Programa de Pós-Graduação em Computação Aplicada
Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto
"Campus de Ribeirão Preto"

DANIEL AUGUSTO DOS SANTOS

UM MODELO PARA EXPLICAÇÃO DE DECISÕES LOCAIS DE CLASSIFICADORES BASEADO EM ALGORITMOS GENÉTICOS COM PRESERVAÇÃO DA DIVERSIDADE DE POPULAÇÕES

RIBEIRÃO PRETO – SP
2022

DANIEL AUGUSTO DOS SANTOS

UM MODELO PARA EXPLICAÇÃO DE DECISÕES LOCAIS DE CLASSIFICADORES BASEADO EM ALGORITMOS GENÉTICOS COM PRESERVAÇÃO DA DIVERSIDADE DE POPULAÇÕES

Dissertação apresentada à banca examinadora do Programa de Pós-Graduação em Computação Aplicada, como parte dos requisitos para a obtenção do título de Mestre em Computação Aplicada pela Faculdade de Filosofia, Ciência e Letras de Ribeirão Preto–USP.

Orientador: Prof. Dr. Renato Tinós

RIBEIRÃO PRETO – SP
2022

Santos, Daniel Augusto dos
Aspectos eletroneuromiográficos da hanseníase. Ribeirão Preto, 2022.
87 p. : il.

Dissertação de Mestrado, apresentada à Faculdade de Filosofia, Ciência e Letras de Ribeirão Preto /USP. Área de concentração: Computação Aplicada.
Orientador: Tinós, Renato.

1. Inteligência Artificial Explicável. 2. Algoritmos Genéticos. 3. *fitness sharing*. I. Título

RESUMO

SANTOS, D. A. **Um modelo para explicação de decisões locais de classificadores baseado em algoritmos genéticos com preservação da diversidade de populações**. 2022. Dissertação (Mestrado em Computação Aplicada) - Faculdade de Filosofia, Ciência e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2022.

O uso de aprendizado de máquina em domínios diversos é cada vez mais comum. No entanto, muitas aplicações críticas não podem usufruir desta tecnologia sem que as decisões de um classificador sejam interpretáveis. O problema é que a maioria dos modelos se comporta como uma caixa-preta cujas decisões não são facilmente interpretáveis, o que limita sua efetividade apesar do bom desempenho. Diversos trabalhos na literatura buscam resolver este problema propondo técnicas de decisão que explicam o comportamento de modelos caixa-preta quando estes são aplicados a um determinado exemplo. Uma delas é a técnica *Local Rule Based Explanations* (LORE) que gera explicações locais, utilizando uma Árvore de Decisão treinada a partir de dados artificiais gerados por um algoritmo genético (AG). O método LORE utiliza um algoritmo genético padrão, que não preserva necessariamente a diversidade das soluções na população final. A hipótese investigada neste trabalho é que a diversidade é importante para gerar árvores de decisão que consigam reproduzir com maior precisão as fronteiras de decisão do classificador localizadas perto do exemplo a ser explicado. Este trabalho mostra que os exemplos artificiais gerados pelos AGs em LORE não são necessariamente diversos. É proposto então o uso da técnica de *fitness sharing* no AG para gerar exemplos artificiais mais diversos. Consequentemente, as fronteiras de decisão locais da Árvore de Decisão devem ser mais semelhantes aos do classificador caixa-preta. Resultados experimentais com dois classificadores (Perceptron Multicamadas e Florestas Aleatórias) e quatro problemas de classificação indicam que LORE com *fitness sharing* produz populações de AG mais diversas e melhores explicações locais.

Palavras-chave: Caixa-preta. Explicação. Inteligência Artificial Explicável. Algoritmos Genéticos. *fitness sharing*.

ABSTRACT

SANTOS, D. A. **A model for explaining local classifier decisions based on genetic algorithms with preservation of population diversity**. 2022. Dissertação (Mestrado em Computação Aplicada) - Faculdade de Filosofia, Ciência e Letras de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto, 2022.

The use of machine learning in diverse domains is increasingly common. However, many critical applications cannot take advantage of this technology without the decisions of a classifier being interpretable. The problem is that most models behave like a black box whose decisions are not easily interpretable, which limits its effectiveness despite good performance. Several works in the literature seek to solve this problem by proposing decision techniques that explain the behavior of black box models when they are applied to a given example. One of them is the Local Rule Based Explanations (LORE), a technique that generates local explanations using a Decision Tree trained on artificial data generated by a genetic algorithm. The LORE method uses a standard genetic algorithm, which does not necessarily preserve the diversity of solutions in the final population. The hypothesis investigated in this work is that diversity is important to generate decision trees that can more accurately reproduce the classifier's decision boundaries located close to the example to be explained. This work shows that the artificial examples generated by the GA in LORE are not necessarily diverse. The Fitness Sharing technique is then used to generate more diverse artificial GA examples. Consequently, the local decision boundaries of the Decision Tree should be more similar to those of the black-box classifier. Experimental results with two classifiers (Multilayer Perceptron and Random Forests) and four classification problems indicate that LORE with fitness sharing produces more diverse GA populations and better local explanations.

Keywords: Black Box. Explanation. Explainable Artificial Intelligence. Genetic Algorithms. Fitness Sharing.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

Apesar de haver estudos desde 1958 (ROSENBLATT, 1958), a adoção ampla de sistemas baseados em aprendizado de máquina para a tomada de decisões é recente. Em uma pesquisa feita em 2015 pela empresa de consultoria Gartner, apenas 10% das organizações entrevistadas alegaram usar algum tipo de Inteligência Artificial (IA) em suas operações, enquanto que em uma nova versão da pesquisa em 2019 a adoção subiu para 37% (GARTNER, 2019).

Tal aumento pode ser atribuído aos ótimos resultados obtidos em aplicações de algoritmos de aprendizado de máquina em diversas áreas. O uso disseminado da IA tornou-se mais viável com o avanço da tecnologia. O grande volume de informação disponível e a maior facilidade de coleta de bases de dados aliados ao crescimento do poder computacional teve como consequência a criação de modelos computacionais que utilizam algoritmos de maior complexidade e com melhor desempenho nas tarefas de classificação e regressão (ABADI, BARHAM, *et al.*, 2016)(IBM, 2020).

Embora estes modelos de IA apresentem bons resultados em muitas aplicações, o uso de aprendizado de máquina em muitas áreas tem algumas restrições. Vários modelos conseguem uma boa acurácia, mas muitas vezes apenas ótima o suficiente para serem usados em aplicações não críticas como recomendações de filmes em serviços de *streaming* ou de produtos por empresas de varejo online. Nestas aplicações a predição do modelo não oferece riscos ou impactos significativos caso a recomendação não seja útil ao cliente. Para muitas empresas, já que não há riscos, é melhor tomar uma ação ruim do que não tomar nenhuma ação.

O mesmo não pode ser dito quando a aplicação é um problema crítico onde a decisão de um software pode impactar significativamente um indivíduo. Na medicina, por exemplo, uma predição errada como um falso negativo pode ter como consequência a morte ao deixar de enviar um paciente para tratamento.

Ainda que a decisão final seja de um humano e a IA atue apenas como um suporte à decisão, como confiar que sua predição é correta? Muitos dos modelos eficientes de aprendizado de máquina, como Redes Neurais Convolucionais e Florestas Aleatórias (FA), não são interpretáveis. Ou seja, suas decisões não são facilmente interpretadas por seres humanos. Sem uma forma de compreender uma predição, mesmo que o modelo apresente uma boa acurácia, a

falta de confiança do usuário o leva a não utilizar o suporte (RIBEIRO, SINGH e GUESTRIN, 2016).

Além disso, com a popularização da IA novas leis surgem para controlar o seu uso, como fez a União Europeia ao criar no Regulamento Geral sobre a Proteção de Dados (GDPR, do inglês *General Data Protection Regulation);* o artigo 22, que define o direito de explicação, garante que qualquer pessoa afetada pela decisão de um algoritmo tenha o direito de saber o porquê aquela decisão foi tomada (COUNCIL OF EUROPEAN UNION, 2016).

Diante desses cenários, entender como o modelo de aprendizado tomou uma decisão para uma determinada entrada é essencial. O usuário final deste modelo está interessado no motivo que levou a uma decisão e se ele é justo ou possui algum tipo de viés. Não basta apenas transparência sobre o modelo – funcionamento do algoritmo, a base de dados usada para treino e os parâmetros de treinamento - é preciso uma explicação de como o sistema chegou àquela conclusão. A explicação atua como um meio de comunicação entre um sistema computacional e a pessoa que o utilizará e, portanto, precisa ser interpretável por humanos.

Diversos pesquisadores se dedicam ao desenvolvimento de métodos e técnicas para criar explicações para modelos já existentes, cujas decisões são de difícil interpretação. É possível encontrar na literatura uma variedade de métodos com diferentes características, vantagens e desvantagens. Um desses métodos é o *LOcal Rule-based Explanations* (LORE) (GUIDOTTI, MONREALE, *et al.*, 2019), que será o objeto de estudo deste trabalho.

LORE busca explicar a decisão de um modelo de IA para uma entrada em particular e não o modelo como um todo. Por exemplo, dado um sistema que decide se uma pessoa é elegível ou não a um empréstimo, o LORE busca explicar por que o sistema decidiu que uma determinada pessoa é elegível ou não.

Para gerar a explicação, da decisão do modelo caixa-preta para um determinado exemplo LORE gera uma base de dados artificiais com o objetivo de treinar uma réplica local e interpretável do modelo. A geração da base de dados artificiais é feita por meio de um Algoritmo Genético que gera exemplos artificiais semelhantes ao exemplo a ser explicado. Os dados artificiais gerados pelo AG são alimentados para treinamento de uma Árvore de Decisão para reconstrução da réplica. Uma das principais propriedades de Árvores de Decisão é que, quando comparada com modelos do tipo caixa-preta como as redes neurais artificiais, suas decisões são mais fáceis de serem interpretadas por seres humanos.

O algoritmo utilizado no método LORE é um AG padrão. É conhecido que o AG padrão não preserva necessariamente a diversidade das soluções na população final, ou seja, os

indivíduos da população final ficam agrupados em uma região só, ou em poucas, de forma que eles são quase idênticos uns aos outros. A hipótese investigada neste trabalho é que a diversidade é importante para reproduzir com maior precisão a decisão do classificador para exemplos perto do exemplo a ser explicado.

A motivação do trabalho é aprimorar o método LORE, primeiramente investigando o impacto do uso do algoritmo genético (padrão) no funcionamento do método. Então, baseado nas deficiências detectadas, propõe-se o uso de algoritmos genéticos eficientes para o problema tratado. Propõe-se o uso de algoritmos genéticos que utilizam o conceito de nicho (*niching*) para preservar a diversidade das soluções da população. Tais mudanças são feitas com o objetivo de obter explicações mais robustas e precisas para as decisões do classificador para um determinado exemplo.

Os capítulos deste trabalho estão organizados como segue. No capítulo 2 são introduzidas as fundamentações teóricas dos principais temas usados no trabalho: Interpretabilidade, Algoritmos Genéticos e Árvores de Decisão. No capítulo 3 é descrito o funcionamento do LORE e a metodologia empregada nos experimentos para comparar o LORE e o novo método proposto neste trabalho, LOREfs. No capítulo 4 são expostos os experimentos e resultados da comparação entre os métodos qualitativamente e quantitativamente. Por fim, no capítulo 5 são apresentadas as conclusões do trabalho e sugestões de trabalhos futuros.

1.1 OBJETIVOS

O objetivo principal deste trabalho é investigar o algoritmo genético utilizado no método LORE e aplicar melhorias na geração do conjunto de dados artificiais na vizinhança do exemplo a ser explicado. Mais precisamente aplicar a técnica de nicho *fitness sharing* sob a hipótese de que a construção de uma vizinhança mais diversa leva a melhores explicações locais por cobrir uma maior parte da superfície de decisão e, consequentemente, a construção de um modelo substituto mais fiel ao modelo original.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 INTERPRETABILIDADE E EXPLICAÇÃO DE MODELOS DE INTELIGÊNCIA ARTIFICIAL

Como visto na Seção 1, há uma grande necessidade de modelos de IA serem interpretáveis. Neste contexto, em (DOSHI-VELEZ e KIM, 2017) interpretabilidade é definida como a habilidade de explicar ou prover significado em termos entendíveis por humanos. Isto não é uma tarefa simples, uma vez que muitos modelos se comportam como caixas-pretas cujos processos de decisão são bastante complexos para serem compreendidos por humanos (MITTELSTADT , RUSSELL e WACHTER, 2018).

Um modelo caixa-preta é um modelo cujo funcionamento não é interpretável, seja porque seu mecanismo é obscuro (como um software proprietário) ou porque suas decisões são difíceis de serem explicadas em termos utilizados por especialistas humanos (como, por exemplo, em um modelo de Aprendizado Profundo). Praticamente tudo o que se sabe sobre a caixa-preta é que para um determinado conjunto de dados de entrada ela retornará um conjunto de dados de saída (Figura 1).

Figura 1 – Representação de um Modelo Caixa-Preta.

Fonte: Elaborado pelo autor.

Um modelo de IA interpretável seria o oposto: seu funcionamento é conhecido e interpretável por humanos, o que permite explicar as decisões quando dados de entrada são transformados para obter os dados de saída. Em aplicações críticas, sacrificar a interpretabilidade para alcançar uma maior precisão não é uma possibilidade, pois é essencial

que o modelo preditor seja preciso e interpretável. O que acaba por ser conflitante pois muitas vezes os melhores modelos em performance são os menos interpretáveis, e os modelos mais interpretáveis possuem menor precisão (DARPA, 2016).

Algumas técnicas eficientes de Aprendizado de Máquina, como as Redes Neurais Convolucionais, não possuem um mecanismo lógico interno para explicar como um determinado resultado é obtido. No entanto, a interpretabilidade é uma questão essencial para muitas áreas, por exemplo, Medicina e Saúde (LAKHANI, PRATER, *et al.*, 2018; EL SHAWI, SHERIF, *et al.*, 2020). Apesar de apresentar resultados notáveis em muitas aplicações médicas, algoritmos inteligentes não são facilmente aceitos na Medicina porque carecem de interpretabilidade. Por exemplo, na análise de imagens médicas, as Redes Neurais Convolucionais superam os radiologistas em muitos casos de diagnóstico (RAJPURKAR, IRVIN, *et al.*, 2017). No entanto, pacientes e profissionais médicos raramente aceitam seus resultados (sem intervenção humana), porque as Redes Neurais Convolucionais não explicam como o resultado foi alcançado da mesma forma que os radiologistas fazem facilmente.

A falta de interpretabilidade ainda impacta profundamente as possibilidades de integração da inteligência humana e artificial. Hoje, o conhecimento de humanos em muitas áreas raramente é usado quando os modelos de Aprendizado de Máquina tomam decisões. Por outro lado, o conhecimento obtido por algoritmos de Aprendizado de Máquina a partir de conjuntos de dados grandes e complexos raramente são usados para melhorar o conhecimento em áreas específicas.

Em (GUIDOTTI, MONREALE , *et al.*, 2018) os autores classificam problema de interpretabilidade em duas vertentes: como criar um modelo já interpretável que apresente boa performance (que foge do escopo deste trabalho); ou como explicar os modelos caixa- preta já existentes. Este último pode ser resolvido por métodos de explicação que auditam a caixa-preta e procuram por uma explicação para o comportamento dela.

Existem diversos métodos de explicação na literatura (GUIDOTTI, MONREALE , *et al.*, 2018), divididos em agnósticos e específicos. Os métodos específicos são usados para explicar uma classe de algoritmos em particular (como redes neurais, redes neurais profundas, máquinas de vetores-suporte, e comitês de classificadores). Já os métodos agnósticos não possuem esta restrição e podem ser usados para explicar decisões geradas por qualquer modelo de aprendizado de máquina, seja caixa-preta ou não.

Um modelo de explicação ainda pode ser global ou local, independentemente de ser agnóstico ou específico. Modelos globais visam explicar todo o funcionamento de uma caixa-

preta por meio de um modelo interpretável que imita seu comportamento para todo um conjunto de dados (GUIDOTTI, MONREALE , *et al.*, 2018). Em contrapartida, modelos locais visam explicar a predição de uma caixa-preta para um caso de entrada (exemplo) específica (GUIDOTTI, MONREALE , *et al.*, 2018). Por exemplo, em um sistema de crédito o interesse é explicar a predição do modelo para os dados de um cliente em específico.

O campo de estudo sobre explicações de modelos de IA ainda é recente. Não existe um consenso do que é uma explicação e o que é um modelo interpretável, e nem um modo preciso de mensurar e comparar dois modelos quanto a essa questão. Com isso, diversos trabalhos na literatura não necessariamente se embasam nos conhecimentos dos campos da Psicologia e das Ciências Sociais e seus autores se apoiam na intuição do que é uma boa explicação (MILLER, 2018). A interpretabilidade é subjetiva, sendo que a explicação dada por um modelo pode ser interpretável para um especialista, mas não para uma pessoa leiga que está sendo afetada pela decisão do algoritmo.

Contudo, o campo de estudo tem ganho cada vez mais importância e o número de publicações tem aumentado significativamente. Até mesmo a Agência de Projetos de Pesquisa Avançada de Defesa (DARPA - *Defense Advanced Research Projects Agency*) dos Estados Unidos publicou uma chamada em busca de pesquisadores em explicações de modelos de IA (DARPA, 2016). Na chamada, a agência defende que modelos de IA explicáveis serão essenciais para que usuários entendam, confiem, e administrem efetivamente a próxima geração de sistemas computacionais inteligentes.

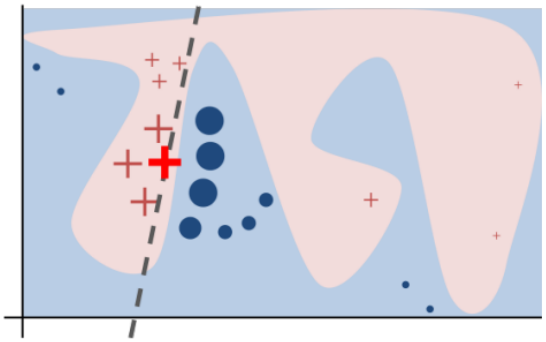
A partir dessa mesma motivação, Ribeiro, Singh e Guestrin (2016) propuseram em seu artigo " '*Why Should I Trust You?' Explaining the Predictions of Any Classifier*" (" 'Por quê eu deveria acreditar em você?' Explicando as Predições de Qualquer Classificador") o LIME *(Local Interpretable Model-agnostic Explanations),* um dos modelos de explicação mais populares atualmente. O LIME é um modelo de explicação agnóstico e local que assume que todo modelo complexo (modelo caixa-preta, por exemplo) é localmente linear. Assim, LIME busca ajustar um modelo linear simples na vizinhança de um único exemplo a ser explicado de forma que esse modelo linear imite localmente o comportamento global do modelo complexo, e, portanto, possa ser usado para explicar localmente as previsões do modelo mais complexo.

O ajuste do modelo linear é feito em cima de um conjunto de dados artificiais gerados por meio da perturbação do caso a ser explicado. Cada ponto gerado pela perturbação é classificado pelo modelo caixa-preta e é calculada sua distância do exemplo a ser explicado (distância Euclidiana). É feita então uma seleção dos pontos mais relevantes para o ajuste do modelo linear de acordo com esta distância. Quanto menor a distância, maior é a probabilidade de um ponto

pertencer à vizinhança do exemplo a ser explicado, logo, maior é sua relevância para uma explicação local.

Na Figura 2 é apresentado um exemplo do LIME para explicar a decisão de uma caixa-preta complexa (regiões azul e rosa) para o exemplo cruz vermelha em destaque. As outras cruzes e os círculos representam os dados gerados artificialmente, onde seus tamanhos representam sua relevância local (quanto maior, mais relevante). A reta tracejada é o modelo linear criado pelo LIME para explicar a decisão localmente.

Figura 2 – Modelo linear gerado pelo LIME para a decisão de um modelo complexo para o exemplo cruz vermelha em destaque.



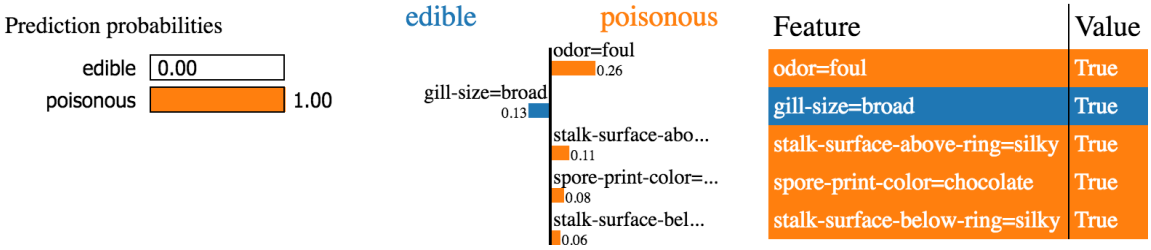
Fonte: (RIBEIRO, 2016).

Como visto na Figura 2, não é possível explicar o espaço global por meio de um modelo linear, mas localmente o modelo linear produzido pelo LIME consegue reproduzir localmente com fidelidade o modelo complexo. No entanto, o modelo linear por si só não é interpretável, principalmente se o alvo da explicação for uma pessoa leiga. Assim, o LIME deriva a explicação a partir dos pesos de cada característica do modelo linear de forma a ser facilmente interpretada por humanos, como mostrado na Figura 3. O peso de cada característica é uma medida de quanto aquela característica influenciou na decisão do modelo.

Contudo, o LIME apresenta algumas desvantagens apontadas por (MOLNAR, 2019). A primeira delas é a falta de definição do que é uma vizinhança para conjuntos de dados tabulares, sendo preciso alterar e testar empiricamente um hiperparâmetro para cada aplicação. A segunda desvantagem é que o método de perturbação ignora a correlação entre a características dos dados o que leva a geração de dados que não correspondem à distribuição original. Por fim, a última desvantagem é a instabilidade nas explicações: diferentes execuções do algoritmo para um mesmo caso podem gerar explicações diferentes.

Apesar destas desvantagens, o LIME é promissor e serve como inspiração para o desenvolvimento de novos métodos e estudos, contribuindo para o crescimento do conhecimento do campo.

Figura 3 – Exemplo de explicação gerada pelo LIME para um classificador de cogumelos (comestível x venenoso). À esquerda, a probabilidade de cada classificação para um tipo de cogumelo. No meio, a contribuição de cada característica (dada seu valor) para a decisão do modelo. À direita, os valores das características do caso a ser explicado.



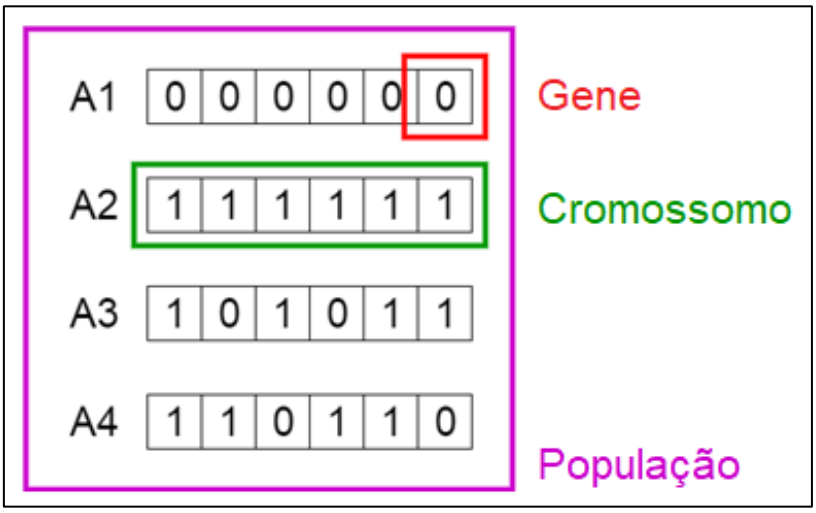
Fonte: (RIBEIRO, 2016).

2.2 ALGORITMOS GENÉTICOS

Algoritmos Genéticos (AGs) são algoritmos de busca baseados na genética e na seleção natural (GOLDBERG, 1989). A ideia é evoluir uma população onde cada indivíduo representa uma possível solução de um determinado problema. Os indivíduos mais aptos têm maiores chances de serem selecionados para reprodução para gerar a próxima população de indivíduos. Na reprodução, as soluções sofrem mutação e recombinação e o processo se repete até que uma condição de parada seja satisfeita (normalmente um número máximo de gerações). Após algumas gerações, a população evolui de forma que seus indivíduos representem soluções mais aptas ao problema.

Em AGs, cada indivíduo da população é representado por um cromossomo, que, por sua vez, é dividido em genes, que são características particulares de um indivíduo (Figura 4). O processo de evolução de uma população ocorre por meio de operadores de seleção, recombinação e de mutação. Na seleção, cromossomos da população são selecionados de acordo com sua aptidão, sendo quanto maior sua aptidão maior a chance de ser selecionado. A aptidão de cada indivíduo é dada por uma função de aptidão (*fitness*) que avalia o quão bom aquele cromossomo representa a solução do problema (MITCHELL, 1998).

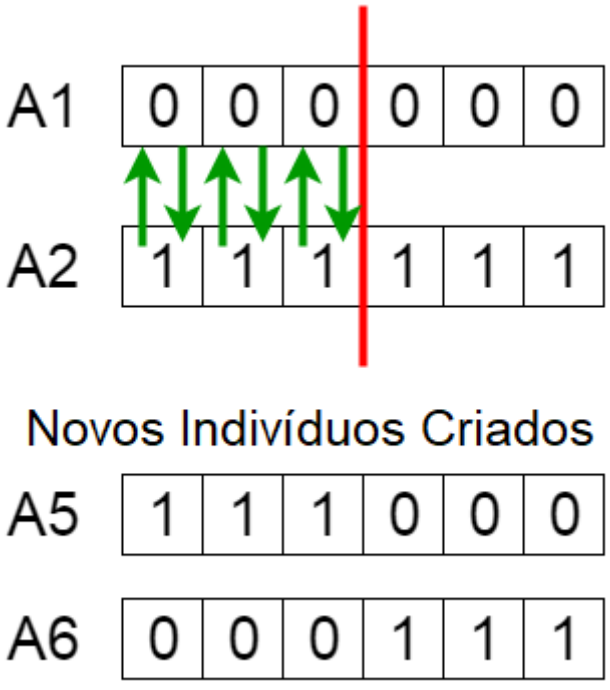
Figura 4 – Representação da População, Cromossomo (representados por A1, A2, A3 e A4) e Gene de um Algoritmo Genético.



Fonte: Adaptado de (MALLAWAARACHCHI, 2017).

Na recombinação, há uma troca de informação genética (genes) entre dois cromossomos pais para gerar novos cromossomos (Figura 5).

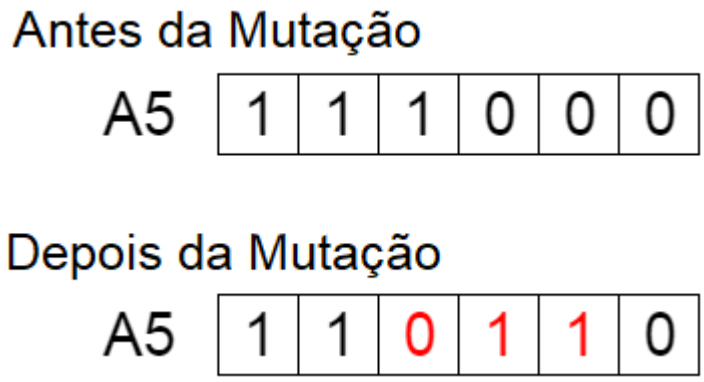
Figura 5 - Recombinação de um ponto entre Cromossomos (*crossover*). Os pais são A1 e A2 e os filhos são A5 e A6.



Fonte: Adaptado de (MALLAWAARACHCHI, 2017).

Na mutação, o novo indivíduo é gerado a partir de mutações aleatórias nos genes do cromossomo (Figura 6). Assim como na natureza, esse processo de evolução pode levar à criação de indivíduos mais aptos do que aqueles que os geraram a cada nova geração da população.

Figura 6 – Mutação binária de um Cromossomo.



Fonte: Adaptado de (MALLAWAARACHCHI, 2017).

Na Figura 7 é apresentado um pseudocódigo de um algoritmo genético simples (BACK, FOGEL e MICHALEWICZ, 2000). O algoritmo primeiro inicializa a população inicial P_0 de tamanho N (linha 1) e avalia a aptidão de cada membro de P_0 (linha 2). Em seguida, entra-se no loop responsável pela evolução da população, sendo a condição de parada definida pelo número de gerações G (linha 3). Dentro do loop ocorrem as operações de seleção (linha 4), de recombinação (linha 5) e de mutação (linha 6). A operação de seleção seleciona N indivíduos da população de acordo com sua aptidão obtida no processo de avaliação (linha 2 ou 7) e retorna a próxima geração P_{i+1}. As operações de recombinação e mutação dependem dos parâmetros p_c (probabilidade de ocorrer recombinação) e p_m (probabilidade de ocorrer mutação), respectivamente. O processo se repete até a condição de parada ser alcançada e o algoritmo retorna o melhor indivíduo encontrado ou toda a população.

Figura 7 - Algoritmo Genético Simples

Algoritmo 1 Algoritmo Genético Padrão

Entrada: N - tamanho da população, G - número de gerações,
p_c - probabilidade de recombinação, p_m - probabilidade de mutação
Saída: Z - população final

1: $P_0 \leftarrow inicializaPopulação(N)$;
2: avaliação(P_0);
3: **for** $i \leftarrow 0$ até $G - 1$ **do**
4: $P_{i+1} \leftarrow$ seleção(P_i);
5: $P'_{i+1} \leftarrow$ recombinação(P_{i+1}, p_c);
6: $P''_{i+1} \leftarrow$ mutação(P'_{i+1}, p_m);
7: avaliação(P''_{i+1});
8: $P_{i+1} = P''_{i+1}$;
9: **end for**
10: $Z \leftarrow P_{i+1}$
11: retorne Z

Fonte: (GUIDOTTI, MONREALE, *et al.*, 2019).

2.3 ÁRVORE DE DECISÃO

A Árvore de Decisão é um modelo de aprendizado de máquina supervisionado usado para regressões e classificações. A ideia do algoritmo é dividir o conjunto de dados em conjuntos menores com características semelhantes até alcançar um conjunto que contém objetos de uma única classe.

Árvores de Decisão são compostas por nós, galhos e folhas. Cada nó representa uma característica ou atributo de um objeto, cada galho uma regra de decisão e cada folha uma decisão (classe ou valor final da regressão).

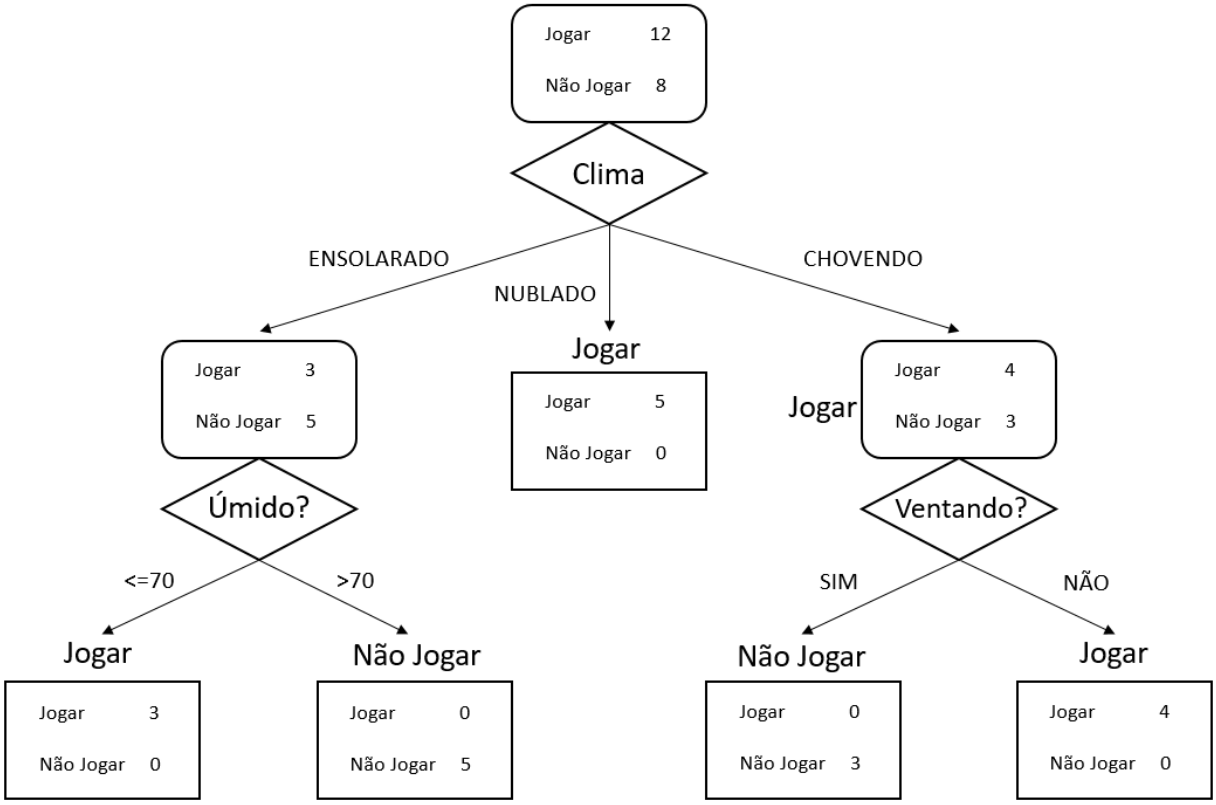
De acordo com (MITCHELL, 1997) a construção de uma Árvore de Decisão é realizada da seguinte forma. Usando o conjunto de treinamento, um atributo é escolhido para particionar as amostras em subconjuntos, de acordo com os valores desse atributo. Para cada subconjunto, outro atributo é escolhido para o particionamento. Este processo continua se um dos subconjuntos contiver uma mistura de amostras pertencentes a diferentes classes. Uma vez obtido um subconjunto uniforme no qual todas as amostras desse subconjunto pertencem à mesma classe, um nó folha é criado e rotulado com o respectivo nome de classe.

Um novo objeto é classificado pelo caminho percorrido na árvore, da raiz (primeiro nó) até a folha. Cada nó da árvore testa um atributo do objeto, e cada galho que sai de um nó corresponde a um possível valor para o atributo deste nó. Ao chegar a um nó folha a classe deste nó folha será atribuída ao novo objeto.

Na Figura 8, é apresentado um exemplo clássico de Árvore de Decisão que classifica se é um bom dia para se jogar tênis dadas as condições climáticas. Os retângulos com bordas arredondadas representam os conjuntos/subconjuntos de amostras com diferentes classes, os retângulos representam as folhas (subconjuntos de amostras puros, ou seja, somente uma classe), os losangos representam os nós de decisão e as flechas representam os galhos.

A Árvore de Decisão é um modelo inerentemente interpretável por seres humanos. Por exemplo, cada caminho da raiz até uma folha da Árvore de Decisão pode ser facilmente convertido em uma regra se-então, também um modelo interpretável.

Figura 8 – Esquematização de uma Árvore de Decisão



Fonte: Adaptado de (MITCHELL, 1997).

3 METODOLOGIA

3.1 LOCAL RULE-BASED EXPLANATIONS – LORE

O LORE é um modelo agnóstico de explicação local onde é usado um modelo substituto (Árvore de Decisão) para explicar a decisão de um modelo caixa-preta para um exemplo em específico. A Árvore de Decisão é treinada a partir de um conjunto de dados de entrada artificial gerado por um Algoritmo Genético com o intuito de reproduzir o comportamento do modelo caixa-preta ao redor do exemplo a ser explicado e, assim, extrair regras que explicam a decisão do modelo caixa-preta e contrafatuais que permitem inverter esta decisão.

Apesar de ser um método agnóstico em que qualquer modelo pode ser explicado o LORE tem algumas restrições para seu funcionamento: a base de dados deve ser do tipo tabular, o modelo deve ser consultado livremente e as tarefas de classificação devem ser binárias (GUIDOTTI, MONREALE, *et al.*, 2019).

O objetivo do método é explicar a decisão de um modelo caixa-preta *b* para um exemplo *x*. Para isso, é construída uma Árvore de Decisão *c* a partir de uma base de dados artificiais (vizinhança) *Z*, que contém indivíduos similares ao exemplo *x* gerados por um algoritmo genético. Dessa árvore, é extraída a explicação local composta de um conjunto de regras lógicas *r*, que indica o caminho que o exemplo *x* percorre na árvore *c*; e um conjunto de regras contrafatuais Φ, contendo condições que podem ser mudadas em *x* para que a decisão de *b* seja invertida.

Na Figura 9 é mostrado o algoritmo LORE. Na linha 1 são definidos os parâmetros do algoritmo genético: *G*, o número de gerações; p_c, a probabilidade de recombinação; e p_m, a probabilidade de mutação. Nas linhas 2 e 3 são criadas duas vizinhanças por meio do algoritmo genético: a primeira com indivíduos similares a *x* e que possuem a mesma classificação pela caixa-preta *b*, ou seja, que estão do mesmo lado da fronteira de decisão; e a segunda com indivíduos similares a *x* mas que possuem uma classificação diferente pela caixa-preta *b*, ou seja, estão do outro lado da fronteira de decisão. Na Figura 10 é ilustrado um exemplo da população gerada pelo AG.

Figura 9 – Algoritmo LORE

Algoritmo 2 LORE

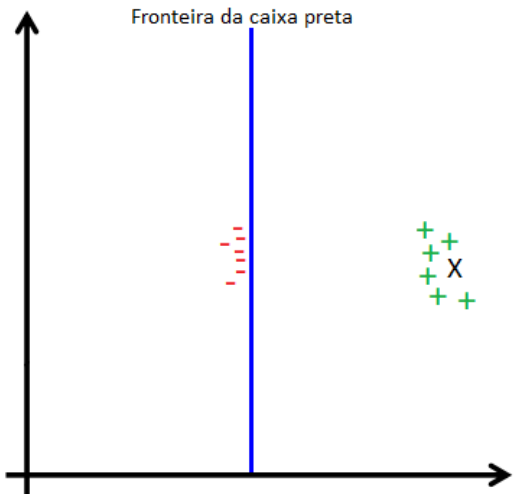
Entrada: $\mathbf{x}$ - exemplo a ser explicado, b - modelo caixa preta,
N - tamanho da população, G - número de gerações,
p_c - probabilidade de recombinação, p_m - probabilidade de mutação
aptidão - função de aptidão

Output: $e_c(\mathbf{x})$ - explicação da decisão de b para x

1: $G \leftarrow 10;\ p_c \leftarrow 0.5;\ p_m \leftarrow 0.2;$
2: $Z_1 \leftarrow GA_1(\mathbf{x}, b, N/2, G, p_c, p_m);$
3: $Z_2 \leftarrow GA_2(\mathbf{x}, b, N/2, G, p_c, p_m);$
4: $Z \leftarrow Z_1 \cup Z_2;$
5: $c \leftarrow$ árvoreDecisão$(Z);$
6: $r_c(\mathbf{x}) \leftarrow$ extraiRegras$(c, \mathbf{x});$
7: $\Phi_c(\mathbf{x}) \leftarrow$ extraiContrafatuais$(c, r_c(\mathbf{x}), \mathbf{x});$
8: $e_c(\mathbf{x}) = \langle r_c(\mathbf{x}), \Phi_c(\mathbf{x}) \rangle;$
9: retorne $e_c(\mathbf{x})$

Fonte: (GUIDOTTI, MONREALE, *et al.*, 2019).

Figura 10 – Exemplo da população gerada pelo AG, na qual os pontos em verde são da mesma classe do ponto a ser explicado (X), enquanto que os pontos em vermelho são de outra classe.



Fonte: Elaborado pelo autor.

Para gerar cada vizinhança, o algoritmo genético usa as equações (1) e (2) para avaliar os indivíduos gerados na evolução, onde *I* é uma função Indicadora que assume 1 caso o índice seja verdadeiro e 0 caso o contrário, e *d* é uma função de distância. A Distância Euclidiana Normalizada é usada para atributos contínuos, onde a variância do conjunto de dados é usada

para normalizar os valores. Para atributos categóricos é usada a função de Casamento Simples (se dois atributos possuem o mesmo valor a distância é 0, caso o contrário 1).

$$fitness^{x}_{=}(\mathrm{z}) = I_{\mathrm{b(x)=b(z)}} + (1 - \mathrm{d(x,z)}) - I_{x=z} \quad (3)$$

$$fitness^{x}_{\neq}(\mathrm{z}) = I_{\mathrm{b(x)\neq b(z)}} + (1 - \mathrm{d(x,z)}) - I_{x=z} \quad (4)$$

A equação (3) procura por indivíduos *z* semelhantes a *x* (termo $1 - d(x, z)$), mas não iguais a *x* (termo $I_{x=z}$), no qual a classe atribuída pela caixa-preta *b* para *z* seja a mesma que a classe de *x* (termo $I_{b(x)=b(z)}$). A equação (4) procura por indivíduos *z* semelhantes a *x*, mas não iguais a *x*, no qual a classe atribuída pela caixa-preta *b* para *z* seja diferente da classe de *x*.

Com isso, o termo $1 - d(x, z)$ implica que a chamada ao algoritmo genético na linha 2 (Figura 10) retorna a vizinhança dos indivíduos mais próximos de *x* e também que a chamada na linha 3 retorna a vizinhança dos indivíduos da classe oposta à de *x*, mais próximos de *x* e, portanto, da fronteira de decisão.

As vizinhanças são concatenadas na linha 4 totalizando cerca de 1000 indivíduos criados pelas 2 execuções do AG, cerca de 500 indivíduos com mesma decisão que a classificação do indivíduo a ser explicado e cerca de 500 indivíduos com a decisão oposta. Em seguida, na linha 5, a Árvore de Decisão *c* é construída. As linhas 6 e 7 correspondem à extração dos conjuntos de regras lógicas e contrafatuais, respectivamente, da Árvore de Decisão *c*. As regras *r* são extraídas percorrendo o caminho para decidir a classificação do exemplo *x* pela Árvore de Decisão *c*. O subconjunto contrafatual *Φ*, com condições que alteram a decisão de classificação de *x* pela caixa-preta *b*, é extraído percorrendo o caminho em *c* que resulta em uma classificação diferente. Por fim, na linha 8 é retornada a explicação local *e*.

Na Figura 12 é representado uma explicação dada pelo LORE da decisão de um modelo caixa-preta para um indivíduo *x* de 22 anos de idade, desempregado, que gostaria de pedir $10.000 emprestado ao banco e não possui carro. Para essa entrada, a explicação *e* dada pelo LORE é que *x* foi negado um empréstimo por conta das regras *r* (menor de 25 anos, desempregado e gostaria de uma quantia superior a $5000), e para reverter essa decisão as regras *Φ* devem ser satisfeitas para que um empréstimo seja concedido a este indivíduo (ou *x* precisa ser maior de 25 anos e pedir uma quantia menor ou igual a $5000; ou *x* precisa ter um trabalho de atendente e um carro).

Figura 11 – Exemplo de Explicação do método LORE

$$x = \{(age{=}22), (job = none), (amount{=}10k), (car{=}no):$$
$$e = \langle r = \{age{\leq}25, job{=}none, amount{>}5k\} \rightarrow deny,$$
$$\Phi = \{(\{age{>}25, amount{\leq}5k\} \rightarrow grant),$$
$$(\{job{=}clerk, car{=}yes\} \rightarrow grant)\}\rangle$$

Fonte: Adaptado de (GUIDOTTI, MONREALE, *et al.*, 2019).

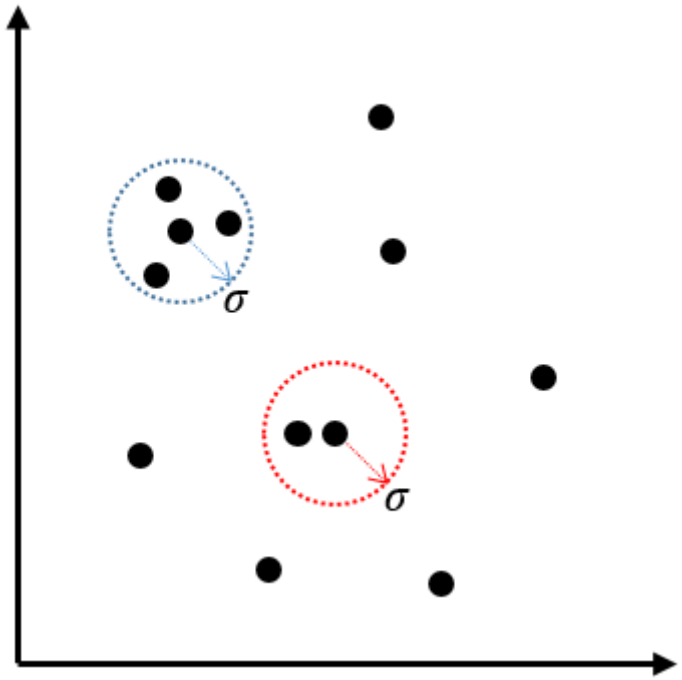
3.2 LOREFS – LORE COM *FITNESS SHARING*

A cada geração, o algoritmo genético geralmente produz indivíduos cada vez mais semelhantes, que convergem para ótimos locais e globais. A convergência prematura pode ser causada pela seleção e recombinação de parentes em uma população finita (EIBEN e SMITH, 2015).

Em problemas multimodais, com vários pontos de ótimos locais, isso é um desafio pois a população deixa de explorar o espaço de busca e passa a explorar uma única região de ótimo local. Uma das soluções adotadas para evitar a convergência prematura é utilizar a estratégia de nichos (*niching*) a fim de preservar a diversidade da população de maneira a representar diversos pontos de ótimos locais (EIBEN e SMITH, 2015). Existem diversas técnicas de nichos na literatura (MAHFOUD, 1995; DEJONG, 1975; BEASLEY, BULL e MARTIN, 1993) e uma deles é o *fitness sharing* (HOLLAND, 1975; GOLDBERG e RICHARDSON, 1987), talvez a mais conhecida e usada entre as técnicas de nicho.

Fitness sharing é usado para preservar a diversidade da população ao longo das gerações. É uma estratégia de nicho que penaliza soluções (indivíduos) em regiões densamente povoadas. A técnica calcula a distância de cada par de indivíduos da população e a aptidão de cada indivíduo é penalizada conforme o número de outros indivíduos dentro de um mesmo nicho definido por uma distância σ (Figura 12).

Figura 12 – Nicho definido pelo parâmetro σ no *fitness sharing*



Fonte: Elaborado pelo autor.

A preservação de nichos é feita por meio do cálculo de uma nova aptidão (fitness) para cada solução da população baseada no valor da aptidão original. A nova aptidão do i-ésimo indivíduo z_i da população *P* do AG é dada por:

$$aptidão'(z_i) = \frac{aptidão(z_i)}{\sum_{j=1}^{N} compartilhamento(d(z_i\,;z_j))} \qquad (1)$$

onde N é o tamanho da população, $d(z_i; z_j)$ é a distância entre os indivíduos z_i e z_j, e compartilhamento(·) é uma função de compartilhamento que mede a similaridade entre as soluções.

A função de compartilhamento é dada por:

$$compartilhamento(d) = \begin{cases} 1 - \left(\frac{d}{\sigma}\right)^{\propto}, & d < \sigma \\ 0, & caso\ contrário \end{cases} \qquad (2)$$

onde o parâmetro σ define o raio do nicho e ∝ controla a forma da função de compartilhamento. Neste trabalho, ∝ = 1, ou seja, a função de compartilhamento é linear. No *fitness sharing*, soluções dentro de uma distância menor que σ estão localizadas no mesmo nicho e, portanto, são penalizadas. A técnica de *fitness sharing* é proposta neste trabalho para ser utilizada para melhorar a eficiência do AG utilizado em LORE.

Técnicas especializadas de Nicho (*niching*) são mais apropriadas para manter a diversidade da população em um AG, e, dessa forma, podem potencialmente aprimorar o método LORE. Assim, o próximo passo é implementar a técnica de *fitness sharing* no AG e comparar a nova superfície de decisão da Árvore de Decisão com os experimentos da seção 4.1.

LORE com *fitness sharing* é chamado de LOREfs (LORE *fitness sharing*) e sua diferença para o LORE é a implementação da linha 8 no AG conforme a Figura 13. Após o processo de avaliação da aptidão dos indivíduos gerados em uma geração é verificada a proximidade destes indivíduos de acordo com o parâmetro σ, indivíduos dentro do raio σ tem sua aptidão penalizada.

Figura 13 - Algoritmo Genético LOREfs

Algoritmo 3 Algoritmo Genético LOREfs

Entrada: $\mathbf{x}$ - exemplo a ser explicado, b - modelo caixa preta,
N - tamanho da população, G - número de gerações,
p_c - probabilidade de recombinação, p_m - probabilidade de mutação
aptidão - função de aptidão, σ - raio do nicho
Saída: Z - conjunto de dados artificiais

1: $P_0 \leftarrow inicializaPopulação(\mathbf{x})$;
2: avaliação(P_0, b, *aptidão*);
3: **for** $i \leftarrow 0$ até $G - 1$ **do**
4: $P_{i+1} \leftarrow$ seleção(P_i);
5: $P'_{i+1} \leftarrow$ recombinação(P_{i+1}, p_c);
6: $P''_{i+1} \leftarrow$ mutação(P'_{i+1}, p_m);
7: avaliação(P''_{i+1}, b, *aptidão*);
8: $fitnessSharing(P''_{i+1}, \sigma)$;
9: $P_{i+1} = P''_{i+1}$;
10: **end for**
11: $Z \leftarrow P_{i+1}$
12: retorne Z

Fonte: Elaborado pelo autor.

4 EXPERIMENTOS E RESULTADOS

Experimentos foram realizados neste trabalho para verificar a diversidade do conjunto de dados artificias gerados pelo AG empregado em LORE. Os experimentos foram realizados utilizando como base o artigo que descreve o LORE e o código em linguagem Python disponibilizado pelos autores[1]. O mesmo código é também utilizado como base para as mudanças no AG propostas aqui.

Para realizar os experimentos foi utilizada uma máquina de configuração: Processador Core I7-2600K @4.3Ghz, 16GB de memória RAM, placa de vídeo GTX 2060, Windows 10 Professional. Linguagem e tecnologia utilizadas: Python e Jupyter Notebook.

No código do LORE, para gerar a vizinhança do exemplo a ser explicado é utilizado o *framework Distributed Evolutionary Algorithms in Python* (DEAP)[2], que foca em prover operadores básicos da Computação Evolutiva e também mecanismos gerais para o desenvolvimento de algoritmos genéticos (AGs) complexos, além de ser explícito e fácil de compreender e ler (FORTIN, DE RAINVILLE, *et al.*, 2012).

Apesar de facilitar a implementação, o LORE não utiliza todo o poder do DEAP e dos algoritmos genéticos, pois utiliza um algoritmo simples pré-implementado pelo framework, idêntico ao algoritmo genético mais simples apresentado na Figura 7. Experimentos apresentados a seguir mostram que o AG empregado em LORE acarreta na perda da diversidade da população e, consequentemente, em uma explicação pobre pois a fronteira de decisão da Árvore de Decisão construída não representa fielmente a fronteira de decisão local da caixa-preta.

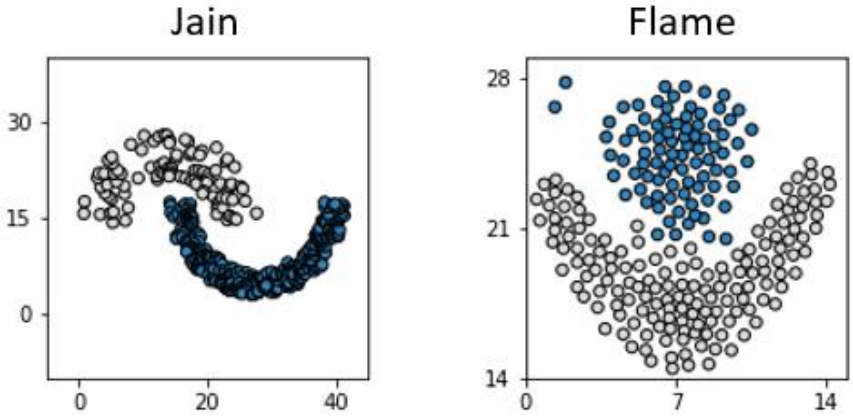
Em um primeiro momento, foi investigada a semelhança das superfícies de decisão da caixa-preta *b* e da Árvore de Decisão *c* para analisar o quão fiel é a reconstrução local antes de extrair uma explicação. Para isso, foram feitos experimentos (apresentados na Seção 4) para explicar as decisões de um modelo *Perceptron* multicamadas (PMC) e de uma FA.

Neste trabalho, os algoritmos LORE e LOREfs foram comparados utilizando-se quatro bases de dados. As duas primeiras bases de dados são *Jain*[3] e *Flame*[4] (Figura 14). Estes conjuntos foram escolhidos por: i) serem bidimensionais, o que permite visualizar no plano as fronteiras de decisão geradas pelos classificadores e plotar os dados, e as superfícies de decisão

[1] Disponível em https://github.com/riccotti/LORE.
[2] Disponível em https://github.com/DEAP/deap.
[3] Disponíveis em http://cs.joensuu.fi/sipu/datasets/jain.txt
[4] Disponíveis em http://cs.joensuu.fi/sipu/datasets/flame.txt

da caixa-preta e da Árvore de Decisão, e ii) por serem compostos de aglomerados com formatos diversos. *Jain* possui 373 amostras e *Flame* possui 240 amostras.

Figura 14 – Plot dos conjuntos de dados *Jain* e *Flame*.



Fonte: Elaborado pelo autor.

Em seguida, foi investigada a modificação do algoritmo genético de forma a implementar técnicas de nicho (*niching*) para preservar a diversidade da população e garantir que a fronteira local esteja bem representada. Foi usado os mesmos conjuntos de dados do experimento anterior a fim de comparar qualitativamente os resultados entre os experimentos.

Para comparar o LORE e LOREfs quantitativamente foram usados mais dois conjuntos de dados além do *Jain* e *Flame.* Os dois conjuntos de dados são *Breast Cancer* e *Heart Disease*, ambos do UCI Machine Learning Repository (DUA e GRAFF, 2017). O conjunto de dados *Breast Cancer* tem 699 amostras e 10 atributos, enquanto o conjunto de dados *Heart Disease* tem 303 amostras e 14 atributos. Consideraremos neste trabalho apenas bases com valores dados por números reais, apesar de outros tipos de bases de dados poderem ser considerados no futuro. Como no LORE, as bases devem possuir apenas duas classes.

A Seção 4.1 mostra resultados de experimentos que avaliam o impacto do número de gerações (G) e taxa de mutação (p_m) no LORE. Neste experimento, o modelo de caixa-preta é um Perceptron Multicamadas (PMC). A Seção 4.2 mostra resultados de experimentos que avaliam o impacto do número de gerações (G) e taxa de mutação (p_m) no LOREfs. O LOREfs possui um parâmetro, σ, que controla o tamanho dos nichos de indivíduos da população do AG. Na Seção 4.3, é apresentada uma estratégia que encontra empiricamente bons valores de σ.

LOREfs é comparado quantitativamente com LORE na Seção 4.4. A comparação é feita em relação ao erro de classificação para amostras do conjunto de dados original próximas ao exemplo a ser explicado. Também são apresentados resultados de FA como modelos de caixa-

preta. Por fim, na Seção 4.5, é feita a comparação entre LORE e LOREfs quanto à diversidade local dos conjuntos de dados artificiais gerados pelos AGs.

4.1 IMPACTO DOS PARÂMETROS DO AG NO LORE

Os primeiros experimentos foram realizados para entender o comportamento do AG na geração dos indivíduos que compõem o conjunto de treinamento da Árvore de Decisão. Foram feitas alterações no código original do autor do LORE para plotar os indivíduos gerados pelo AG e também a superfície de decisão da Árvore de Decisão. No LORE (Figura 9), é utilizado um AG com número de gerações baixo (10) e uma taxa de mutação alta (0.2). Com o objetivo de entender a escolha dos autores para tais parâmetros não-usuais e o impacto destes parâmetros na população final (vizinhança gerada do exemplo a ser explicado), foram feitos experimentos aqui variando-se o número de gerações e taxa de mutação.

Foram realizados 20 experimentos divididos em 2 para os conjuntos de dados, *Jain* e *Flame.* Para cada conjunto de dados, os experimentos foram subdivididos para explicar um modelo FA e um modelo PMC (5 experimentos para cada). Por fim, para cada modelo, os experimentos foram novamente subdivididos em 1 experimento de variação na taxa de mutação (valores testados 0.05, 0.10, 0.15 e 0.20) com número de gerações constante e igual a 10; e 4 experimentos de variação no número de gerações (valores testados 10, 50, 100 e 150), onde cada um dos 4 experimentos foi testado com uma taxa de mutação diferente (0.05, 0.10, 0.15, 0.20). A Tabela 1 mostra a divisão dos experimentos.

Os valores para a taxa de mutação e número de gerações foram escolhidos de modo a entender o impacto destes parâmetros na população final gerada pelo AG do LORE. Os indivíduos a serem explicados foram selecionados manualmente na tentativa de explorar diferentes explicações.

Na Figura 15 são mostrados os resultados do primeiro experimento realizado com: variação da taxa de mutação (*mutpb – mutation probability*) do AG nos valores 0.05, 0.10, 0.15 e 0.20, com número de gerações (*G*) igual a 10, para um modelo caixa-preta do tipo FA e conjunto de dados *Jain*. Resultados para outros modelos e conjuntos de dados são apresentados no Apêndice A. As colunas representam diferentes indivíduos que estão sendo explicados, enquanto as linhas representam os diferentes testes realizados nestes indivíduos. Na primeira linha, mostra-se o conjunto de dados original e também a fronteira de decisão da caixa-preta, neste caso uma FA. A estrela vermelha indica o indivíduo que está sendo explicado.

Tabela 1 – Divisão dos Experimentos.

Conjunto de Dados = *Jain*		Conjunto de Dados = *Flame*	
FA	PMC	FA	PMC
Experimento 1 Variação na Taxa de Mutação	Experimento 6 Variação na Taxa de Mutação	Experimento 11 Variação na Taxa de Mutação	Experimento 16 Variação na Taxa de Mutação
Experimento 2 Variação no nº de Gerações Taxa mutação = 0.20	Experimento 7 Variação no nº de Gerações Taxa mutação = 0.20	Experimento 12 Variação no nº de Gerações Taxa mutação = 0.20	Experimento 17 Variação no nº de Gerações Taxa mutação = 0.20
Experimento 3 Variação no nº de Gerações Taxa mutação = 0.15	Experimento 8 Variação no nº de Gerações Taxa mutação = 0.15	Experimento 13 Variação no nº de Gerações Taxa mutação = 0.15	Experimento 18 Variação no nº de Gerações Taxa mutação = 0.15
Experimento 4 Variação no nº de Gerações Taxa mutação = 0.10	Experimento 9 Variação no nº de Gerações Taxa mutação = 0.10	Experimento 14 Variação no nº de Gerações Taxa mutação = 0.10	Experimento 19 Variação no nº de Gerações Taxa mutação = 0.10
Experimento 5 Variação no nº de Gerações Taxa mutação = 0.05	Experimento 10 Variação no nº de Gerações Taxa mutação = 0.05	Experimento 15 Variação no nº de Gerações Taxa mutação = 0.05	Experimento 20 Variação no nº de Gerações Taxa mutação = 0.05

Fonte: Elaborado pelo autor.

As linhas subsequentes apresentam as populações geradas pelo AG para cada valor da taxa de mutação (a partir do valor default 0.20) e a fronteira de decisão da Árvore de Decisão gerada pelo LORE, que explica a decisão para um indivíduo (estrela vermelha). Nas laterais de cada gráfico, foram plotados histogramas a fim de observar a distribuição de indivíduos ao longo dos eixos *X* e *Y*. Os valores apresentados nos histogramas representam o máximo de indivíduos encontrados em uma mesma abscissa ou ordenada.

Na Figura 16 é apresentado o segundo experimento feito: variação do número de gerações do AG nos valores 10, 50, 100 e 150, com taxa de mutação igual a 0.2, para um modelo caixa-preta do tipo FA e conjunto de dados *Jain*. A leitura dos gráficos é similar ao experimento anterior.

A partir dos dois experimentos, é possível observar uma alta concentração de indivíduos em uma mesma ordenada ou abscissa. Isto confirma a hipótese de convergência prematura que acarreta em perda de diversidade da população. A falta de diversidade pode ser um problema pois o AG deixa de explorar outras áreas por estar concentrado em uma região só, o que afeta a explicação dada, como pode ser visto no Experimento 1, Indivíduo 1 e *mutpb* = 0.2.

Como o AG não conseguiu evoluir um indivíduo que explorasse a área acima do indivíduo a ser explicado (estrela vermelha), a superfície de decisão da Árvore de Decisão construída pelo

LORE não é fiel à fronteira original. Isto afeta a explicação dada, mais precisamente, afeta os contrafatuais, uma vez que o LORE deixa de retornar uma possível regra que faz com que a decisão seja revertida.

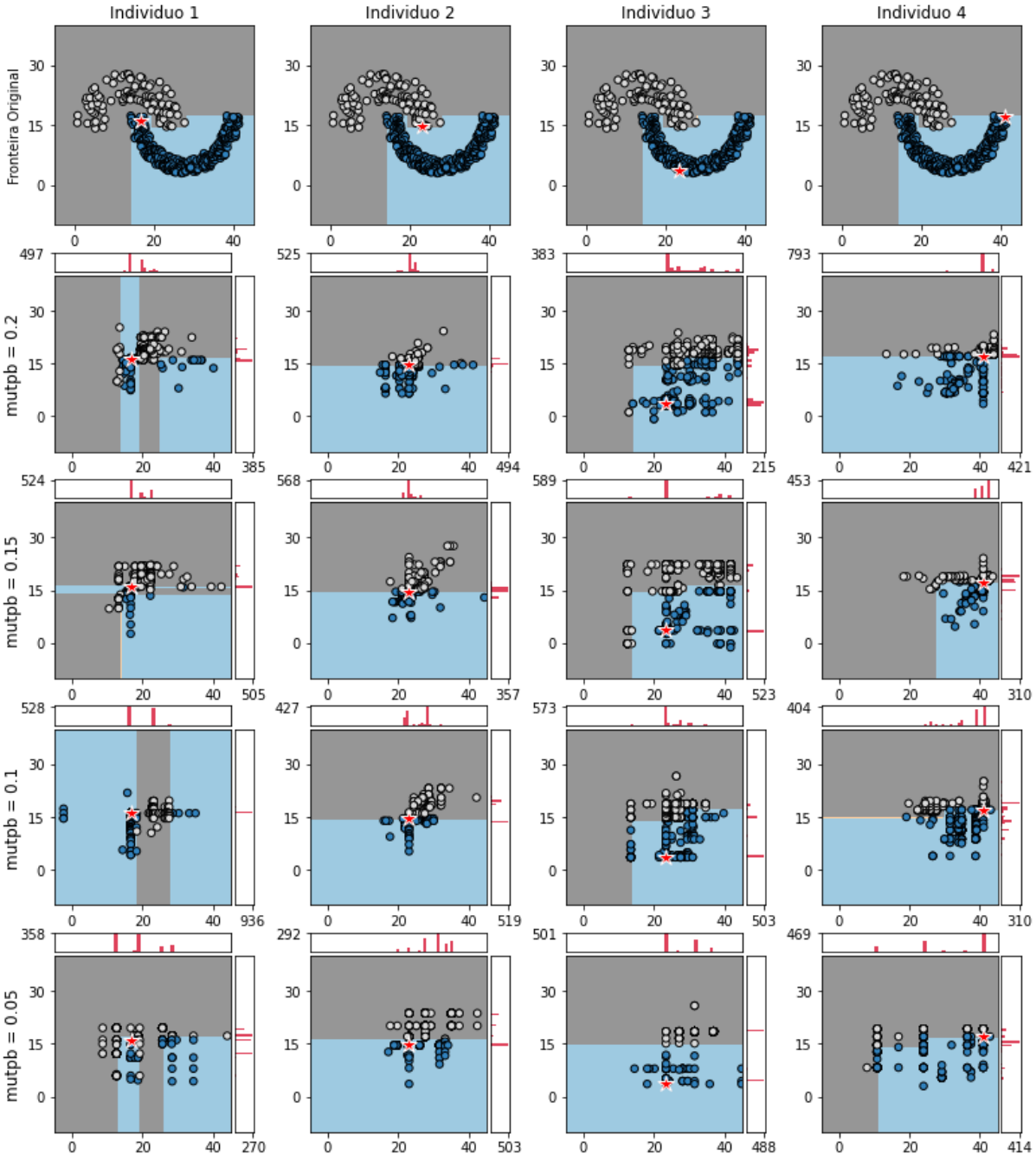
Em uma aplicação real, a consequência disto é que o poder de ação de uma pessoa afetada por essa decisão pode ficar limitado. A situação se agrava em problemas de maiores dimensões, pois quanto maior o número de dimensões, maior é o espaço de busca (problema da Maldição da Dimensionalidade), o que aumenta a probabilidade de um AG com perda de diversidade de população não explorar de maneira eficaz o espaço de entradas.

Da análise do Experimento 1, é possível observar que na maior parte dos casos quanto maior é a taxa de mutação, maior é a diversidade da população. Como o AG é inicializado com cópias idênticas do indivíduo a ser explicado, a evolução da população inicial depende principalmente da operação de mutação, uma vez que a recombinação de dois indivíduos iguais não produz variabilidade genética. Assim, taxas de mutação mais baixas implicam em uma menor probabilidade do AG explorar a caixa-preta.

Da análise do Experimento 2, é possível observar que na maior parte dos casos quanto menor é o número de gerações, maior é a diversidade da população. Este comportamento é devido as funções de aptidão (Equações 1 e 2) do AG que dão maior prioridade a indivíduos mais próximos do indivíduo a ser explicado por meio da função de distância. Isto sugere que o AG está sofrendo uma parada prematura a fim de preservar a diversidade gerada pela a alta taxa de mutação no início da evolução. Os demais experimentos encontram-se no Apêndice A e seus resultados corroboram com as análises feitas a partir dos dois experimentos apresentados.

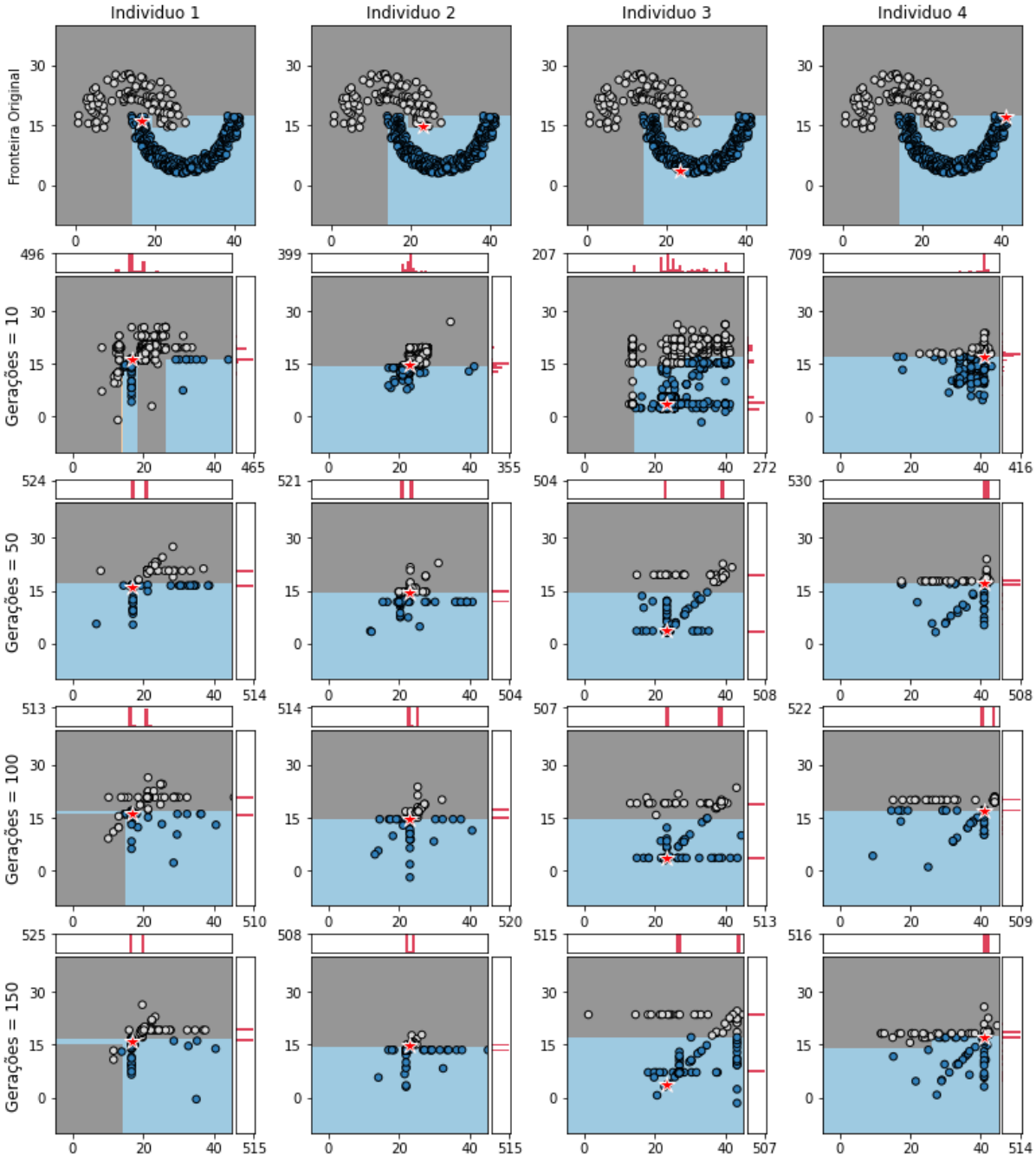
No entanto, em alguns casos, há uma falha na geração da Árvore de Decisão onde ela é treinada com indivíduos de uma única classe, como mostrado na Figura 17. Apesar de não ser documentado no artigo em sua implementação o LORE reduz o tamanho do conjunto de dados artificial usado para treinar a Árvore de Decisão, o seguinte procedimento é aplicado à população de cada execução dos AGs do LORE original: i) ordena a lista de indivíduos da população final de acordo com sua aptidão; ii) encontra a maior diferença de aptidão entre dois indivíduos consecutivos da lista; iii) define a aptidão do indivíduo com a maior diferença de aptidão (passo ii) como limiar; iv) remove do conjunto de dados artificiais todos os indivíduos com fitness menor que o limite definido na etapa iii.

Figura 15 – Experimento 1: LORE - Alteração da Taxa de Mutação, Gerações = 10, FA, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

Figura 16 – Experimento 2: LORE - Alteração do Número de Gerações, *mutpb* = 0.20, FA, Conjunto de Dados = *Jain*



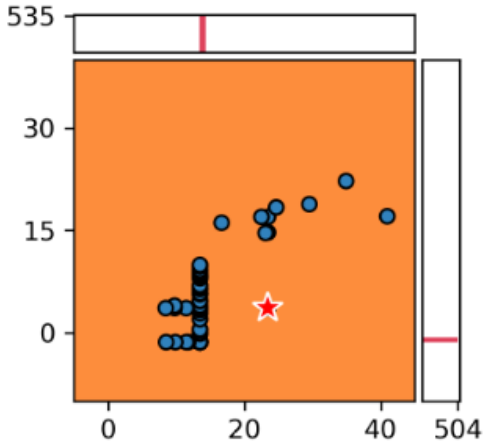
Fonte: Elaborado pelo autor.

Em experimentos iniciais com LORE, observamos que, para alguns conjuntos de dados, o número de indivíduos removidos por esse procedimento é muito alto. Como a população inicial do AG é composta de cópias do exemplo a ser explicado pode ocorrer do AG não conseguir evoluir um indivíduo diferente da cópia, ou ainda evoluir para indivíduos exatamente iguais. Consequentemente, um pequeno número de exemplos é usado para treinar a Árvore de Decisão,

resultando em um desbalanceamento entre as classes. Para evitar esse efeito, o procedimento foi modificado para que o conjunto de treinamento seja composto de pelo menos 100 indivíduos gerados por cada AG para as execuções do novo algoritmo LOREfs.

Conclui-se que, ao usar um AG padrão a Árvore de Decisão gerada pelo LORE não representa com fidelidade a caixa-preta, o que impacta na qualidade da explicação. Os valores da taxa de mutação e número de geração do AG foram escolhidos para dar uma maior ênfase à exploração do espaço de busca (*Exploration*) do que a seu aproveitamento (*Exploitation*).

Figura 17 – Exemplo de falha na execução do AG.



Fonte: Elaborado pelo autor.

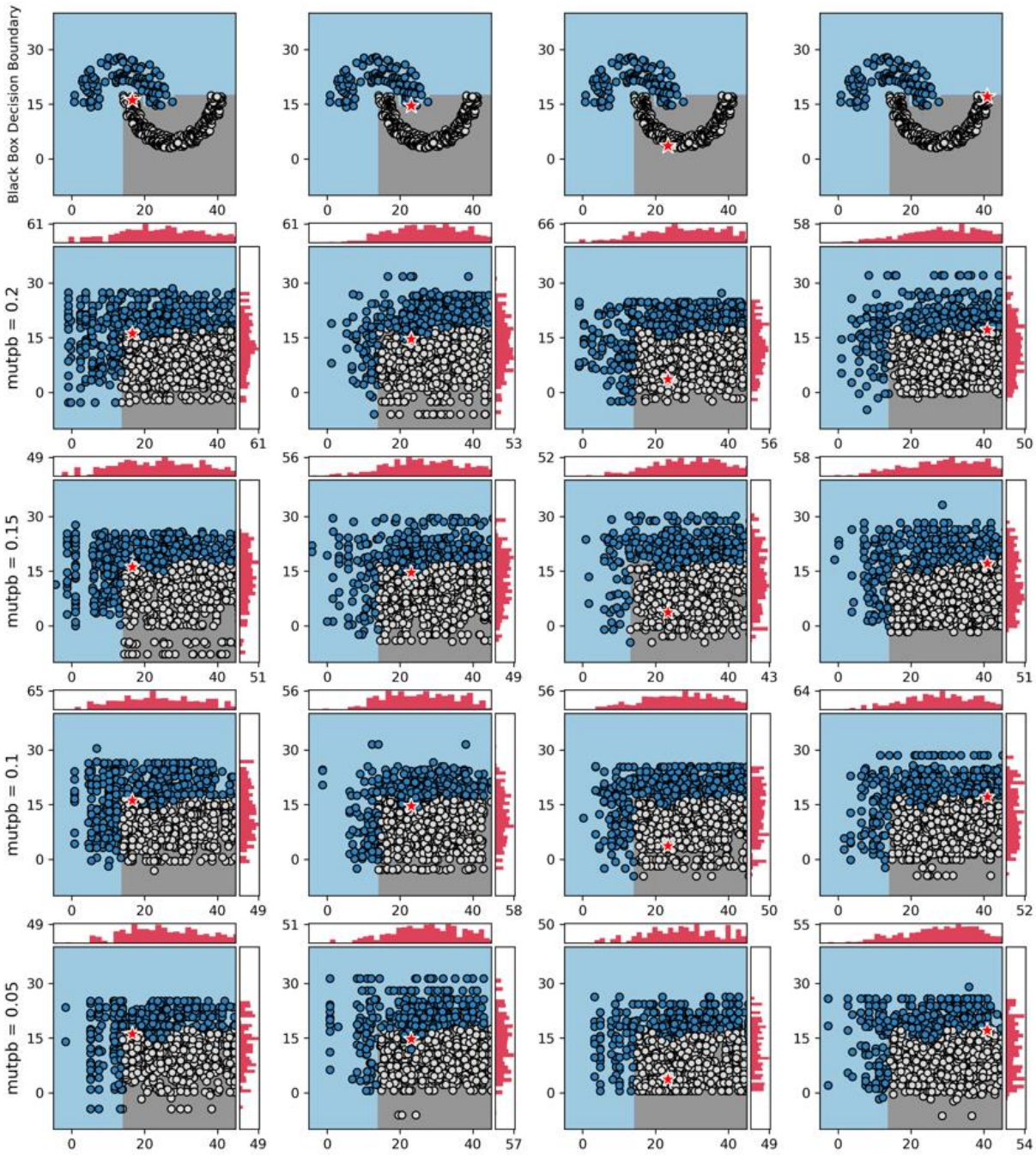
4.2 LORE COM *FITNESS SHARING*

A fim de comparar os dois métodos, os experimentos do LOREfs seguem o mesmo design da Seção 4.1: 20 experimentos com os conjuntos de dados *Jain* e *Flame* onde é variada a taxa de mutação e número de gerações, e plotado os indivíduos gerados pelo AG e também a superfície de decisão da Árvore de Decisão. Para o novo parâmetro σ *é* usado σ =1.

Os resultados dos LOREfs são apresentados nas Figuras 18 e 19, os demais experimentos encontram-se no Apêndice A. Nesses experimentos, o LOREfs produziu superfícies de decisão para o modelo substituto (Árvore de Decisão) que são mais semelhantes aos do modelo caixa-preta. Isso pode ser explicado pela maior diversidade populacional quando o *fitness sharing* é adotado. Os histogramas indicam que indivíduos artificiais mais diversos são gerados pelos AGs. Consequentemente, as superfícies de decisão próximas ao exemplo a ser explicado são mais semelhantes aos do modelo caixa-preta.

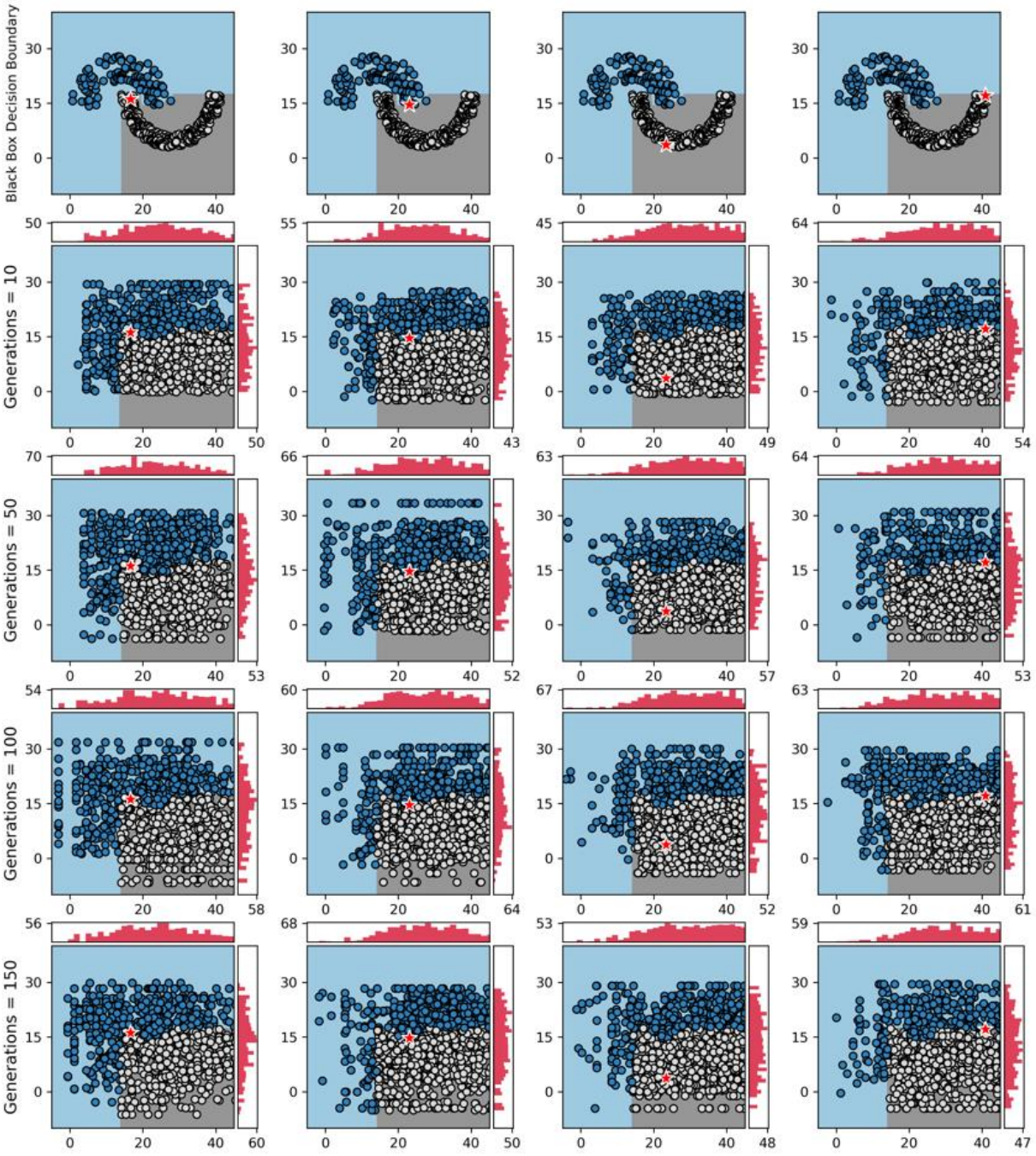
Observa-se que os parâmetros dos AGs também impactam as superfícies de decisão. No entanto, o impacto é menor quando comparado ao LORE, principalmente quanto ao número de gerações.

Figura 18 – Experimento 21: LOREfs - Alteração da Taxa de Mutação, Gerações = 10, FA, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

Figura 19 – Experimento 22: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.20, FA, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

4.3 DETERMINANDO O HIPERPARÂMETRO σ

Ao implementar o *fitness sharing*, é inserido um novo hiperparâmetro σ que precisa ser determinado antes da execução do LOREfs. Nesta seção é proposto encontrar empiricamente o valor de σ no LOREfs.

Além dos experimentos com os conjuntos de dados *Jain* e *Flame*, são realizados experimentos com dois conjuntos de dados públicos relacionados à Medicina *Breast Cancer* e *Heart Disease*.

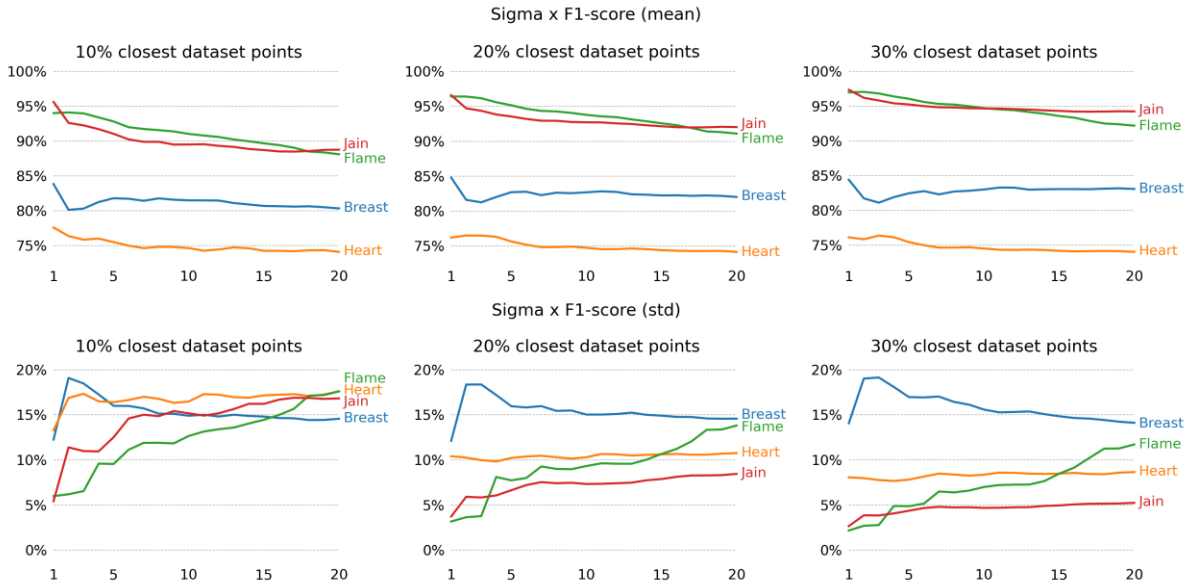
No procedimento para encontrar σ, cada conjunto de dados foi dividido aleatoriamente em conjuntos de treinamento (80%) e de teste (20%). O conjunto de treinamento foi usado para treinar um PMC (modelo caixa-preta), enquanto o conjunto de teste foi usado para explicar a decisão do modelo caixa-preta. Consequentemente, os conjuntos de dados têm um número diferente de exemplos a serem explicados (conjunto de teste): *Flame* tem 48 instâncias, *Jain* tem 75 instâncias, *Heart Disease* tem 61 instâncias e *Breast Cancer* tem 114 instâncias. Para cada exemplo *x* do conjunto de teste, um modelo substituto é construído e os resultados médios são relatados. Os parâmetros do AG são os mesmos usados na implementação original do LORE: $p_m = 0.2$, $p_c = 0.5$, G = 10 e N = 1000.

A comparação do LOREfs com diferentes valores de σ é feita em relação ao F1-score. O F1-score é calculado comparando a classificação do modelo caixa-preta *b* (PMC) e o modelo substituto *c* (Árvore de Decisão) para uma porcentagem de amostras do conjunto de treinamento que estão mais próximas do exemplo *x*. Por exemplo, quando a porcentagem é 20%, o subconjunto de amostras para calcular o F1-score é formado pela união das 10% amostras do conjunto de dados que estão mais próximas de *x* e possuem a mesma classe de *x* e 10% amostras do conjunto de dados que estão mais próximos de *x*, mas têm a classe oposta. Este método é adotado pelo interesse em reproduzir as superfícies de decisão próximas do exemplo a ser explicado. O número de execuções para cada conjunto de dados é 20, uma para cada valor de σ variando de 1 a 20. O F1-score médio é calculado para todos os exemplos do conjunto de teste. Os resultados são apresentados na Figura 20.

Na maioria dos experimentos, $\sigma = 1$ resultou no melhor F1-score médio e no desvio padrão mínimo. Mesmo quando o melhor F1-score não é obtido quando $\sigma = 1$, os resultados são ligeiramente piores. De qualquer forma, este procedimento pode ser usado para encontrar bons valores de σ para diferentes conjuntos de dados. É importante observar que os melhores valores de σ podem mudar devido a diferentes propriedades, por exemplo, a distribuição dos exemplos

no espaço de classificação e o valor máximo e mínimo de cada atributo. Nos experimentos apresentados nas seções a seguir, $\sigma = 1$.

Figura 20 – σ vs F1-score médio e σ vs Desvio Padrão médio para 4 conjuntos de dados diferentes.



Fonte: Elaborado pelo Autor

4.4 ANÁLISE QUANTITATIVA ENTRE LORE E LOREfs

Para a análise quantitativa da comparação do LORE e do LOREfs foi usado o mesmo modelo de experimento anterior (Seção 4.3). Além dos resultados para um modelo caixa-preta PMC também foi utilizado um modelo FA.

Ao treinar o PMC como modelo caixa-preta, a precisão obtida foi: 1 para o conjunto de dados *Flame*, 0.987 para *Jain*, 0.965 *Breast Cancer*, e 0.770 para *Heart Disease*. Ao treinar a FA como modelo caixa-preta, a acurácia obtida foi: 0.937 para *Flame*, 0.920 para *Jain*, 0.965 para *Breast Cancer*, e 0.787 para *Heart Disease*.

Nos experimentos com LORE e LOREfs, diferentes porcentagens do conjunto de dados (conjunto de dados de treinamento) são consideradas para o cálculo do F1-score. Também são comparados estatisticamente os resultados de LORE e LOREfs.

Os resultados para o experimento com o PMC são apresentados na Tabela 1, enquanto os resultados para a FA são apresentados na Tabela 2. A média e o desvio padrão do F1-score são apresentados. O F1-score é calculado comparando a classificação do modelo caixa-preta *b* e do modelo substituto *c*. Para explicar cada exemplo *x* do conjunto de teste, um modelo substituto é construído. O F1-score é calculado para os modelos *b* e *c* aplicados às seguintes porcentagens

do conjunto de dados: 10%, 20% e 30% amostras do conjunto de dados para cada classe que estão mais próximas de *x* são usadas. Os símbolos '=', '+' e '−' respectivamente indicam que os resultados de LOREfs são iguais, melhores ou piores que os resultados de LORE. O teste de postos sinalizados de Wilcoxon (teste não paramétrico), com $\alpha = 0.05$, é utilizado para comparar estatisticamente os resultados. A letra *s* indica que o valor-p é menor que α.

O algoritmo proposto, LOREfs, apresentou resultados significativamente melhores para o F1-score em todos os casos, tanto para experimentos de PMC quanto de FA (tabelas 1 e 2). Portanto, LOREfs obteve uma classificação média melhor que a do LORE para todos os conjuntos de dados. O melhor desempenho é explicado pela maior diversidade das populações finais dos AGs. Uma maior diversidade resultou em superfícies de decisão em torno do exemplo *x* mais semelhantes aos do modelo caixa-preta. Os tempos mínimo e máximo para explicar uma decisão do modelo caixa-preta nos experimentos são apresentados na Tabela 3.

Tabela 2 – LORE x LOREfs PMC

	10% of the dataset		20% of the dataset		30% of the dataset	
Dataset	LORE	LOREfs	LORE	LOREfs	LORE	LOREfs
Flame	0.783±0.278	0.925±0.062(s+)	0.829±0.21	0.954±0.038(s+)	0.855±0.164	0.964±0.03(s+)
Jain	0.766±0.322	0.894±0.092(s+)	0.832±0.235	0.928±0.043(s+)	0.882±0.147	0.946±0.027(s+)
BreastCancer	0.736±0.243	0.859±0.142(s+)	0.717±0.246	0.865±0.136(s+)	0.707±0.249	0.855±0.138(s+)
HeartDisease	0.571±0.303	0.704±0.192(s+)	0.571±0.293	0.707±0.158(s+)	0.565±0.289	0.714±0.128(s+)

Fonte: Elaborado pelo Autor

Tabela 3 – LORE x LOREfs FA

	10% of the dataset		20% of the dataset		30% of the dataset	
Dataset	LORE	LOREfs	LORE	LOREfs	LORE	LOREfs
Flame	0.810±0.274	0.932±0.08(s+)	0.854±0.226	0.955±0.044(s+)	0.873±0.177	0.965±0.031(s+)
Jain	0.834±0.253	0.969±0.062(s+)	0.878±0.196	0.978±0.038(s+)	0.901±0.176	0.983±0.025(s+)
BreastCancer	0.744±0.200	0.805±0.162(s+)	0.762±0.179	0.838±0.138(s+)	0.775±0.173	0.863±0.106(s+)
HeartDisease	0.542±0.305	0.697±0.169(s+)	0.539±0.278	0.747±0.096(s+)	0.546±0.272	0.750±0.088(s+)

Fonte: Elaborado pelo Autor

Tabela 4 – LORE x LOREfs tempo de execução (em segundos).

	MLP		RF	
Dataset	LORE	LOREfs	LORE	LOREfs
Flame	3-4	12-14	13-17	27-31
Jain	3-4	12-17	14-17	27-33
BreastCancer	5-7	14-16	15-16	34-48
HeartDisease	3-4	11-14	12-14	27-30

Fonte: Elaborado pelo Autor

Exemplos de explicação de modelos de caixa-preta para casos dos conjuntos de dados *Breast Cancer* e *Heart Disease* são apresentados respectivamente nas figuras 21 e 22. Nesses exemplos, as explicações *r* geradas pelo LORE são muito simples: apenas 1 atributo é usado para explicar três das decisões, enquanto 3 atributos são usados para explicar uma decisão. As regras contrafatuais Φ também são muito simples: apenas 1 ou 2 regras. Por outro lado, a explicação *r* e as regras contrafatuais Φ produzidas por LOREfs são mais complexas. Nos exemplos, 3, 5 ou 6 recursos são usados para explicar as decisões dos modelos caixa-preta, indicando superfícies de decisão mais complexas do modelo substituto. Além disso, existem mais regras contrafatuais (2, 4 ou 5). Quando comparado ao LORE, o LOREfs gera superfícies de decisão locais mais semelhantes aos produzidos pelo modelo caixa-preta. Isso é explicado novamente pela maior diversidade dos conjuntos de dados artificiais gerados pelos AGs com *fitness sharing*.

Figura 21 – Exemplos de Explicações do LORE e LOREfs para o conjunto de dados *Heart Disease*

Heart Disease Dataset

Example 1

LORE

$r = (\{cp : <= 2.00\} \rightarrow diagnosis = \text{" } > 50\% \ diameter \ narrowing\text{"})$

$\Phi = (\{cp : > 2.00\})$

LOREfs

$r = (\{thal : > 5.069187,$
$cp : <= 2.625233,$
$thalach : <= 159.433245,$
$fbs : -0.16829 < fbs <= 0.21,$
$ca : <= 1.597054,$
$chol : > 303.517165\} \rightarrow diagnosis = \text{" } > 50\% \ diameter \ narrowing\text{"})$

$\Phi = (\{thal : <= 5.069187, thalach : > 144.251838\},$
$\{fbs : 0.21 < fbs <= 0.234588\},$
$\{fbs : > 0.234588, oldpeak : > 1.870153\},$
$\{chol : 303.145116 < chol <= 303.517165\},$
$\{fbs : <= -0.16829, oldpeak : > 2.202191\})$

Example 2

LORE

$r = (\{oldpeak : <= 0.984268\} \rightarrow diagnosis = \text{" } < 50\% \ diameter \ narrowing\text{"})$

$\Phi = (\{oldpeak : > 0.984268\})$

LOREfs

$r = (\{thal : <= 4.504506,$
$oldpeak : <= 1.691220,$
$thalach : > 152.417628,$
$ca : <= 1.415176,$
$exang : <= 0.716649\} \rightarrow diagnosis = \text{" } < 50\% \ diameter \ narrowing\text{"})$

$\Phi = (\{exang : 0.716649 < exang <= 0.798229\},$
$\{thalach : <= 152.417628, exang : > 0.705035\})$

Fonte: Elaborado pelo Autor

Figura 22 – Exemplos de Explicações do LORE e LOREfs para o conjunto de dados *Breast Cancer*

Breast Cancer Dataset

Example 1

LORE

$$r = (\{radius\_worst : > 12.014649, perimeter\_mean : > 82.104949, symmetry\_worst > 0.387108\} \rightarrow diagnosis = Benign)$$
$$\Phi = (\{radius\_worst : <= 12.014649\}, \{perimeter\_mean : <= 82.104949\})$$

LOREfs

$$r = (\{perimeter\_mean : > 85.201299, perimeter\_worst : <= 106.167881, texture\_se : <= 1.943543, radius\_mean : > 11.824107, area\_mean : <= 615.259162\} \rightarrow diagnosis = Benign)$$
$$\Phi = (\{perimeter\_mean : <= 85.201299, perimeter\_worst : > 99.62338\}, \{perimeter\_mean : <= 67.222484, perimeter\_worst : <= 99.62338\}, \{radius\_mean : <= 3.075754, compactness\_se : > 0.028771\}, \{area\_mean : > 615.259162, area\_se : > 110.491737\})$$

Example 2

LORE

$$r = (\{perimeter\_mean : <= 149.696084\} \rightarrow diagnosis = Malign)$$
$$\Phi = (\{perimeter\_mean : > 149.696084\})$$

LOREfs

$$r = (\{area\_se : > 138.741714, area\_mean : > 599.103741, perimeter\_worst : > 92.434470\} \rightarrow diagnosis = Malign)$$
$$\Phi = (\{area\_mean : 486.895625 < area\_mean <= 599.103741, perimeter\_mean : > 89.620855\}, \{perimeter\_worst : <= 92.43447, perimeter\_mean : > 138.909976, fractal\_dimension\_mean : <= 0.060839\})$$

Fonte: Elaborado pelo Autor

4.5 ANÁLISE DA DIVERSIDADE DA POPULAÇÃO DO AG

Para quantificar a diversidade populacional dos AGs, é medida as distâncias médias entre cada par de indivíduos gerados pelos AGs em cada execução dos experimentos apresentados na Seção 4.4; ou seja, após treinar o modelo substituto, são calculadas as distâncias entre cada par de indivíduos gerados pelo AG.

Todos os quatro conjuntos de dados têm atributos reais. Assim, a distância Euclidiana é usada como métrica de distância. A média e o desvio padrão da distância Euclidiana (para diferentes exemplos a serem explicados) são apresentados na Tabela 4. LOREfs apresentou resultados significativamente melhores para diversidade populacional em todos os experimentos. Esses resultados corroboram com os resultados apresentados nas seções anteriores.

Tabela 5 – Diversidade entre LORE e LOREfs para dois tipos de caixa-preta (PMC e FA)

	MLP		RF	
Dataset	LORE	LOREfs	LORE	LOREfs
Flame	3.057±1.222	8.452±0.300(s+)	2.726±1.122	8.443±0.307(s+)
Jain	7.814±3.476	17.919±1.18(s+)	7.655±3.483	17.793±0.988(s+)
BreastCancer	131.609±159.83	364.751±203.236(s+)	164.384±155.722	444.99±208.967(s+)
HeartDisease	11.237±12.885	28.413±13.071(s+)	7.11±11.093	23.497±5.09(s+)

Fonte: Elaborado pelo Autor

5 CONCLUSÕES

Nos experimentos realizados neste trabalho foi analisado o impacto do número de gerações e taxa de mutação no AG do LORE. Melhores resultados são obtidos quando o número de gerações é pequeno e a taxa de mutação é alta. De fato, os autores em (GUIDOTTI, MONREALE, *et al.*, 2019) propõem o uso de AGs no LORE com poucas gerações (10) e com alta taxa de mutação (0,2). Ambas as estratégias são utilizadas para preservar a diversidade da população. Em vez disso, este trabalho usou o *fitness sharing* no LORE para obter conjuntos de dados artificiais mais diversos (compostos pelos indivíduos nas populações finais dos AGs).

Os resultados experimentais indicam que LORE com *fitness sharing* (LOREfs) produz conjuntos de dados artificiais com diversidade significativamente maior. Consequentemente, as superfícies de decisão criadas pelo modelo substituto (Árvore de Decisão) são mais semelhantes às superfícies de decisão locais gerados pelo modelo caixa-preta.

Nos experimentos com quatro conjuntos de dados de classificação e dois modelos caixa-preta (PMC e FA), resultados de F1-score significativamente melhores foram obtidos por LOREfs quando comparados a LORE.

Por fim, os experimentos e resultados deste trabalho resultaram na publicação de um artigo nos anais da conferência 2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI) (SANTOS, BARANAUSKAS e TINÓS, 2021).

5.1 TRABALHOS FUTUROS

Nos experimentos deste trabalho foram testados apenas conjuntos de dados com atributos reais. No futuro, o LOREfs pode ser adaptado para problemas com atributos categóricos.

Quando comparado ao LORE, o LOREfs é particularmente interessante para conjuntos de dados com mais atributos. O desempenho de LOREfs em problemas com centenas ou milhares de características deve ser investigado futuramente, assim como a investigação de problemas de classificação que não sejam binários.

O impacto do uso de diferentes métricas de distância para calcular a função de compartilhamento deve ser investigado. O uso de operadores eficientes de recombinação e de mutação também é um trabalho futuro adicional, além de alterar outras estratégias utilizadas no LORE, por exemplo, o procedimento de inicialização da população.

6 REFERÊNCIAS

ABADI, M. et al. TensorFlow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16). 2016. Disponivel em: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>.

BACK, T.; FOGEL, D. B.; MICHALEWICZ, Z. Evolutionary Computation 1: Basic Algorithms and Operators. Institute Of Physics Publishing, 2000.

BEASLEY, D.; BULL, D. R.; MARTIN, R. R. A Sequential Niche Technique for Multimodal Function Optimization. Evolutionary Computation, v. 1, n. 2, p. 101-125, June 1993.

COUNCIL OF EUROPEAN UNION. Council regulation (EU) no 279/2016. Official website of the European Union, 2016. Disponivel em: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>. Acesso em: 15 maio 2020.

DARPA. Explainable Artificial Intelligence (XAI). Defense Advanced Research Projects Agency, 2016. Disponivel em: <https://www.darpa.mil/attachments/DARPA-BAA-16-53.pdf>.

DEJONG, K. A. “An analysis of the behavior of a class of genetica daptative systems”. Ph.D. dissertation, Univ. of Michigan, Ann Arbor. 1975.

DOSHI-VELEZ, F.; KIM, B. Towards A Rigorous Science of Interpretable Machine Learning, 2017. Disponivel em: <https://arxiv.org/abs/1702.08608>.

DUA, D.; GRAFF, C. UCI Machine Learning Repository, 2017. Disponivel em: <http://archive.ics.uci.edu/ml>.

EIBEN, A. E.; SMITH, J. E. Introduction to Evolutionary Computing. 2ª. ed.: Springer Publishing Company, Incorporated, 2015.

EL SHAWI, R. et al. Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. Computational Intelligence, v. 37, n. 4, p. 1633-1650, 2021.

FORTIN, F.-A. et al. DEAP: Evolutionary Algorithms Made Easy. Journal of Machine Learning Research, v. 13, n. 1, p. 2171-2175, 2012.

GARTNER. Gartner Survey Shows 37 Percent of Organizations Have Implemented AI in Some Form. Gartner, 21 jan. 2019. Disponivel em: <https://www.gartner.com/en/newsroom/press-releases/2019-01-21-gartner-survey-shows-37-percent-of-organizations-have>. Acesso em: 14 abr. 2020.

GOLDBERG, D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. New York: Addison-Wesley, 1989.

GOLDBERG, D. E.; RICHARDSON, J. Genetic Algorithms with Sharing for Multimodal Function Optimization. Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application. Cambridge: L. Erlbaum Associates Inc. 1987. p. 41-49.

GUIDOTTI, R. et al. A Survey Of Methods For Explaining Black Box Models. ACM computing surveys (CSUR), v. 51, n. 5, p. 1-42, 2018.

GUIDOTTI, R. et al. Factual and counterfactual explanations for black box decision making. IEEE Intelligent Systems, v. 34, n. 6, p. 14-23, 2019.

HOLLAND, J. H. Adaptation in Natural and Artificial Systems. Ann Arbor: Univ. of Michigan Press, 1975.

LAKHANI, P. et al. Machine Learning in Radiology: Applications Beyond Image Interpretation. Journal of the American College of Radiology, v. 15, n. 2, p. 350–359, 2018.

MAHFOUD, S. W. “Niching methods for genetic algorithms,”. Ph.D. dissertation, Univ. of Illinois, Urbana-Champaign. 1995.

MALLAWAARACHCHI,. Introduction to Genetic Algorithms — Including Example Code. Towards Data Science, 2017. Disponivel em: <https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3>. Acesso em: 21 abr. 2020.

MILLER, T. Explanation in Artificial Intelligence: Insights from the Social Sciences. CoRR, abs/1706.07269, 2018. Disponivel em: <http://arxiv.org/abs/1706.07269>. Acesso em: 10 dez. 2020.

MITCHELL, M. An Introduction to Genetic Algorithms. Cambridge: MIT Press, 1998.

MITCHELL, T. M. Machine Learning. McGraw-Hill Science/Engineering/Math, 1997.

MITTELSTADT , B. D.; RUSSELL , ; WACHTER, S. Explaining Explanations in AI, 2018. Disponivel em: <https://arxiv.org/abs/1811.01439>.

RAJPURKAR, P. et al. CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning, 2017.

RIBEIRO, M. T. LIME - Local Interpretable Model-Agnostic Explanations – Marco Tulio Ribeiro, 2016. Disponivel em: <https://homes.cs.washington.edu/~marcotcr/blog/lime/>. Acesso em: nov. 2022.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. 2016. p. 1135-1144.

ROSENBLATT, F. The Perceptron: A Probabilistic Model For Information Storage And Organization. Psychological Review, 65, 1958.

SANTOS, D. A.; BARANAUSKAS, J. A.; TINÓS, R. Use of Fitness Sharing in the Local Rule-Based Explanations Method. 2021 IEEE Latin American Conference on Computational Intelligence (LA-CCI). 2021. p. 1-6.

THE new AI innovation equation. IBM. Disponivel em: <https://www.ibm.com/watson/advantage-reports/future-of-artificial-intelligence/ai-innovation-equation.html>. Acesso em: 14 abr. 2020.

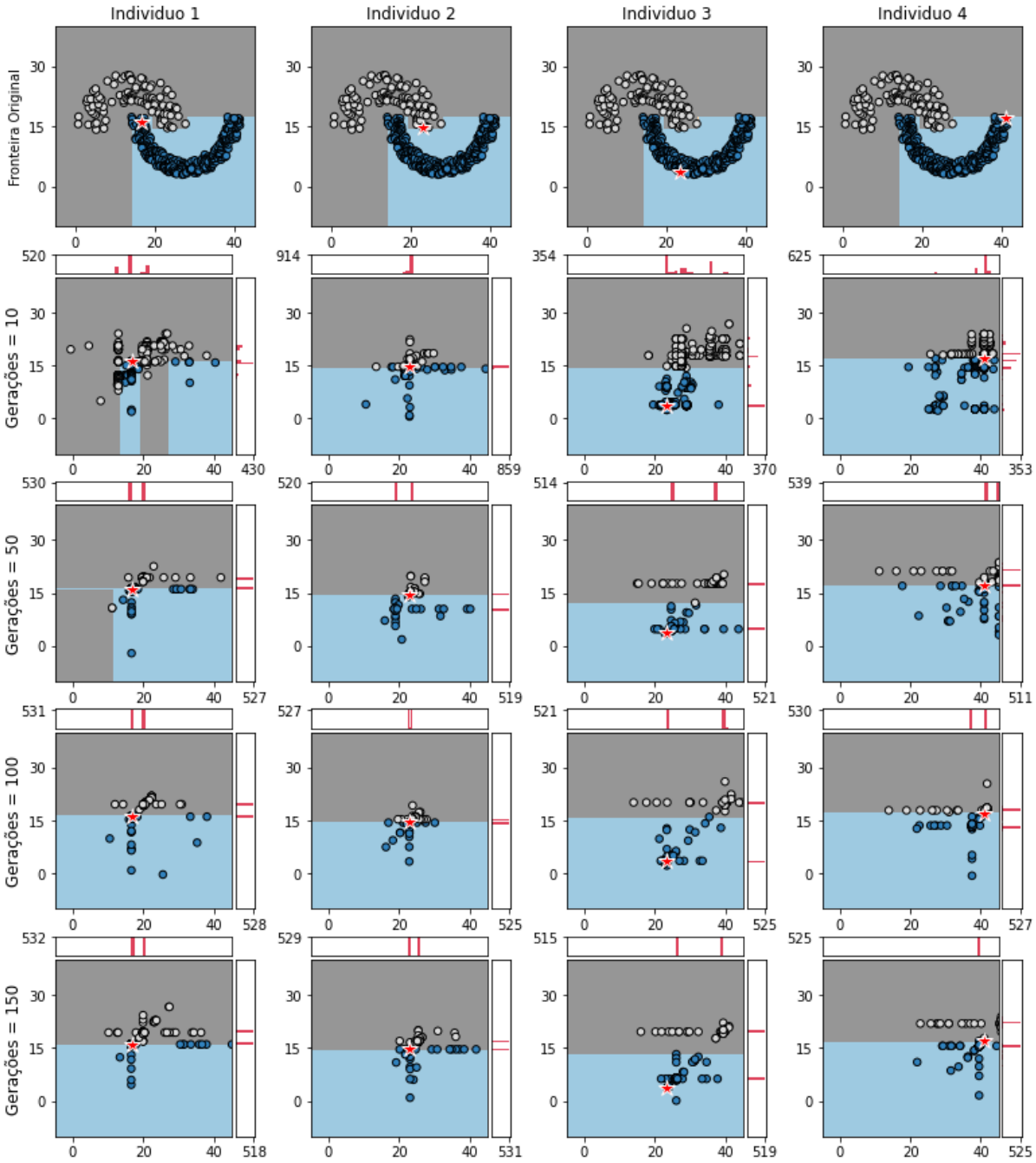
A APÊNDICE

A.1 EXPERIMENTOS LORE COM CONJUNTO DE DADOS *JAIN*

A.1.1 *EXPERIMENTOS COM MODELO CAIXA-PRETA FA*

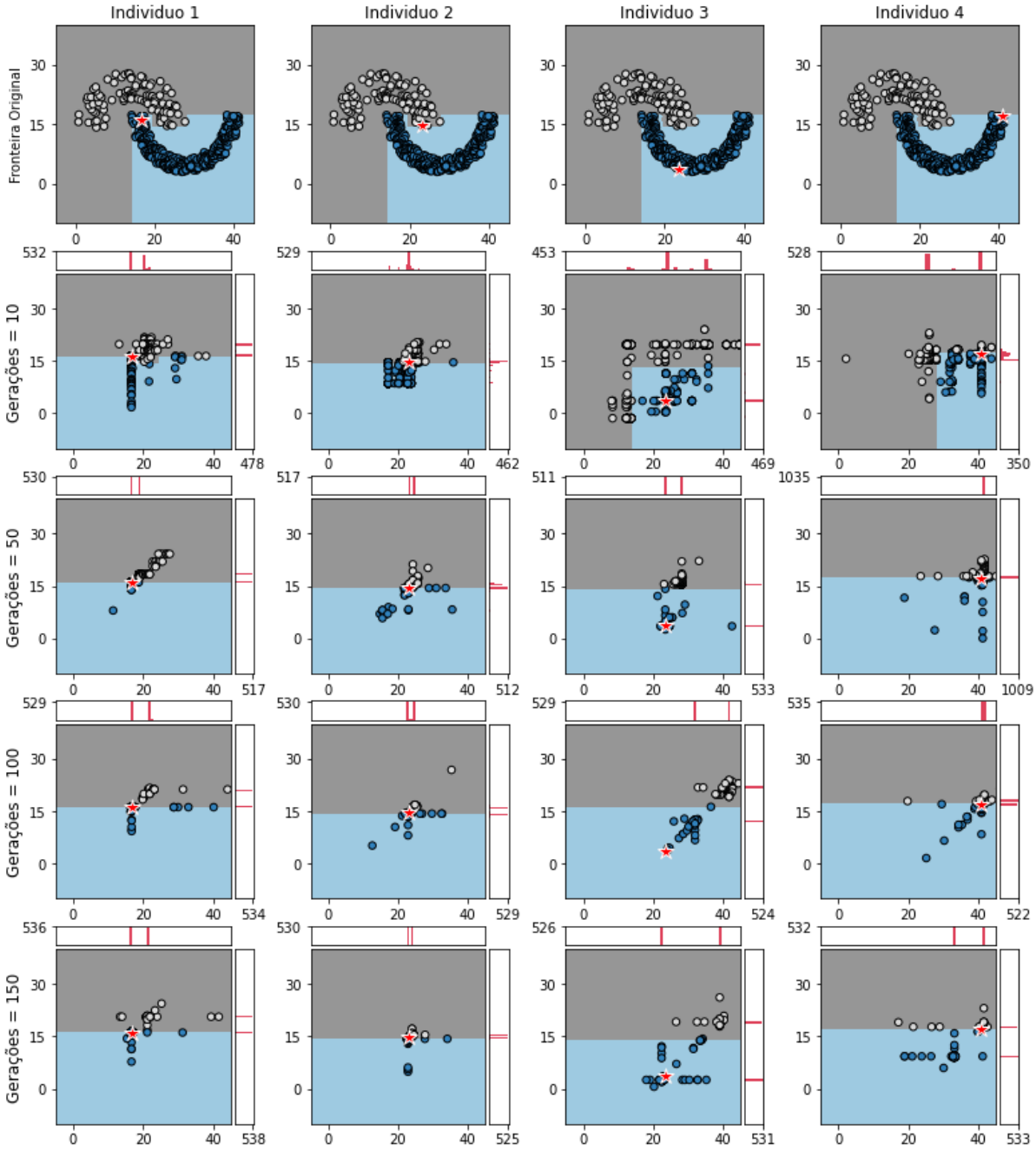
A.1.1.1 *Experimentos Alteração do Número de Gerações*

Figura 23 – Experimento 3: LORE - Alteração do Número de Gerações, *mutpb* = 0.15, FA, Conjunto de Dados = *Jain*



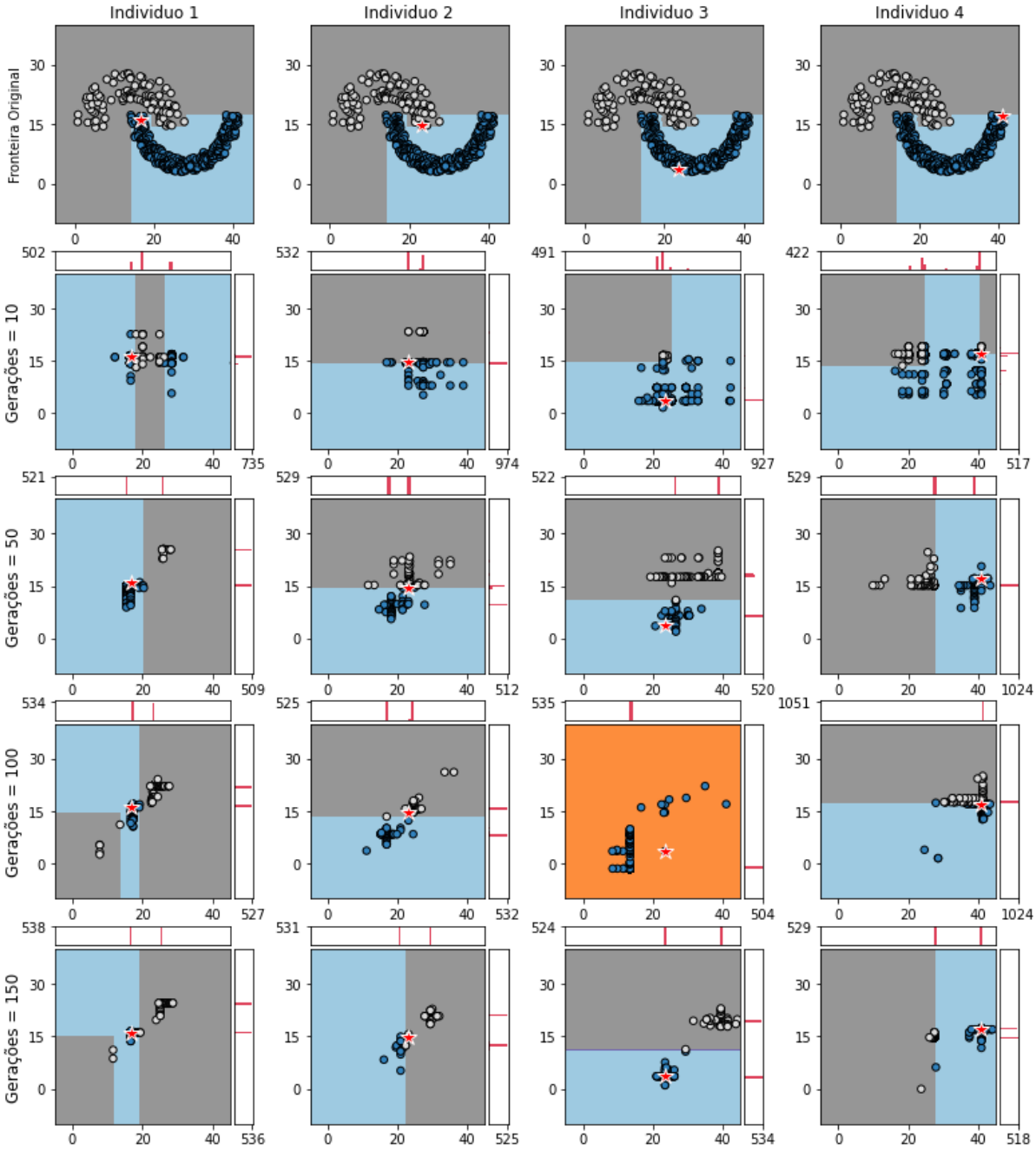
Fonte: Elaborado pelo autor.

Figura 24 – Experimento 4: LORE - Alteração do Número de Gerações, *mutpb* = 0.10, FA, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

Figura 25 – Experimento 5: LORE - Alteração do Número de Gerações, *mutpb* = 0.05, FA, Conjunto de Dados = *Jain*

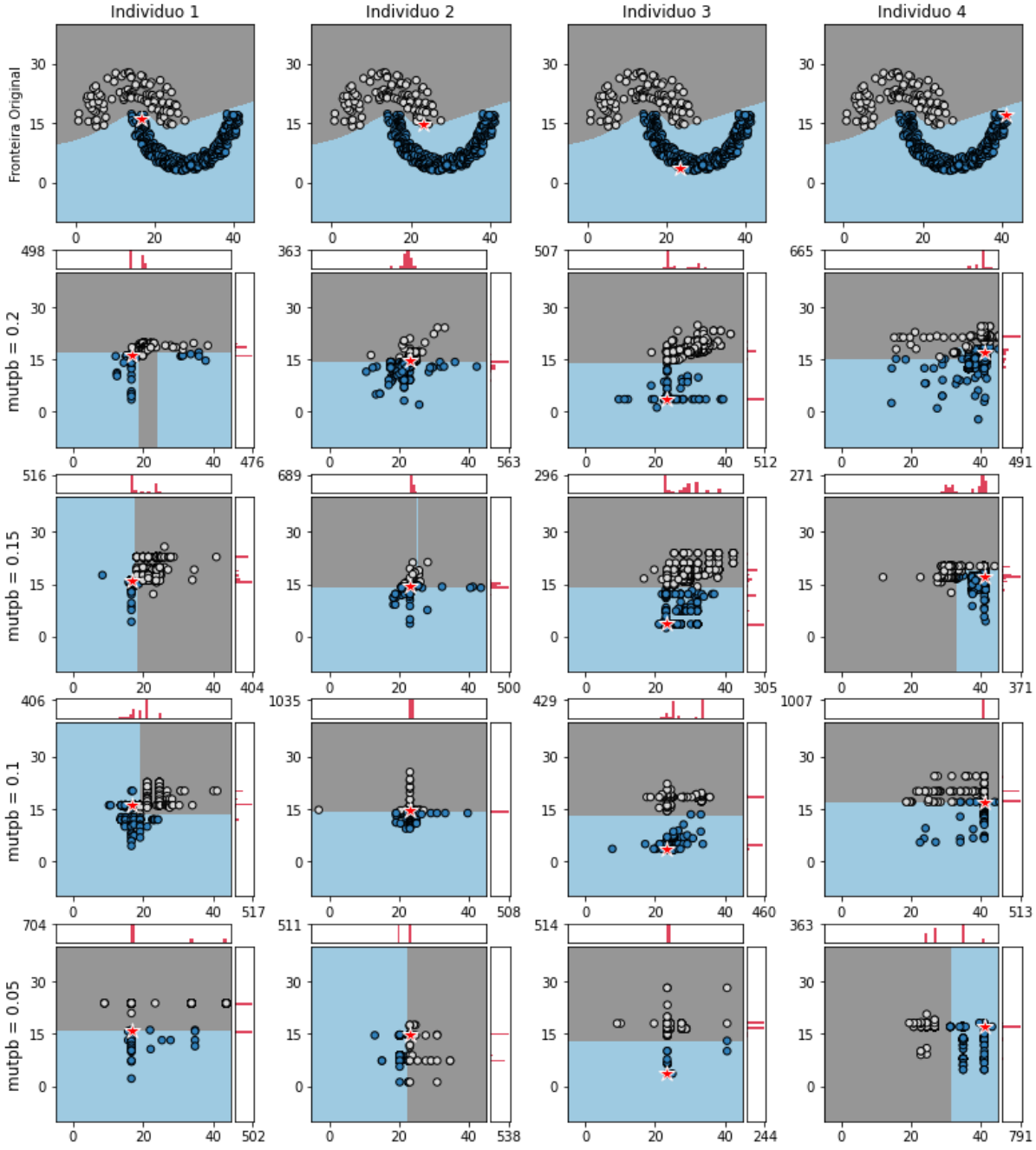


Fonte: Elaborado pelo autor.

A.1.2 *EXPERIMENTOS COM MODELO CAIXA-PRETA PMC*

A.1.2.1 *Experimentos Alteração da Taxa de Mutação*

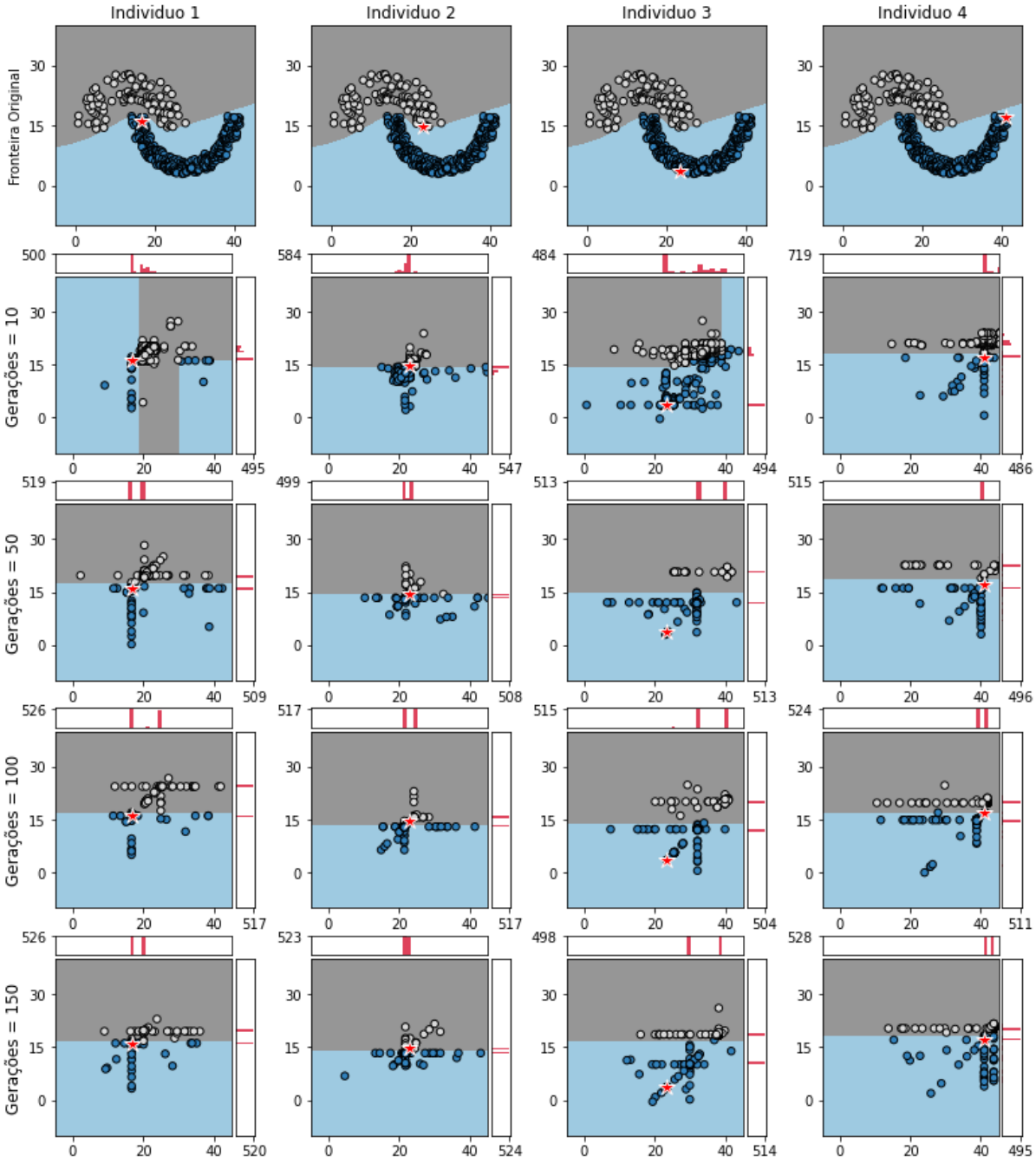
Figura 26 – Experimento 6: LORE - Alteração da Taxa de Mutação, Gerações = 10, PMC, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

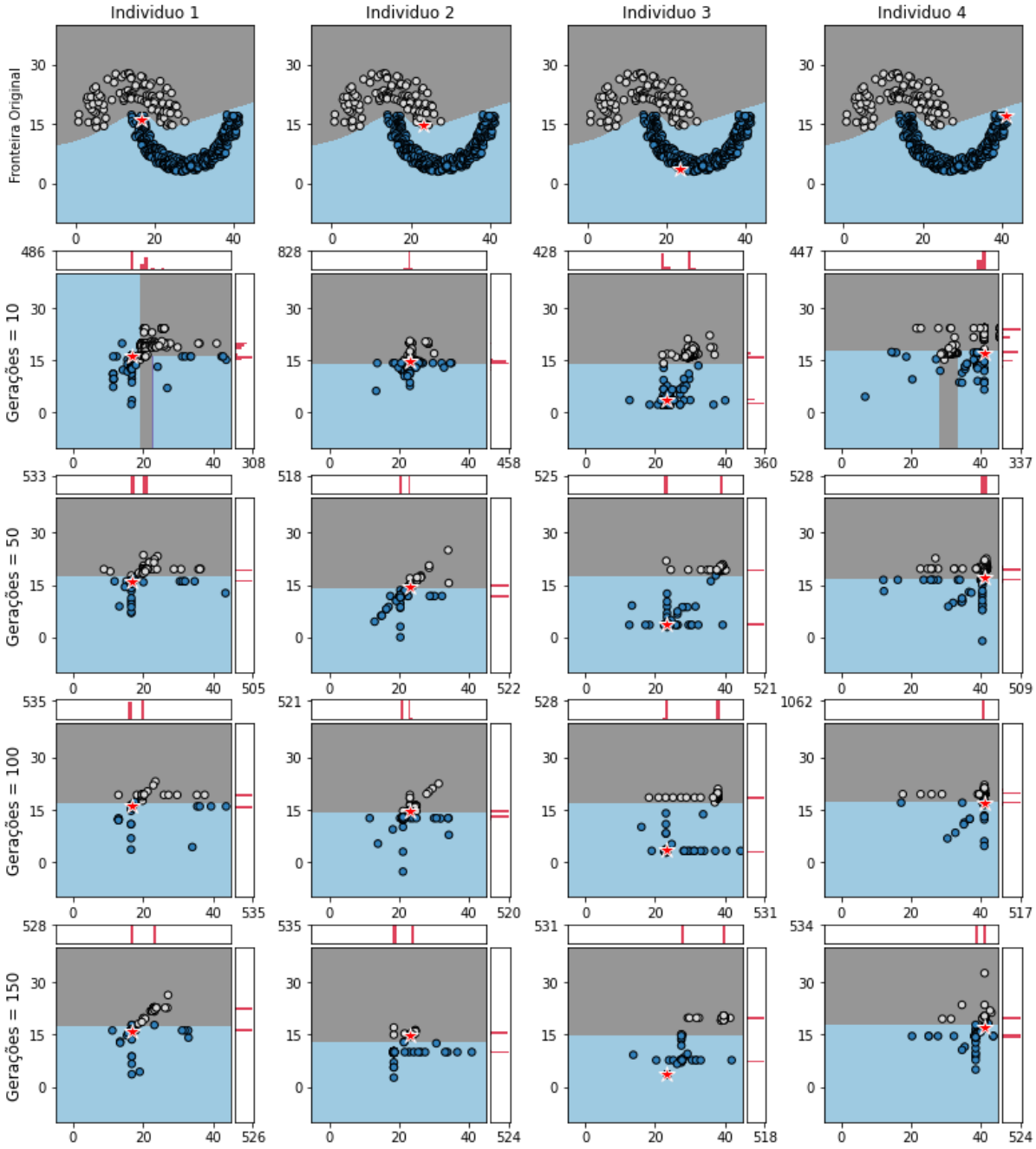
A.1.2.2 *Experimentos Alteração do Número de Gerações*

Figura 27 – Experimento 7: LORE - Alteração do Número de Gerações, *mutpb* = 0.20, PMC, Conjunto de Dados = *Jain*



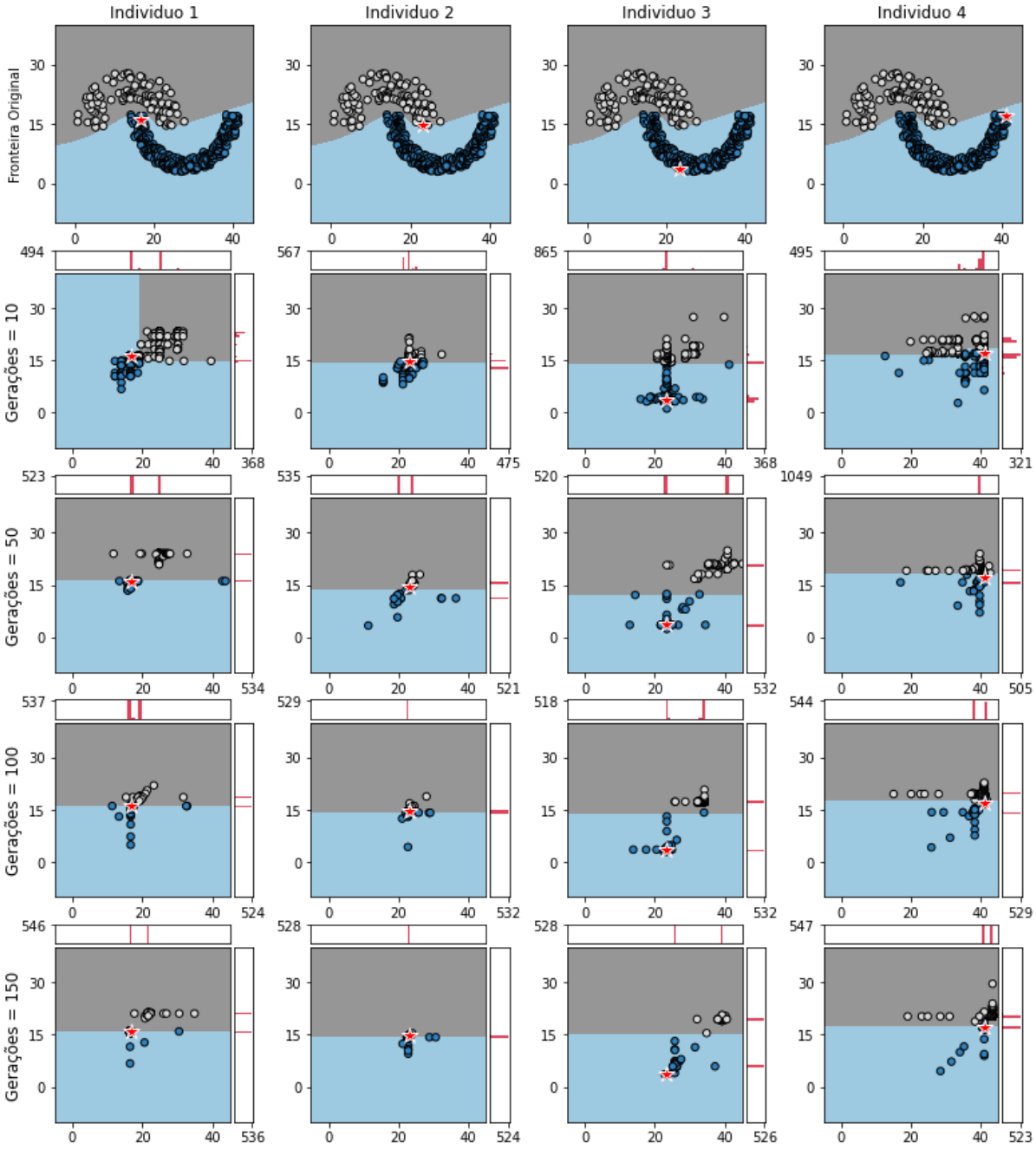
Fonte: Elaborado pelo autor.

Figura 28 – Experimento 8: LORE - Alteração do Número de Gerações, *mutpb* = 0.15, PMC, Conjunto de Dados = *Jain*



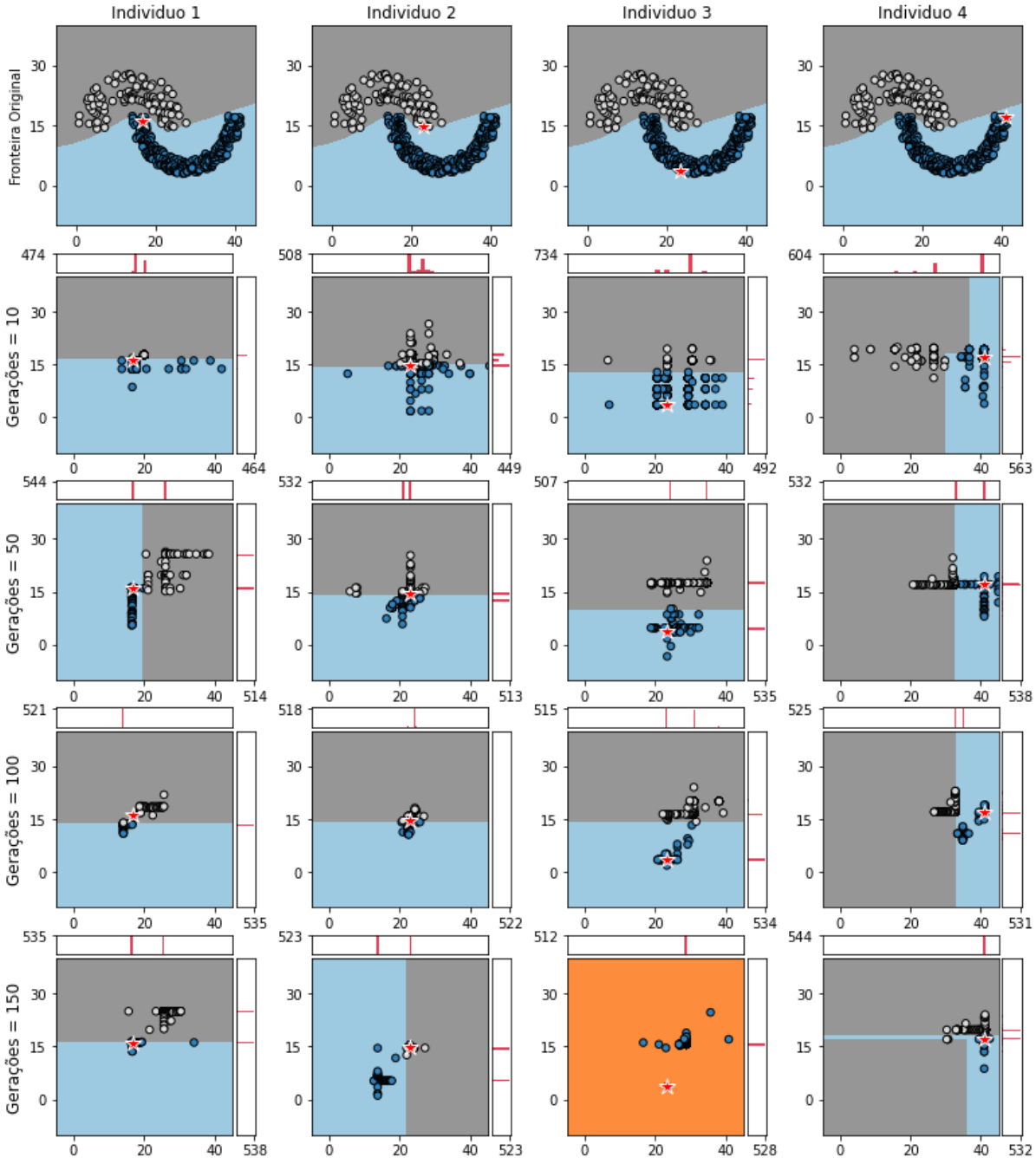
Fonte: Elaborado pelo autor.

Figura 29 – Experimento 9: LORE - Alteração do Número de Gerações, *mutpb* = 0.10, PMC, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

Figura 30 – Experimento 10: LORE - Alteração do Número de Gerações, *mutpb* = 0.05, PMC, Conjunto de Dados = *Jain*



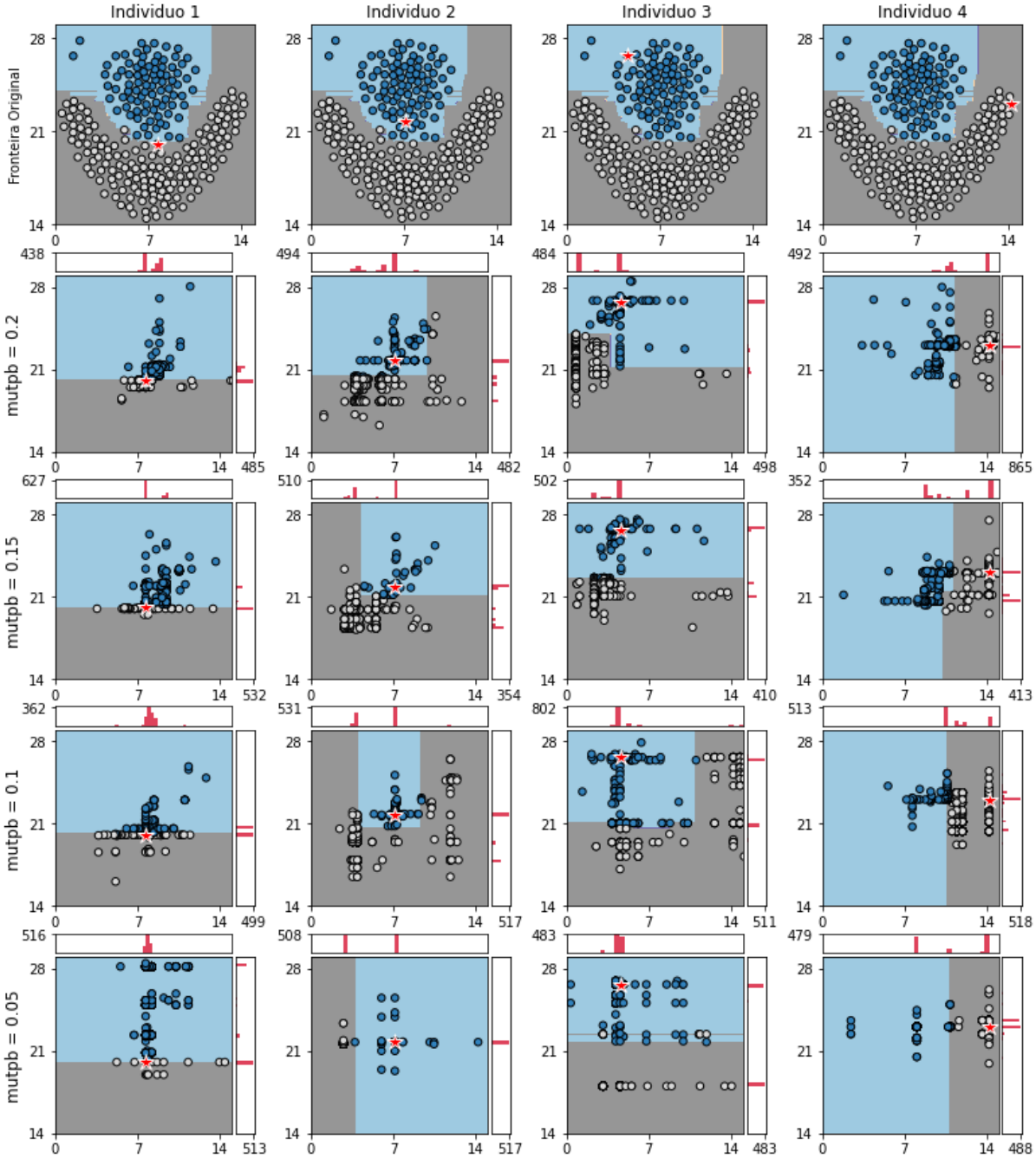
Fonte: Elaborado pelo autor.

A.2 EXPERIMENTOS LORE COM CONJUNTO DE DADOS *FLAME*

A.2.1 *EXPERIMENTOS COM MODELO CAIXA-PRETA FA*

A.2.1.1 *Experimentos Alteração da Taxa de Mutação*

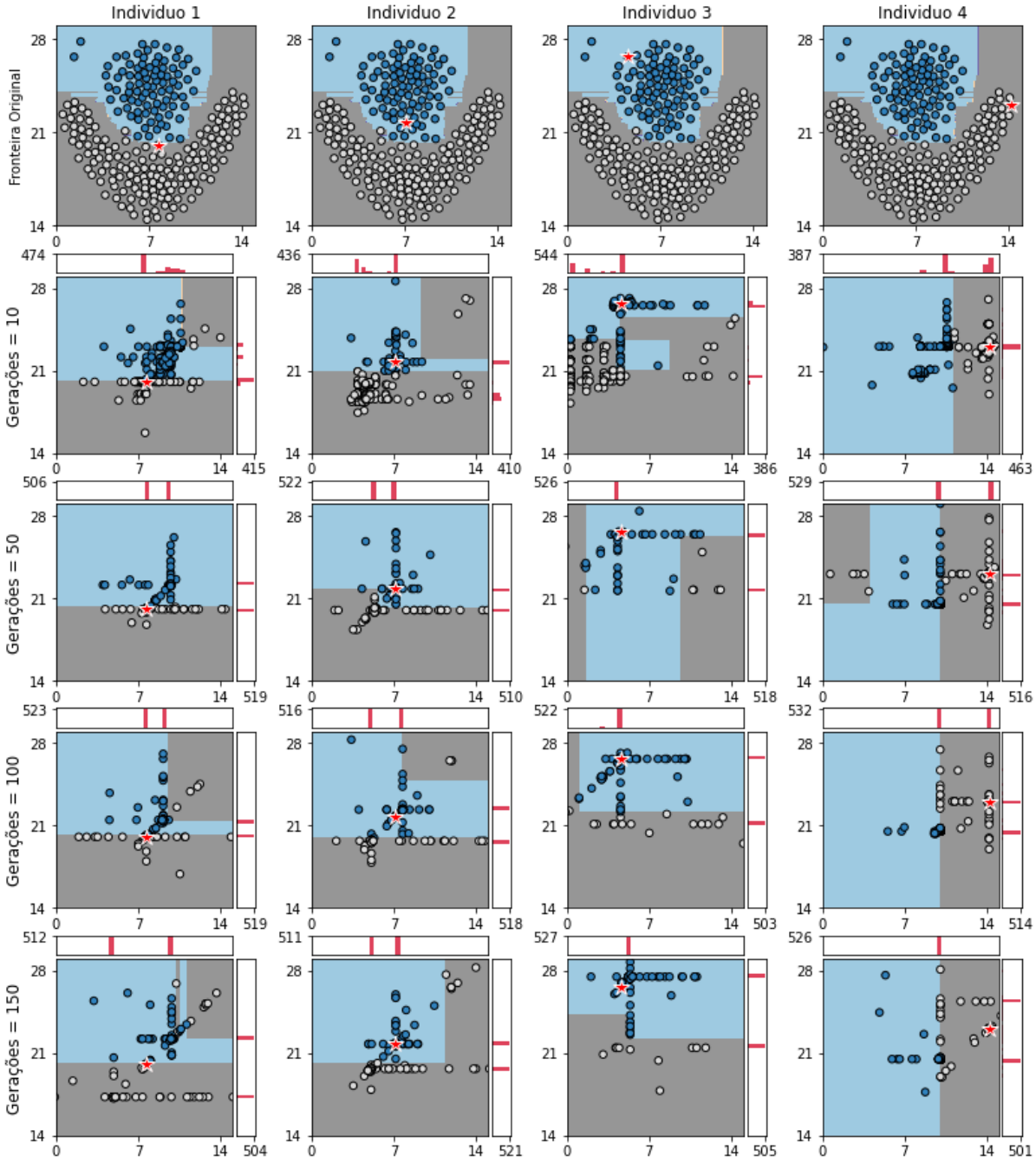
Figura 31 – Experimento 11: LORE - Alteração da Taxa de Mutação, Gerações = 10, FA, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

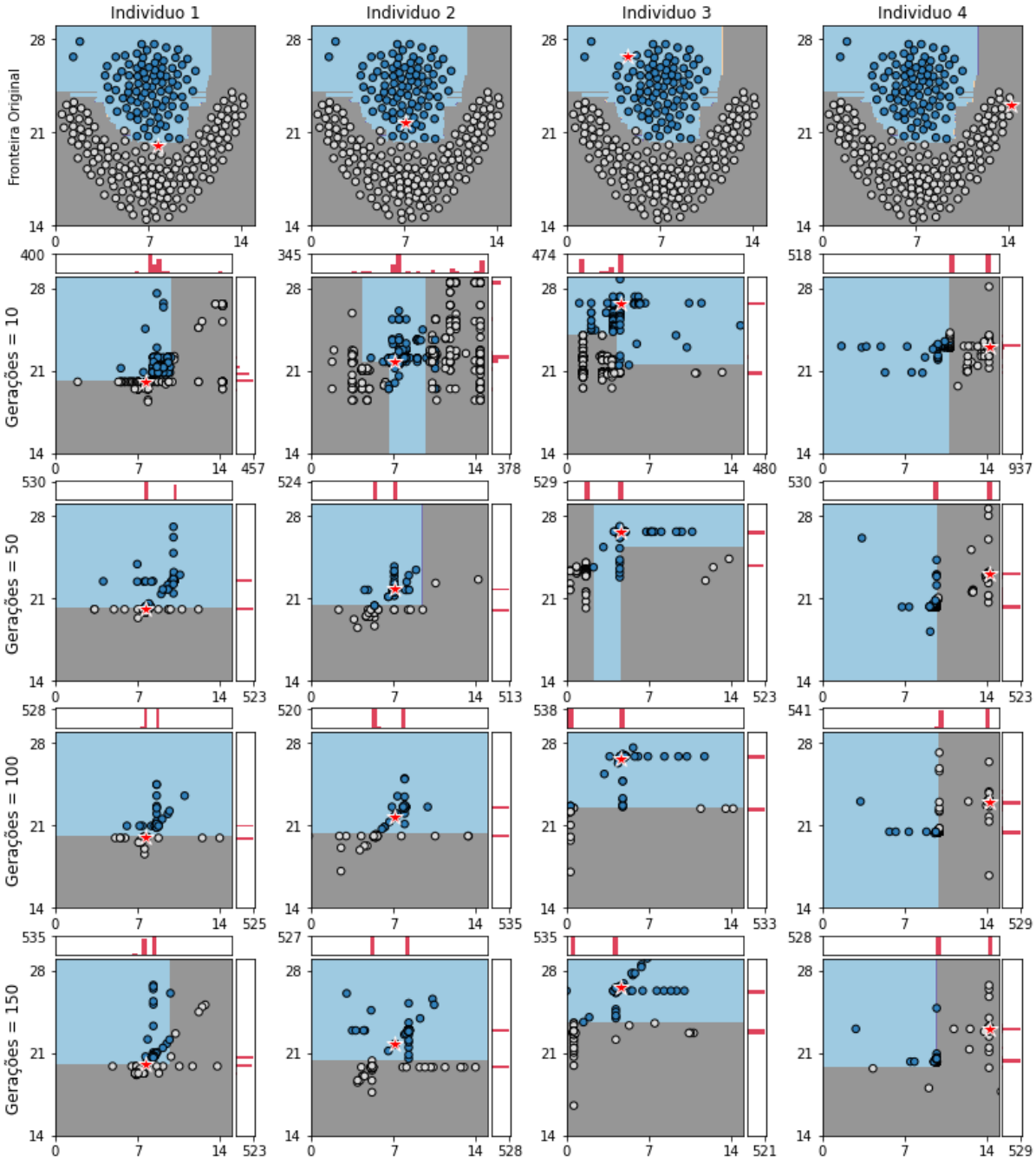
A.2.1.2 *Experimentos Alteração do Número de Gerações*

Figura 32 – Experimento 12: LORE - Alteração do Número de Gerações, *mutpb* = 0.20, FA, Conjunto de Dados = *Flame*



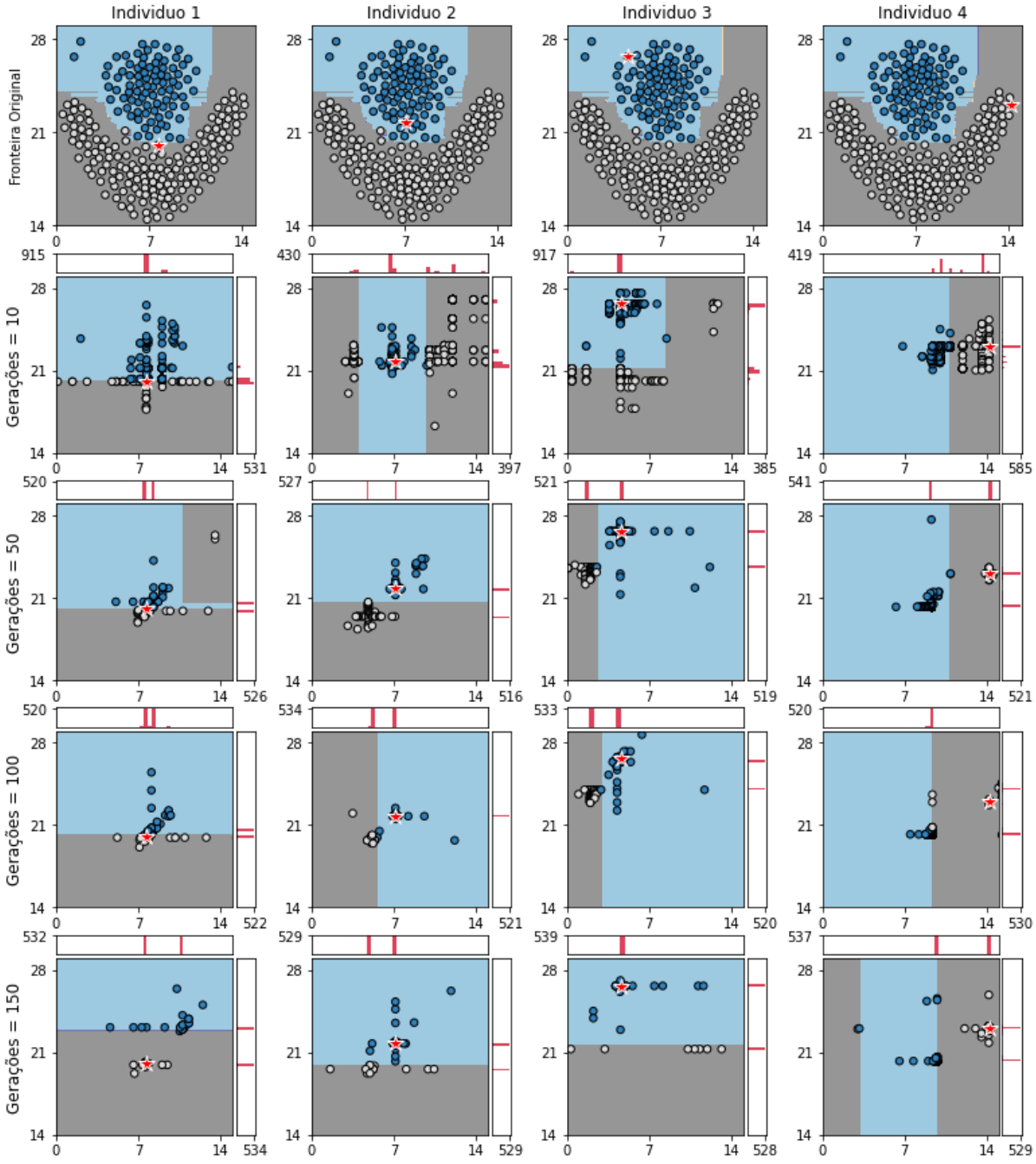
Fonte: Elaborado pelo autor.

Figura 33 – Experimento 13: LORE - Alteração do Número de Gerações, *mutpb* = 0.15, FA, Conjunto de Dados = *Flame*



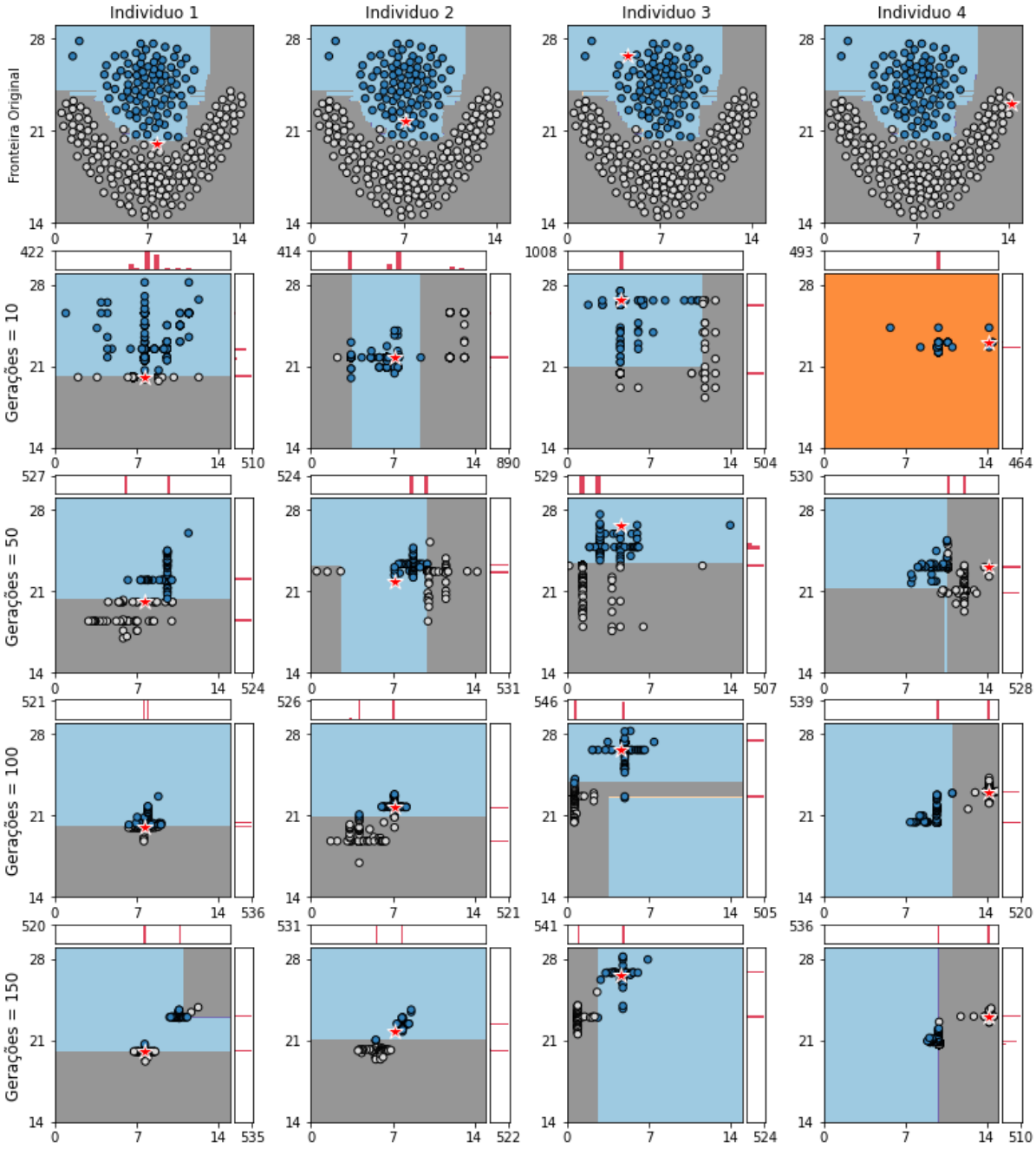
Fonte: Elaborado pelo autor.

Figura 34 – Experimento 14: LORE - Alteração do Número de Gerações, *mutpb* = 0.10, FA, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

Figura 35 – Experimento 15: LORE - Alteração do Número de Gerações, *mutpb* = 0.05, FA, Conjunto de Dados = *Flame*

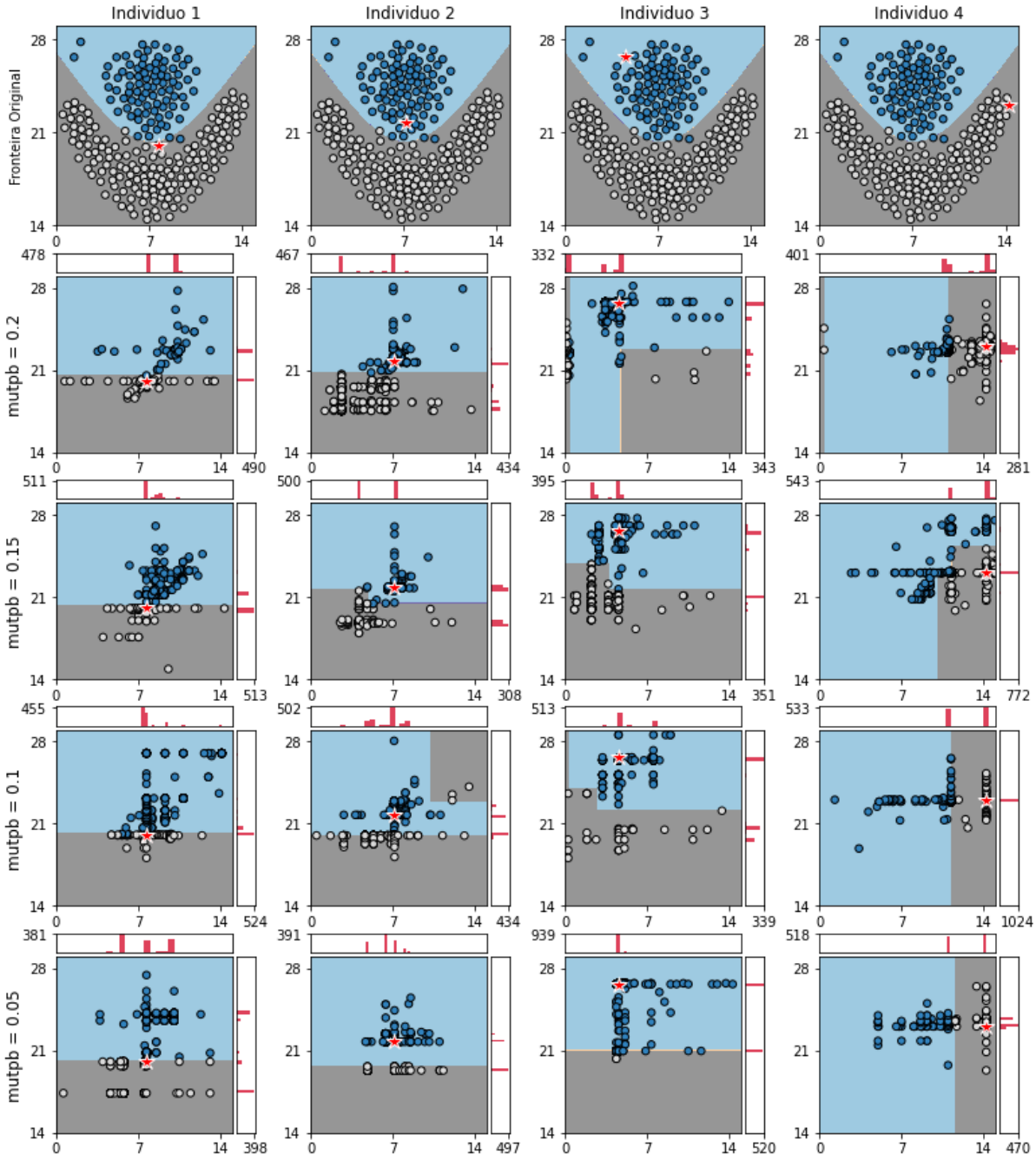


Fonte: Elaborado pelo autor.

A.2.2 *EXPERIMENTOS COM MODELO CAIXA-PRETA PMC*

A.2.2.1 *Experimentos Alteração da Taxa de Mutação*

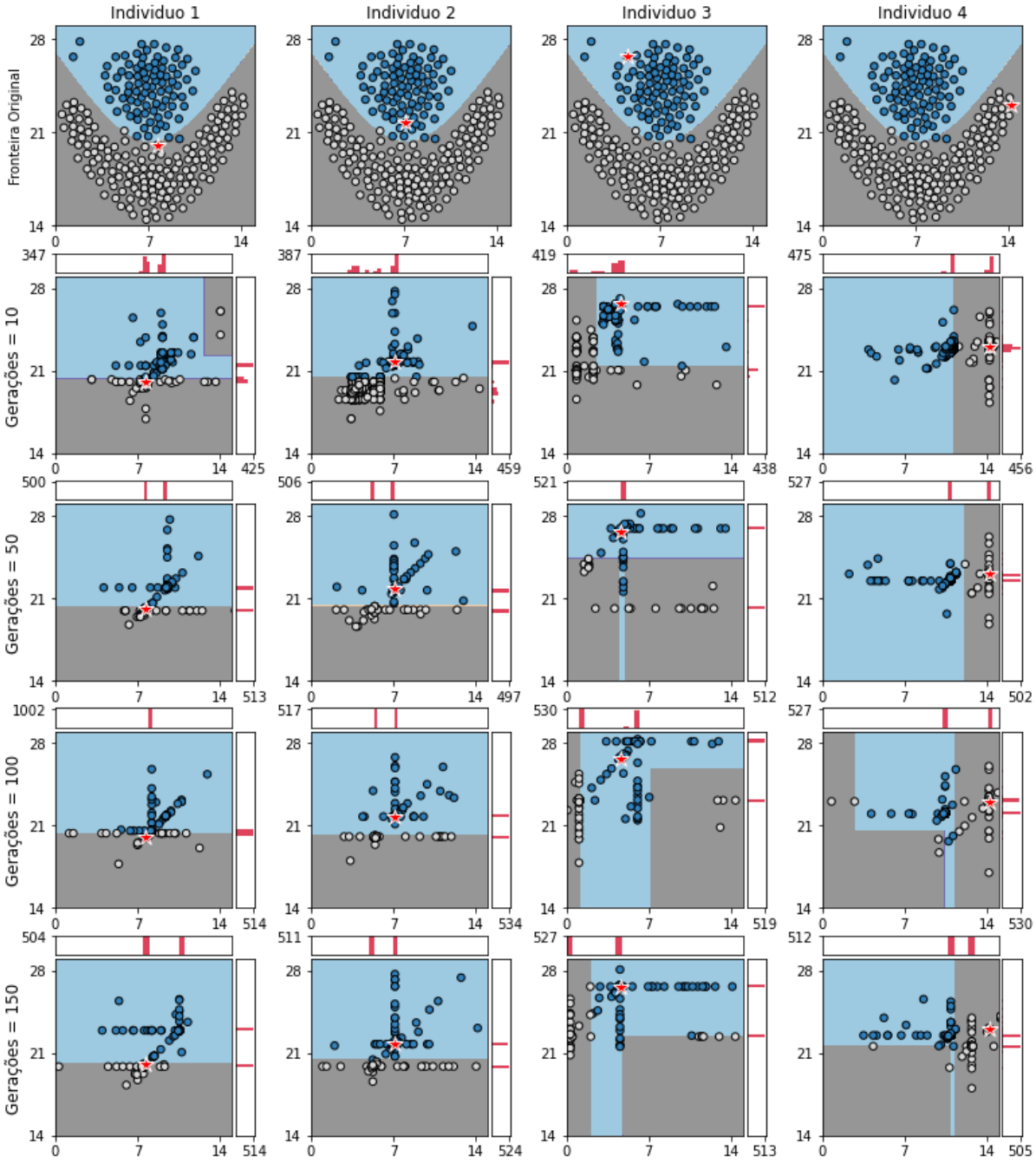
Figura 36 – Experimento 16: LORE - Alteração da Taxa de Mutação, Gerações = 10, PMC, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

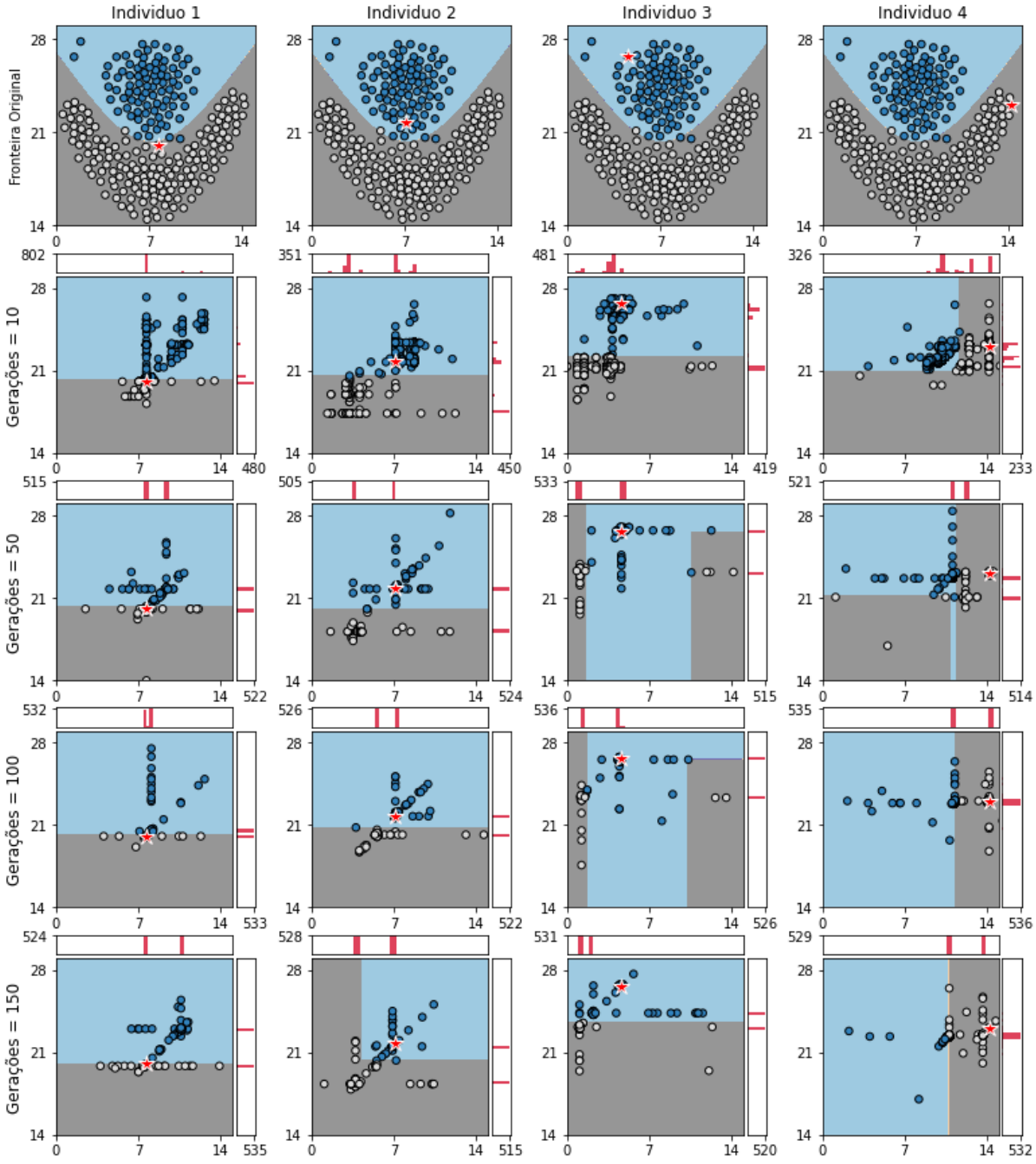
A.2.2.2 *Experimentos Alteração do Número de Gerações*

Figura 37 – Experimento 17: LORE - Alteração do Número de Gerações, *mutpb* = 0.20, PMC, Conjunto de Dados = *Flame*



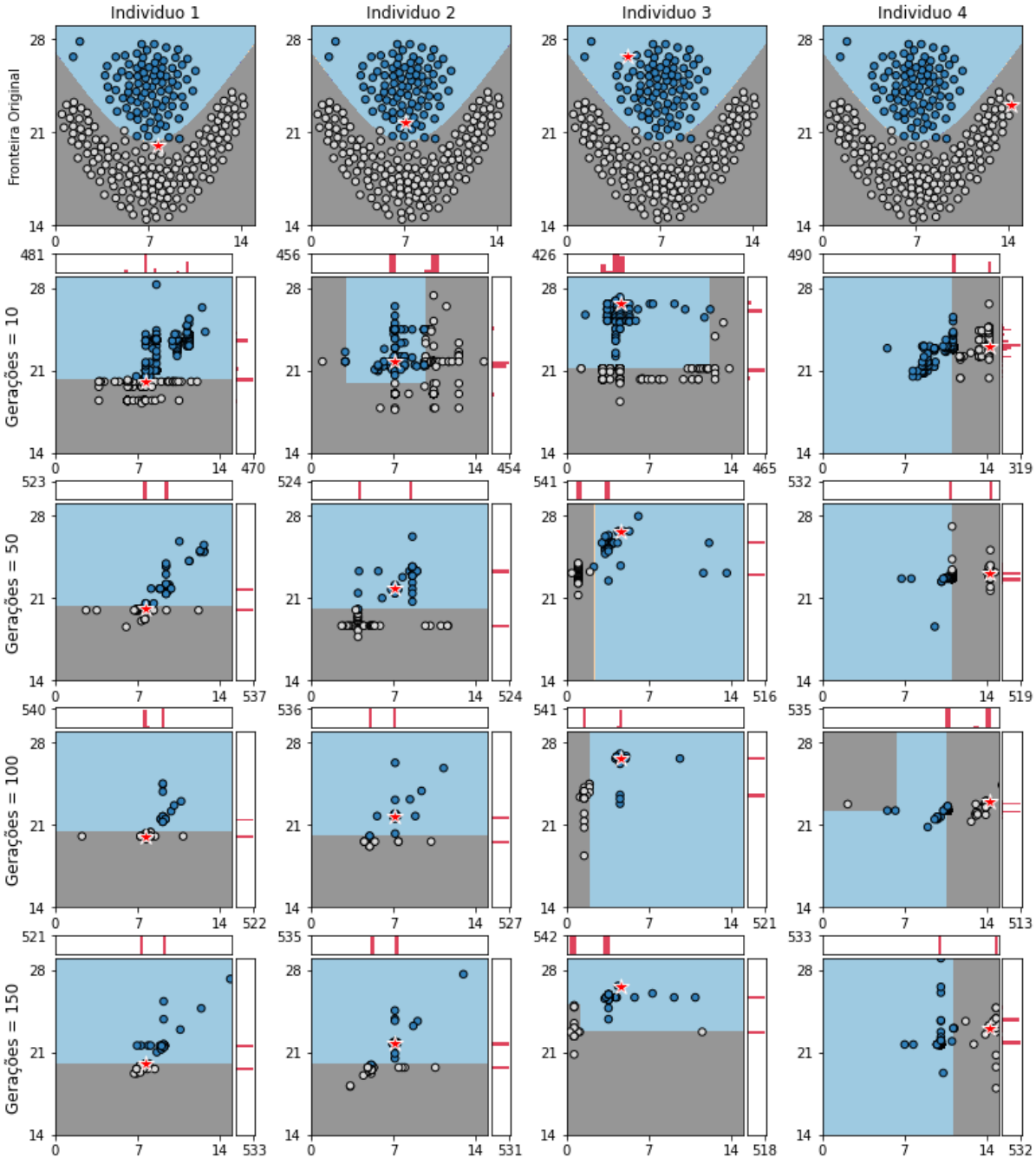
Fonte: Elaborado pelo autor.

Figura 38 – Experimento 18: LORE - Alteração do Número de Gerações, *mutpb* = 0.15, PMC, Conjunto de Dados = *Flame*



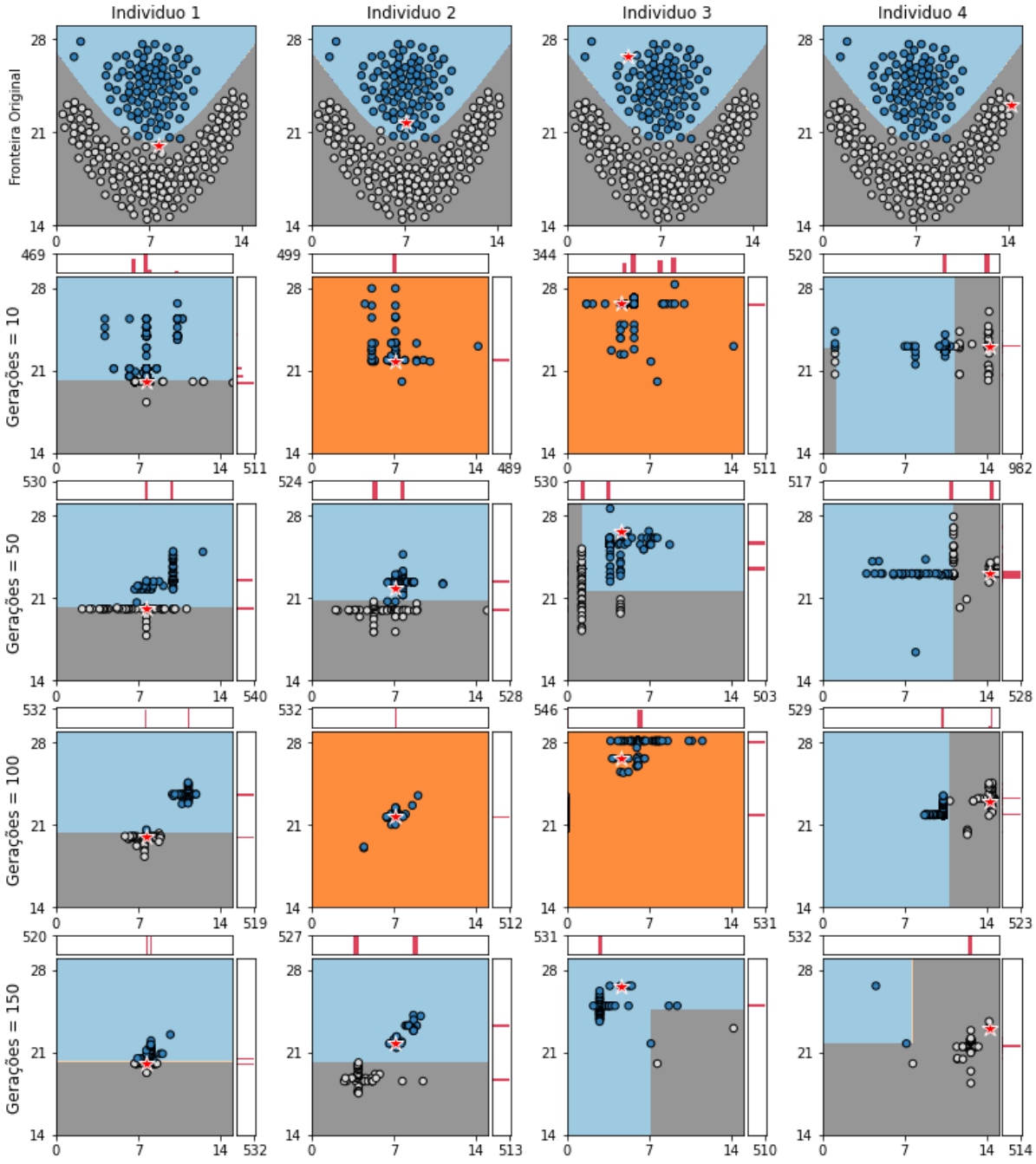
Fonte: Elaborado pelo autor.

Figura 39 – Experimento 19: LORE - Alteração do Número de Gerações, *mutpb* = 0.10, PMC, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

Figura 40 – Experimento 20: LORE - Alteração do Número de Gerações, *mutpb* = 0.05, PMC, Conjunto de Dados = *Flame*



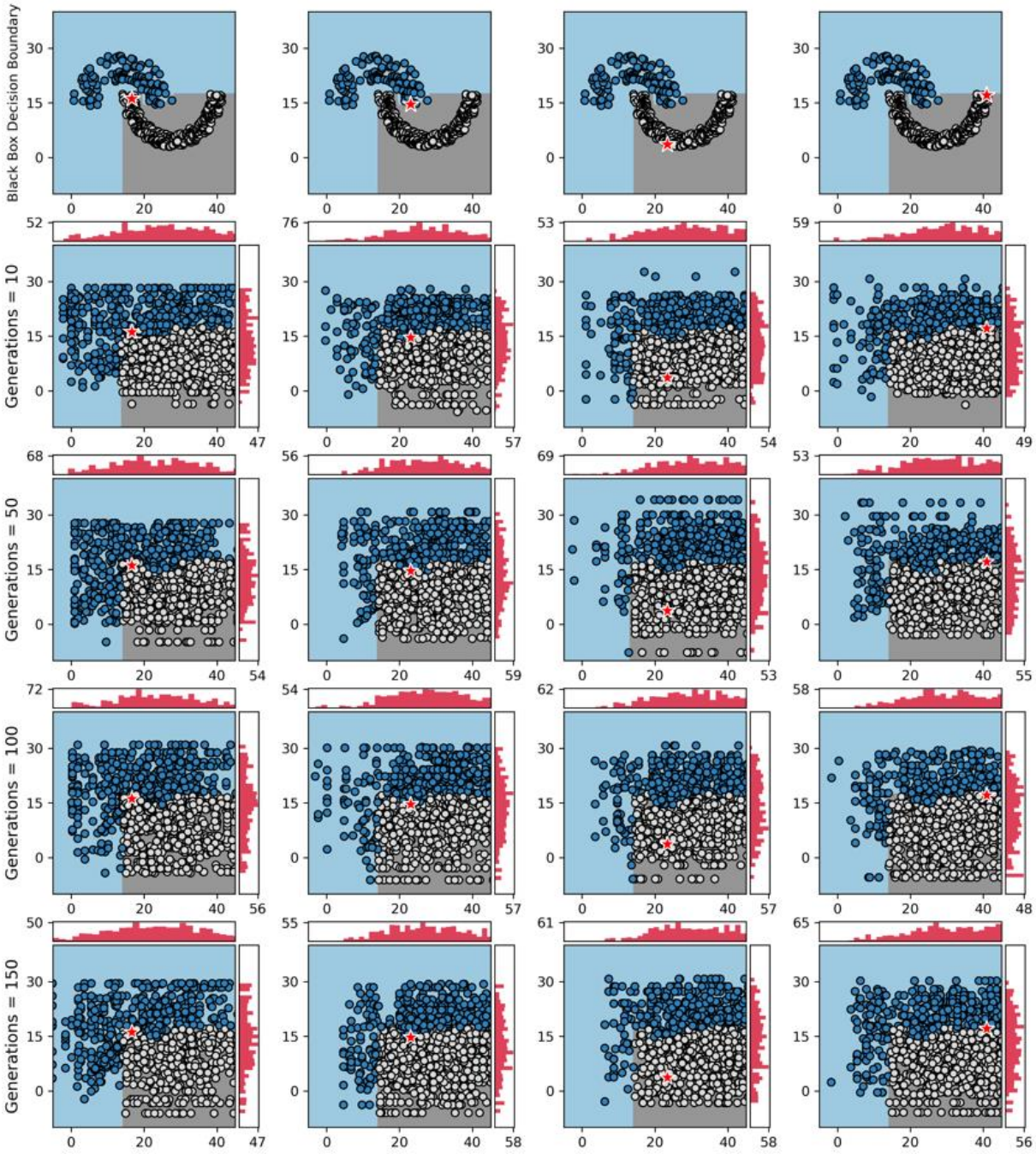
Fonte: Elaborado pelo autor.

A.3 EXPERIMENTOS LOREFS COM CONJUNTO DE DADOS *JAIN*

A.3.1 *EXPERIMENTOS COM MODELO CAIXA-PRETA FA*

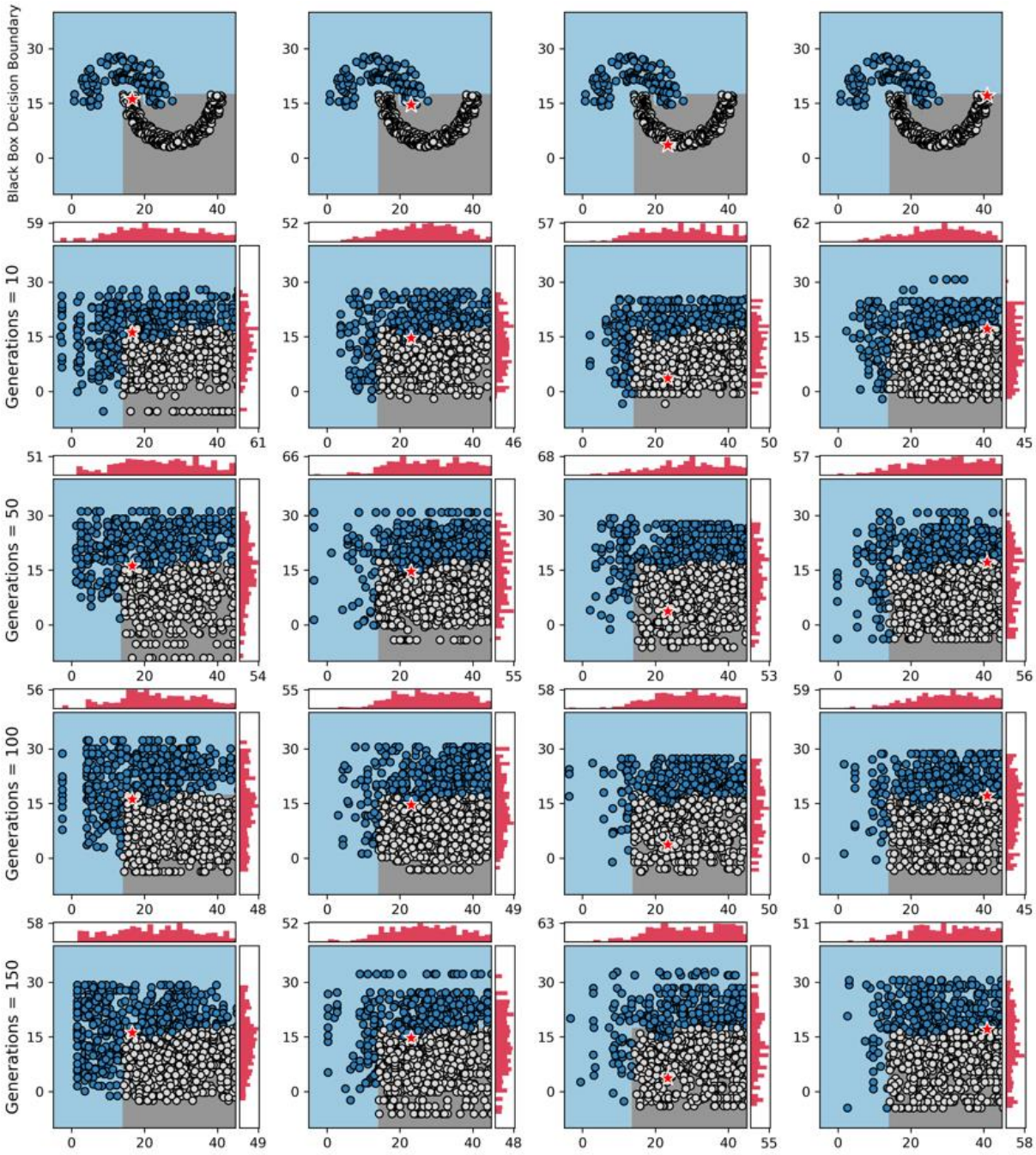
A.3.1.1 *Experimentos Alteração do Número de Gerações*

Figura 41 – Experimento 23: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.15, FA, Conjunto de Dados = *Jain*



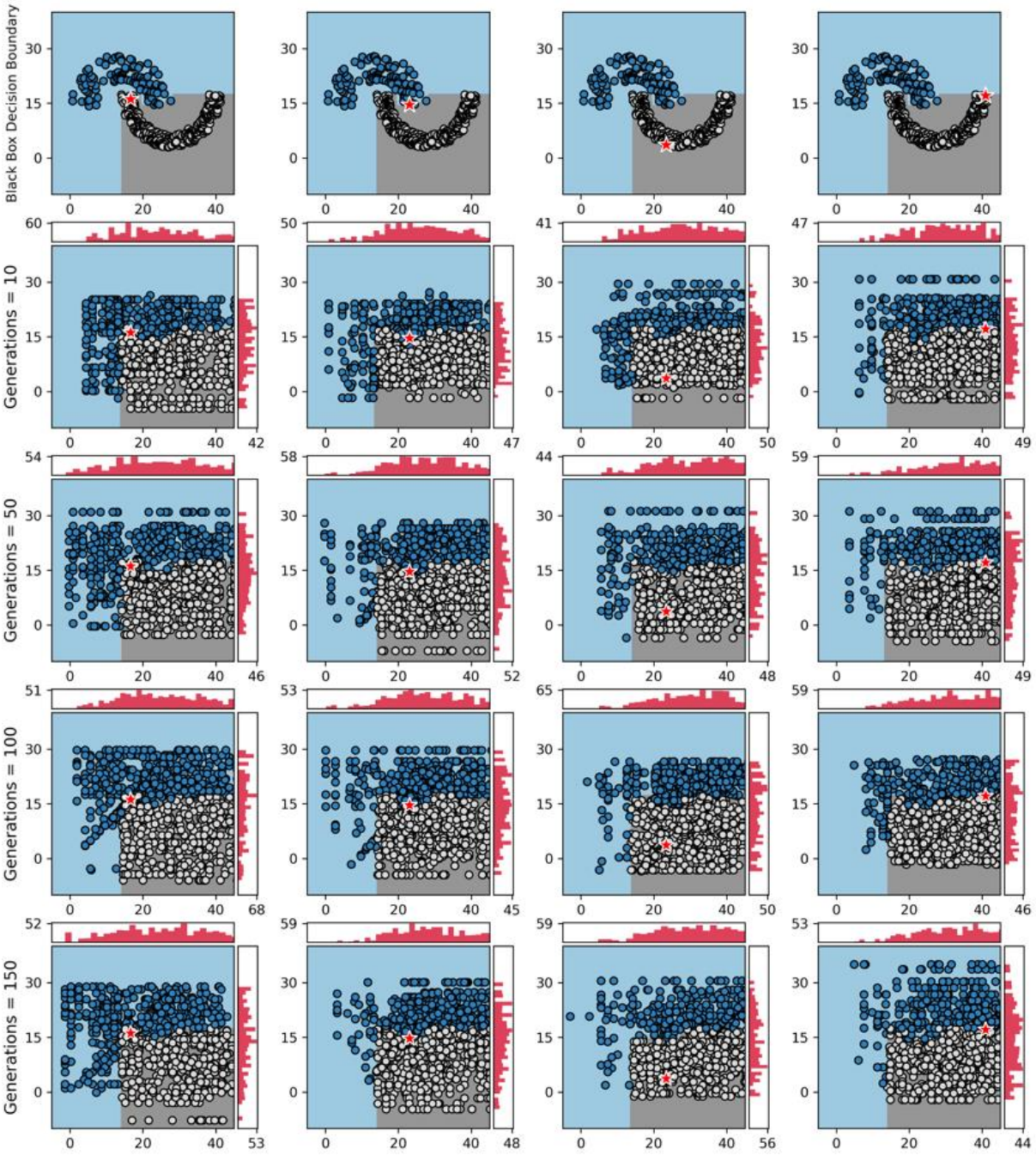
Fonte: Elaborado pelo autor.

Figura 42 – Experimento 24: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.10, FA, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

Figura 43 – Experimento 25: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.05, FA, Conjunto de Dados = *Jain*

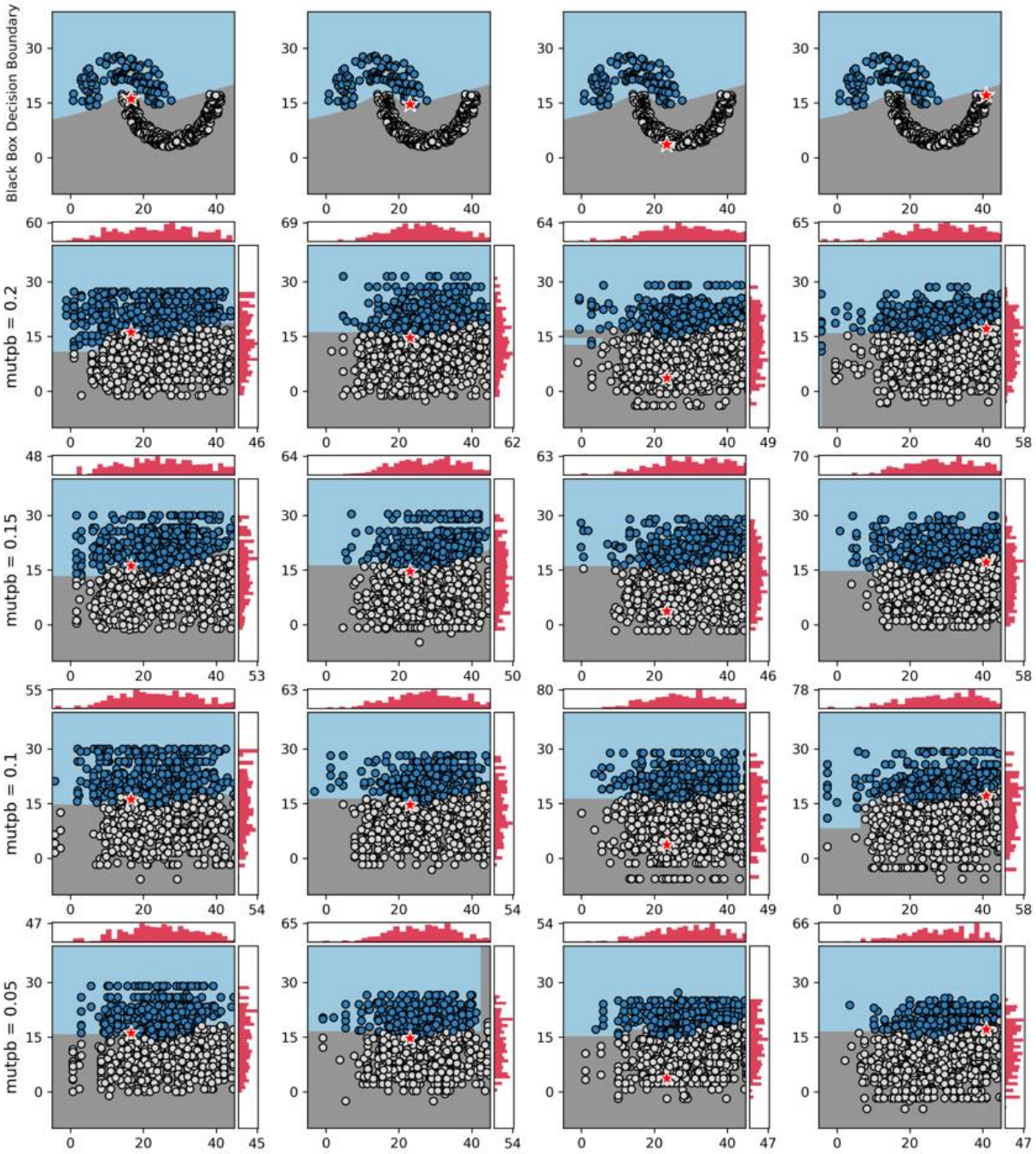


Fonte: Elaborado pelo autor.

A.3.2 *EXPERIMENTOS COM MODELO CAIXA-PRETA PMC*

A.3.2.1 *Experimentos Alteração da Taxa de Mutação*

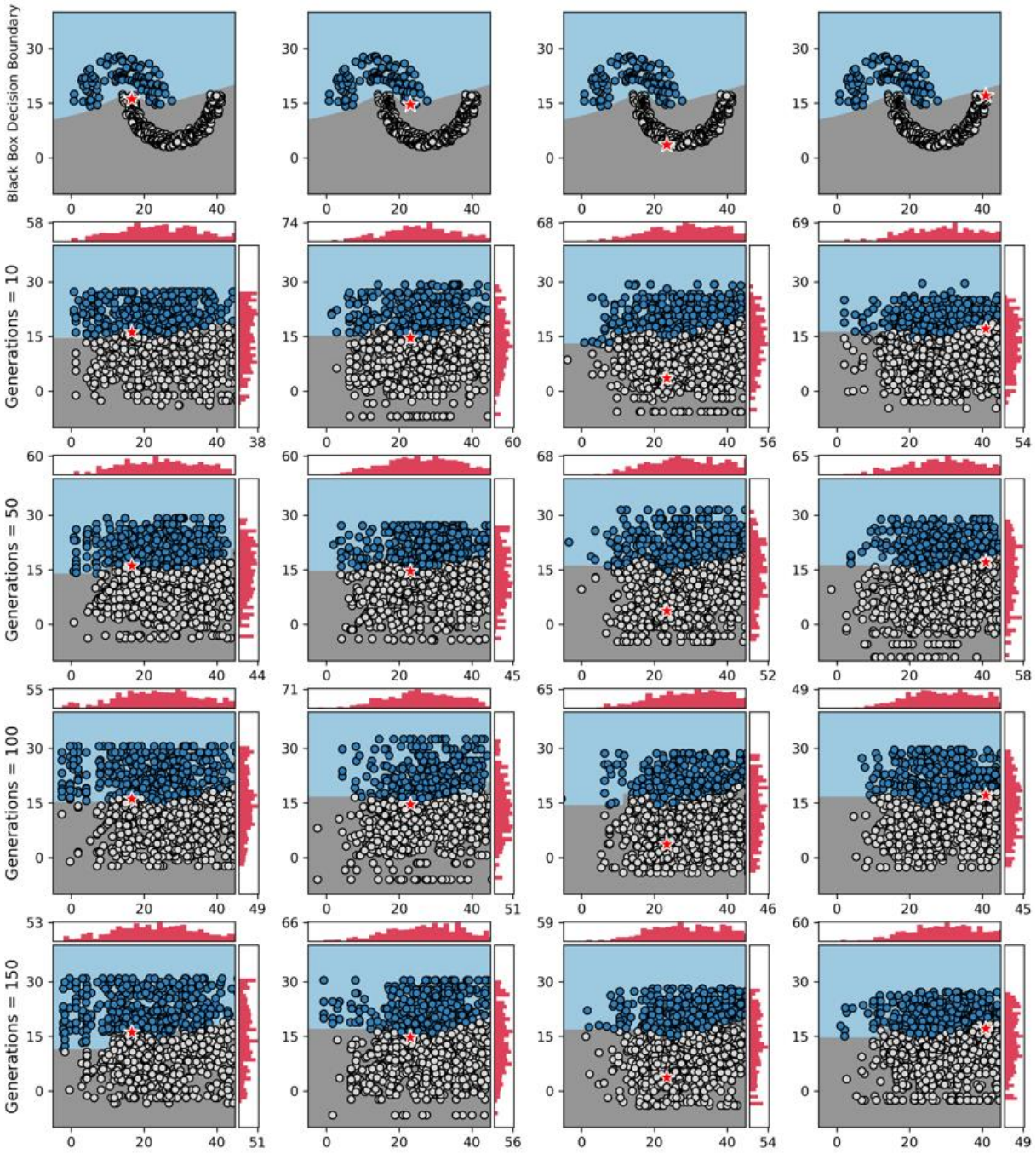
Figura 44 – Experimento 26: LOREfs - Alteração da Taxa de Mutação, Gerações = 10, PMC, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

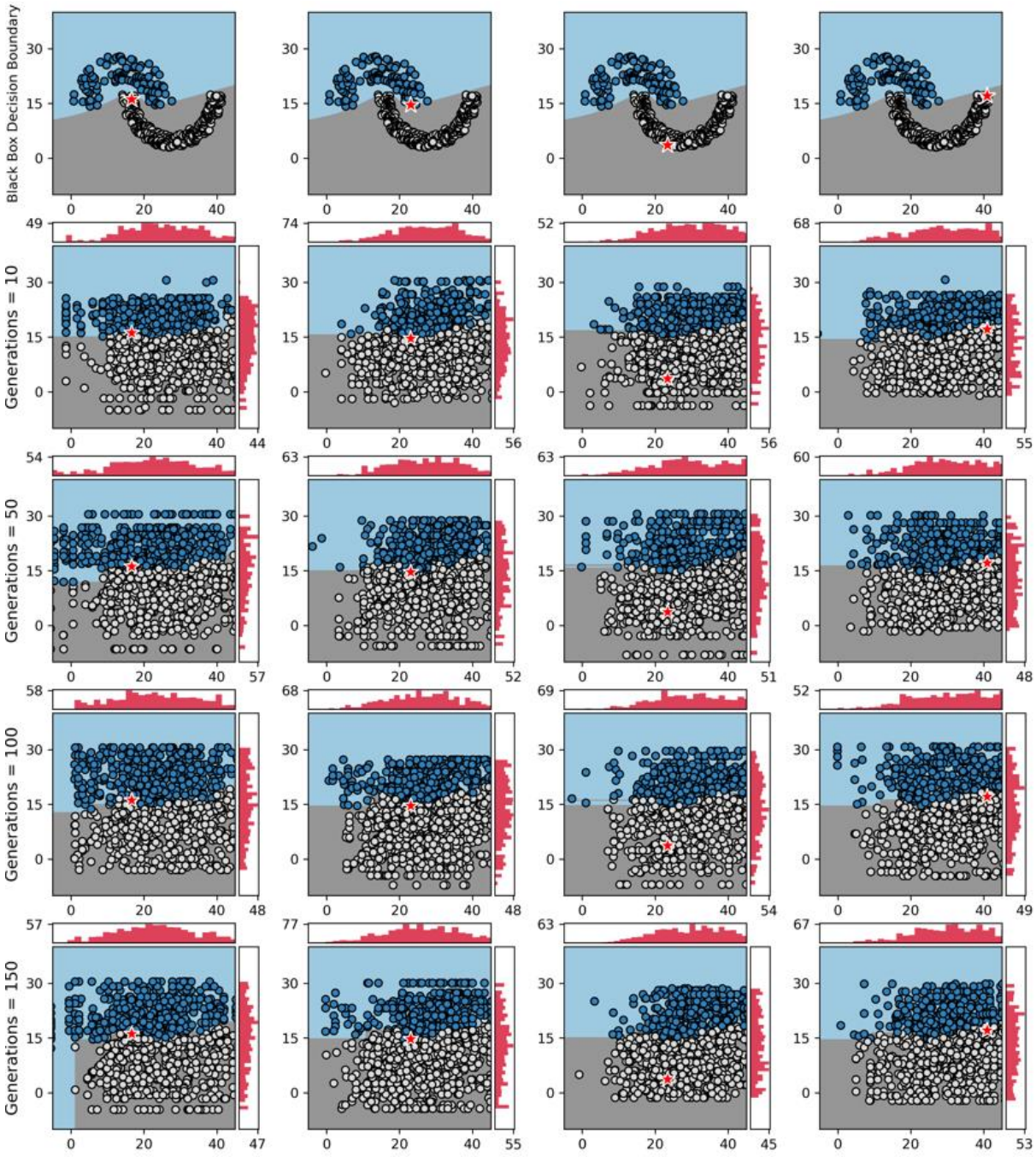
A.3.2.2 *Experimentos Alteração do Número de Gerações*

Figura 45 – Experimento 27: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.20, PMC, Conjunto de Dados = *Jain*



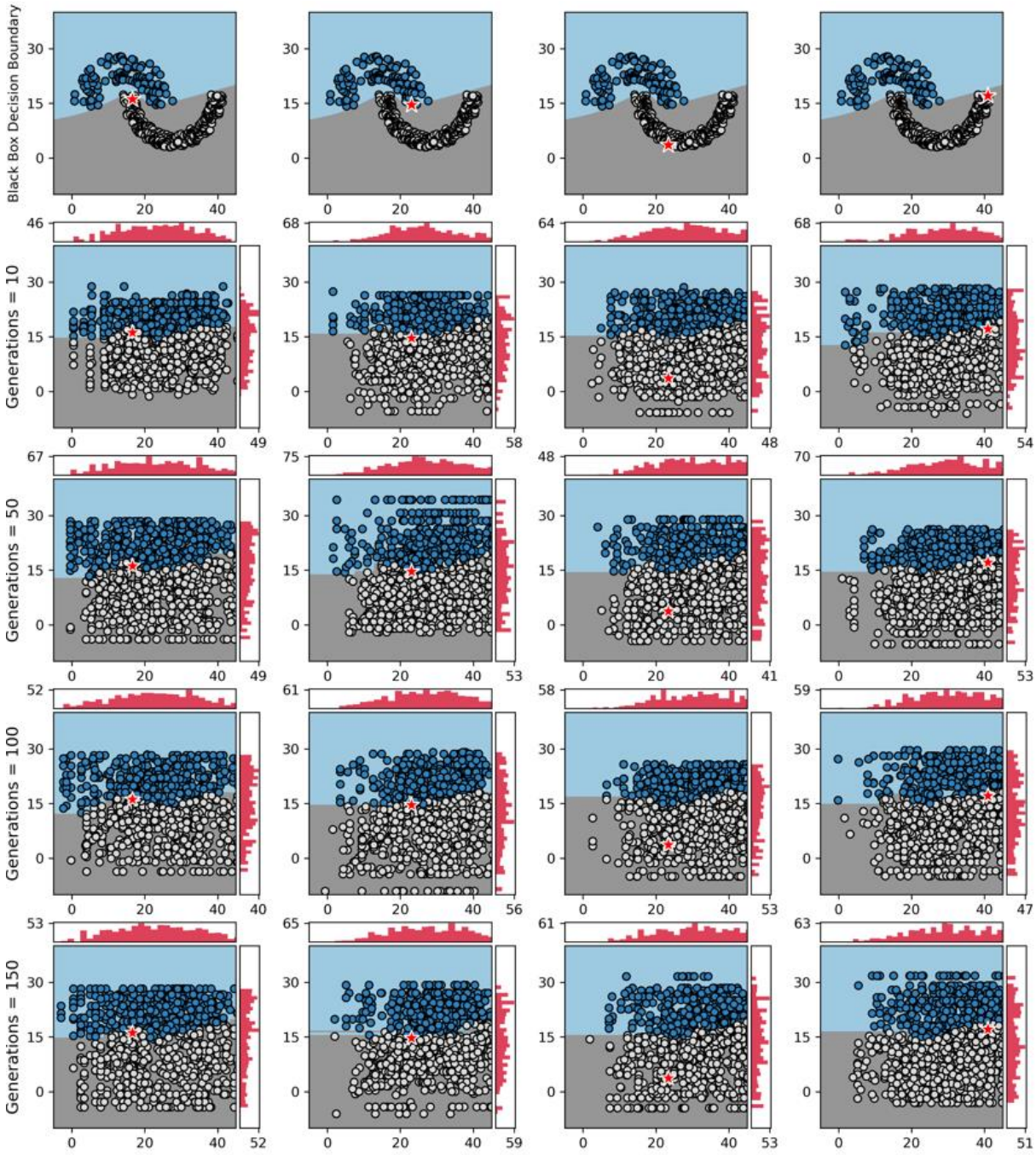
Fonte: Elaborado pelo autor.

Figura 46 – Experimento 28: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.15, PMC, Conjunto de Dados = *Jain*



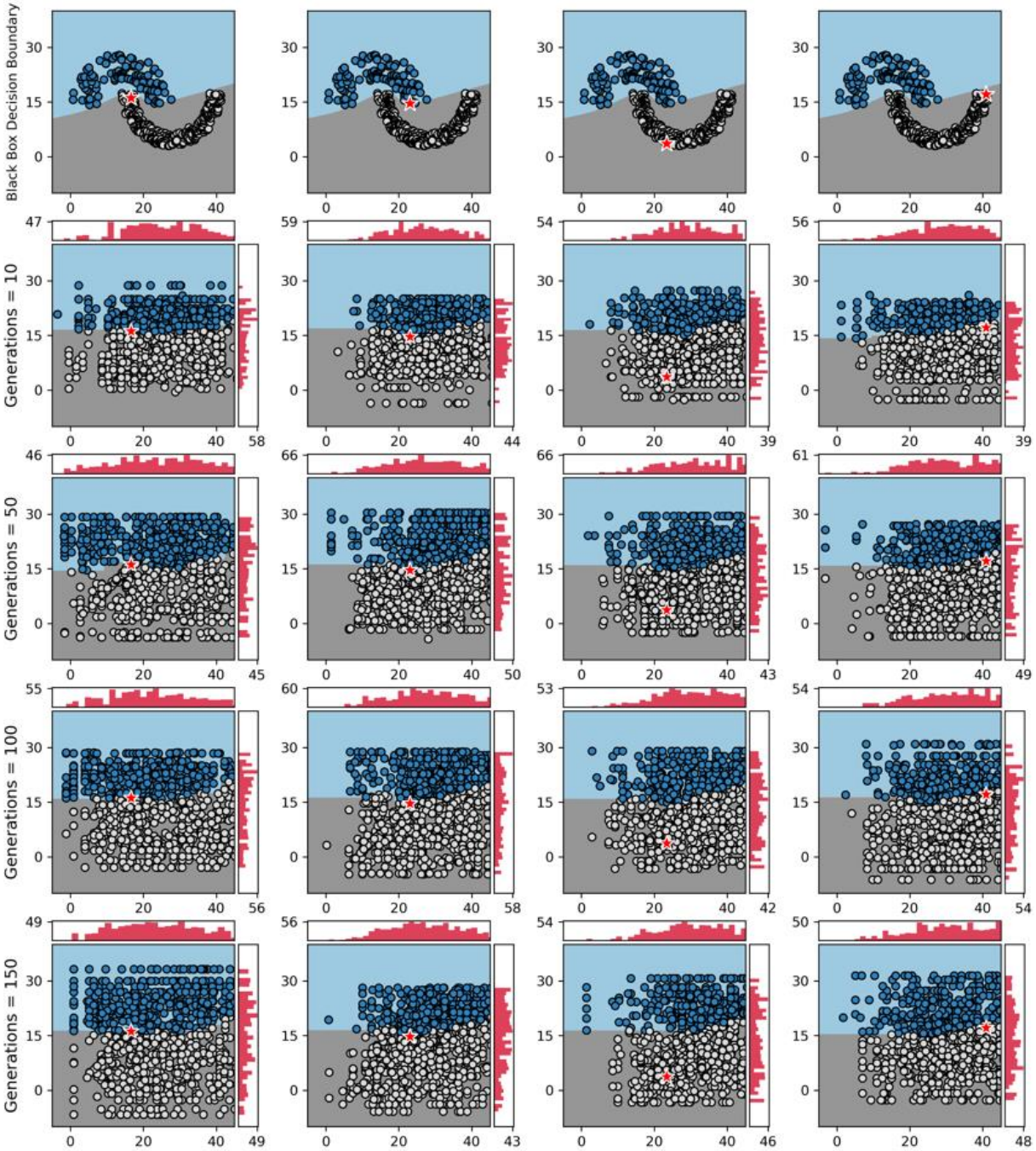
Fonte: Elaborado pelo autor.

Figura 47 – Experimento 29: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.10, PMC, Conjunto de Dados = *Jain*



Fonte: Elaborado pelo autor.

Figura 48 – Experimento 30: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.05, PMC, Conjunto de Dados = *Jain*



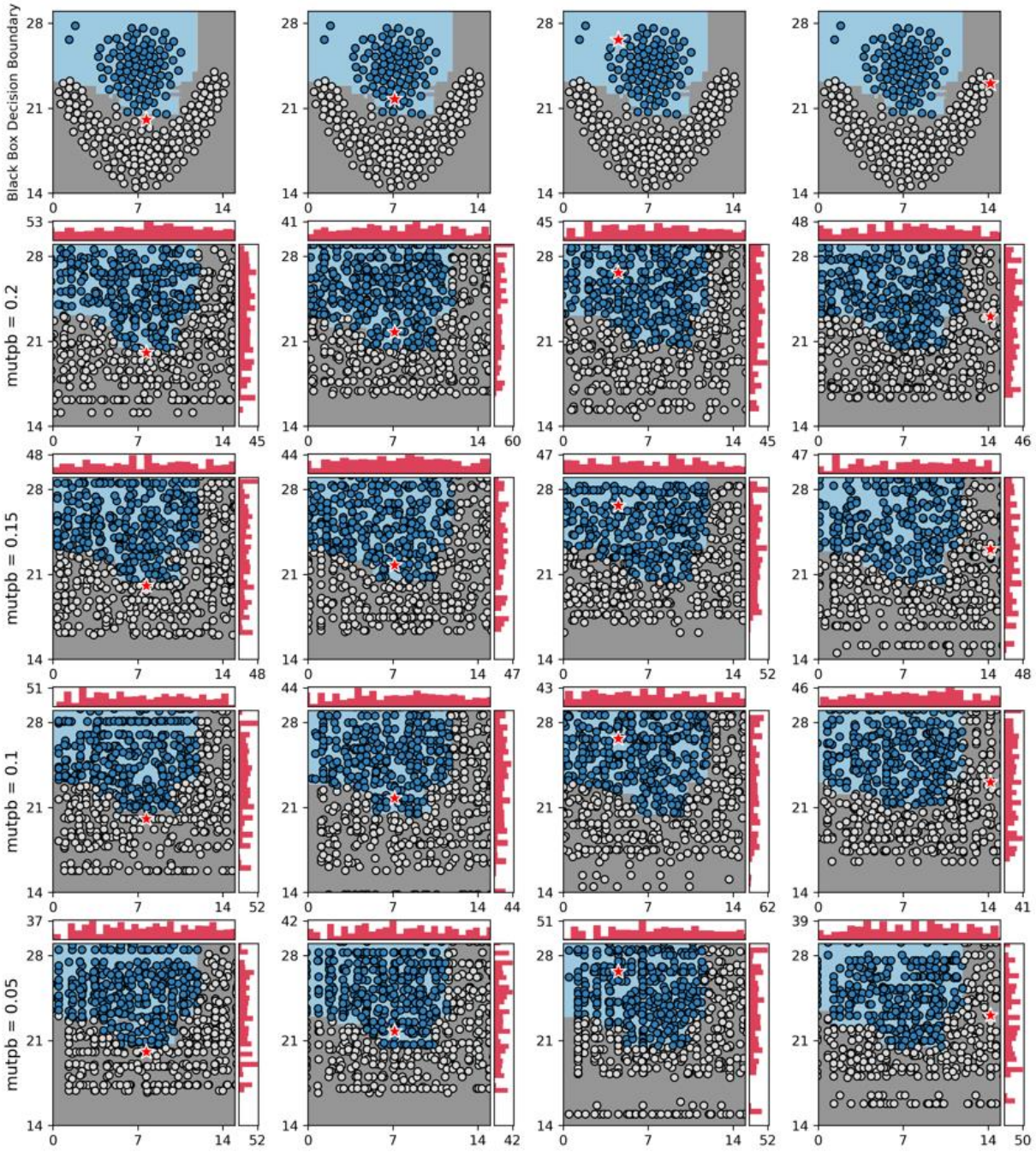
Fonte: Elaborado pelo autor.

A.4 EXPERIMENTOS LOREFS COM CONJUNTO DE DADOS *FLAME*

A.4.1 *EXPERIMENTOS COM MODELO CAIXA-PRETA FA*

A.4.1.1 *Experimentos Alteração da Taxa de Mutação*

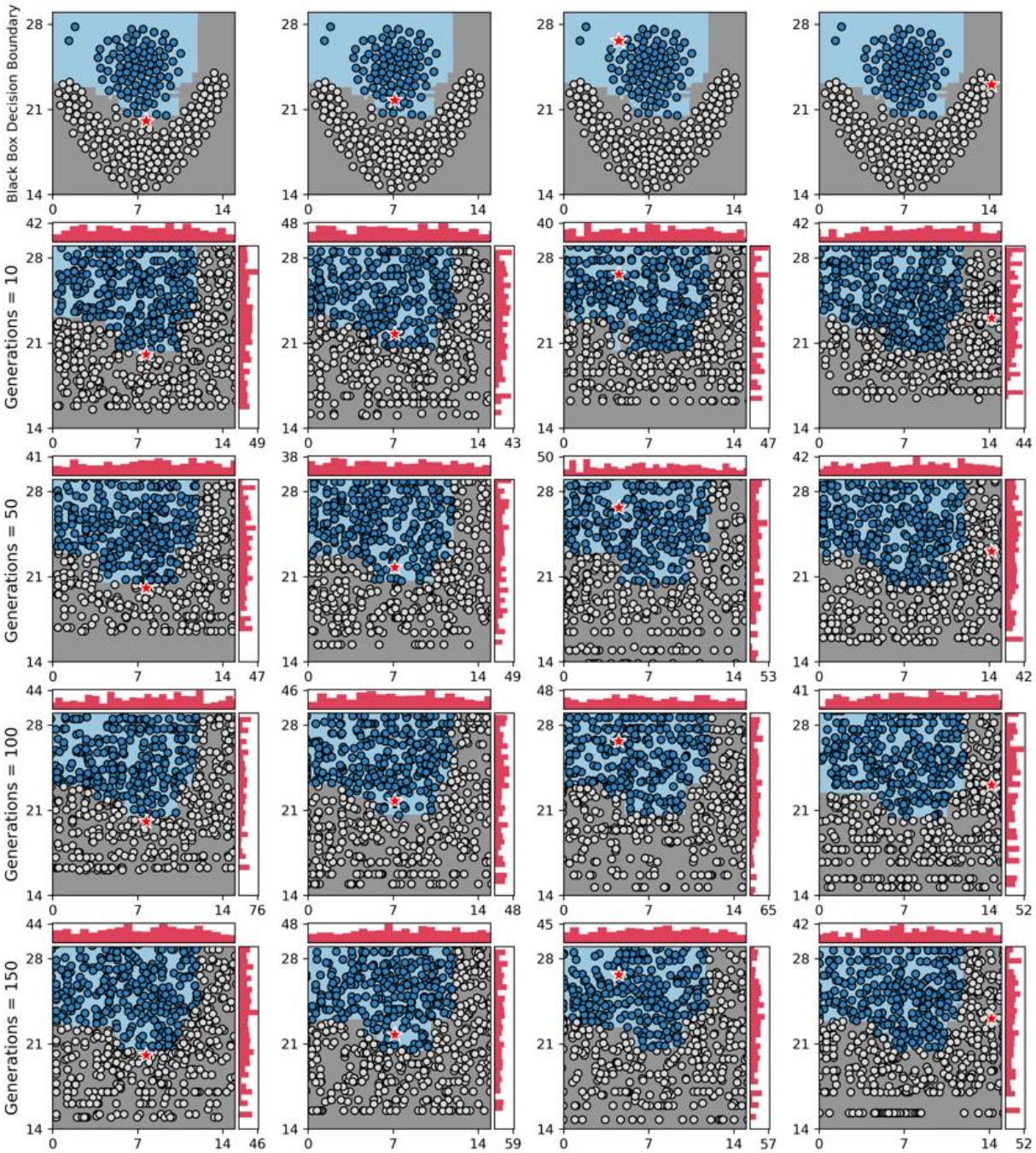
Figura 49 – Experimento 31: LOREfs - Alteração da Taxa de Mutação, Gerações = 10, FA, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

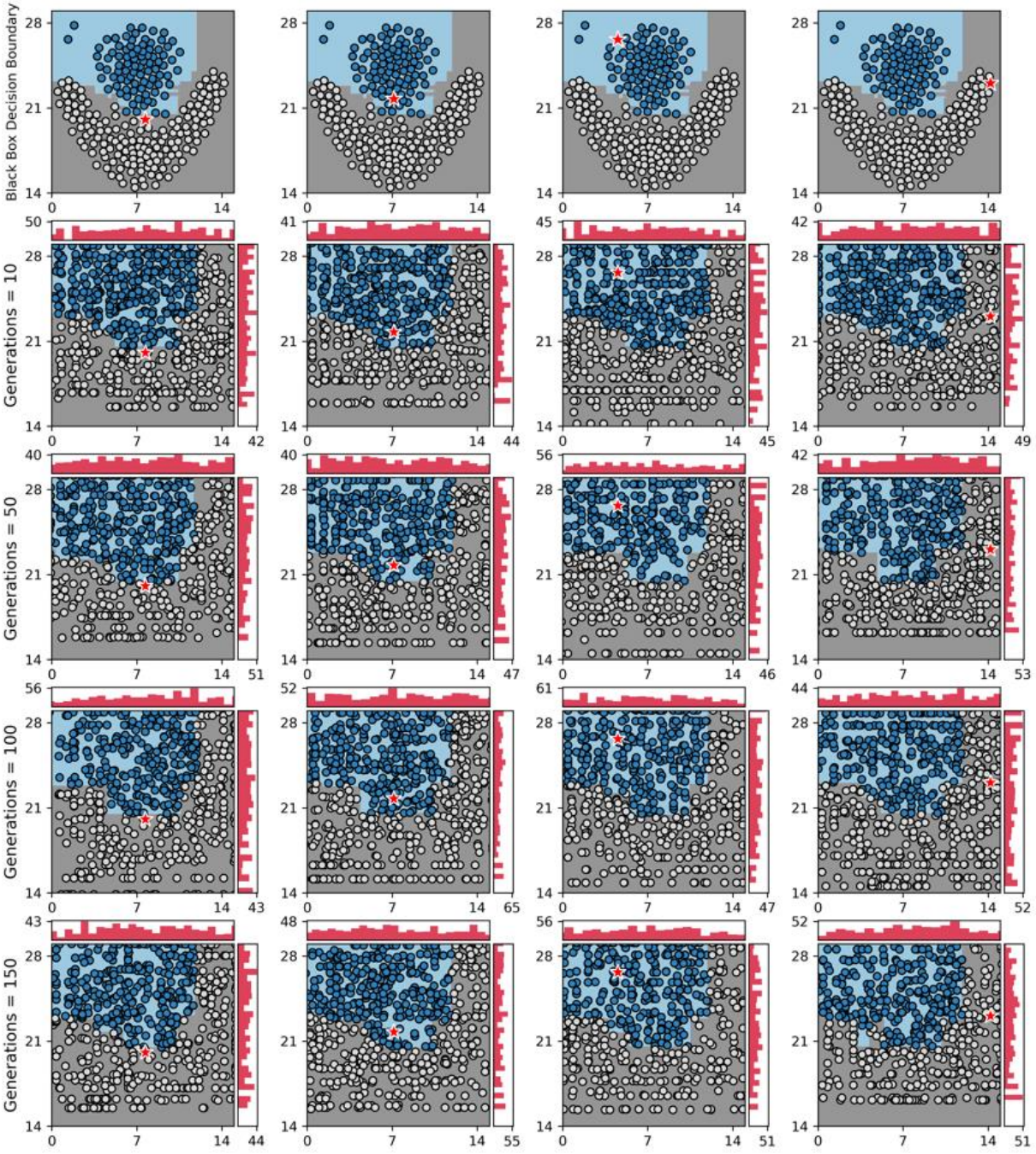
A.4.1.2 *Experimentos Alteração do Número de Gerações*

Figura 50 – Experimento 32: LORElfs - Alteração do Número de Gerações, *mutpb* = 0.20, FA, Conjunto de Dados = *Flame*



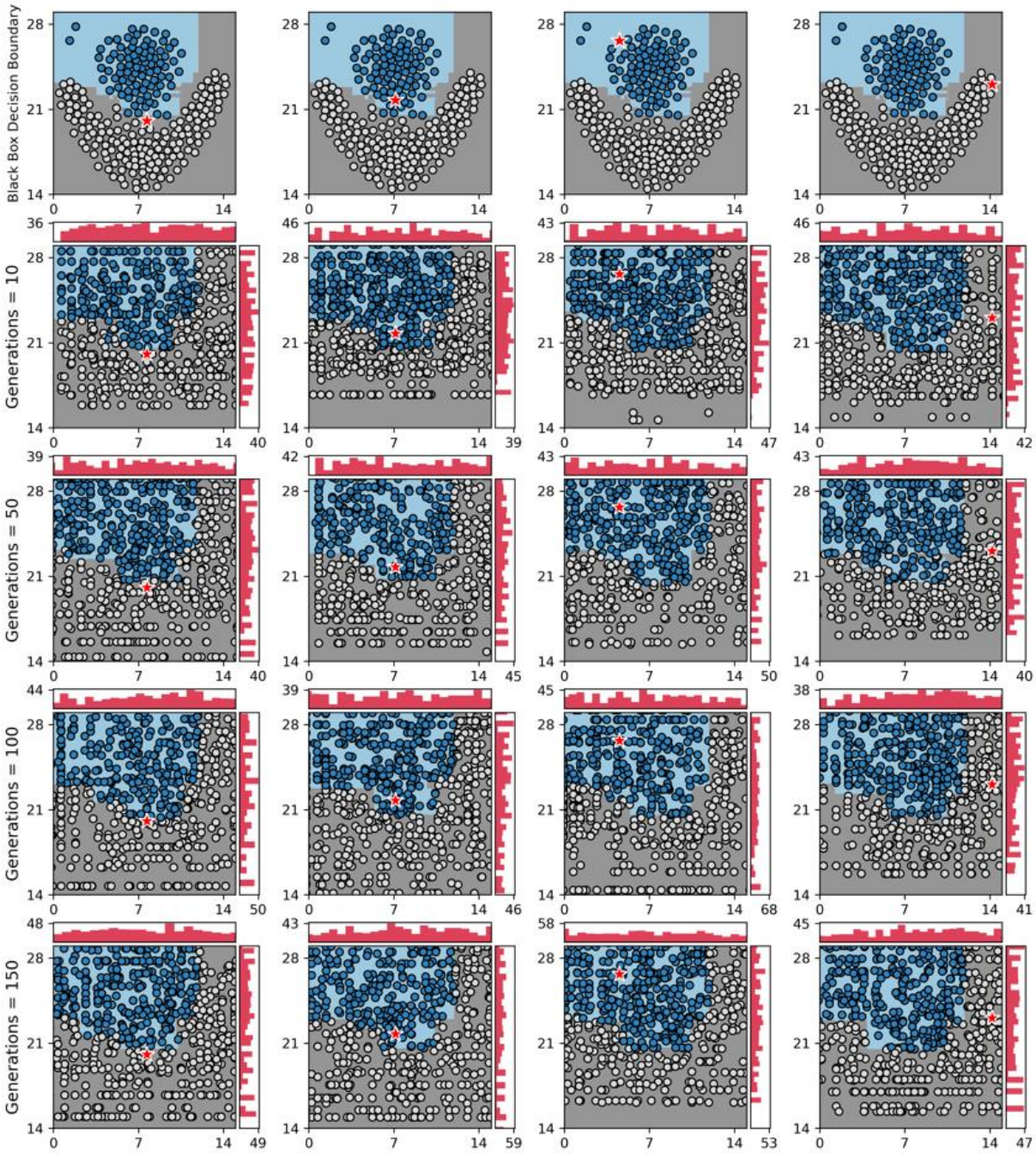
Fonte: Elaborado pelo autor.

Figura 51 – Experimento 33: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.15, FA, Conjunto de Dados = *Flame*



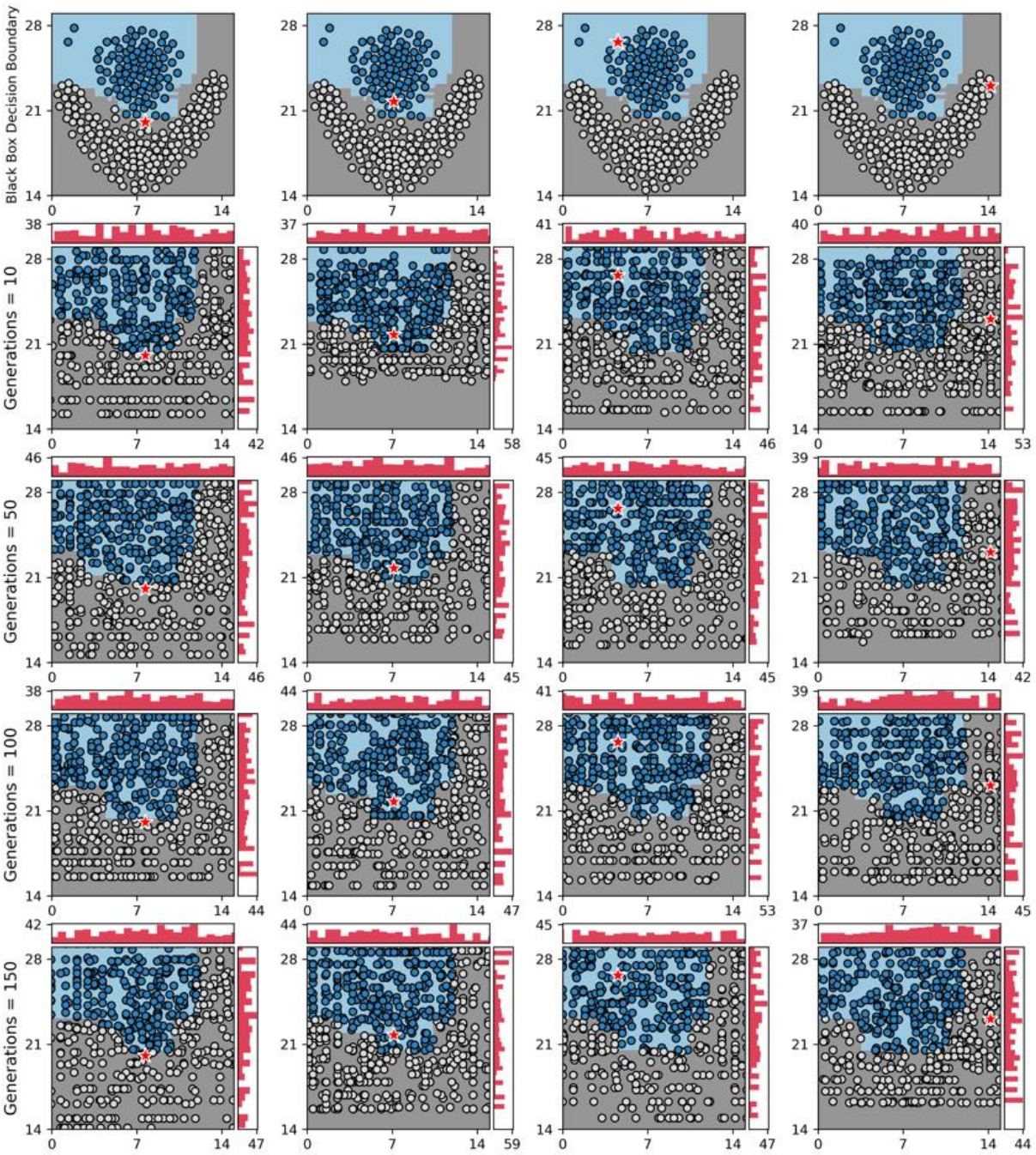
Fonte: Elaborado pelo autor.

Figura 52 – Experimento 34: LORElfs - Alteração do Número de Gerações, *mutpb* = 0.10, FA, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

Figura 53 – Experimento 35: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.05, FA, Conjunto de Dados = *Flame*

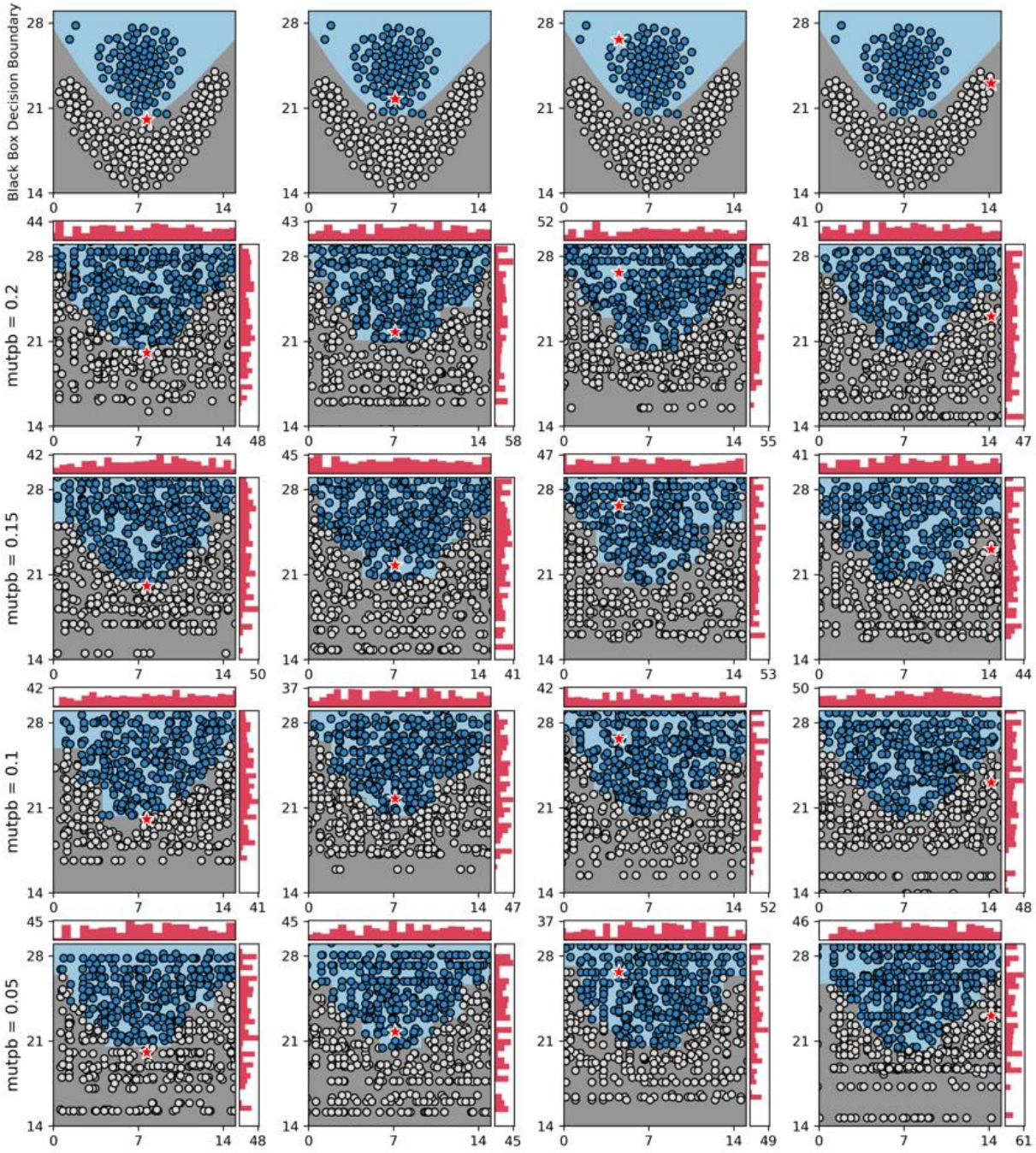


Fonte: Elaborado pelo autor.

A.4.2 *EXPERIMENTOS COM MODELO CAIXA-PRETA PMC*

A.4.2.1 *Experimentos Alteração da Taxa de Mutação*

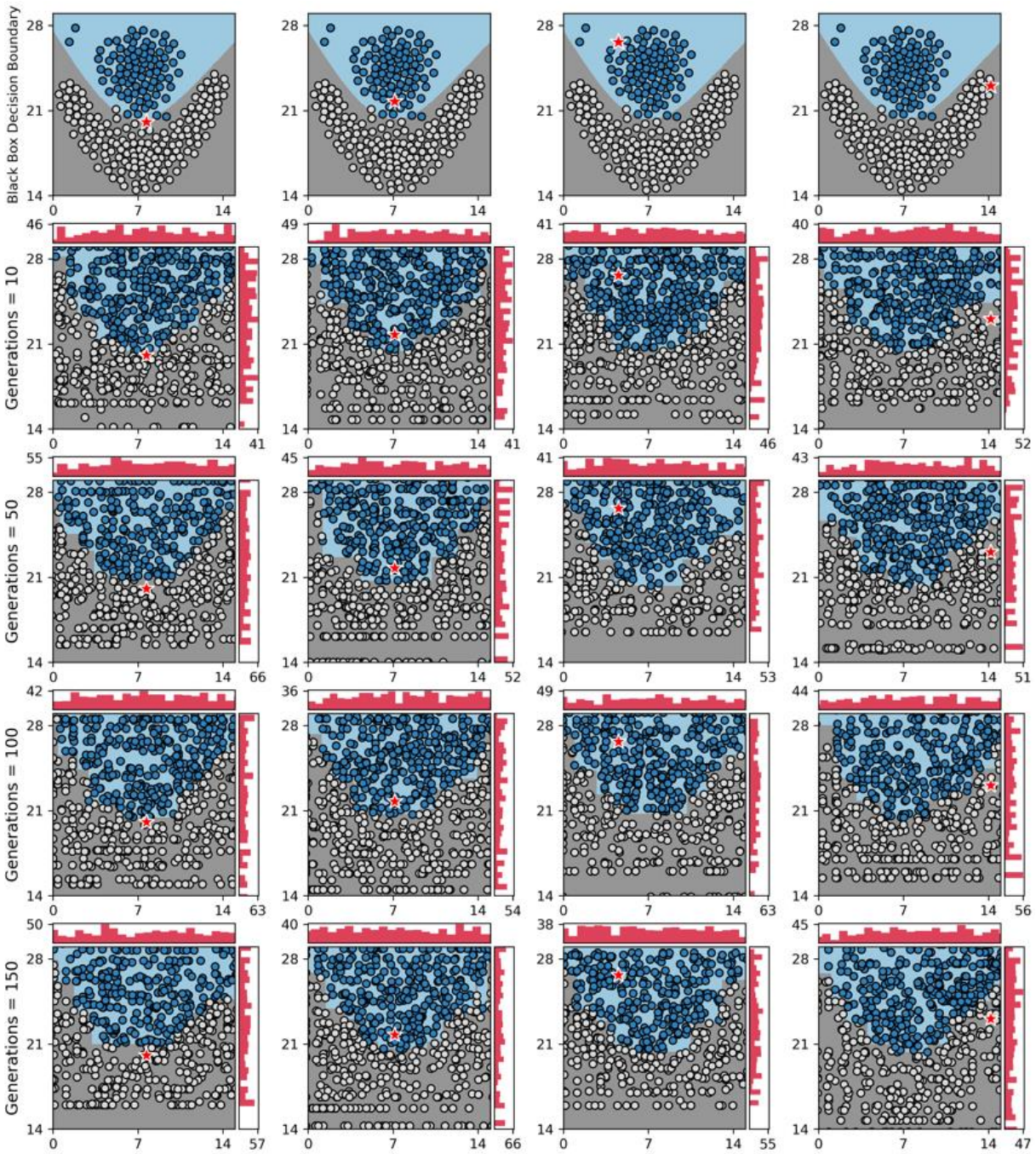
Figura 54 – Experimento 36: LOREfs - Alteração da Taxa de Mutação, Gerações = 10, PMC, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

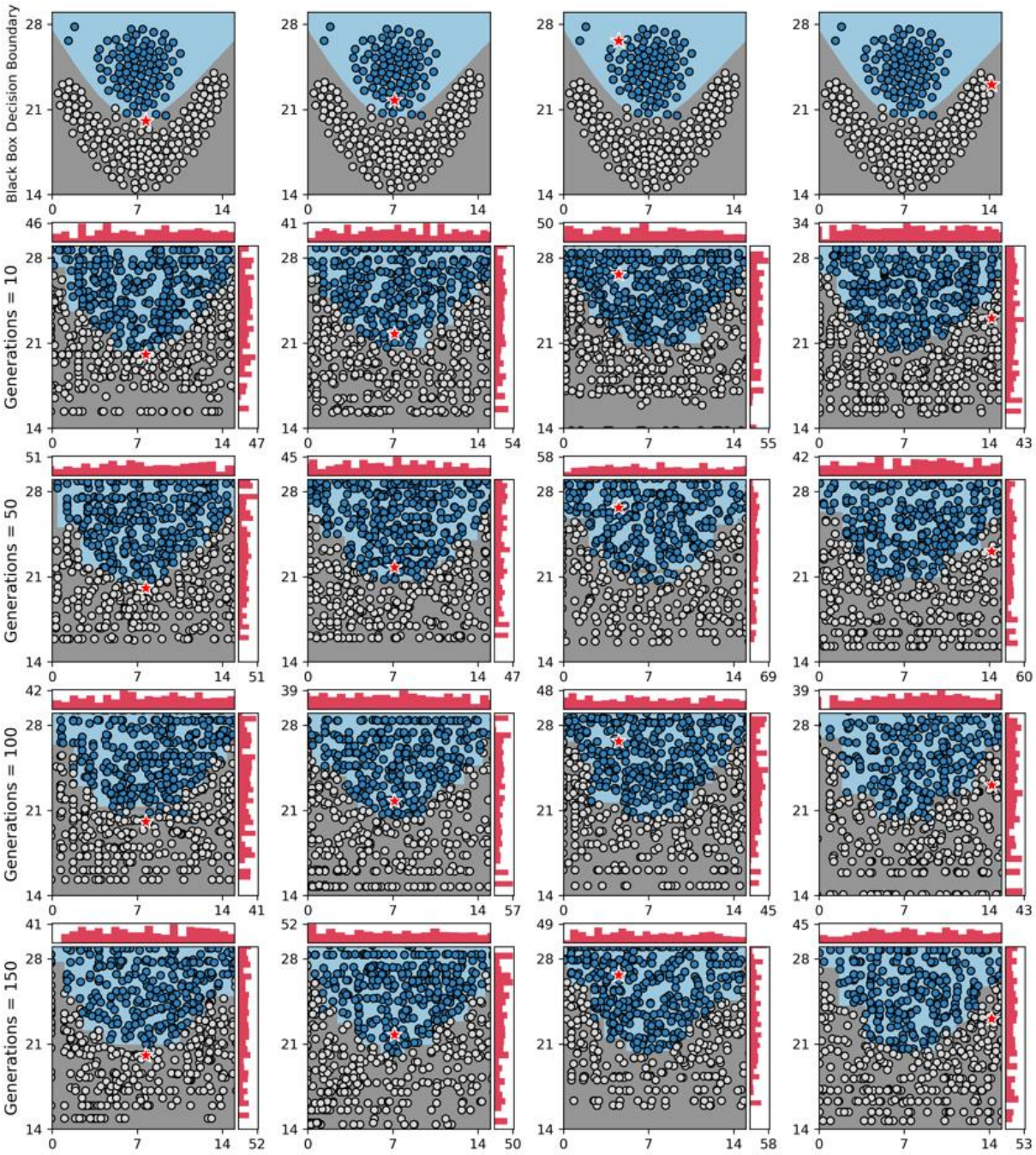
A.4.2.2 *Experimentos Alteração do Número de Gerações*

Figura 55 – Experimento 37: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.20, PMC, Conjunto de Dados = *Flame*



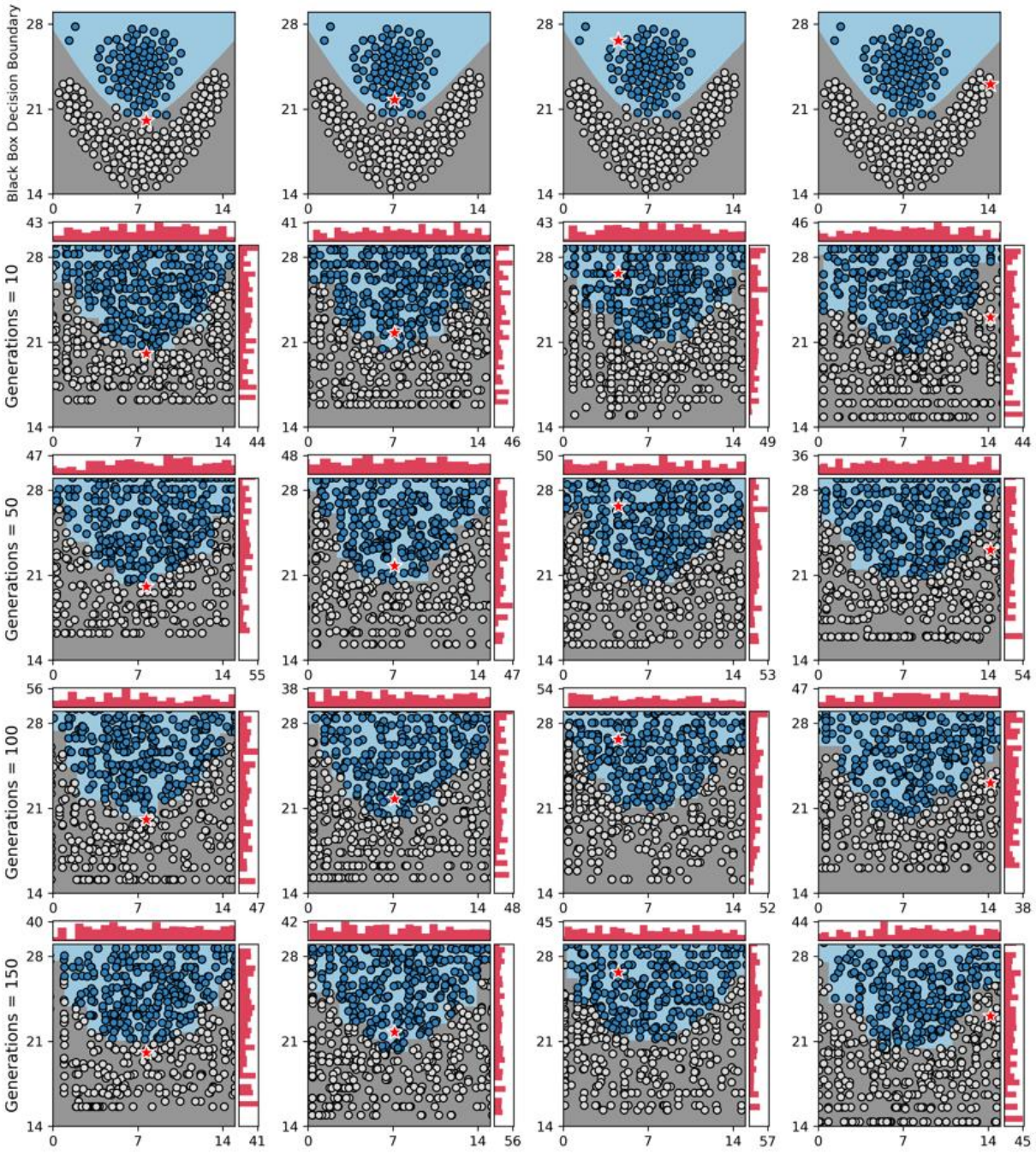
Fonte: Elaborado pelo autor.

Figura 56 – Experimento 38: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.15, PMC, Conjunto de Dados = *Flame*



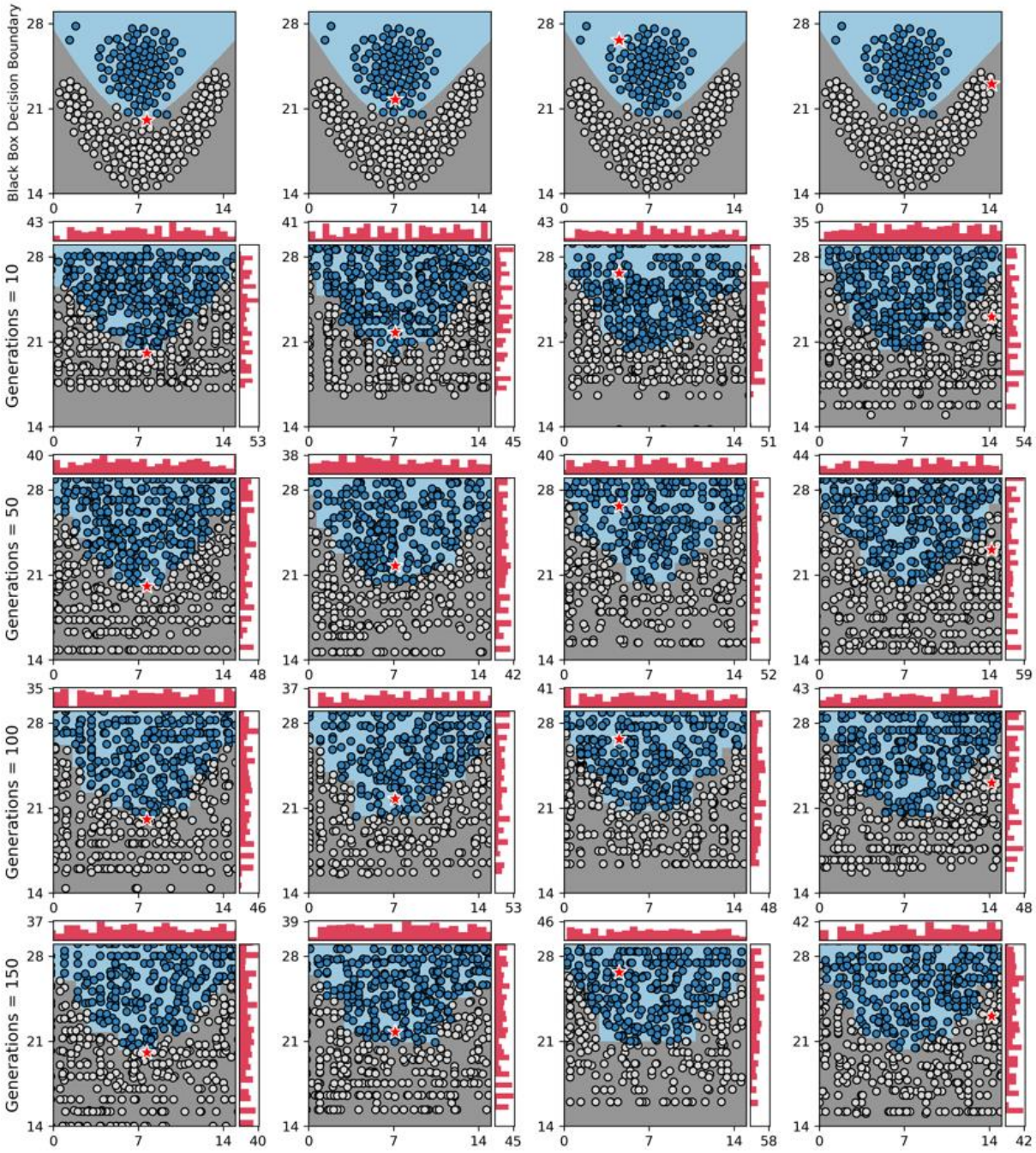
Fonte: Elaborado pelo autor.

Figura 57 – Experimento 39: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.10, PMC, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.

Figura 58 – Experimento 40: LOREfs - Alteração do Número de Gerações, *mutpb* = 0.05, PMC, Conjunto de Dados = *Flame*



Fonte: Elaborado pelo autor.