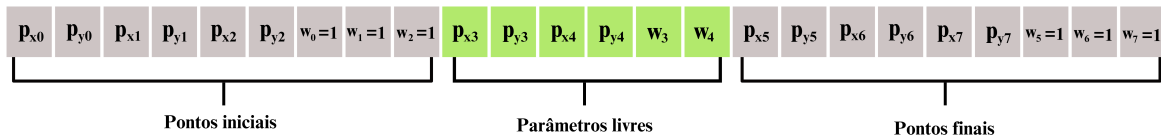
$$\mathbf{P}_{N_p-2} = \mathbf{P}_{N_p-1} - \frac{2}{3}\lambda_2 \begin{bmatrix} \cos \gamma_{\text{goal}} \\ \sin \gamma_{\text{goal}} \end{bmatrix}, \quad (3.10)$$

$$\mathbf{P}_{N_p-3} = \mathbf{P}_{N_p-2} - \frac{1}{3}\lambda_2 \begin{bmatrix} \cos \gamma_{\text{goal}} \\ \sin \gamma_{\text{goal}} \end{bmatrix}. \quad (3.11)$$

Aqui, \mathbf{P}_0 e \mathbf{P}_{N_p-1} representam, respectivamente, os pontos inicial e final do caminho. As constantes λ_1 e λ_2 ajustam a distância entre esses pontos, permitindo gerar dois pares de três pontos colineares de controle: $\{(\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2); (\mathbf{P}_{N_p-3}, \mathbf{P}_{N_p-2}, \mathbf{P}_{N_p-1})\}$. Os pesos associados a esses pontos de controle, definidos como $w_0 = w_1 = w_2 = w_{N_p-3} = w_{N_p-2} = w_{N_p-1} = 1$, conferem maior importância a esses pontos ao determinar a curva final.

Para a implementação da curva NURBS nesse trabalho, foi utilizada uma curva de grau $k = 5$ com o auxílio da GEOMDL, uma biblioteca em Python que oferece suporte para a criação e manipulação de curvas NURBS. Na Figura 15, é possível visualizar seis parâmetros livres da curva NURBS que podem ser otimizados pelos algoritmos DEs. Esses parâmetros livres representam dois pontos de controle, (P_{x2}, P_{y2}) e (P_{x3}, P_{y3}) , e dois pesos correspondentes (w_2 e w_3).

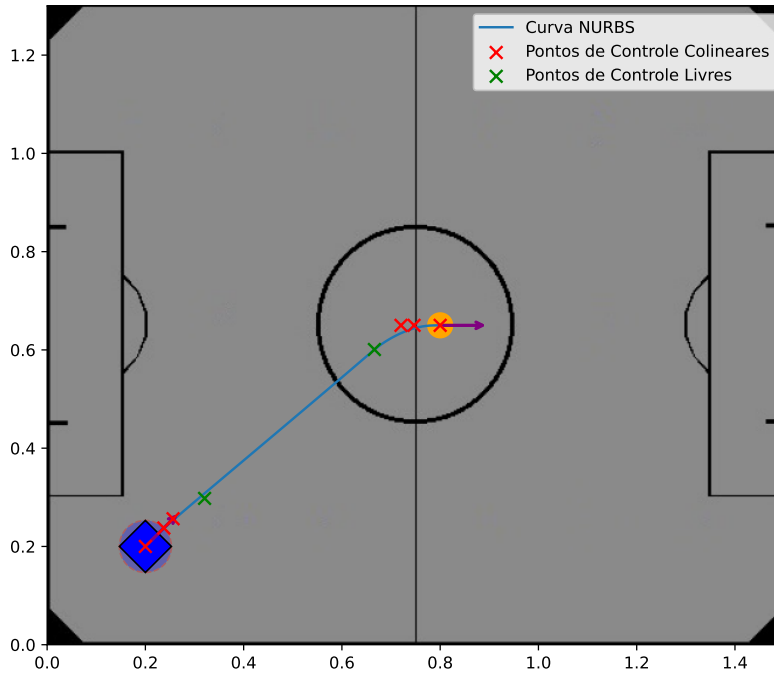
Figura 15 – Parâmetros de controle utilizado para uma curva NURBS.



Fonte: Elaborada pelo autor.

Na Figura 16, as marcações em vermelho, representadas por "x", indicam os pontos de controle colineares (iniciais e finais), enquanto os pontos em verde correspondem aos pontos de controle otimizados pelos algoritmos DEs (parâmetros livres).

Figura 16 – Pontos de controle colineares e otimizados pelos algoritmos DEs.



Fonte: Elaborada pelo autor.

Além disso, vale ressaltar que os algoritmos DEs também foram utilizados para otimizar os valores das constantes λ_1 e λ_2 , empregadas no cálculo dos pontos colineares. Durante os testes realizados, adotou-se a simplificação de considerar $\lambda_1 = \lambda_2 = \lambda$, facilitando a análise e o ajuste desses parâmetros de forma consistente em relação ao desempenho do sistema.

3.2.3 Problema de otimização

A função de custo f_T a ser minimizada para o problema, composta por uma função objetivo f e três funções de restrição h , é dada por:

$$f_T = f_1(\vec{r}(t)) + \alpha \cdot h_1(\vec{r}(t)) + h_2(\vec{r}(t), \vec{o}) + h_3(\vec{r}(t)), \quad (3.12)$$

onde f_1 retorna o comprimento total do caminho $\vec{r}(t)$, calculado com k termos:

$$f_1 = \sum_{j=0}^{k-1} |\vec{r}[k]|. \quad (3.13)$$

A função h_1 penaliza a curvatura excessiva do caminho, promovendo suavidade ao longo dos pontos discretos da curva. Ela é ativada quando a curvatura máxima κ_{\max} não é

respeitada, sendo definida como:

$$h_1 = \begin{cases} f_1 \cdot \left(\sum (k_i - \kappa_{\max})^2 \right), & \forall k_i \geq \kappa_{\max} \\ 0, & \text{caso contrário.} \end{cases} \quad (3.14)$$

Para transformar o problema original em um problema irrestrito, utiliza-se uma função de penalidade proporcional ao termo $\alpha \cdot h_1$. O parâmetro α é responsável por balancear a importância da suavidade do caminho em relação ao comprimento total. Essa abordagem permite lidar com restrições implícitas no problema original ao incorporá-las diretamente na função de custo. Assim, ao ajustar o valor de α , é possível penalizar caminhos que não sejam suficientemente suaves, promovendo uma solução que atenda aos requisitos de movimentação suave do robô.

A função h_2 é utilizada para penalizar soluções que atravessam obstáculos. Essa penalidade é aplicada sempre que a distância euclidiana entre um ponto da curva e o centro de um obstáculo $|\vec{r}_i - \vec{o}_j|$ for menor que o dobro do raio do obstáculo. P é um valor de penalidade elevado (exemplo, 10^6) aplicado em caso de colisão:

$$h_2 = \begin{cases} P, & \text{se } |\vec{r}_i - \vec{o}_j| < 2r_j, \forall \vec{o}_j \in \text{obstáculos} \\ 0, & \text{caso contrário,} \end{cases} \quad (3.15)$$

Por fim, a função h_3 penaliza soluções que possuem pontos fora dos limites permitidos do ambiente N_{h_3} , ou seja, fora dos limites do campo de futebol:

$$h_3 = \begin{cases} f_1 \cdot (N_{h_3}), & \forall [x, y, z]^T \notin \Omega_{\text{lim}} \\ 0, & \text{caso contrário.} \end{cases} \quad (3.16)$$

4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados e discutidos os resultados alcançados durante o desenvolvimento deste trabalho.

Na Seção 4.1 são apresentadas as configurações dos experimentos, incluindo os parâmetros, *hardware* e *software* empregados na execução dos testes. Já na Seção 4.2 é avaliado o desempenho dos operadores de mutação aplicados ao algoritmo L-SHADE nos problemas do *benchmark* CEC2013, incluindo análises estatísticas e visuais dos resultados.

Na Seção 4.3 é abordada a influência dos operadores de mutação quando utilizados especificamente para o planejamento de caminho dos robôs futebolistas, identificando as combinações mais eficazes, em um primeiro momento, apenas para o algoritmo L-SHADE. Por fim, na Seção 4.4 é realizada uma análise mais ampla, comparando não apenas os operadores de mutação, mas também diferentes algoritmos DEs para o planejamento de caminho, destacando os métodos mais eficientes.

4.1 Configuração dos experimentos

Para a obtenção dos resultados apresentados neste capítulo, todos os experimentos foram executados em uma CPU Intel Core i9-12900KF rodando até 5,1 GHz com sistema operacional Ubuntu/Linux 20.04. Esses resultados são divididos em duas etapas: problemas genérico de otimização e o problema de planejamento de caminho. Nas seguintes subseções são apresentadas as configurações utilizadas nestes experimentos.

4.1.1 Problemas genéricos de otimização

Na primeira etapa deste trabalho, foi analisado o algoritmo L-SHADE para a solução de problemas de otimização genéricos, por meio da base de dados do CEC2013. Para tal, utilizou-se a linguagem de programação C++ na implementação das 28 funções disponibilizadas no *benchmark* de acordo com a implementação do Renkavieski (2019). Além disso, o algoritmo L-SHADE e os mutadores utilizados também foram implementados em C++. As funções foram otimizadas para as diferentes dimensões, $D = 2$, $D = 10$, $D = 30$ e $D = 50$.

Para cada uma dessas dimensões, o algoritmo L-SHADE foi submetido a 30 execuções distintas para cada problema do *benchmark* CEC2013 e para cada operador de mutação a fim de uma análise estatística dos resultados. A população inicial do algoritmo, foi definida de acordo com a dimensão D do problema, sendo definida como $N = 18 \cdot D$, e o critério de pa-

rada do algoritmo foi definido com base no número máximo de avaliações, neste caso, 10.000 avaliações.

Com relação às análises dos resultados encontrados para esses problemas, utilizou-se a linguagem Python em conjunto com a biblioteca Matplotlib para a construção dos diagramas de caixa (*boxplot*).

4.1.2 Planejamento de caminho

Na segunda etapa deste trabalho, foram analisados os algoritmos DEs descritos na Subseção 2.3.1 (DE, SaDE, JADE, SHADE e L-SHADE) para o planejamento de caminho aplicado ao futebol de robôs. Os algoritmos foram implementados utilizando linguagem Python. Para facilitar o desenvolvimento, utilizou-se a biblioteca Mealpy, que fornece um *framework* para implementação de algoritmos meta-heurísticos.

Foi realizada a implementação dos diferentes operadores de mutação descritos na Subseção 2.4 ("rand/1", "rand/2", "best/1", "best/2", "either-or", "current-to-rand/1", "current-to-best/1" e "current-to-pbest/1"), realizando as devidas adaptações nos algoritmos fornecidos pela Mealpy. Esses mutadores foram desenvolvidos com base nas estratégias propostas na literatura, permitindo explorar diferentes abordagens de variação dentro da população de soluções.

Cada combinação possível do algoritmo com operador de mutação foi executada 30 vezes em cinco cenários distintos no contexto do futebol de robôs, sendo esses descritos na Seção 4.3. Esses cenários variaram quanto ao número de obstáculos, às posições iniciais e finais, bem como à orientação dessas posições, com o objetivo de testar a eficácia e a robustez dos algoritmos e mutadores. Para cada execução, foram considerados 126 indivíduos na população inicial (Tanabe; Fukunaga, 2014) e um limite máximo de $N_{max} = 250$ gerações.

Para a realização dos experimentos, consideraram-se os seguintes aspectos: (i) o tamanho do campo, adotando as dimensões padrão (1,5 m × 1,3 m); (ii) o diâmetro dos robôs e dos obstáculos, de 10,6 cm (um pouco maior do que o estabelecido pelas normas do VSSS, a fim de garantir uma margem de segurança); e (iii) partidas com equipes compostas por três e cinco robôs.

4.2 Comparação dos operadores de mutação em problemas genéricos de otimização

Com base no *benchmark* CEC2013, foi realizada uma comparação dos operadores de mutação, considerando o algoritmo L-SHADE, que apresenta vantagens em relação ao DE

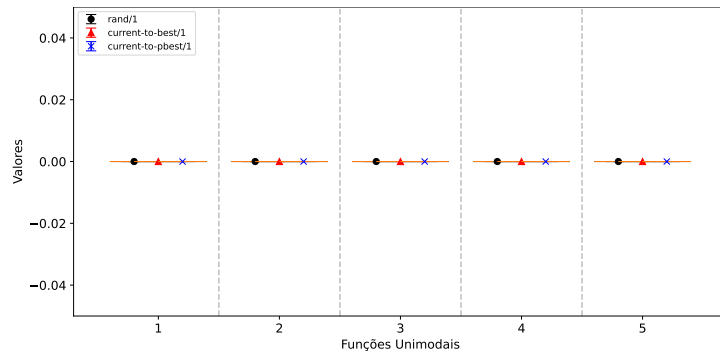
original, como mencionado por Tanabe e Fukunaga (2014). Os resultados dessa comparação foram publicados em (Silva; Souza; Freitas, 2024).

As Tabelas 5, 6, 7 e 8 do Apêndice A apresentam os resultados, considerando as seguintes dimensões dos problemas de otimização: $D = 2$, $D = 10$, $D = 30$ e $D = 50$.

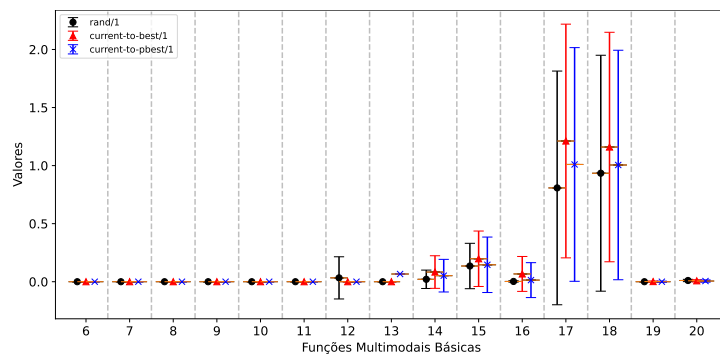
Para facilitar a visualização desses resultados, a Figura 17 mostra os diagramas de caixa com respeito as 30 execuções e o *fitness* retornado pelo algoritmo, considerando o caso a dimensão $D = 2$. Os dados são representados de acordo com os diferentes operadores de mutação utilizados: o operador “rand/1” é indicado por círculos pretos, “current-to-best/1” é representado por triângulos vermelhos e “current-to-pbest/1” é identificado por marcadores ‘x’ de cor azul.

Ao analisar os resultados, observa-se que, para a maioria das funções, considerando um problema de baixa dimensão ($D = 2$), os três operadores de mutação apresentam desempenhos semelhantes. No entanto, a partir da função 14 (que são funções mais complexas), torna-se mais evidente que o operador “rand/1” obteve os melhores resultados, uma vez que apresentou valores médios menores quando comparado aos outros operadores. Verificou-se que, entre os 28 valores de *fitness*, “rand/1” obteve 24 melhores. Assim, para esse tipo de problema de baixa dimensão, observa-se que uma abordagem de exploração aleatória proporciona um desempenho superior.

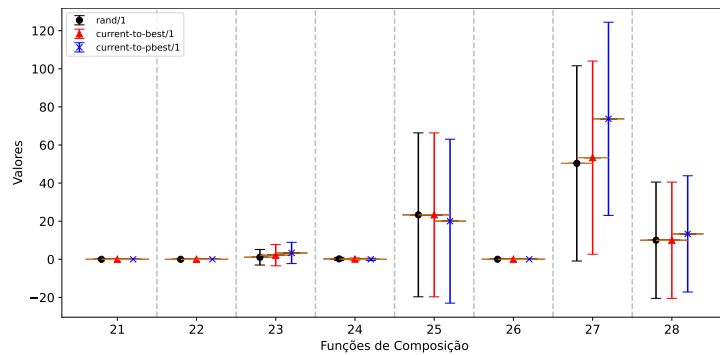
Figura 17 – Diagrama de caixa das funções CEC2013 para $D = 2$, onde cada cor representa um mutador diferente aplicado ao algoritmo L-SHADE. (a) Funções Unimodais; (b) Funções Multimodais Básicas; (c) Funções de Composição.



(a)

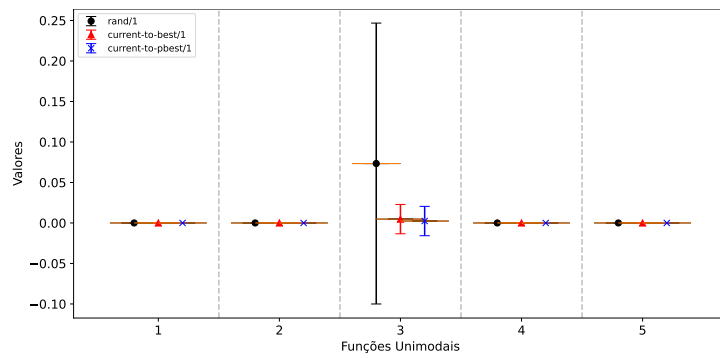


(b)

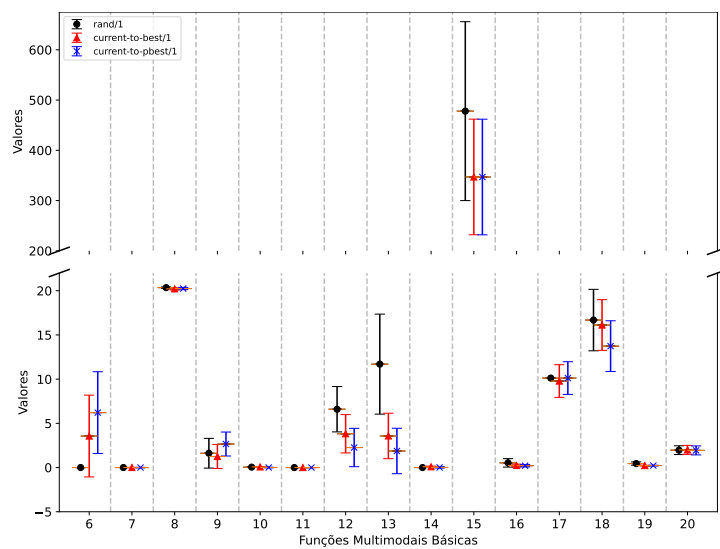


(c)

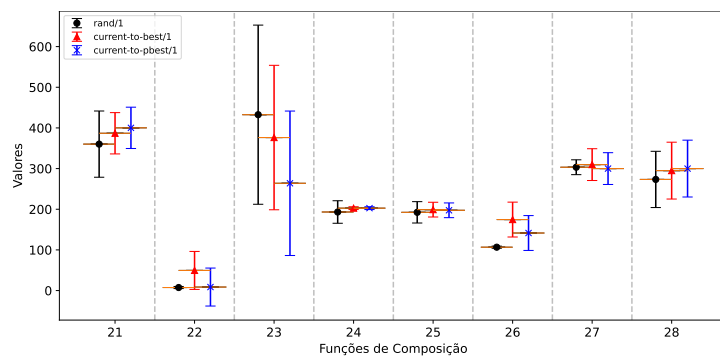
Figura 18 – Diagrama de caixa das funções CEC2013 para $D = 10$, onde cada cor representa um mutador diferente aplicado ao algoritmo L-SHADE. (a) Funções Unimodais; (b) Funções Multimodais Básicas; (c) Funções de Composição.



(a)

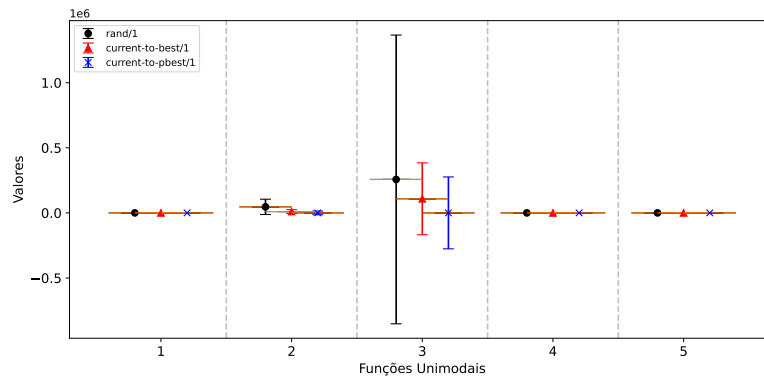


(b)

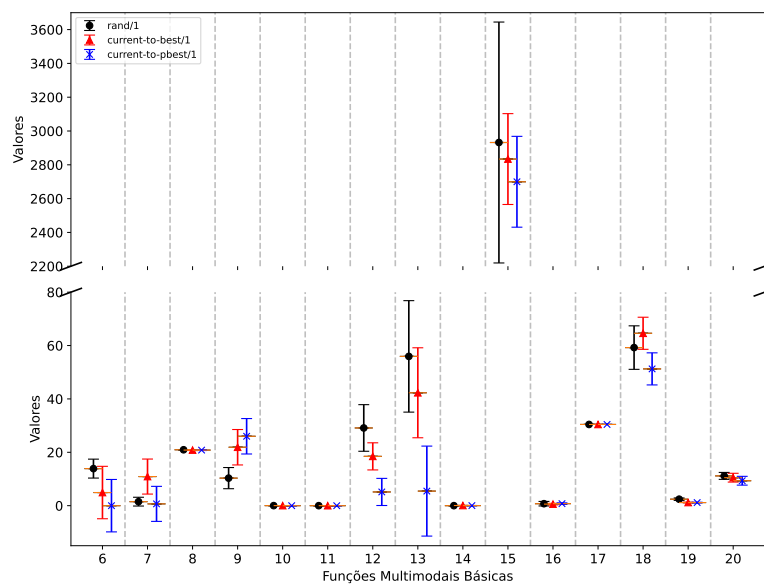


(c)

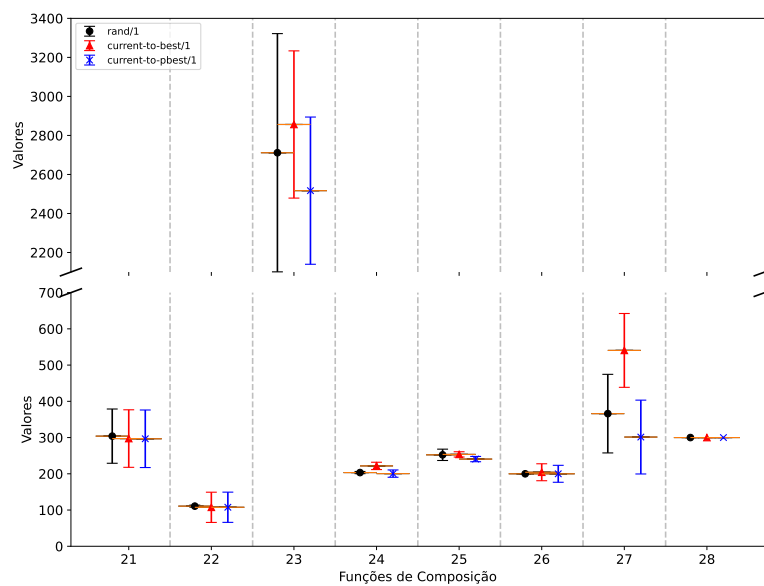
Figura 19 – Diagrama de caixa das funções CEC2013 para $D = 30$, onde cada cor representa um mutador diferente aplicado ao algoritmo L-SHADE. (a) Funções Unimodais; (b) Funções Multimodais Básicas; (c) Funções de Composição.



(a)



(b)



(c)

Figura 20 – Diagrama de caixa das funções CEC2013 para $D = 50$, onde cada cor representa um mutador diferente aplicado ao algoritmo L-SHADE. (a) Funções Unimodais; (b) Funções Multimodais Básicas; (c) Funções de Composição.

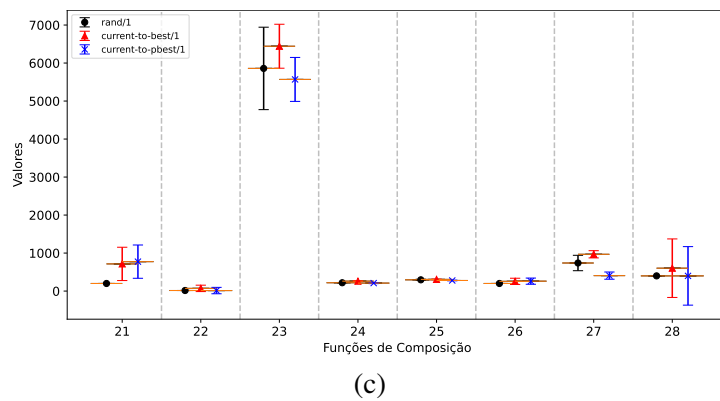
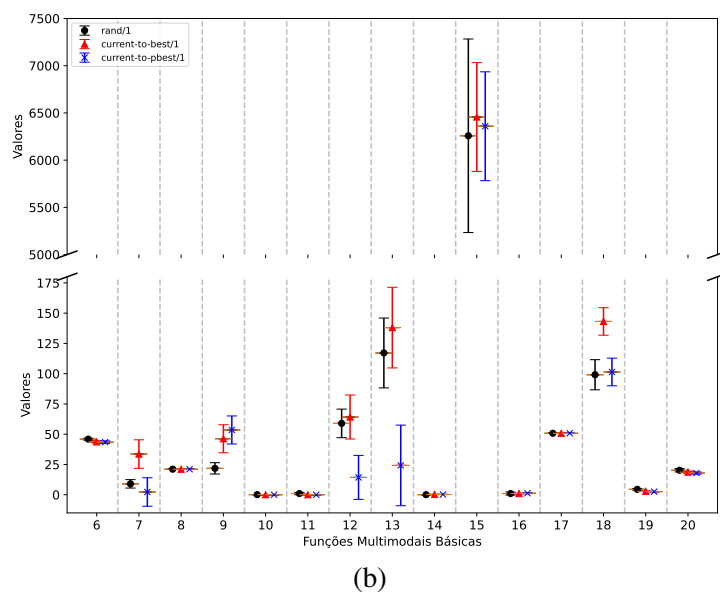
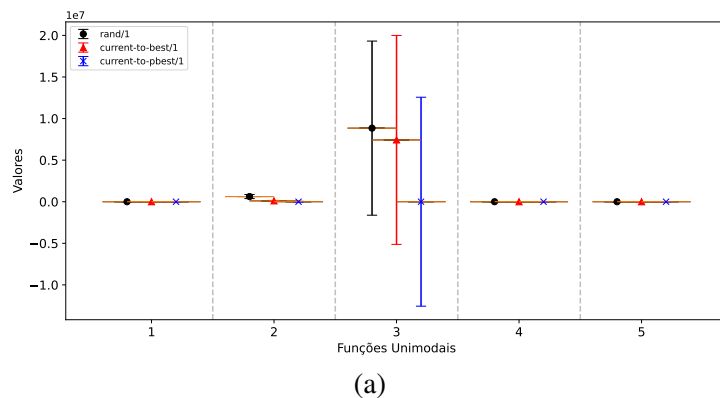
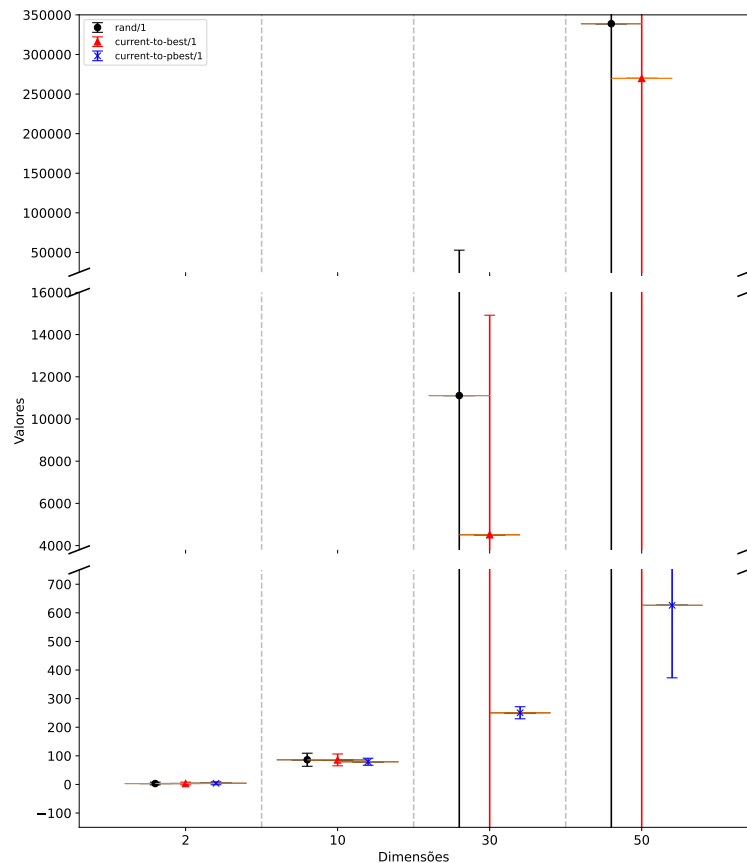


Figura 21 – Diagramas de caixa (média e desvio padrão geral) para cada dimensão.



Fonte: Elaborada pelo autor.

Na Figura 18, são exibidos Diagramas de Caixa que mostram a média e o desvio padrão dos valores de *fitness* correspondentes à dimensão 10. Observou-se que os resultados dos três operadores, tanto em termos de média quanto de desvio padrão, são também semelhantes, sem apresentar diferenças significativas.

Considerando quando a complexidade do problema aumenta, como no caso das dimensões $D = 30$ e $D = 50$, os resultados apresentados nas Figuras 19 e 20 apontam que o operador “current-to-pbest/1” obteve um desempenho significativamente melhor, em termos de média, em comparação com os outros operadores. Para $D = 30$, esse operador obteve o melhor resultado em 22 dos 28 valores de *fitness*, enquanto para $D = 50$, alcançou o melhor valor em 19 dos 28 valores, confirmando a tendência. Esses resultados podem indicar que, ao guiar os indivíduos na mutação para regiões mais próximas dos melhores indivíduos (pbest), além de manter a diversidade (*exploration*), o operador facilita o direcionamento para áreas do espaço de busca mais próximas do ótimo da função de custo (*exploitation*).

Por fim, com base na média geral das execuções para cada dimensão, o operador "current-to-pbest/1" também obteve o melhor desempenho, como ilustrado na Figura 21.

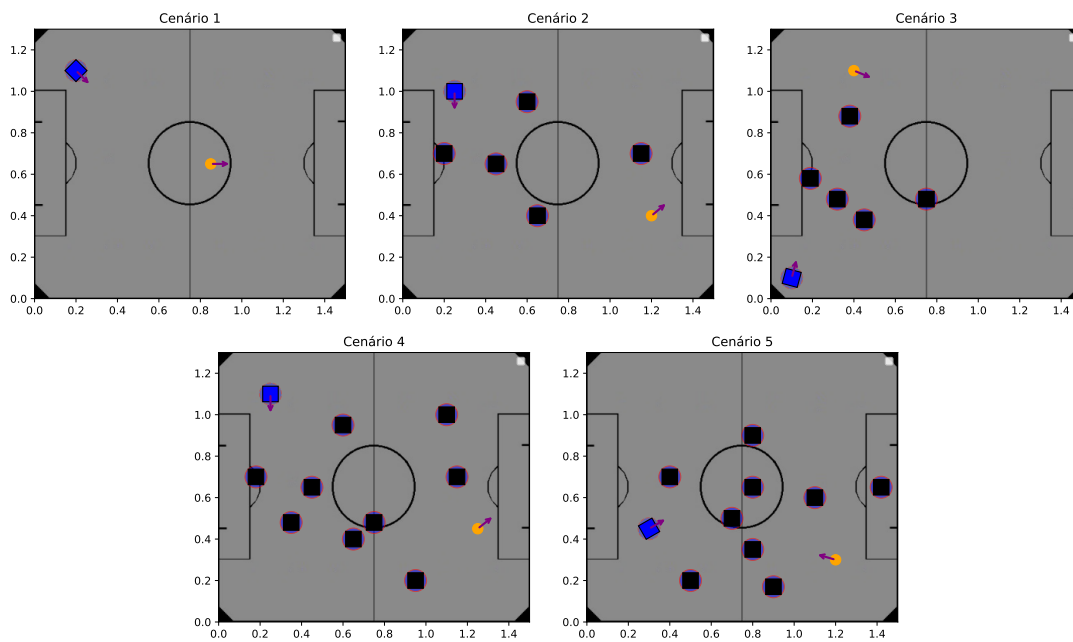
4.3 Comparação dos operadores de mutação aplicados ao L-SHADE para o problema de planejamento de caminho

Antes de comparar outros algoritmos DEs aplicados ao problema de planejamento de caminho, optou-se por avaliar inicialmente apenas os resultados obtidos pelo L-SHADE. O objetivo dessa análise é verificar se, assim como em problemas genéricos do *benchmark* CEC2013, o mutador "current-to-pbest/1" continua sendo o que gera as melhores soluções, agora aplicado a um problema real com restrições, analisando o desempenho com mais operadores de mutação. Além disso, essa abordagem permite expandir as análises realizadas em (Freitas; Cohen; Guimarães, 2023), onde o L-SHADE também foi utilizado para otimizar os parâmetros do planejador DE-NURBS.

Os cinco cenários empregados para testar os algoritmos e mutadores neste trabalho, indicados na Figura 22, apresentam um aumento progressivo de complexidade em cada situação. Esses cenários e os objetivos que se esperam com cada um deles são descritos a seguir:

- Cenário 1: Nesta situação, o robô é conduzido em direção a bola, que está orientada para o gol adversário, considerando uma pequena distância até ela. Nesse caso, é necessário que o robô altere a sua orientação para a desejada a fim de direcionar a bola para o gol adversário;
- Cenário 2: Neste cenário, o robô deve percorrer uma longa distância para alcançar a bola, evitando obstáculos ao longo do caminho. Além disso, o robô também deve alterar a sua orientação para direcionar a bola até o gol adversário;
- Cenário 3: Neste cenário, o robô está cercado por obstáculos. A ideia aqui, é observar se ele consegue chegar na bola com a orientação desejada para direcioná-la para o gol adversário, desviando dos obstáculos e sem sair para fora do campo, sendo obrigado a buscar um caminho mais estreito;
- Cenário 4: Este cenário é semelhante ao Cenário 2, porém, aqui foi pensado em uma partida composta por cinco robôs cada. A ideia é observar se o robô ainda alcançará a bola com a orientação desejada para marcar o gol, desviando dos obstáculos em um cenário ainda mais complexo, com mais obstáculos ao longo do percurso.

Figura 22 – Cenários do futebol de robôs utilizados nos testes.



Fonte: Elaborada pelo autor.

- **Cenário 5:** Neste caso, o robô também precisa lidar com mais obstáculos em uma partida de cinco robôs cada time. Além disso, o robô e a bola estão com as suas orientações opostas, assim como no cenário 3. O objetivo é verificar se o robô consegue alcançar a bola com orientação desejada para direcioná-la ao gol adversário, desviando dos obstáculos e tendo que caminhar por passagens estreitas para encontrar o caminho mais curto.

A Tabela 1 apresenta um resumo das soluções obtidas para cada mutador aplicado em conjunto com o algoritmo L-SHADE, mostrando a média e o desvio padrão de *fitness* para cada cenário. Além disso, nessa tabela, a solução com melhor valor de *fitness* (menor média), para cada cenário, é destacada. Vale ressaltar que os valores de média e desvio padrão apresentados nessa tabela consideram apenas as soluções factíveis, ou seja, aquelas que não violam as restrições propostas (funções de restrição iguais a zero).

Nos Cenários 1 e 2, todos os mutadores alcançaram uma taxa de sucesso de 100%, ou seja, em todas as 30 execuções do problema, todas as melhores soluções encontradas foram factíveis. Além disso, as médias de *fitness* apresentaram grande consistência entre os mutadores, com valores próximos de 0,803 no Cenário 1 e em torno de 1,44 no Cenário 2.

Tabela 1 – Desempenho do algoritmo L-SHADE com diferentes mutadores na otimização do planejador de caminho DE-NURBS aplicado em cenários específicos de futebol de robôs. A média e o desvio padrão são apenas dos caminhos factíveis e são destacadas as menores médias de *fitness* para cada cenário.

Mutador	Cenário	Fitness (média \pm desvio padrão)	Factível (n/30)
rand/1	1	0,803904 \pm 0,001913	30/30
	2	1,443588 \pm 0,007382	30/30
	3	1,258553 \pm 0,013596	7/30
	4	1,342335 \pm 0,032496	8/30
	5	N/A	0/30
rand/2	1	0,803746 \pm 0,001934	30/30
	2	1,443061 \pm 0,004127	30/30
	3	1,262462 \pm 0,013165	11/30
	4	1,330956 \pm 0,025731	12/30
	5	N/A	0/30
best/1	1	0,803281 \pm 0,001990	30/30
	2	1,444234 \pm 0,004545	30/30
	3	1,264904 \pm 0,011146	9/30
	4	1,342883 \pm 0,030652	9/30
	5	N/A	0/30
best/2	1	0,803218 \pm 0,001808	30/30
	2	1,445215 \pm 0,016435	30/30
	3	1,253069 \pm 0,002976	8/30
	4	1,351461 \pm 0,036120	10/30
	5	N/A	0/30
either-or	1	0,803838 \pm 0,001793	30/30
	2	1,442390 \pm 0,003139	30/30
	3	1,259478 \pm 0,011254	4/30
	4	1,346081 \pm 0,036077	6/30
	5	N/A	0/30
current-to-rand/1	1	0,803893 \pm 0,001974	30/30
	2	1,440891 \pm 0,002922	30/30
	3	1,260789 \pm 0,019178	5/30
	4	1,334839 \pm 0,027912	9/30
	5	N/A	0/30
current-to-best/1	1	0,803922 \pm 0,001883	30/30
	2	1,444248 \pm 0,005251	30/30
	3	1,254749 \pm 0,011348	8/30
	4	1,329873 \pm 0,024786	9/30
	5	N/A	0/30
current-to-pbest/1	1	0,803166 \pm 0,002200	30/30
	2	1,443444 \pm 0,004088	30/30
	3	1,258204 \pm 0,006853	9/30
	4	1,347370 \pm 0,034510	11/30
	5	N/A	0/30

No Cenário 3 há uma queda significativa no número de soluções factíveis. O mutador "rand/2" é o responsável pelo maior número de soluções factíveis, cerca de 33,67%, enquanto o "either-or" apresenta apenas 13,33% de suas soluções como factíveis, sendo o mutador responsável pelo menor número de soluções factíveis. Em relação ao *fitness*, o mutador "best/2" obteve o menor valor médio, $1,253069 \pm 0,002976$, enquanto o "best/1" apresentou o maior, $1,264904 \pm 0,011146$.

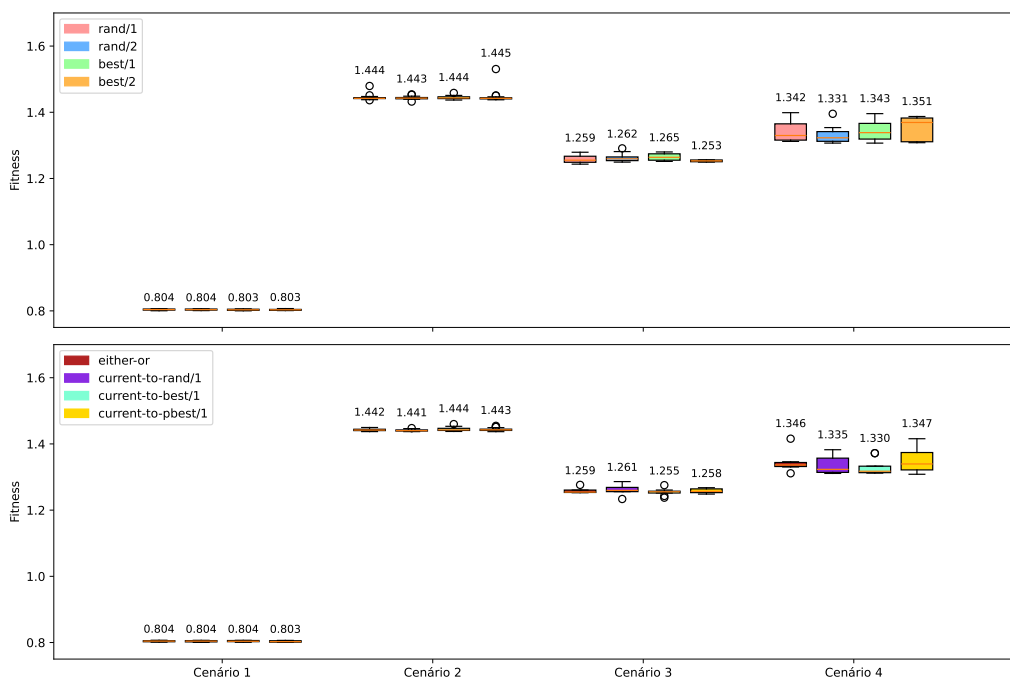
No Cenário 4, assim como no anterior, o mutador com o maior número de soluções factíveis foi o "rand/2", com 40% das soluções encontradas, enquanto o "either-or" obteve o menor número, com apenas 20% de soluções factíveis. O mutador "current-to-best/1" apresentou o menor valor médio de *fitness*, $1,329873 \pm 0,024786$, enquanto o "best/2" obteve o maior, $1,351461 \pm 0,036120$.

Com relação ao Cenário 5, vale destacar que nenhuma combinação de mutador com o L-SHADE foi capaz de encontrar uma solução factível, o que demonstra a complexidade desse cenário proposto.

Em relação à quantidade de soluções factíveis encontradas, independentemente do cenário analisado, os mutadores podem ser classificados em um *ranking*. O "rand/2" obteve a maior quantidade de soluções factíveis, com um total de 83. Em seguida, o "current-to-pbest/1" com 80, enquanto "best/1" e "best/2" alcançaram 78 cada. O "current-to-best/1" registrou 77 soluções factíveis, seguido por "rand/1" com 75 e "current-to-rand/1" com 74. Por fim, o "either-or" apresentou a menor quantidade, com 70 soluções factíveis.

Na Figura 23, são apresentados diagramas de caixa que ilustram as soluções factíveis nos diferentes cenários testados. De modo geral, não há grandes variações aparentes entre os mutadores, o que pode indicar que o L-SHADE não é sensível a variação de mutador para o problema proposto de planejamento de caminho.

Figura 23 – Diagramas de caixas dos mutadores com o L-SHADE.

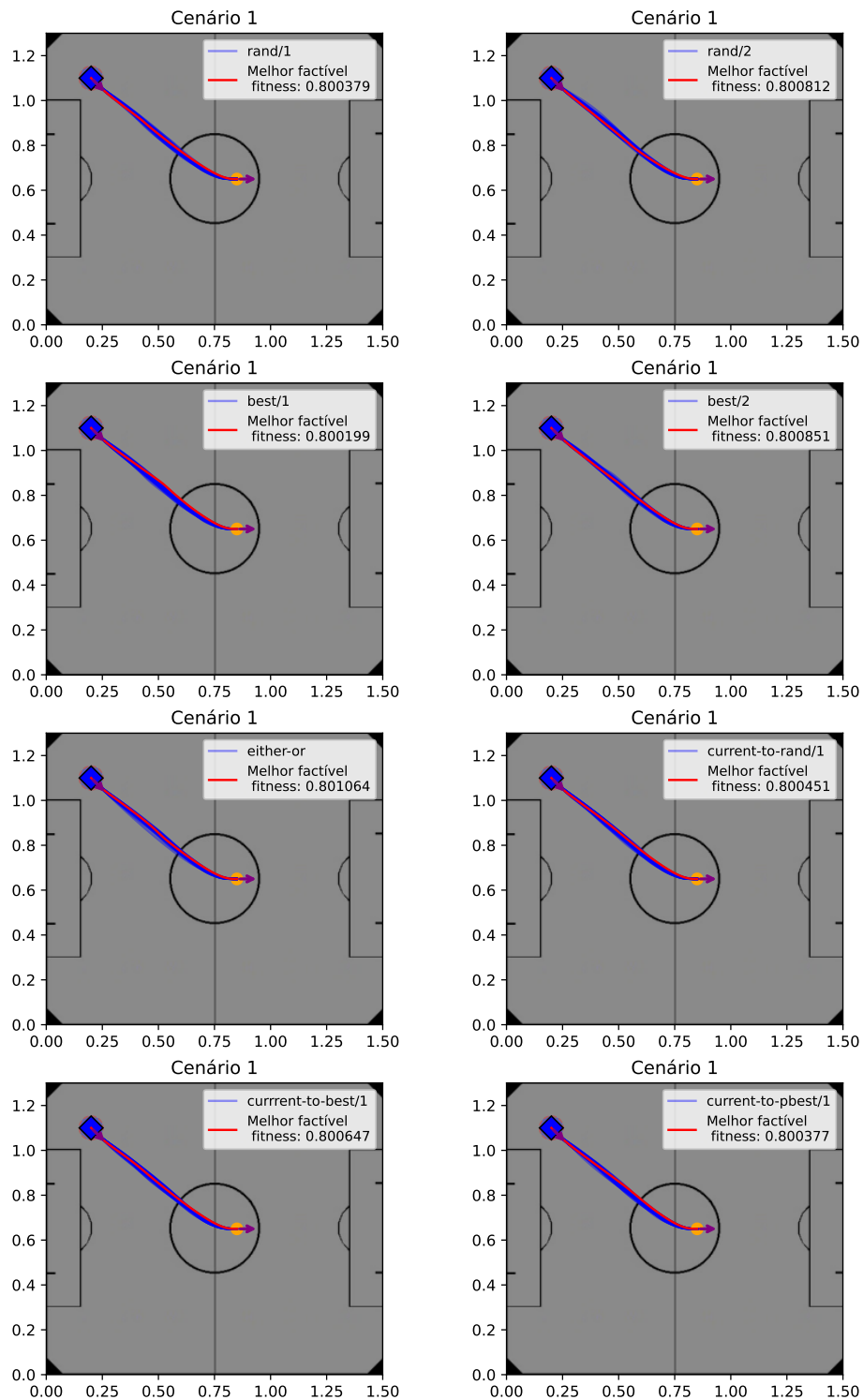


Fonte: Elaborada pelo autor.

As Figuras 24, 25, 26, 27 e 28 ilustram todos os melhores caminhos gerados, incluindo os caminhos não factíveis. Cada uma dessas figuras é referente a um dos cenários, apresentando as soluções para os oito mutadores utilizados. Os caminhos em azul representam todas as soluções geradas, sendo indicado na legenda o mutador utilizado. Já o caminho em vermelho, destaca o melhor caminho factível, apresentando o valor de *fitness* correspondente também na legenda.

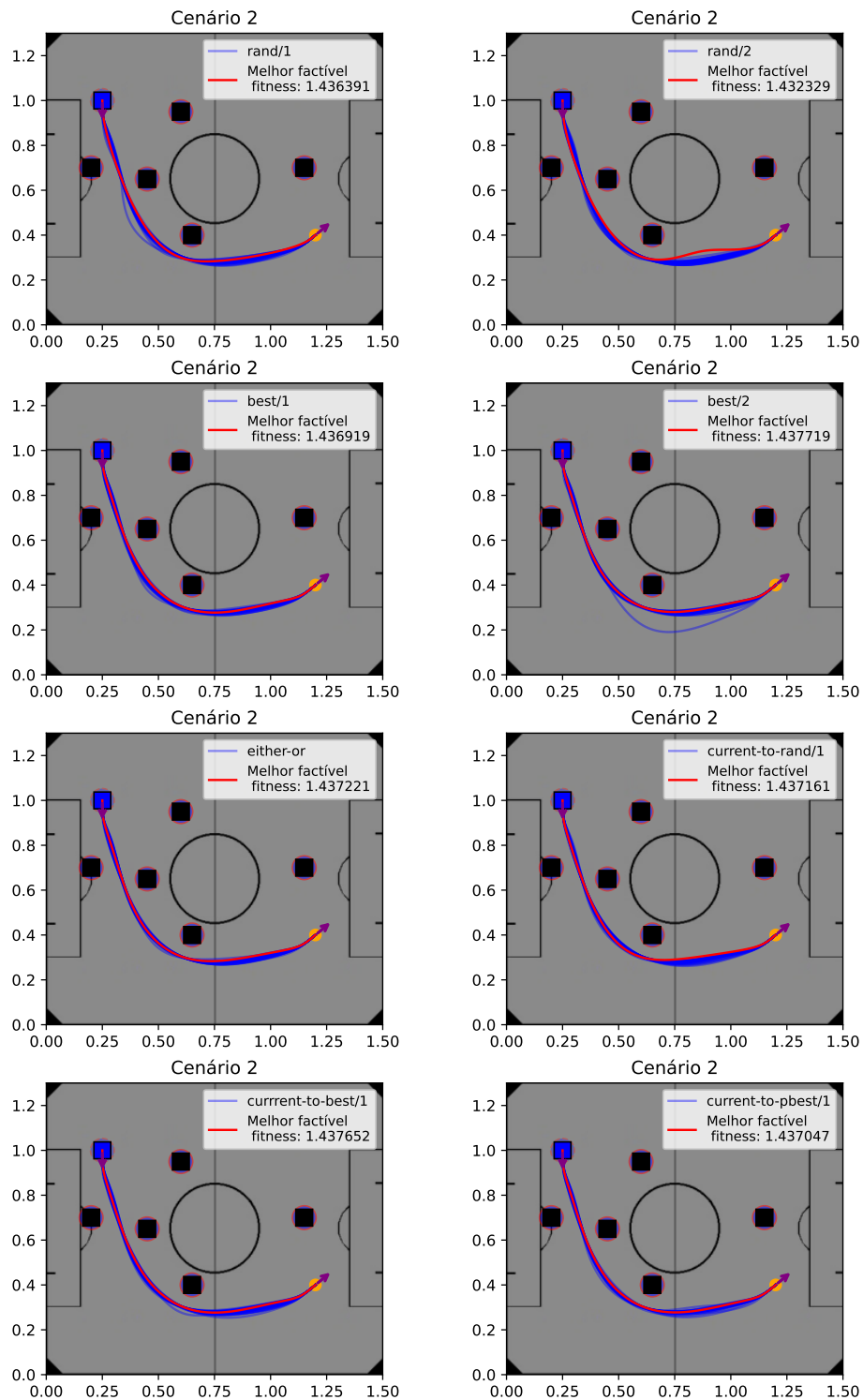
Independentemente do operador de mutação utilizado, observa-se nas Figuras 24 e 25 que, tanto no Cenário 1 quanto no Cenário 2, todos os caminhos são factíveis, como evidenciado também pela Tabela 1. Isso porque, todos os caminhos estão sendo direcionados para a bola com curvas suaves de maneira a guiar a bola em direção ao gol adversário. Os caminhos gerados são bastante semelhantes para ambos cenários, passando pelas mesmas regiões e apresentando pontos próximos uns dos outros. Isso indica uma convergência do algoritmo mesmo com as diferentes abordagens para um cenário livre de obstáculos.

Figura 24 – Caminhos, factíveis e infactíveis, obtidos com L-SHADE e mutadores para o Cenário 1, considerando 30 execuções do planejador para cada mutador. Em vermelho é destacada a melhor solução factível, caso ela tenha sido encontrada ou exista.



Fonte: Elaborada pelo autor.

Figura 25 – Caminhos, factíveis e infactíveis, obtidos com L-SHADE e mutadores para o Cenário 2, considerando 30 execuções do planejador para cada mutador. Em vermelho é destacada a melhor solução factível, caso ela tenha sido encontrada ou exista.



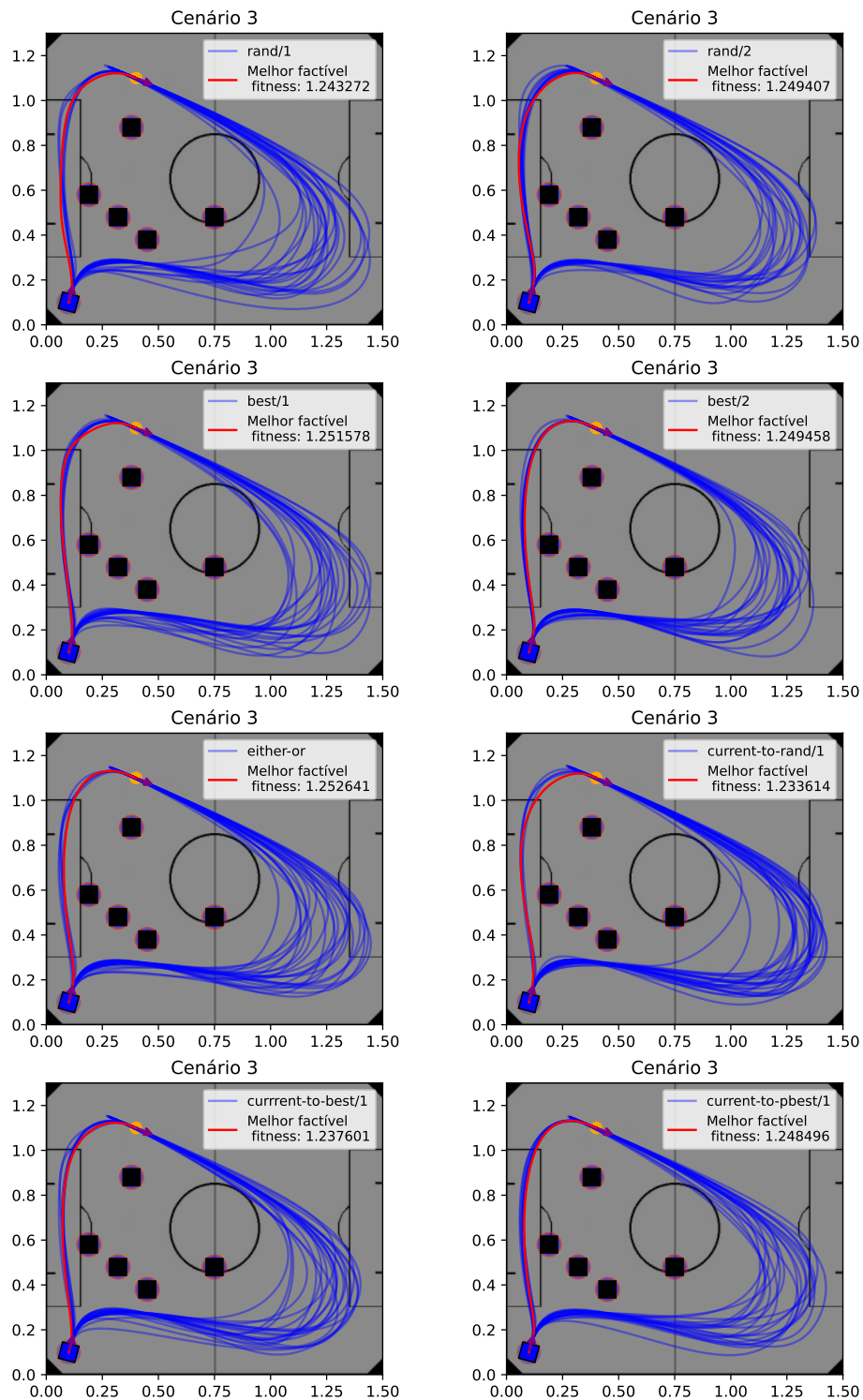
Fonte: Elaborada pelo autor.

Observando os caminhos relativos ao Cenário 3, Figura 26, note que a maioria dos caminhos optou por um percurso mais longo, desviando dos obstáculos pelo lado direito do campo. Além disso, esses caminhos direcionam o robô para a bola em um ângulo oposto ao desejado, fazendo com que o robô não consiga direcionar a bola para o gol adversário. Esses caminhos, que são a maioria nesse cenário, são os não factíveis. Já os caminhos factíveis conseguiram chegar na bola por uma passagem estreita do lado esquerdo do campo, apresentando um caminho mais curto, suave e com um ângulo adequado. Assim como nos cenários anteriores, entre os melhores caminhos factíveis não foram identificadas, visualmente, diferenças entre os caminhos.

O Cenário 4, indicado na Figura 27, é interessante de se observar, uma vez que visualmente todos os caminhos gerados aparentam ser factíveis. No entanto, como mostrado na Tabela 1, isso não é verdade e ocorre, pois a maioria dos caminhos não atenderam à restrição de curvatura máxima próxima à bola. Nesse cenário também não é perceptível diferenças visuais nos melhores caminhos factíveis.

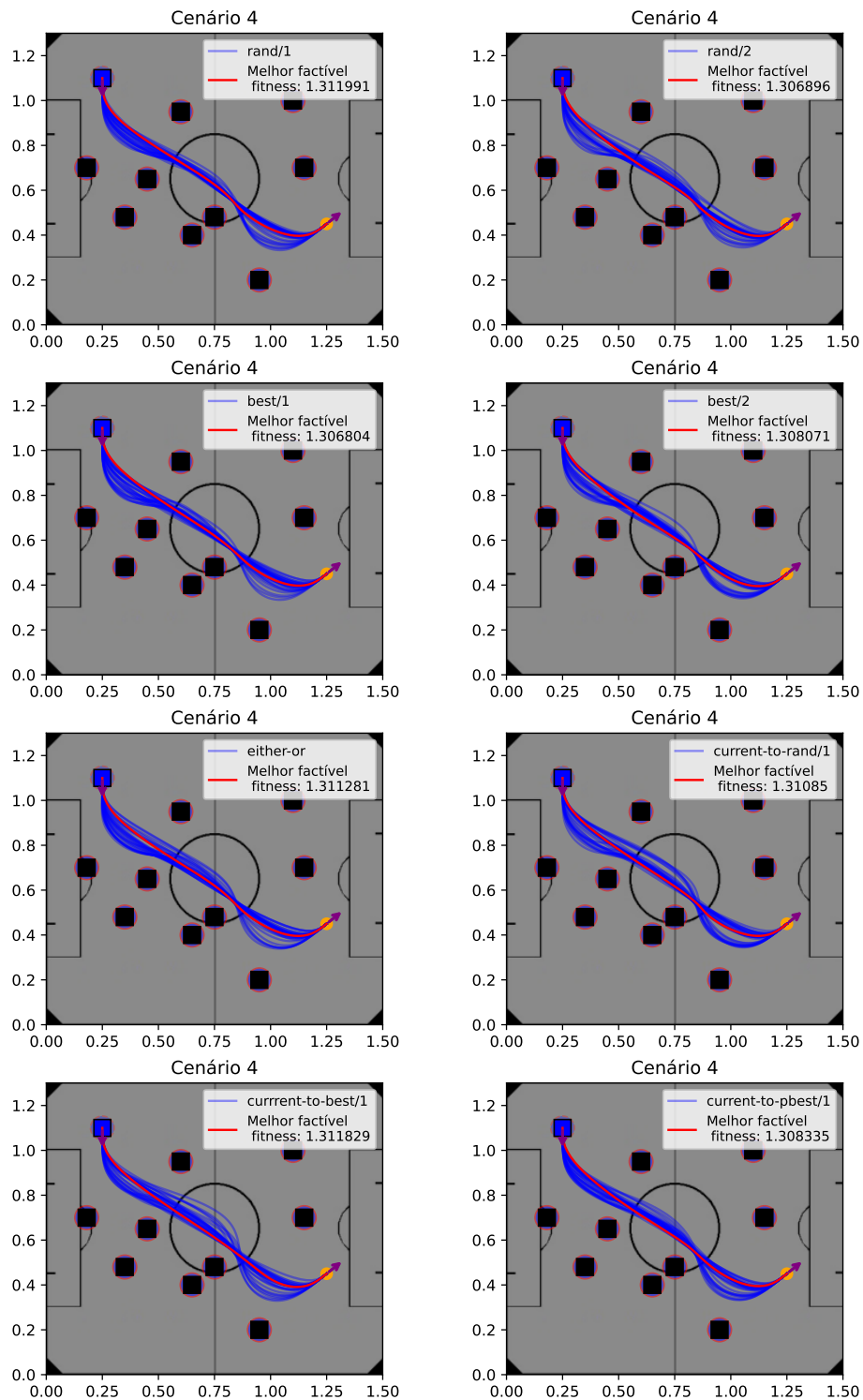
Por fim, é importante destacar como os resultados encontrados para o Cenário 5, na Figura 28, também são interessantes. Eles possuem comportamentos semelhantes aos caminhos não factíveis do Cenário 4. Isto é, embora os caminhos gerados tenham sido consistentes e o robô tenha alcançado a bola com a orientação correta e desviando dos obstáculos, nenhum dos caminhos foi factível devido à curvatura acentuada próxima à bola, o que impede o robô de cumprir a restrição da curvatura máxima, tornando as soluções inviáveis.

Figura 26 – Caminhos, factíveis e infactíveis, obtidos com L-SHADE e mutadores para o Cenário 3, considerando 30 execuções do planejador para cada mutador. Em vermelho é destacada a melhor solução factível.



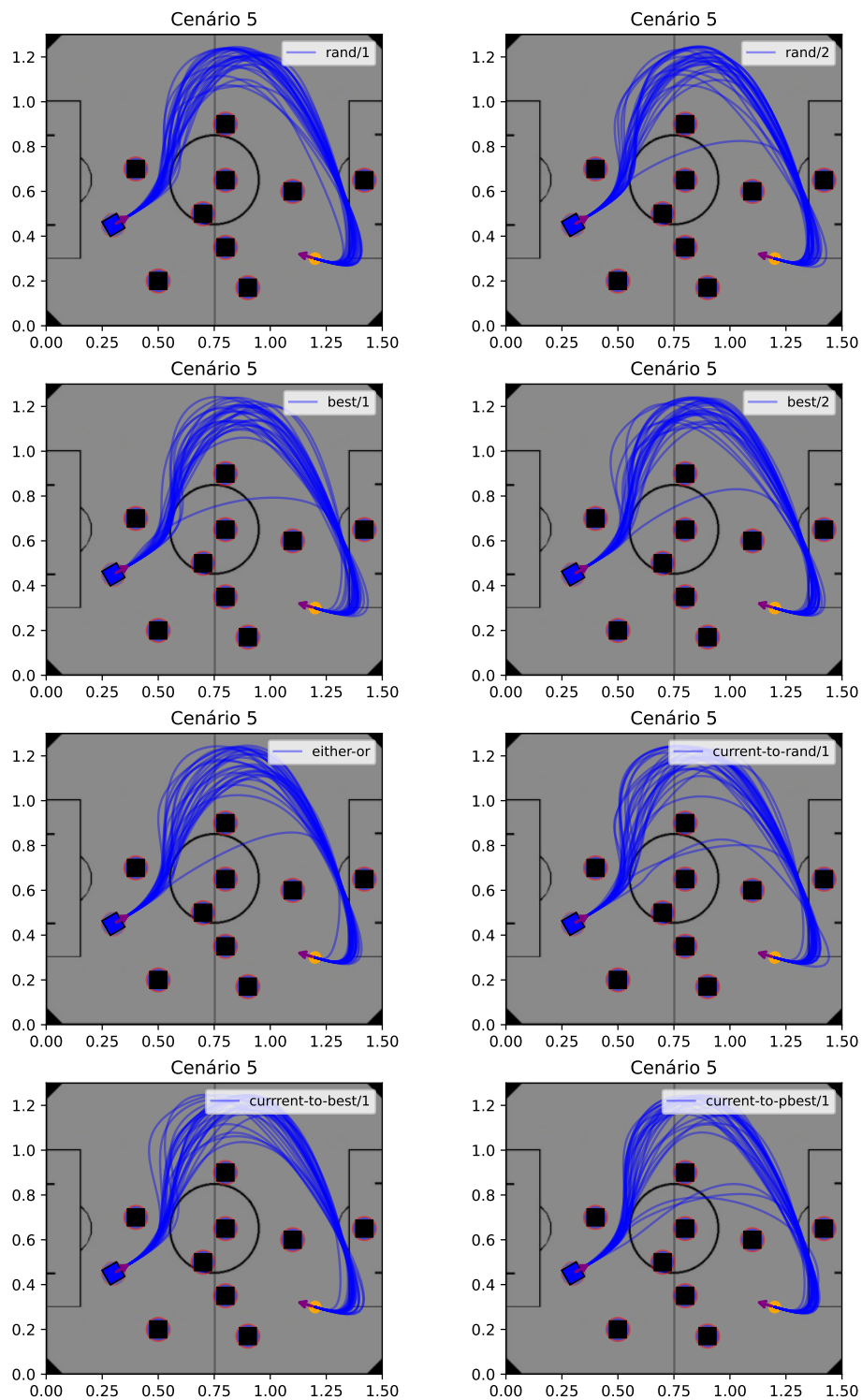
Fonte: Elaborada pelo autor.

Figura 27 – Caminhos, factíveis e infactíveis, obtidos com L-SHADE e mutadores para o Cenário 4, considerando 30 execuções do planejador para cada mutador. Em vermelho é destacada a melhor solução factível.



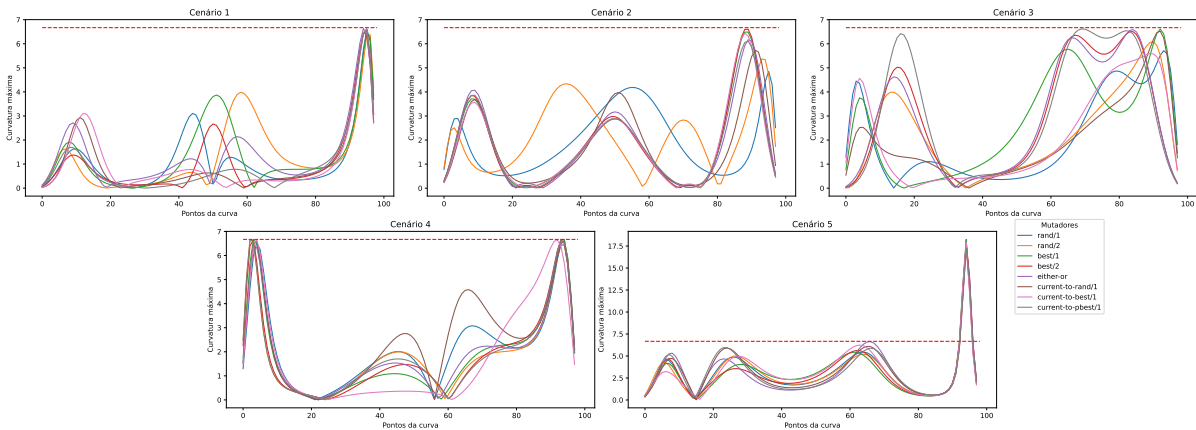
Fonte: Elaborada pelo autor.

Figura 28 – Caminhos infactíveis obtidos com L-SHADE e mutadores para o Cenário 5, considerando 30 execuções do planejador para cada mutador.



Fonte: Elaborada pelo autor.

Figura 29 – Curvatura dos melhores caminhos obtidos com o L-SHADE para cada cenário, onde o tracejado vermelho representa a curvatura máxima permitida e as curvas coloridas a curvatura obtida com cada mutador.



Fonte: Elaborada pelo autor.

Como já mencionado, a curvatura do caminho é uma das restrições do problema, sendo importante na determinação dos caminhos factíveis. Isso foi identificado no Cenário 4, para as soluções não factíveis encontradas, e, principalmente, no Cenário 5, onde não foram encontradas soluções factíveis. No entanto, em algumas situações, pode não ser imediatamente perceptível se essa restrição está sendo atendida.

Por esse motivo, as curvaturas das melhores soluções factíveis obtidas com o L-SHADE foram analisadas para cada cenário e representadas graficamente na Figura 29. No caso do Cenário 5, onde nenhuma solução factível foi encontrada, analisou-se a curvatura da solução com o menor valor de *fitness* para evidenciar o não atendimento à restrição. Nessa figura, a linha tracejada vermelha representa a curvatura máxima permitida, enquanto as curvas coloridas ilustram a curvatura obtida por cada mutador utilizado. Essa visualização facilita a verificação do atendimento à restrição de curvatura do robô futebolista, garantindo um caminho suave.

Nos Cenários 1, 2, 3 e 4, observa-se que as linhas de curvatura não ultrapassam a linha tracejada que representa a curvatura máxima, indicando que as melhores soluções factíveis encontradas por cada mutador respeitaram a restrição. No entanto, no Cenário 5, o comportamento observado é o esperado pelos resultados apresentados anteriormente, ou seja, as curvas de curvatura ultrapassam a linha tracejada, sugerindo que nenhum dos mutadores conseguiu atender à restrição nesse cenário específico.

Em resumo, com base nos testes realizados para o algoritmo L-SHADE, a escolha do mutador apresenta ter pouca influência sobre o caminho factível, pois não foram identificadas

diferenças significativas entre os caminhos encontrados. Nesse contexto, a seleção de mutadores, que apresentaram um maior número de soluções factíveis, como "rand/2" e "current-to-pbest/1", parece ser uma possível escolha para o problema de planejamento de caminho para o futebol de robôs utilizando o algoritmo L-SHADE, baseando-se na factibilidade. Na Seção 4.4, os testes serão expandidos para os algoritmos DE, SaDE, JADE e SHADE, o que permitirá verificar se essa tendência se mantém.

4.4 Comparação dos operadores de mutação e algoritmos diferenciais evolutivos aplicados ao problema de planejamento de caminho

Após avaliar apenas os resultados obtidos pelo L-SHADE, expandiu-se a análise para os demais algoritmos DEs: DE, SaDE, JADE e SHADE. Aqui, além de tentar identificar qual combinação de algoritmo e mutador apresenta os melhores resultados no planejamento de caminho, também será verificado se os outros algoritmos terão resultados melhores de desempenho do que o L-SHADE.

Na Tabela 2 são apresentadas as médias e os desvios padrão dos valores de *fitness* para cada mutador e cenário, considerando apenas as soluções factíveis. Além disso, é destacado nessa tabela, para cada cenário e mutador, o algoritmo que apresentou a menor média de *fitness*. Note que, nos testes com todos os algoritmos, nenhum conseguiu obter uma solução factível para o Cenário 5, assim como se observou na Tabela 1 com apenas o L-SHADE.

Foi analisada, inicialmente, a quantidade de vezes em que um algoritmo apresentou a menor média de *fitness* em relação aos demais para cada problema (Cenário e mutador), sendo essas médias destacadas na Tabela 1. O L-SHADE se sobressaiu como o melhor algoritmo, registrando a menor média de *fitness* em 16 problemas. Em seguida, o SHADE com a menor média em 7 problemas, o JADE em 5, o DE em 3 e, por último, o SaDE com apenas apenas 1 problema. Essa informação mostra uma diferença considerável entre o L-SHADE e os demais algoritmos.

Tabela 2 – Desempenho dos algoritmos DEs implementados com diferentes mutadores na otimização do planejador de caminho DE-NURBS aplicado em cenários específicos de futebol de robôs.

Mutador	Cenário	Média ± Desvio padrão (n/30) factíveis				
		DE	SaDE	JADE	SHADE	LSHADE
rand/1	1	1,692557 ± 0,811438 (21/30)	0,804481 ± 0,001803 (30/30)	0,838252 ± 0,182840 (30/30)	0,803560 ± 0,001946 (30/30)	0,803904 ± 0,001913 (30/30)
	2	1,466067 ± 0,039909 (3/30)	1,446124 ± 0,011980 (30/30)	1,442477 ± 0,002322 (30/30)	1,443863 ± 0,003756 (30/30)	1,443588 ± 0,007382 (30/30)
	3	N/A (0/30)	1,258903 ± 0,005716 (8/30)	1,259309 ± 0,018173 (12/30)	1,261764 ± 0,010656 (7/30)	1,258553 ± 0,013596 (7/30)
	4	1,341253 ± 0,027175 (9/30)	1,332753 ± 0,019451 (10/30)	1,369522 ± N/A (1/30)	1,337823 ± 0,037736 (9/30)	1,342335 ± 0,032496 (8/30)
	5	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)
rand/2	1	1,738424 ± 0,677728 (22/30)	0,805504 ± 0,009393 (30/30)	0,804955 ± 0,001508 (30/30)	0,804291 ± 0,001862 (30/30)	0,803746 ± 0,001934 (30/30)
	2	1,457478 ± 0,025373 (4/30)	1,445420 ± 0,005388 (30/30)	1,443689 ± 0,004792 (30/30)	1,442580 ± 0,003219 (30/30)	1,443061 ± 0,004127 (30/30)
	3	N/A (0/30)	1,270421 ± 0,025159 (6/30)	1,270452 ± 0,016215 (8/30)	1,262508 ± 0,011696 (5/30)	1,262462 ± 0,013165 (11/30)
	4	1,318699 ± 0,011816 (4/30)	1,354175 ± 0,040344 (7/30)	N/A (0/30)	1,348469 ± 0,028411 (9/30)	1,330956 ± 0,025731 (12/30)
	5	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)
best/1	1	1,579566 ± 0,695902 (25/30)	0,804170 ± 0,001796 (30/30)	0,804762 ± 0,001799 (30/30)	0,847995 ± 0,239749 (30/30)	0,803281 ± 0,001990 (30/30)
	2	1,479089 ± 0,044334 (6/30)	1,444342 ± 0,004479 (30/30)	1,442815 ± 0,002700 (30/30)	1,444014 ± 0,009140 (30/30)	1,444234 ± 0,004545 (30/30)
	3	N/A (0/30)	1,266637 ± 0,015445 (12/30)	1,266908 ± 0,010724 (14/30)	1,275182 ± 0,023397 (6/30)	1,264904 ± 0,011146 (9/30)
	4	1,344682 ± 0,028793 (7/30)	1,362854 ± 0,032612 (7/30)	N/A (0/30)	1,347271 ± 0,039546 (6/30)	1,342883 ± 0,030652 (9/30)
	5	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)
best/2	1	2,039276 ± 0,634661 (24/30)	0,804350 ± 0,001752 (30/30)	0,805182 ± 0,001658 (30/30)	0,803341 ± 0,001789 (30/30)	0,803218 ± 0,001808 (30/30)
	2	1,501018 ± 0,067261 (4/30)	1,445418 ± 0,007025 (30/30)	1,442586 ± 0,003530 (30/30)	1,437643 ± 0,030865 (30/30)	1,445215 ± 0,016435 (30/30)
	3	1,290259 ± N/A (1/30)	1,273632 ± 0,019391 (4/30)	1,260524 ± 0,014299 (9/30)	1,272591 ± 0,020704 (13/30)	1,253069 ± 0,002976 (8/30)
	4	1,339243 ± 0,037038 (5/30)	1,345426 ± 0,029165 (7/30)	1,405687 ± N/A (1/30)	1,347033 ± 0,028099 (10/30)	1,351461 ± 0,036120 (10/30)
	5	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)
either-or	1	1,811030 ± 0,696115 (26/30)	0,803958 ± 0,001893 (30/30)	0,805174 ± 0,001643 (30/30)	0,803387 ± 0,001712 (30/30)	0,803838 ± 0,001793 (30/30)
	2	1,456381 ± 0,023655 (3/30)	1,442851 ± 0,002862 (30/30)	1,442658 ± 0,004063 (30/30)	1,443794 ± 0,004100 (30/30)	1,442390 ± 0,003139 (30/30)
	3	1,284016 ± N/A (1/30)	1,264276 ± 0,020564 (8/30)	1,270388 ± 0,025366 (13/30)	1,269240 ± 0,011997 (4/30)	1,259478 ± 0,011254 (4/30)
	4	1,345498 ± 0,019480 (4/30)	1,345842 ± 0,049703 (8/30)	1,364392 ± 0,071071 (3/30)	1,343292 ± 0,043342 (8/30)	1,346081 ± 0,036077 (6/30)
	5	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)
current-to-rand/1	1	1,863524 ± 0,656473 (24/30)	0,805151 ± 0,006001 (30/30)	0,805244 ± 0,001818 (30/30)	0,803530 ± 0,002106 (30/30)	0,803893 ± 0,001974 (30/30)
	2	1,506028 ± 0,137740 (7/30)	1,446272 ± 0,006876 (30/30)	1,443083 ± 0,003158 (30/30)	1,443926 ± 0,006668 (30/30)	1,440891 ± 0,002922 (30/30)
	3	N/A (0/30)	1,264241 ± 0,011774 (7/30)	1,270735 ± 0,012194 (12/30)	1,261292 ± 0,010695 (5/30)	1,260789 ± 0,019178 (5/30)
	4	1,329342 ± 0,030450 (6/30)	1,347235 ± 0,031502 (9/30)	1,415347 ± N/A (1/30)	1,331588 ± 0,035252 (5/30)	1,334839 ± 0,027912 (9/30)
	5	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)
current-to-best/1	1	1,904193 ± 0,680065 (27/30)	0,804361 ± 0,001781 (30/30)	0,804903 ± 0,001727 (30/30)	0,837040 ± 0,183481 (30/30)	0,803922 ± 0,001883 (30/30)
	2	1,451682 ± 0,008778 (2/30)	1,444108 ± 0,003597 (30/30)	1,443321 ± 0,003868 (30/30)	1,443937 ± 0,005672 (30/30)	1,444248 ± 0,005251 (30/30)
	3	N/A (0/30)	1,262291 ± 0,014970 (5/30)	1,264175 ± 0,021473 (7/30)	1,259889 ± 0,006380 (5/30)	1,254749 ± 0,011348 (8/30)
	4	1,344491 ± 0,024133 (8/30)	1,347067 ± 0,039647 (6/30)	1,322374 ± N/A (1/30)	1,339680 ± 0,036730 (9/30)	1,329873 ± 0,024786 (9/30)
	5	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)
current-to-pbest/1	1	1,887368 ± 0,675456 (24/30)	0,804408 ± 0,003525 (30/30)	0,805389 ± 0,001343 (30/30)	0,803760 ± 0,001869 (30/30)	0,803166 ± 0,002200 (30/30)
	2	1,467060 ± 0,045153 (7/30)	1,445940 ± 0,004702 (30/30)	1,442626 ± 0,002684 (30/30)	1,444440 ± 0,005602 (30/30)	1,443444 ± 0,004088 (30/30)
	3	N/A (0/30)	1,270836 ± 0,015060 (8/30)	1,261997 ± 0,016736 (9/30)	1,276388 ± 0,044245 (5/30)	1,258204 ± 0,006853 (9/30)
	4	1,527375 ± 0,438551 (6/30)	1,342022 ± 0,029565 (7/30)	1,380536 ± 0,040134 (3/30)	1,333760 ± 0,028921 (10/30)	1,347370 ± 0,034510 (11/30)
	5	N/A (0/31)	N/A (0/30)	N/A (0/30)	N/A (0/30)	N/A (0/30)

Observa-se que o algoritmo DE obteve menor média para o Cenário 4 e o algoritmo JADE para o Cenário 5. Porém, o L-SHADE é o único algoritmo a encontrar a menor média de *fitness* para todos os quatro cenários que obteve soluções factíveis, considerando as diferentes abordagens utilizadas. Isso demonstra que, para um ambiente onde o ambiente não é fixo, como no caso do futebol de robôs, o L-SHADE pode se destacar em relação aos demais. Sendo que a combinação do L-SHADE com o operador de mutação "best/1" apresentou o melhor desempenho em três dos cinco cenários com soluções factíveis.

Para determinar, de maneira geral, a combinação de algoritmo e mutador com o melhor desempenho (menor média de *fitness*), considerando os resultados para os cinco cenários, foi realizada uma classificação baseada no desempenho. Assim, foi atribuída uma pontuação a cada combinação, iniciando com o valor 100 para a combinação com a menor média, diminuindo progressivamente conforme as médias se tornavam maiores. Posteriormente, foi realizada a soma dessas pontuações para cada combinação, e os resultados foram organizados pela soma total da classificação, do maior para o menor.

Assim, foi possível verificar que, para o problema de planejamento de caminho proposto, a combinação com melhor desempenho (menor *fitness*) é o L-SHADE com o operador de mutação "current-to-rand/1", conforme é possível verificar na Tabela 3.

Com relação a quantidade de soluções factíveis obtida por cada combinação de algoritmo e mutador, a Tabela 4 exibe, de forma percentual, os valores correspondentes. As combinações com as maiores quantidades de caminhos factíveis estão em destaque na tabela, são elas: (i) DE e "best/1" com 25,33% das soluções factíveis; (ii) SaDE e "best/1" com 52,67%; (iii) JADE e "either-or" com 50,67%; (iv) SHADE e "best/2" com 55,3%; e (v) L-SHADE e "rand/2" também com 55,33%.

Observe que o algoritmo DE é o que gera a menor quantidade de caminhos factíveis em comparação com os demais, mesmo quando combinado com o mutador de melhor desempenho. Por outro lado, SHADE e L-SHADE apresentam as combinações com o maior número de soluções factíveis, ficando empatados nesse aspecto.

Tabela 3 – *Ranking* da combinação algoritmo e mutador com menores médias de fitness independentemente do cenário.

Ranking	Algoritmo e Mutador	Pontuação
1º	LSHADE + current-to-rand/1	372
2º	LSHADE + rand/2	365
3º	SHADE + current-to-rand/1	362
4º	LSHADE + either-or	361
5º	LSHADE + current-to-best/1	361
6º	LSHADE + rand/1	358
7º	LSHADE + current-to-pbest/1	358
8º	SHADE + rand/1	357
9º	JADE + current-to-best/1	350
10º	SHADE + best/2	346
11º	LSHADE + best/2	344
12º	LSHADE + best/1	342
13º	SHADE + either-or	342
14º	SADE + rand/1	341
15º	SADE + either-or	339
16º	SHADE + rand/2	338
17º	SHADE + current-to-best/1	334
18º	SHADE + current-to-pbest/1	329
19º	JADE + rand/1	329
20º	JADE + best/2	327
21º	SADE + current-to-best/1	325
22º	JADE + current-to-pbest/1	321
23º	JADE + either-or	314
24º	SADE + current-to-pbest/1	313
25º	JADE + best/1	312
26º	SADE + best/1	311
27º	SADE + best/2	309
28º	SADE + current-to-rand/1	304
29º	JADE + current-to-rand/1	301
30º	JADE + rand/2	300
31º	DE + rand/2	297
32º	SHADE + best/1	293
33º	SADE + rand/2	290
34º	DE + current-to-rand/1	286
35º	DE + rand/1	286
36º	DE + best/2	280
37º	DE + either-or	280
38º	DE + best/1	277
39º	DE + current-to-best/1	275
40º	DE + current-to-pbest/1	251

Tabela 4 – Quantidade percentual de caminhos factíveis por combinação de algoritmo e mutador, destacando a melhor combinação para cada algoritmo.

Mutador	DE	SaDE	JADE	SHADE	L-SHADE
rand/1	22%	52%	48,67%	50,67%	50%
rand/2	20%	48,67%	45,33%	49,33%	55,33%
best/1	25,33%	52,67%	49,33%	48%	52%
best/2	22,67%	47,33%	46,67%	55,33%	52%
either-or	22,67%	50,67%	50,67%	48%	46,67%
current-to-rand/1	24,67%	50,67%	48,67%	46,67%	49,33%
current-to-best/1	24,67%	47,33%	45,33%	49,33%	51,33%
current-to-pbest/1	24,67%	50%	48%	50%	53,33%

Nas Figuras 30, 31, 32, 33 e 34 são apresentados os caminhos com menor valor de *fitness* para cada combinação de algoritmo e mutador, mesmo que essa combinação não tenha um melhor factível, como ocorre nos caminhos do Cenário 5. Cada figura corresponde à um dos cenários, apresentando os resultados para cada um dos mutadores em análise. Cada algoritmo é representado por uma cor e um marcador diferente e, além disso, para cada caminho também é exibido o valor de *fitness* na legenda.

Com relação ao Cenário 1, Figura 30, não há diferença perceptível entre os algoritmos e mutadores utilizados. Nesse caso, todos conseguem alcançar o objetivo esperado. Os caminhos com os menores valores de *fitness* foram obtidos, para a maioria dos mutadores, por meio do algoritmo DE, embora a diferença em relação aos demais algoritmos seja muito pequena.

Assim como no Cenário 1, o Cenário 2 apresentado na Figura 31, de modo geral, não apresenta diferenças significativas entre os mutadores, com exceção da combinação SHADE e "best/2". Essa configuração se destaca por superar todas as demais, propondo um caminho factível mais curto e com um valor de *fitness* consideravelmente menor.

Figura 30 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 1, considerando 30 execuções do planejador para cada mutador. Cada cor representa um algoritmo diferencial evolutivo e o melhor *fitness* é indicado na legenda.

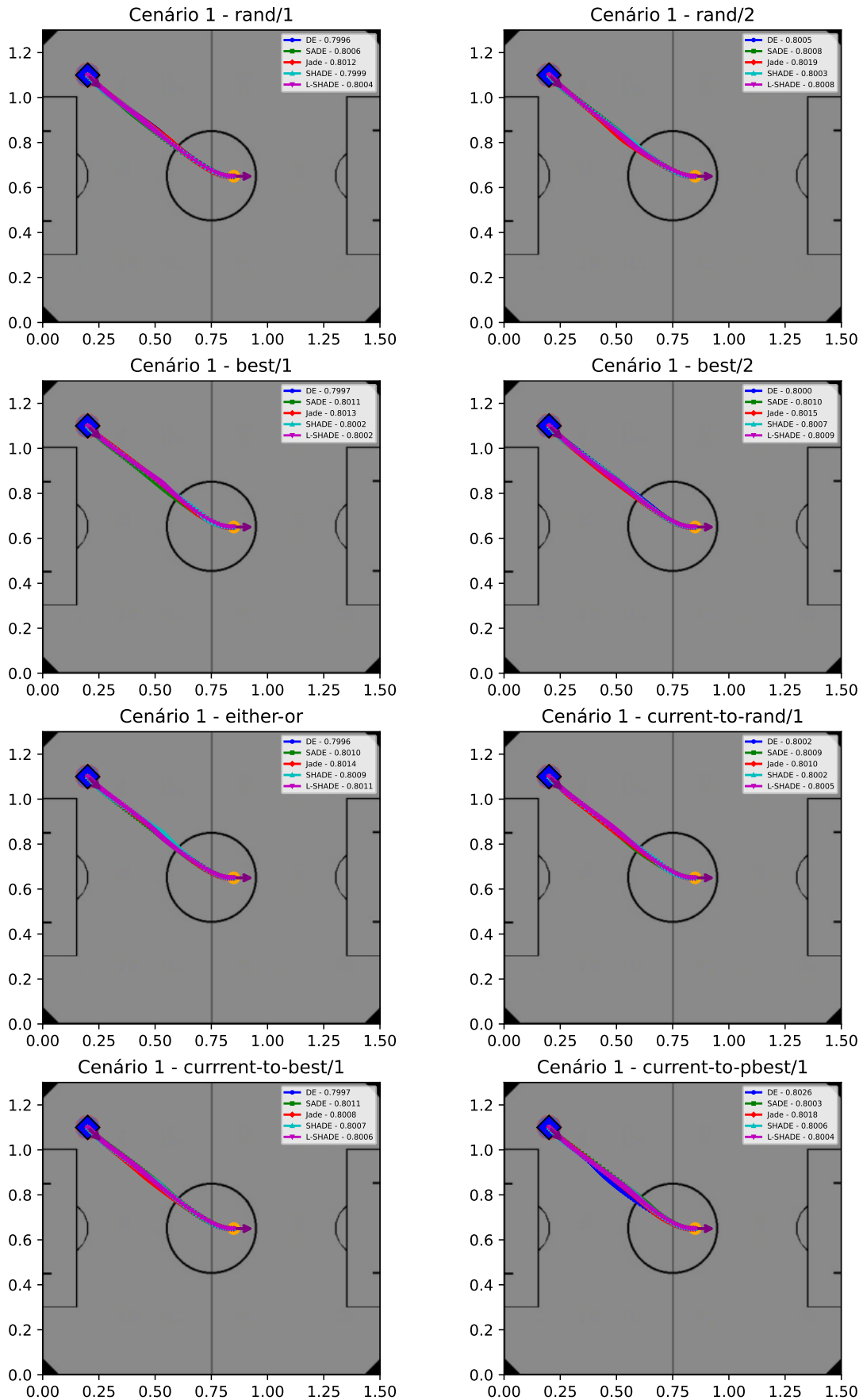
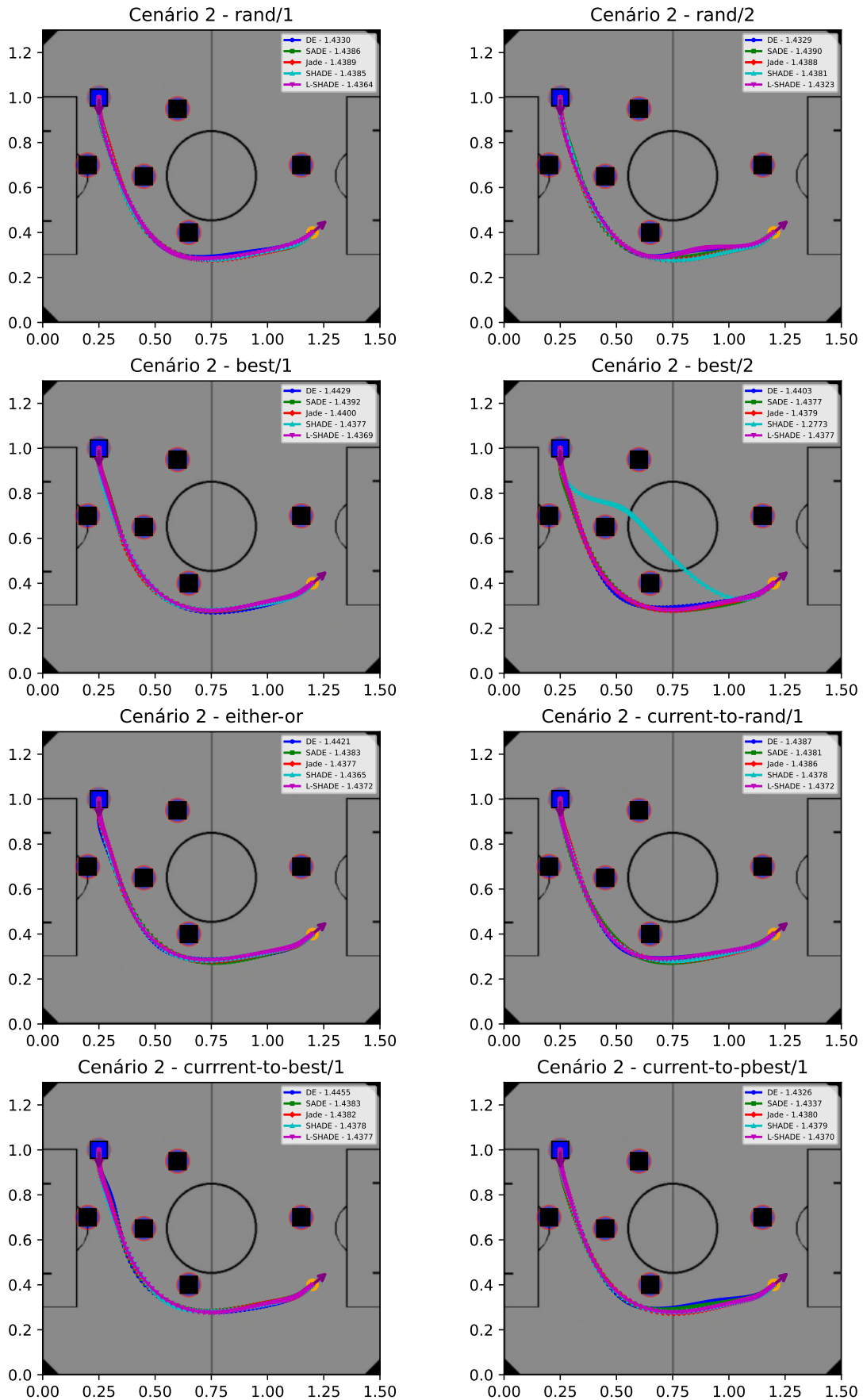


Figura 31 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 2, considerando 30 execuções do planejador para cada mutador. Cada cor representa um algoritmo diferencial evolutivo e o melhor *fitness* é indicado na legenda.



No Cenário 3, Figura 32, é possível observar com mais clareza a influência dos operadores de mutação no planejamento de caminho. Nesse cenário, o algoritmo DE não obteve um caminho factível na maioria dos mutadores, falhando no atendimento a algumas restrições. No entanto, ao utilizar os mutadores "best/2" e "either-or", o DE foi capaz de encontrar uma solução factível. Além disso, nota-se que o operador "current-to-pbest/1" resultou, para o DE, um caminho pior, apresentando um maior valor de *fitness*.

Quanto ao Cenário 4, ilustrado na Figura 33, foram observadas diferenças significativas no desempenho dos mutadores ao gerar caminhos com o algoritmo JADE. Especificamente, os mutadores "rand/2" e "best/1" não conseguiram produzir um caminho factível (identificado pelo alto valor de *fitness* apresentado nas legendas), possivelmente devido à incapacidade de atender à restrição de curvatura. Em contraste, os demais algoritmos não apresentaram variações perceptíveis em seus resultados, mantendo um desempenho consistente.

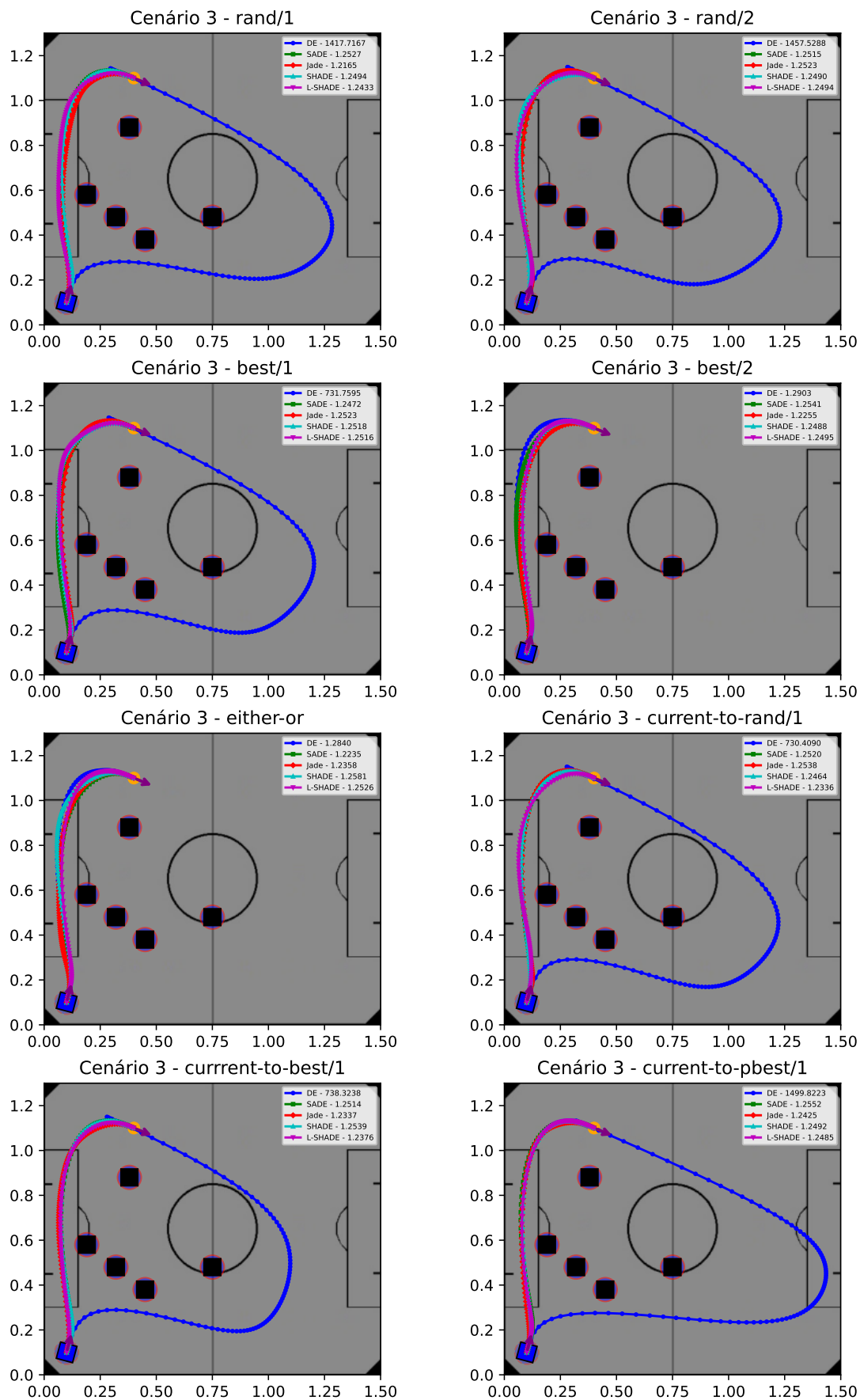
Por fim, no Cenário 5, ilustrado na Figura 34, confirma-se que nenhum dos algoritmos foi capaz de gerar um caminho factível, uma vez que todos apresentaram valores de *fitness* significativamente elevados devido à incapacidade de atender à restrição de curvatura máxima do robô futebolista. No entanto, ao analisar os caminhos gerados por cada mutador, observam-se algumas tentativas de caminhos mais curtos e uma maior abertura no ângulo de chegada à bola, indicando esforços parciais para otimizar o movimento, ainda que sem sucesso total em relação à restrição de curvatura.

Diante de cenários complexos, como o Cenário 5, a variação do parâmetro α talvez poderia auxiliar na geração de caminhos mais adequados, permitindo um ajuste que favorecesse a obtenção de soluções viáveis.

De modo geral, a análise dos resultados apresentados nas Tabelas 2, 3 e 4, juntamente com a comparação dos caminhos gerados em cada cenário e mutador, indica que o algoritmo L-SHADE superou os demais. Ele obteve, com maior frequência, os menores valores médios de *fitness*, consolidando seu desempenho superior, além de apresentar como melhor combinação de algoritmo e mutador (L-SHADE e "current-to-rand/1").

Conforme observado na seção anterior, os diferentes mutadores não geraram variações visualmente perceptíveis nos caminhos obtidos pelo L-SHADE. Dessa forma, a escolha de um mutador que retorna mais soluções factíveis também pode ser uma boa opção para o problema de planejamento de caminho de um robô futebolista, sendo nesse caso o "rand/2". Além da alta factibilidade, esse mutador também proporcionou valores médios de *fitness* inferiores aos demais operadores de mutação aplicados ao L-SHADE, sendo o segundo colocado no *ranking* exibido na Tabela 3.

Figura 32 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 3, considerando 30 execuções do planejador para cada mutador. Cada cor representa um algoritmo diferencial evolutivo e o melhor *fitness* é indicado na legenda.



Fonte: Elaborada pelo autor.

Figura 33 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 4, considerando 30 execuções do planejador para cada mutador. Cada cor representa um algoritmo diferencial evolutivo e o melhor *fitness* é indicado na legenda.

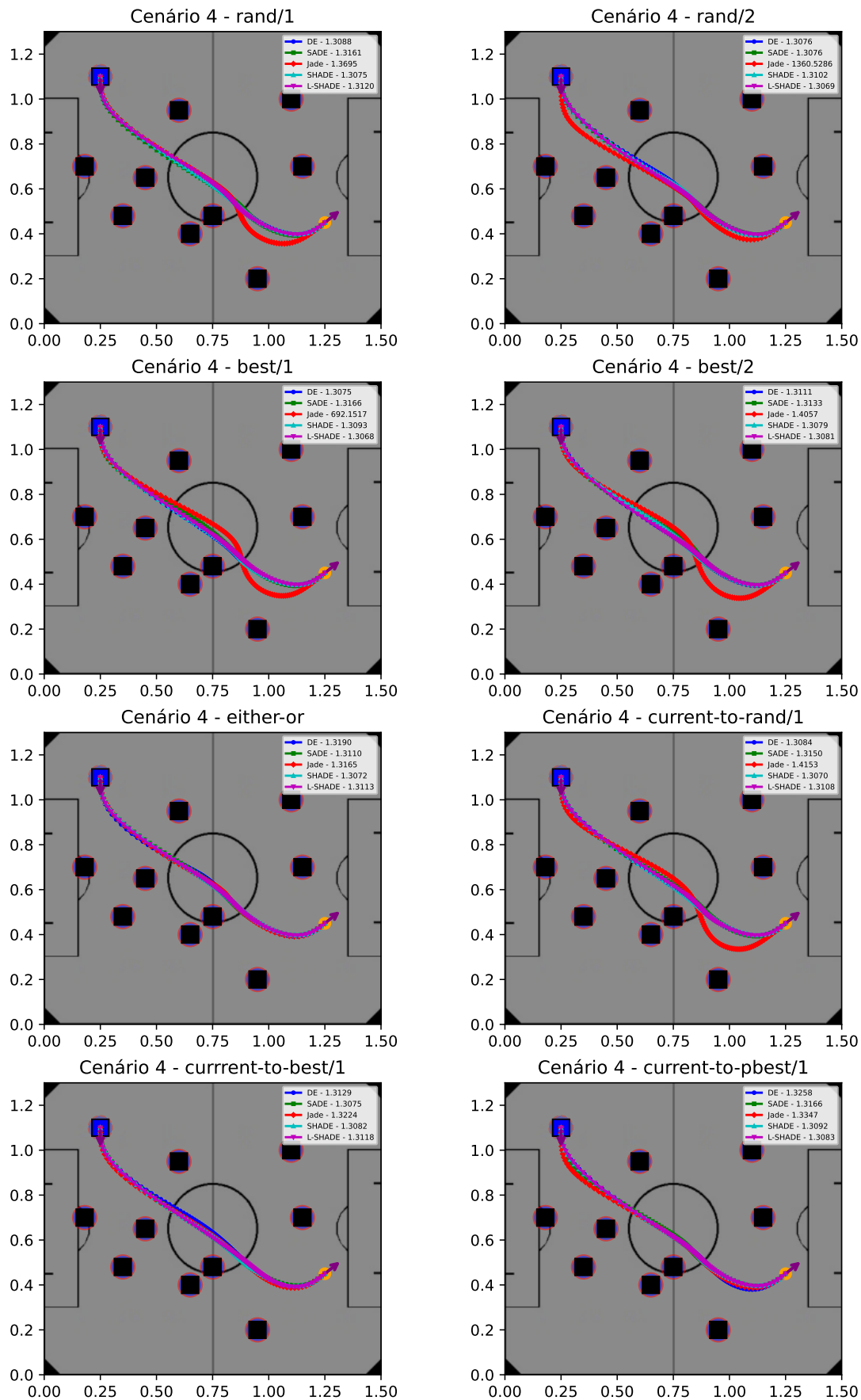
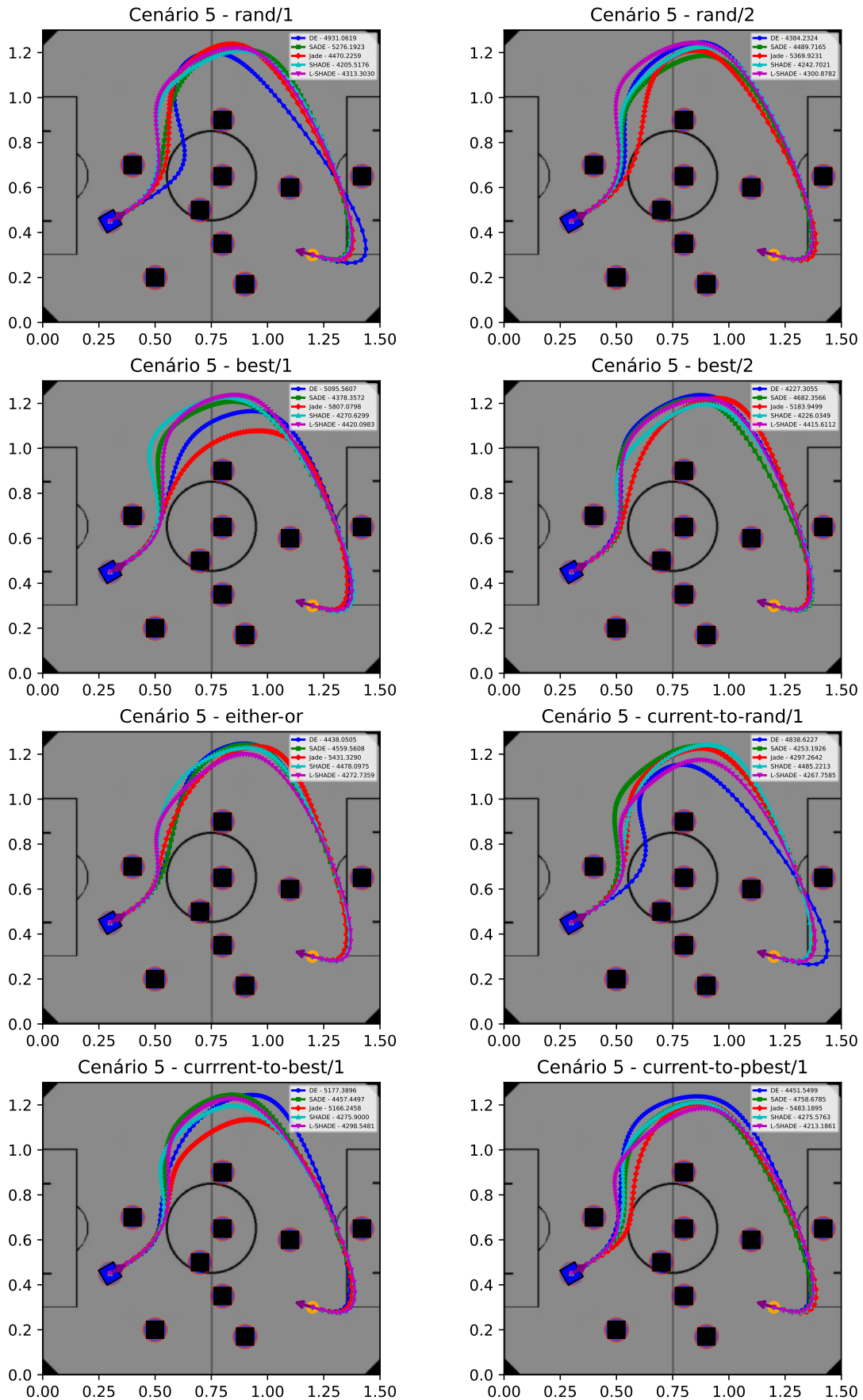


Figura 34 – Melhores caminhos obtidos com DEs e mutadores para o Cenário 5, considerando 30 execuções do planejador para cada mutador. Cada cor representa um algoritmo diferencial evolutivo e o melhor *fitness* é indicado na legenda.



5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho foi realizada a análise de desempenho de algoritmos diferenciais evolutivos, especificamente: DE, SaDE, JADE, SHADE e L-SHADE. Essa análise foi realizada variando diferentes tipos de operadores de mutação, encontrados na literatura, para os diferentes algoritmos DEs a fim de identificar qual a melhor combinação de algoritmo e mutador para o problema específico de planejamento de caminho de um robô futebolista.

Primeiramente, comparou-se a influência dos operadores de mutação "rand/1", "current-to-best/1" e "current-to-pbest"/1 e o algoritmo L-SHADE aplicados a problemas irrestritos do *benchmark* CEC2013.

Nessa comparação, verificou-se que, para a dimensão $D = 2$ (problemas de baixa complexidade), o operador "rand/1" obteve, em média, um desempenho superior aos demais operadores. Em problemas com dimensionalidade $D = 10$, os resultados dos diferentes operadores não mostraram diferenças significativas. No entanto, para problemas de maior complexidade ($D = 30$ e $D = 50$), o operador "current-to-pbest/1" se destacou, apresentando um desempenho significativamente melhor do que os outros, ressaltando a importância de sua escolha para esse tipo de problema.

Após essa etapa, foram realizadas comparações, em um segundo momento, dos algoritmos DEs e dos mutadores aplicados ao problema específico de planejamento de caminho de um robô futebolista. Primeiro foi comparado apenas o L-SHADE para cada mutador e, depois, foram comparados todos os demais algoritmos.

Pode-se concluir que, para esse problema, alguns algoritmos são mais sensíveis à diferentes mutadores, como o DE, JADE e o SHADE. Também se conclui que o algoritmo L-SHADE demonstrou desempenho superior em relação aos demais, alcançando, com maior frequência, os menores valores médios de *fitness*. A combinação L-SHADE e "current-to-rand/1" obteve o melhor desempenho geral. Já a combinação L-SHADE e "rand/2" obteve a maior quantidade de soluções factíveis, 55,33%, considerando todos os cenários. Essa combinação também se destacou como no desempenho geral, obtendo o segundo lugar geral. Assim, ambas opções de combinações são as escolhas mais recomendadas para o planejamento de caminho de um robô futebolista.

5.1 Trabalhos Futuros

Como propostas para trabalhos futuros, sugere-se a exploração e implementação de abordagens que possam aprimorar e expandir os resultados obtidos neste trabalho. As princi-

país direções incluem:

- Implementação de uma análise estatística para validar se existem diferenças significativas de desempenho dos algoritmos.
- Incorporação de uma análise multiobjetivo, considerando não apenas a minimização do comprimento do caminho, mas também a minimização da curvatura. Essa abordagem permitiria a geração de caminhos mais suaves, adequadas para situações mais complexas, como a apresentada no Cenário 5 deste trabalho;
- Implementação e avaliação de algoritmos que considerem restrições durante o processo de evolução, como o CDE (*Constraint Differential Evolution*) introduzido em (Yu; Li; Zhou, 2020) e L-SHADE-COP proposto em (Freitas *et al.*, 2024). Esses métodos podem melhorar a capacidade de gerar soluções factíveis em cenários complexos, considerando as restrições do problema de otimização;
- Validação dos caminhos gerados pelos diferentes algoritmos e operadores de mutação propostos em ambientes reais, utilizando robôs móveis, sejam eles futebolistas ou não. Dessa forma, será possível verificar na prática se os robôs conseguirão percorrer os caminhos gerados, respeitando as restrições.

5.2 Publicações, participação em projetos de pesquisa e premiações

Os seguintes trabalhos foram originados diretamente do trabalho com futebol de robôs:

1. OLIVEIRA, V. C. R. de *et al.* Protótipo de robô para a categoria ieev very small size soccer: design mecânico, eletrônica embarcada e simulação. Anais do III Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia, p. 1–7, 2022.
2. FERREIRA, R. H. X. *et al.* Sistema de visão computacional para a categoria ieev very small size soccer: implementação e desenvolvimento. Anais do IV Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia, p. 1-6, 2023.
3. FERREIRA, R. H. X. *et al.* Initial development of ieev very small size soccer robots: Electromechanics, computer vision system and navigation strategy. The Journal of Engineering and Exact Sciences –jCEC, p. 1–16, 2023.
4. SILVA, P. L. da; SOUZA, D. S. C.; FREITAS, E. J. de R. Análise dos operadores de mutação no desempenho do algoritmo l-shade em problemas irrestritos. Anais do V Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia (CoBICET), p. 1–9, 2024.

Outros trabalhos também foram publicados durante a graduação:

1. SILVA, P. L. da *et al.* Modelagem de um sistema de aquecimento solar de água. *Eletricidade Moderna*, p. 1-4, 2023.
2. SILVA, P. L. da *et al.* Implementação Inicial de um painel de monitoramento (dashboard) para um ambiente maker. *Anais do IV Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia*, p. 1-6, 2023.

Sendo que alguns desses trabalhos e outras participações foram premiadas:

1. Primeiro lugar na categoria ENSINO na Semana Nacional de Ciência e Tecnologia do Instituto Federal de Minas Gerais (2020).
2. Quarto lugar no Ideathon, competição de ideias realizada em meio às atividades da Semana Global de Empreendedorismo e Semana Nacional de Ciência e Tecnologia do IFMG campus Ibirité (2022).
3. Terceiro lugar pelo artigo "Sistema de Visão Computacional para a Categoria IEEE Very Small Size Soccer: Implementação e Desenvolvimento" no evento IV CoBICET (2023).
4. Menção honrosa pelo artigo "Implementação Inicial de um Painel de Monitoramento (dashboard) para um Ambiente Maker" no evento IV CoBICET (2023).
5. Sexto lugar com a equipe Tech Fish na V Olimpíada de Inovação do IFMG (2024).

Por fim, são elencados os projetos de pesquisa com participação como bolsista e/ou voluntário, incluindo o projeto relacionado ao futebol de robôs:

1. Estudo da incidência de descargas atmosféricas em MG: Verificação da polaridade prevalente e variação do número de incidência entre os anos de 2012 e 2017. (jul. de 2021 à mar. de 2022).
2. Inserção Profissional da Mulher na Engenharia: um estudo com egressas dos cursos de Engenharia do IFMG. (mai. de 2020 à jul. de 2022).
3. Desenvolvimento de robôs futebolistas da categoria IEEE Very Small Size Soccer. (jul. de 2021 à set. de 2023).

REFERÊNCIAS

- ALAM, T.; OKAFOR, F.; PATEL, A.; NEWAZ, A. A. R. Dvf-rrt: Randomized path planning on predictive vector fields. **Sixth IEEE International Conference on Robotic Computing (IRC)**, p. 403–410, 2022. Citado na página 16.
- AYAWLI, B. B. K. *et al.* Mobile robot path planning in dynamic environment using voronoi diagram and computation geometry technique. **IEEE Access**, v. 7, p. 86026–86040, 2019. Citado na página 16.
- CHEN, C.-A.; CHIANG, T.-C. Adaptive differential evolution: A visual comparison. **2015 IEEE Congress on Evolutionary Computation (CEC), Sendai**, p. 401–408, 2015. Citado na página 26.
- EIBEN, A.; SMITH, J. **Introduction to Evolutionary Computing**. Amsterdam: Natural Computing Series - Springer, 2015. ISBN 978-3-662-44873-1. Citado na página 27.
- FERREIRA, R. H. X. *et al.* Initial development of ieee very small size soccer robots: Electromechanics, computer vision system and navigation strategy. **The Journal of Engineering and Exact Sciences –jCEC**, p. 1–16, 2023. Citado na página 14.
- FERREIRA, W. R. B.; CARVALHO, J. C. M. Curvas utilizadas no planejamento de trajetórias de um robô industrial. **VII Congresso Nacional de Engenharia Mecânica**, p. 1–9, 2012. Citado na página 15.
- FREITAS, E. J. *et al.* De3d-nurbs: A differential evolution-based 3d path-planner integrating kinematic constraints and obstacle avoidance. **Knowledge-Based Systems**, v. 300, p. 1–16, 2024. Citado 3 vezes nas páginas 17, 36 e 72.
- FREITAS, E. J. de R.; COHEN, M. W.; GUIMARÃES, F. G. An autonomous mobile robot path planner using spline curves and differential evolution. **IEEE Latin American Conference on Computational Intelligence (LA-CCI)**, p. 1–6, 2023. Citado 2 vezes nas páginas 17 e 48.
- GANESAN, S.; RAMALINGAM, B.; MOHAN, R. E. A hybrid sampling-based rrt* path planning algorithm for autonomous mobile robot navigation. **Expert Systems With Applications**, v. 258, p. 1–12, 2024. Citado na página 15.
- GHALLABI, F. **Precise self-localization of autonomous vehicles using lidar sensors and highly accurate digital maps on highway roads**. Dissertação (Mestrado) — Université Paris sciences et lettres, 2020. Citado na página 23.
- GOMEZ, V. *et al.* Robominer: Development of a highly configurable and modular scaled-down prototype of a mining robot. **MDPI: Machines**, v. 11, n. 809, p. 1–23, 2023. Citado na página 19.
- GONG, W.; CAI, Z. Differential evolution with ranking-based mutation operators. **IEEE TRANSACTIONS ON CYBERNETICS**, v. 43, n. 6, p. 2066–2081, 2013. Citado 3 vezes nas páginas 17, 27 e 30.

IEEE VSSS. **Liga Very Small Size Soccer**. 2024. Disponível em: <https://ieeevss.github.io/vss/index_ptbr.html>. Acesso em: 18 de dezembro 2024. Citado na página 14.

JI, H.; XIE, H.; WANG, C.; YANG, H. E-rrt*: Path planning for hyper-redundant manipulators. **IEEE Robotics and Automation Letters**, v. 8, n. 12, p. 8128–8135, 2023. Citado na página 15.

LI, X. *et al.* Ai-empowered intelligent search for path planning in uav-assisted data collection networks. **IEEE INTERNET OF THINGS JOURNAL**, v. 11, n. 21, p. 34492–34503, 2024. Citado na página 17.

LIANG, J. J.; QU, B. Y.; SUGANTHAN, P. N.; HERNÁNDEZ-DÍAZ, A. G. Problem definitions and evaluation criteria for the cec2013 special session on real-parameter optimization. **IEEE Congress on Evolutionary Computation, CEC 2013**, p. 1–39, 2013. Citado 3 vezes nas páginas 33, 34 e 82.

LYNCH, K. M.; PARK, F. C. **Modern Robotics: Mechanics, Planning, and Control**. Cambridge: Cambridge University Press, 2017. ISBN 9781107156302. Citado na página 14.

MENDIBURU, F. J. **Planejamento de Trajetória Baseado em Visão Computacional para Manipulador Robótico**. Dissertação (Mestrado) — Universidade Federal de Campina Grande, 2013. Citado na página 14.

MILANO, D. de; HONORATO, L. B. Visão computacional. **Faculdade de Tecnologia, Universidade Estadual de Campinas**, p. 1–7, 2010. Citado na página 14.

MIRELES, C. *et al.* Home-care nursing controlled mobile robot with vital signal monitoring. **Medical & Biological Engineering & Computing**, v. 61, p. 399–420, 2022. Citado na página 19.

MOHAMED, A. W.; HADI, A. A.; JAMBI, K. M. Novel mutation strategy for enhancing shade and lshade algorithms for global numerical optimization. **Swarm and Evolutionary Computation**, v. 50, p. 1–14, 2019. Citado na página 28.

NASCIMENTO, L. B. P. *et al.* Safe path planning algorithms for mobile robots based on probabilistic foam. **Journal of Sensors**, p. 1–20, 2021. Citado na página 16.

NGUYEN, N. T. *et al.* B-spline path planner for safe navigation of mobile robots. **IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**, p. 339–345, 2021. Citado na página 15.

OLIVEIRA, V. C. R. de *et al.* Protótipo de robô para a categoria ieev very small size soccer: design mecânico, eletrônica embarcada e simulação. **Anais do III Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia**, p. 1–7, 2022. Citado 2 vezes nas páginas 22 e 24.

OPARA, K.; ARABAS, J. Comparison of mutation strategies in differential evolution – a probabilistic perspective. **Swarm and Evolutionary Computation**, v. 39, p. 53–69, 2018. Citado 2 vezes nas páginas 17 e 30.

- PATLE, B. *et al.* A review: On path planning strategies for navigation of mobile robot. **Defence Technology**, v. 15, p. 582–606, 2019. Citado na página 19.
- PIEGL, L.; TILLER, W. **The NURBS Book**. Berlin: Springer Berlin, Heidelberg, 1996. ISBN 978-3-540-61545-3. Citado na página 36.
- QIN, A. K.; SUGANTHAN, P. N. Self-adaptive differential evolution algorithm for numerical optimization. **IEEE Congress on Evolutionary Computation (CEC)**, p. 1785–1791, 2005. Citado na página 28.
- RAMOS, H. A.; OLIVEIRA, D. J. de; OLIVEIRA, T. G. de; PUGLIESE, L. F. Implementação em simulação de um controlador pd para rastreamento de trajetória de robôs móveis aplicados ao futebol de robôs. **XV Simpósio Brasileiro de Automação Inteligente - SBAI**, p. 131–137, 2021. Citado na página 21.
- RAVANKAR, A.; RAVANKAR, A.; EMARU, T.; KOBAYASHI, Y. Hpprm: Hybrid potential based probabilistic roadmap algorithm for improved dynamic path planning of mobile robots. **IEEE Access**, 2020. Citado na página 26.
- RENKAVIESKI, C. Evolução diferencial com parametrização adaptativa aplicada a problemas contínuos sem restrições. **Universidade do Estado de Santa Catarina**, 2019. Citado na página 40.
- RIBOLI, M. *et al.* Collision-free and smooth motion planning of dual-arm cartesian robot based on b-spline representation. **Robotics and Autonomous Systems**, p. 1–25, 2023. Citado na página 15.
- ROMERO, R. A. F.; PRESTES, E.; OSÓRIO, F.; WOLF, D. **Robótica Móvel**. Rio de Janeiro: LTC, 2017. ISBN 978-8-521-62303-8. Citado na página 23.
- ROSA, J. F. da; ROSA, P. F. F. Planejamento de trajetória de múltiplos robôs terrestres autônomos em ambientes dinâmicos. **Revista Militar de Ciência e Tecnologia**, p. 27–36, 2021. Citado 2 vezes nas páginas 16 e 24.
- SANTOS, P. H. dos; FILHO, A. M. de S. T.; FREITAS, E. J. de R. Desenvolvimento de robôs diferenciais futebolistas: uma aplicação de campos vetoriais no planejamento de caminho. **Anais do 14º Simpósio Brasileiro de Automação Inteligente**, p. 1313–1318, 2019. Citado 3 vezes nas páginas 16, 19 e 25.
- SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. **Introduction to Autonomous Mobile Robots**. 2. ed. London: The MIT Press, 2011. Citado na página 21.
- SILVA, P. L. da; SOUZA, D. S. C.; FREITAS, E. J. de R. Análise dos operadores de mutação no desempenho do algoritmo l-shade em problemas irrestritos. **Anais do V Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia (CoBICET)**, p. 1–9, 2024. Citado 3 vezes nas páginas 27, 30 e 42.

- STORN, R.; PRICE, K. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, p. 341–359, 1997. Citado na página 27.
- TAGLIAVINI, L. *et al.* Wheeled mobile robots: State of the art overview and kinematic comparison among three omnidirectional locomotion strategies. **Journal of Intelligent & Robotic Systems**, v. 106, n. 57, p. 1–18, 2022. Citado 2 vezes nas páginas 20 e 21.
- TANABE, R.; FUKUNAGA, A. Success-history based parameter adaptation for differential evolution. **IEEE Congress on Evolutionary Computation, CEC 2013**, p. 71–78, 2013. Citado na página 29.
- TANABE, R.; FUKUNAGA, A. S. Improving the search performance of shade using linear population size reduction. **IEEE Congress on Evolutionary Computation (CEC)**, p. 1658–1665, 2014. Citado 4 vezes nas páginas 27, 30, 41 e 42.
- TRIVEDI, A.; SANYAL, K.; VERMA, P.; SRINIVASAN, D. A unified differential evolution algorithm for constrained optimization problems. **IEEE Congress on Evolutionary Computation (CEC)**, p. 1231–1238, 2017. Citado na página 17.
- ULLAH, I. *et al.* Mobile robot localization: Current challenges and future prospective. **Computer Science Review**, v. 53, p. 1–34, 2024. Citado na página 20.
- VIKHAR, P. A. Evolutionary algorithms: A critical review and its future prospects. **2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication**, p. 261–265, 2016. Citado 2 vezes nas páginas 17 e 26.
- XU, X.; ZENG, J.; ZHAO, Y.; Lü, X. Research on global path planning algorithm for mobile robots based on improved a*. **Expert Systems With Applications**, v. 243, p. 1–12, 2024. Citado na página 19.
- YAN, C. Research on path planning of robot based on artificial intelligence algorithm. **5th International Conference on Intelligent Computing and Signal Processing (ICSP)**, p. 1–6, 2020. Citado na página 17.
- YU, X.; LI, C.; ZHOU, J. A constrained differential evolution algorithm to solve uav path planning in disaster scenarios. **Knowledge-Based Systems**, v. 204, 2020. Citado na página 72.
- ZHANG, J.; SANDERSON, A. C. Jade: Adaptive differential evolution with optional external archive. **IEEE Transactions on Evolutionary Computation**, v. 13, n. 5, p. 945–958, 2009. Citado na página 29.
- ŠLAJPAH, S.; MUNIH, M.; MIHELJ, M. Mobile robot system for selective asparagus harvesting. **MDPI: Agronomy**, v. 13, n. 1766, p. 1–17, 2023. Citado na página 19.

APÊNDICE A – RESULTADOS PARA OS PROBLEMAS IRRESTRITOS DO *BENCHMARK* CEC2013

Resultados obtidos para os três operadores de mutação analisados neste trabalho, considerando as 28 funções do *benchmark* CEC2013.

Tabela 5 – Média e desvio padrão do valor de *fitness* para D=2.

Função	rand/1	current-to-best/1	current-to-pbest/1
1	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
2	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
3	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
4	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
5	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
6	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
7	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
8	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
9	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
10	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
11	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
12	3,317e-02 ± 1,817e-01	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
13	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	6,633e-02 ± 3,633e-01
14	2,081e-02 ± 7,920e-02	8,325e-02 ± 1,404e-01	5,203e-02 ± 1,657e-01
15	1,353e-01 ± 1,954e-01	1,977e-01 ± 2,388e-01	1,458e-01 ± 2,127e-01
16	2,305e-03 ± 6,053e-03	6,697e-02 ± 1,503e-01	1,380e-02 ± 3,941e-02
17	8,078e-01 ± 1,006e+00	1,212e+00 ± 1,006e+00	1,010e+00 ± 1,027e+00
18	9,343e-01 ± 1,016e+00	1,160e+00 ± 9,881e-01	1,005e+00 ± 1,019e+00
19	0,000e+00 ± 0,000e+00	1,315e-03 ± 5,005e-03	0,000e+00 ± 0,000e+00
20	1,052e-02 ± 9,724e-03	7,773e-03 ± 9,682e-03	5,830e-03 ± 9,057e-03
21	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
22	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
23	1,075e+00 ± 4,093e+00	2,151e+00 ± 5,578e+00	3,333e+00 ± 1,826e+01
24	2,098e-01 ± 7,985e-01	1,049e-01 ± 5,746e-01	0,000e+00 ± 0,000e+00
25	2,333e+01 ± 4,302e+01	2,333e+01 ± 4,302e+01	2,000e+01 ± 4,068e+01
26	1,990e-02 ± 8,106e-02	2,800e-02 ± 8,275e-02	4,250e-02 ± 1,110e-01
27	5,034e+01 ± 5,124e+01	5,336e+01 ± 5,071e+01	7,373e+01 ± 4,513e+01
28	1,000e+01 ± 3,051e+01	1,000e+01 ± 3,051e+01	1,333e+01 ± 3,457e+01

Tabela 6 – Média e desvio padrão do valor de *fitness* para D=10.

Função	rand/1	current-to-best/1	current-to-pbest/1
1	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
2	3,032e-14 ± 1,299e-13	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
3	7,338e-02 ± 1,734e-01	4,757e-03 ± 1,810e-02	2,379e-03 ± 1,303e-02
4	5,305e-14 ± 1,858e-13	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
5	5,305e-14 ± 1,858e-13	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
6	0,000e+00 ± 0,000e+00	3,566e+00 ± 4,626e+00	6,215e+00 ± 4,809e+00
7	2,517e-03 ± 4,772e-03	4,869e-03 ± 1,999e-02	1,893e-06 ± 2,874e-06
8	2,035e+01 ± 8,236e-02	2,021e+01 ± 1,292e-01	2,025e+01 ± 1,465e-01
9	1,619e+00 ± 1,683e+00	1,248e+00 ± 1,355e+00	2,661e+00 ± 1,359e+00
10	4,470e-02 ± 2,562e-02	6,980e-02 ± 3,703e-02	3,618e-03 ± 7,830e-03
11	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00	0,000e+00 ± 0,000e+00
12	6,600e+00 ± 2,567e+00	3,821e+00 ± 2,168e+00	2,265e+00 ± 1,005e+00
13	1,170e+01 ± 5,656e+00	3,575e+00 ± 2,570e+00	1,873e+00 ± 8,919e-01
14	0,000e+00 ± 0,000e+00	8,952e-02 ± 7,089e-02	1,041e-02 ± 2,367e-02
15	4,780e+02 ± 1,780e+02	3,470e+02 ± 1,151e+02	3,469e+02 ± 9,964e+01
16	5,294e-01 ± 4,775e-01	2,394e-01 ± 1,428e-01	2,070e-01 ± 1,315e-01
17	1,012e+01 ± 2,026e-14	9,785e+00 ± 1,848e+00	1,012e+01 ± 1,807e-15
18	1,668e+01 ± 3,475e+00	1,612e+01 ± 2,867e+00	1,374e+01 ± 9,489e-01
19	4,501e-01 ± 1,872e-01	2,239e-01 ± 3,928e-02	2,249e-01 ± 3,303e-02
20	1,967e+00 ± 4,953e-01	1,999e+00 ± 5,070e-01	1,938e+00 ± 3,447e-01
21	3,602e+02 ± 8,145e+01	3,868e+02 ± 5,079e+01	4,002e+02 ± 2,891e-13
22	7,487e+00 ± 2,554e+00	4,953e+01 ± 4,670e+01	8,655e+00 ± 1,355e+01
23	4,325e+02 ± 2,203e+02	3,763e+02 ± 1,776e+02	2,639e+02 ± 1,526e+02
24	1,932e+02 ± 2,771e+01	2,027e+02 ± 3,442e+00	2,027e+02 ± 3,357e+00
25	1,924e+02 ± 2,628e+01	1,990e+02 ± 1,812e+01	1,974e+02 ± 1,761e+01
26	1,067e+02 ± 2,628e+00	1,746e+02 ± 4,287e+01	1,415e+02 ± 4,858e+01
27	3,033e+02 ± 1,826e+01	3,098e+02 ± 3,910e+01	3,000e+02 ± 0,000e+00
28	2,733e+02 ± 6,915e+01	2,950e+02 ± 6,989e+01	3,000e+02 ± 0,000e+00

Tabela 7 – Média e desvio padrão do valor de *fitness* para D=30.

Função	rand/1	current-to-best/1	current-to-pbest/1
1	0,000e+00 ± 0,000e+00	2,122e-13 ± 5,769e-14	0,000e+00 ± 0,000e+00
2	4,646e+04 ± 5,883e+04	1,025e+04 ± 1,431e+04	5,230e-13 ± 1,095e-12
3	2,570e+05 ± 1,109e+06	1,083e+05 ± 2,761e+05	1,040e-01 ± 5,151e-01
4	1,834e+01 ± 1,738e+01	1,238e-08 ± 2,693e-08	8,337e-14 ± 1,114e-13
5	1,895e-13 ± 4,657e-13	2,956e-13 ± 7,670e-14	1,137e-13 ± 1,284e-29
6	1,389e+01 ± 3,546e+00	4,922e+00 ± 9,821e+00	1,544e-10 ± 5,567e-10
7	1,508e+00 ± 1,643e+00	1,091e+01 ± 6,563e+00	6,732e-01 ± 4,162e-01
8	2,095e+01 ± 3,283e-02	2,083e+01 ± 1,590e-01	2,082e+01 ± 1,372e-01
9	1,032e+01 ± 3,965e+00	2,190e+01 ± 6,637e+00	2,600e+01 ± 1,900e+00
10	1,831e-02 ± 1,542e-02	2,224e-02 ± 1,702e-02	2,465e-04 ± 1,350e-03
11	1,648e-13 ± 6,177e-13	1,345e-13 ± 4,597e-14	8,337e-14 ± 3,248e-14
12	2,912e+01 ± 8,732e+00	1,847e+01 ± 5,081e+00	5,144e+00 ± 1,282e+00
13	5,594e+01 ± 2,088e+01	4,230e+01 ± 1,686e+01	5,461e+00 ± 2,587e+00
14	3,470e-03 ± 9,600e-03	6,107e-02 ± 3,940e-02	3,053e-02 ± 2,169e-02
15	2,932e+03 ± 7,126e+02	2,834e+03 ± 2,687e+02	2,700e+03 ± 2,512e+02
16	7,404e-01 ± 8,481e-01	5,805e-01 ± 2,861e-01	7,663e-01 ± 2,512e-01
17	3,043e+01 ± 2,055e-06	3,043e+01 ± 1,230e-06	3,043e+01 ± 1,230e-06
18	5,923e+01 ± 8,151e+00	6,460e+01 ± 6,012e+00	5,125e+01 ± 3,699e+00
19	2,434e+00 ± 4,590e-01	1,183e+00 ± 1,259e-01	1,164e+00 ± 9,393e-02
20	1,116e+01 ± 1,300e+00	1,049e+01 ± 1,646e+00	9,347e+00 ± 3,700e-01
21	3,039e+02 ± 7,487e+01	2,973e+02 ± 7,939e+01	2,967e+02 ± 1,826e+01
22	1,107e+02 ± 2,793e+00	1,075e+02 ± 4,170e+01	1,077e+02 ± 1,825e+00
23	2,711e+03 ± 6,105e+02	2,856e+03 ± 3,773e+02	2,517e+03 ± 2,961e+02
24	2,034e+02 ± 3,005e+00	2,218e+02 ± 1,011e+01	2,005e+02 ± 1,118e+00
25	2,524e+02 ± 1,565e+01	2,536e+02 ± 7,279e+00	2,406e+02 ± 7,672e+00
26	2,000e+02 ± 2,619e-03	2,043e+02 ± 2,339e+01	2,000e+02 ± 1,156e-13
27	3,660e+02 ± 1,084e+02	5,405e+02 ± 1,019e+02	3,014e+02 ± 3,537e+00
28	3,000e+02 ± 2,542e-13	3,000e+02 ± 2,407e-13	3,000e+02 ± 1,961e-13

Tabela 8 – Média e desvio padrão do valor de *fitness* para D=50.

Função	rand/1	current-to-best/1	current-to-pbest/1
1	2,274e-14 ± 6,938e-14	4,169e-13 ± 1,346e-13	0,000e+00 ± 0,000e+00
2	6,238e+05 ± 2,454e+05	1,032e+05 ± 5,222e+04	8,183e+02 ± 1,072e+03
3	8,848e+06 ± 1,047e+07	7,432e+06 ± 1,256e+07	2,112e+03 ± 4,506e+03
4	3,053e+02 ± 1,033e+02	1,245e-03 ± 1,002e-03	5,811e-11 ± 8,239e-11
5	6,706e-09 ± 2,057e-08	7,200e-13 ± 1,278e-13	1,440e-13 ± 5,113e-14
6	4,594e+01 ± 7,913e-01	4,364e+01 ± 1,037e+00	4,345e+01 ± 0,000e+00
7	9,007e+00 ± 3,489e+00	3,360e+01 ± 1,178e+01	2,277e+00 ± 1,534e+00
8	2,112e+01 ± 4,367e-02	2,105e+01 ± 1,408e-01	2,109e+01 ± 6,813e-02
9	2,179e+01 ± 4,678e+00	4,624e+01 ± 1,158e+01	5,352e+01 ± 1,702e+00
10	8,113e-02 ± 3,406e-02	3,380e-02 ± 2,696e-02	9,444e-03 ± 8,335e-03
11	9,618e-01 ± 9,594e-01	1,370e-10 ± 2,581e-10	1,678e-10 ± 4,858e-10
12	5,893e+01 ± 1,180e+01	6,419e+01 ± 1,817e+01	1,432e+01 ± 2,270e+00
13	1,171e+02 ± 2,885e+01	1,380e+02 ± 3,330e+01	2,423e+01 ± 8,326e+00
14	4,197e-02 ± 1,901e-02	2,453e-01 ± 7,103e-02	2,257e-01 ± 6,099e-02
15	6,258e+03 ± 1,025e+03	6,457e+03 ± 5,763e+02	6,359e+03 ± 4,177e+02
16	9,841e-01 ± 1,246e+00	9,716e-01 ± 3,197e-01	1,241e+00 ± 1,405e-01
17	5,079e+01 ± 4,022e-03	5,079e+01 ± 3,201e-03	5,079e+01 ± 2,615e-03
18	9,912e+01 ± 1,245e+01	1,431e+02 ± 1,140e+01	1,014e+02 ± 6,717e+00
19	4,482e+00 ± 6,307e-01	2,810e+00 ± 2,026e-01	2,562e+00 ± 1,492e-01
20	2,024e+01 ± 1,387e+00	1,866e+01 ± 9,318e-01	1,790e+01 ± 4,653e-01
21	2,000e+02 ± 2,829e-10	7,152e+02 ± 4,380e+02	7,745e+02 ± 4,476e+02
22	1,520e+01 ± 1,526e+00	7,102e+01 ± 8,403e+01	1,345e+01 ± 1,254e+00
23	5,859e+03 ± 1,084e+03	6,442e+03 ± 5,783e+02	5,569e+03 ± 5,078e+02
24	2,194e+02 ± 4,173e+00	2,612e+02 ± 1,354e+01	2,100e+02 ± 5,475e+00
25	2,963e+02 ± 1,563e+01	3,075e+02 ± 8,932e+00	2,787e+02 ± 7,642e+00
26	2,001e+02 ± 4,224e-02	2,564e+02 ± 8,156e+01	2,608e+02 ± 5,413e+01
27	7,387e+02 ± 2,025e+02	9,680e+02 ± 9,519e+01	4,067e+02 ± 5,810e+01
28	4,000e+02 ± 6,210e-11	6,024e+02 ± 7,701e+02	4,000e+02 ± 1,180e-13

ANEXO A – FUNÇÕES DO *BENCHMARK CEC2013*.

Neste anexo serão apresentadas as 28 funções do *benchmark CEC2013* definidas em (Liang *et al.*, 2013). São elas:

1. *Sphere Function*:

$$f_1(x) = \sum_{i=1}^D z_i^2 + f_1^*, \quad z = x - 0 \quad (\text{A.1})$$

2. *Rotated High Conditioned Elliptic Function*:

$$f_2(x) = \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} z_i^2 + f_2^*, \quad z = T_{osz}(M_1(x - o)) \quad (\text{A.2})$$

3. *Rotated Bent Cigar Function*:

$$f_3(x) = z_1^2 + 10^6 \sum_{i=2}^D z_i^2 + f_3^*, \quad z = M_2 T_{asy}^{0,5}(M_1(x - o)) \quad (\text{A.3})$$

4. *Rotated Discus Function*:

$$f_4(x) = 10^6 z_1^2 + \sum_{i=2}^D z_i^2 + f_4^*, \quad z = T_{osz}(M_1(x - o)) \quad (\text{A.4})$$

5. *Different Powers Function*:

$$f_5(x) = \sqrt{\sum_{i=1}^D |z_i|^{2+4\frac{i-1}{D-1}}} + f_5^*, \quad z = x - o \quad (\text{A.5})$$

6. *Rotated Rosenbrock's Function*:

$$f_6(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_6^*, \quad z = M_1\left(\frac{2,048(x - o)}{100}\right) + 1 \quad (\text{A.6})$$

7. *Rotated Schaffers F7 Function*:

$$f_7(x) = \left(\frac{1}{D-1} \sum_{i=1}^{D-1} (\sqrt{z_i} + \sqrt{z_i} \sin^2(50z_i^{0,2}))\right)^2 + f_7^*, \quad (\text{A.7})$$

$$z_i = \sqrt{y_i^2 + y_{i+1}^2}, \quad \text{for } i = 1, \dots, D, \quad y = \Lambda^{10} M_2 T_{asy}^{0,5}(M_1(x - o))$$

8. *Rotated Ackley's Function:*

$$f_8(x) = -20 \exp(-0,2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}) - \exp(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)) + 20 + e + f_8^*, \quad (\text{A.8})$$

$$z = \Lambda^{10} M_2 T_{asy}^{0,5}(M_1(x - o))$$

9. *Rotated Weierstrass Function:*

$$f_9(x) = \sum_{i=1}^D (\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (z_i + 0,5))]) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k \cdot 0,5)] + f_9^*, \quad (\text{A.9})$$

$$a = 0,5, b = 3, kmax = 20, z = \Lambda^{10} M_2 T_{asy}^{0,5}(M_1 \frac{0,5(x - o)}{100})$$

10. *Rotated Griewank's Function:*

$$f_{10}(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos(\frac{z_i}{\sqrt{i}}) + 1 + f_{10}^*, z = \Lambda^{100} M_1 \frac{600(x - o)}{100} \quad (\text{A.10})$$

11. *Rastrigin's Function:*

$$f_{11}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{11}^*, z = \Lambda^{10} T_{asy}^{0,2}(T_{osz}(\frac{5,12(x - o)}{100})) \quad (\text{A.11})$$

12. *Rotated Rastrigin's Function:*

$$f_{12}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{12}^*, \quad (\text{A.12})$$

$$z = M_1 \Lambda^{10} M_2 T_{asy}^{0,2}(T_{osz}(M_1 \frac{5,12(x - o)}{100}))$$

13. *Non-Continuous Rotated Rastrigin's Function:*

$$f_{13}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) + f_{13}^*, \quad (\text{A.13})$$

$$\hat{x} = M_1 \frac{5,12(x - o)}{100}, y_i = \begin{cases} \hat{x}_i & \text{if } |\hat{x}_i| \leq 0,5 \\ \text{round}(2\hat{x}_i)/2 & \text{if } |\hat{x}_i| > 0,5 \end{cases} \text{ for } i = 1, 2, \dots, D$$

$$z = M_1 \Lambda^{10} M_2 T_{asy}^{0,2}(T_{osz}(y))$$

14. Schwefel's Function:

$$f_{14}(z) = 418,9829 \times D - \sum_{i=1}^D g(z_i) + f_{14}^*, \quad (\text{A.14})$$

$$z = \Lambda^{10} \left(\frac{1000(x-o)}{100} \right) + 4,209687462275036e + 002,$$

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{1/2}) & \text{if } |z_i| \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{|500 - \text{mod}(z_i, 500)|}) - \frac{(z_i-500)^2}{10000D} & \text{if } z_i > 500 \\ (\text{mod}(|z_i|, 500) - 500) \sin(\sqrt{|\text{mod}(|z_i|, 500) - 500|}) - \frac{(z_i+500)^2}{10000D} & \text{if } z_i < -500 \end{cases}$$

15. Rotated Schwefel's Function:

$$f_{15}(z) = 418,9829 \times D - \sum_{i=1}^D g(z_i) + f_{15}^*, \quad (\text{A.15})$$

$$z = \Lambda^{10} M_1 \left(\frac{1000(x-o)}{100} \right) + 4,209687462275036e + 002,$$

$$g(z_i) = \begin{cases} z_i \sin(|z_i|^{1/2}) & \text{if } |z_i| \leq 500 \\ (500 - \text{mod}(z_i, 500)) \sin(\sqrt{|500 - \text{mod}(z_i, 500)|}) + \frac{(z_i-500)^2}{10000D} & \text{if } z_i > 500 \\ (\text{mod}(|z_i|, 500) - 500) \sin(\sqrt{|\text{mod}(|z_i|, 500) - 500|}) + \frac{(z_i+500)^2}{10000D} & \text{if } z_i < -500 \end{cases}$$

16. Rotated Katsuura Function:

$$f_{16}(x) = \frac{10}{D^2} \prod_{i=1}^D \left(1 + i \sum_{j=1}^{32} \frac{|2^j z_i - \text{round}(2^j z_i)|}{2^j} \right)^{\frac{10}{D^{1/2}}} - \frac{10}{D^2} + f_{16}^*, \quad (\text{A.16})$$

$$z = M_2 \Lambda^{100} \left(M_1 \frac{5(x-o)}{100} \right)$$

17. Lunacek Bi_Rastrigin Function:

$$f_{17}(x) = \min \left(\sum_{i=1}^D (\bar{x}_i - \mu_0)^2, dD + s \sum_{i=1}^D (\bar{x}_i - \mu_1)^2 \right) + 10 \left(D - \sum_{i=1}^D \cos(2\pi \bar{z}_i) \right) + f_{17}^*, \quad (\text{A.17})$$

$$\mu_0 = 2,5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D} + 20 - 8,2}, d = 1,$$

$$y = \frac{10(x-o)}{100}, \bar{x}_i = 2 \text{sign}(x_i^*) y_i + \mu_0, \text{ for } i = 1, 2, \dots, D,$$

$$z = \Lambda^{100} (\bar{x} - \mu_0)$$

18. *Rotated Lunacek Bi_Rastrigin Function:*

$$f_{18}(x) = \min\left(\sum_{i=1}^D (\bar{x}_i - \mu_0)^2, dD + s \sum_{i=1}^D (\bar{x}_i - \mu_1)^2\right) + 10\left(D - \sum_{i=1}^D \cos(2\pi\bar{z}_i)\right) + f_{18}^*, \quad (\text{A.18})$$

$$\mu_0 = 2,5, \mu_1 = -\sqrt{\frac{\mu_0^2 - d}{s}}, s = 1 - \frac{1}{2\sqrt{D+20}-8,2}, d = 1,$$

$$y = \frac{10(x-o)}{100}, \bar{x}_i = 2\text{sign}(y_i^*)y_i + \mu_0, \text{ for } i = 1, 2, \dots, D, z = M_2\Lambda^{100}(M_1(\bar{x} - \mu_0))$$

19. *Expanded Griewank's plus Rosenbrock's Function:*

$$f_{19}(x) = g_1(g_2(z_1, z_2)) + g_1(g_2(z_2, z_3)) + \dots + g_1(g_2(z_{D-1}, z_D)) + g_1(g_2(z_D, z_1)) + f_{19}^*, \quad (\text{A.19})$$

$$\text{Basic Griewank's Function: } g_1(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$\text{Basic Rosenbrock's Function: } g_2(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2),$$

$$z = M_1\left(\frac{5(x-o)}{100}\right) + 1$$

20. *Expanded Scaffer's F6 Function:*

$$f_{20}(x) = g(z_1, z_2) + g(z_2, z_3) + \dots + g(z_{D-1}, z_D) + g(z_D, z_1) + f_{20}^*, \quad (\text{A.20})$$

$$\text{Scaffer's F6 Function: } g(x, y) = 0,5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0,5)}{(1 + 0,001(x^2 + y^2))^2},$$

$$z = M_2 T_{asy}^{0,5}(M_1(x-o))$$

21. *Composition Function 1 (n=5, Rotated):*

$$f(x) = \sum_{i=1}^n \{w_i^*[\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.21})$$

$$n = 5, \quad \sigma = [10, 20, 30, 40, 50],$$

$$\lambda = [1, 1e-6, 1e-26, 1e-6, 0.1],$$

$$bias = [0, 100, 200, 300, 400],$$

$$g_1 : \text{Rotated Rosenbrock's Function } f_6,$$

g_2 : Rotated Different Powers Function f'_5 ,

g_3 : Rotated Bent Cigar Function f'_3 ,

g_4 : Rotated Discus Function f'_4 ,

g_5 : Sphere Function f'_1 ,

22. Composition Function 2 ($n=3$, Unrotated):

$$f(x) = \sum_{i=1}^n \{w_i^* [\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.22})$$

$n = 3$, $\sigma = [20, 20, 20]$,

$\lambda = [1, 1, 1]$,

$bias = [0, 100, 200]$,

g_{1-3} : Schwefel's Function f'_{14} ,

23. Composition Function 3 ($n=3$, Rotated):

$$f(x) = \sum_{i=1}^n \{w_i^* [\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.23})$$

$n = 3$, $\sigma = [20, 20, 20]$,

$\lambda = [1, 1, 1]$,

$bias = [0, 100, 200]$,

g_{1-3} : Rotated Schwefel's Function f'_{15} ,

24. Composition Function 4 ($n=3$, Rotated):

$$f(x) = \sum_{i=1}^n \{w_i^* [\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.24})$$

$n = 3$, $\sigma = [20, 20, 20]$,

$\lambda = [0.25, 1, 2.5]$,

$bias = [0, 100, 200]$,

g_1 : Rotated Schwefel's Function f'_{15} ,

g_2 : Rotated Rastrigin's Function f'_{12} ,

g_3 : Rotated Weierstrass Function f'_9 ,

25. Composition Function 5 ($n=3$, Rotated):

$$f(x) = \sum_{i=1}^n \{w_i^* [\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.25})$$

- $n = 3, \quad \sigma = [10, 30, 50],$
 $\lambda = [0.25, 1, 2.5],$
 $bias = [0, 100, 200],$
 g_1 : Rotated Schwefel's Function $f'_{15},$
 g_2 : Rotated Rastrigin's Function $f'_{12},$
 g_3 : Rotated Weierstrass Function $f'_9,$

26. Composition Function 6 ($n=5$, Rotated):

$$f(x) = \sum_{i=1}^n \{w_i^* [\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.26})$$

- $n = 5, \quad \sigma = [10, 10, 10, 10, 10],$
 $\lambda = [0.25, 1, 1e-7, 2.5, 10],$
 $bias = [0, 100, 200, 300, 400],$
 g_1 : Rotated Schwefel's Function $f'_{15},$
 g_2 : Rotated Rastrigin's Function $f'_{12},$
 g_3 : Rotated High Conditioned Elliptic Function $f'_2,$
 g_4 : Rotated Weierstrass Function $f'_9,$
 g_5 : Rotated Griewank's Function $f'_{10},$

27. Composition Function 7 ($n=5$, Rotated):

$$f(x) = \sum_{i=1}^n \{w_i^* [\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.27})$$

- $n = 5, \quad \sigma = [10, 10, 10, 20, 20],$
 $\lambda = [100, 10, 2.5, 25, 0.1],$
 $bias = [0, 100, 200, 300, 400],$
 g_1 : Rotated Griewank's Function $f'_{10},$
 g_2 : Rotated Rastrigin's Function $f'_{12},$
 g_3 : Rotated Schwefel's Function $f'_{15},$
 g_4 : Rotated Weierstrass Function $f'_9,$
 g_5 : Sphere Function $f'_1,$

28. Composition Function 8 ($n=5$, Rotated):

$$f(x) = \sum_{i=1}^n \{w_i^* [\lambda_i g_i(x) + bias_i]\} + f^*, \quad (\text{A.28})$$

$$n = 5, \quad \sigma = [10, 20, 30, 40, 50],$$

$$\lambda = [2.5, 2.5e - 3, 2.5, 5e - 4, 0.1],$$

$$\text{bias} = [0, 100, 200, 300, 400],$$

g_1 : Rotated Expanded Griewank's plus Rosenbrock's Function f'_{19} ,

g_2 : Rotated Schaffer's F7 Function f'_7 ,

g_3 : Rotated Schwefel's Function f'_{15} ,

g_4 : Rotated Expanded Scaffer's F6 Function f'_{20} ,

g_5 : Sphere Function f'_1 ,