57º SBPO
SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL
Gramado • RS
DE 5 A 9 DE OUTUBRO DE 2025

ANÁLISE DE INSTÂNCIAS DO PROBLEMA DA DIVERSIDADE MÁXIMA COM APRENDIZADO SUPERVISIONADO

Carlos Vinicio D. Araújo, Jailon William B. Oliveira da Silva, Pablo Luiz Braga Soares
Laboratório de Pesquisa & Desenvolvimento do NEMO, Universidade Federal do Ceará
Avenida Felipe Santiago – Nº 411, 62.900-420, Russas/CE, Brasil
{carlosvinicio, williambrunocc}@alu.ufc.br, pablo.soarea@ufc.br

RESUMO

Este trabalho investiga a aplicação de técnicas de aprendizado de máquina supervisionado na resolução aproximada do problema da diversidade máxima. Foram explorados modelos como Regressão Logística, k-Vizinhos mais Próximos, árvores de decisão, floresta randômica e redes neurais, treinados a partir de 315 instâncias da MDPLib. As instâncias foram pré-processadas com extração de características estruturais dos grafos e balanceamento de classes, além da técnica PCA para redução de dimensionalidade. Os modelos foram avaliados com base na proximidade das soluções obtidas em relação a valores de referência de alta qualidade, fornecidos por uma metaheurística. Os resultados demonstram que, mesmo com acurácias de classificação moderadas, os modelos atingiram até 95.66% da diversidade máxima em instâncias com até 3000 vértices, evidenciando a viabilidade e escalabilidade da abordagem. Este estudo abre caminho para o uso de aprendizado supervisionado em problemas combinatórios de grande escala, reforçando seu potencial como ferramenta complementar às abordagens heurísticas tradicionais.

PALAVRAS CHAVE. Diversidade Máxima, Aprendizado de Máquina Supervisionado, Análise de Instâncias

Tópicos: EST&AM – PO Analytics em Estatística e Aprendizado de Máquina, AS&DS – Análise e Ciência de Dados

ABSTRACT

This paper investigates the application of supervised machine learning techniques to the approximate resolution of the Maximum Diversity Problem. Models such as Logistic Regression, k-Nearest Neighbors, Decision Trees, Random Forests, and Neural Networks were trained on 315 instances from MDPLib. The instances underwent preprocessing steps including extraction of structural graph features, class balancing, and Principal Component Analysis (PCA) for dimensionality reduction. Model performance was assessed based on the proximity of the obtained solutions to high-quality reference values provided by a metaheuristic. Results show that, even with moderate classification accuracies, the models achieved up to 95.66% of the maximum diversity on instances with up to 3,000 vertices, demonstrating the feasibility and scalability of the approach. This study paves the way for the use of supervised learning in large-scale combinatorial problems, reinforcing its potential as a complementary tool to traditional heuristic approaches.

KEYWORDS. Maximum Diversity, Supervised Machine Learning, Instance Analysis.

Paper topics: EST&AM – PO Statistical and Machine Learning Analytics, AS&DS –Data Analysis and Science

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

1. Introdução

Na área da Teoria dos Grafos, diversos problemas combinatórios são classificados como NP-difíceis, ou seja, problemas cuja solução exata é inviável em grandes instâncias devido ao crescimento exponencial do tempo computacional necessário [Garey e Johnson, 1979]. Entre esses problemas, se destaca o Problema da Diversidade Máxima, do inglês MDP, que busca selecionar subconjuntos de vértices em um grafo de modo a maximizar a soma das distâncias entre eles [Gonzalez et al., 2020].

O MDP é formalmente definido no contexto de grafos da seguinte forma: dado um grafo $G = (V, E)$, onde V representa o conjunto de vértices e E o conjunto de arestas, o objetivo é selecionar um subconjunto $S \subseteq V$, de tamanho k, que maximize a soma das distâncias entre os vértices selecionados. Esse problema, por ser NP-difícil, torna inviável o uso de algoritmos exatos para instâncias de grande escala, uma vez que o tempo computacional aumenta exponencialmente com o tamanho do grafo. Isto impõe a necessidade de abordagens aproximadas, como heurísticas e metaheurísticas, para resolver o problema em tempo aceitável, embora com alguma perda de precisão [Blum e Roli, 2003].

A resolução de forma eficiente do MDP tem implicações práticas significativas. Por exemplo, em bioinformática, a escolha de subconjuntos diversos de dados pode acelerar a descoberta científica e melhorar a qualidade dos resultados experimentais [Porter et al., 1975]. Mais recentemente, o MDP teve aplicação na otimização de cadeias de suprimento [Parreno et al., 2024]. Dessa forma, encontrar soluções para o MDP não é apenas um desafio teórico, mas também um fator importante para o avanço de várias áreas de pesquisa e desenvolvimento operacional.

Neste contexto, o Aprendizado de Máquina, do inglês ML surge como uma ferramenta promissora para resolver o MDP de maneira eficiente. Técnicas de ML, tanto supervisionadas quanto não supervisionadas, se destacam pela capacidade de identificar padrões complexos em grandes volumes de dados e gerar soluções aproximadas em tempo computacional reduzido [Faceli et al., 2011].

Diferente de outras abordagens presentes na literatura, este estudo inova ao aplicar técnicas de aprendizado de máquina especificamente ao MDP. Embora o aprendizado de máquina tenha sido utilizado para resolver outros problemas combinatórios, como o Problema do Caixeiro Viajante (TSP) [Shi e Zhang, 2022] e o Problema do Corte Máximo [Gu e Yang, 2020], sua aplicação ao MDP é uma área ainda pouco explorada. Este trabalho busca preencher essa lacuna, investigando como diferentes modelos de aprendizado supervisionado podem ser utilizados para resolver instâncias do MDP, com o objetivo de melhorar a eficiência computacional e a qualidade das soluções, especialmente em grafos de grande escala.

2. Fundamentação Teórica

Este tópico está estruturado da seguinte forma: a Seção 2.1, define o Problema da Diversidade Máxima, bem como suas aplicações. A Seção 2.2 aborda os métodos exatos para a resolução do problema. A Seção 2.3, aborda conceitos necessários sobre os modelos de aprendizado de máquina utilizados. Por fim, a Seção 2.4 aborda duas técnicas necessárias na fase de pré-processamento dos dados.

2.1. Problema da Diversidade Máxima

O Problema da Diversidade Máxima (MDP) é uma formulação combinatória que busca selecionar um subconjunto de vértices de modo a maximizar a diversidade entre eles. Dado um grafo $G = (V, E)$, onde V é o conjunto de vértices e E é o conjunto de arestas, o objetivo do MDP é encontrar um subconjunto $S \subseteq V$ de tamanho k que maximize a soma das diferenças entre os vértices em S. Essas diferenças são geralmente medidas pela quantidade de arestas que conectam

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

dois vértices, ou, em grafos ponderados, pelos pesos das arestas, de modo a maximizar a separação entre os vértices selecionados Gonzalez et al. [2020]; Duarte et al. [2015]; Carrabs et al. [2022].

Formalmente, o problema pode ser expresso da seguinte forma:

$$\max_{S \subseteq V, |S|=k} \sum_{v_i, v_j \in S} d(v_i, v_j) \tag{1}$$

onde $d(v_i, v_j)$ mede a diferença entre vértices. Classificado como NP-difícil, sua resolução exata em grafos de grande escala é inviável, demandando heurísticas e metaheurísticas para soluções aproximadas [Lopez et al., 2019].

O MDP tem grande aplicabilidade prática, desde a locação estratégica de instalações para mitigar riscos operacionais [Erkut e Neuman, 1991] até a otimização de tratamentos médicos por meio de diversificação terapêutica [Kuo et al., 1993]. Na genética e biotecnologia, é utilizado para maximizar variabilidade genética em programas de melhoramento [Porter et al., 1975], enquanto no design de produtos, auxilia na criação de portfólios diversificados para múltiplos mercados [Ghyczy, 1985; Besanko et al., 1990]. Recentemente, aplicou-se à otimização de cadeias de suprimento, garantindo diversidade na seleção de fornecedores para aumentar produtividade [Parreno et al., 2024], reforçando sua versatilidade em problemas multidisciplinares.

2.2. Métodos Exatos para o Problema da Diversidade Máxima

Apesar de sua complexidade NP-difícil, existem métodos exatos para resolver o MDP, especialmente para instâncias de tamanho reduzido. Essas abordagens, em geral, baseiam-se em formulações de programação inteira. O MDP pode ser formulado como um problema de programação quadrática binária [Kuo et al., 1993]. No entanto, a natureza quadrática da função objetivo torna o problema computacionalmente desafiador para solucionadores padrão.

Uma estratégia comum é linearizar a formulação quadrática, transformando-a em um problema de programação linear inteira mista, que pode ser resolvido por algoritmos como o Branch and Bound [Martí et al., 2010; Kuo et al., 1993]. Este processo envolve a introdução de variáveis e restrições adicionais para substituir os termos quadráticos.

Neste contexto, o trabalho de Soares [Soares, 2018] apresenta uma contribuição relevante, propondo um método exato para problemas quadráticos binários que emprega técnicas de *t-linearização* e *suavização hiperbólica*. Tais técnicas mostraram-se eficientes para encontrar a solução ótima para problemas como o MDP, sendo o método utilizado neste trabalho para gerar os rótulos das instâncias de menor porte [Soares, 2018].

2.3. Modelos de Aprendizado Supervisionados

De acordo com Cox [1958] a Regressão Logística (LR) é um modelo estatístico para prever resultados binários, estimando a probabilidade de um evento via função sigmoide:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{p} \beta_i x_i)}},$$

onde β_0 é o intercepto e $\beta_1, \ldots, \beta_p$ são coeficientes das variáveis X, ajustados por máxima verossimilhança. A função sigmoide, em formato de ”S”, mapeia valores contínuos para probabilidades entre 0 e 1, suavizando transições entre categorias.

Já o K-Vizinho mais Próximo, do inglês K-Nearest Neighbors (KNN) é um algoritmo de aprendizado supervisionado que classifica ou regride dados com base na proximidade a amostras conhecidas, atribuindo a classe mais frequente (classificação) ou a média (regressão) entre os k vizinhos mais próximos [Altman, 1992]. A proximidade é calculada pela distância Euclidiana:

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2},$$

onde p e q são pontos no espaço n-dimensional [Cover e Hart, 1967]. A escolha de k é crítica: valores baixos aumentam a sensibilidade a ruídos, enquanto altos introduzem viés.

As Árvores de Decisão (DT) são modelos supervisionados que dividem dados recursivamente em subconjuntos via regras hierárquicas, com nós internos (testes em features), ramos (resultados) e folhas (predições) [Quinlan, 1986]. Embora eficazes em capturar relações não lineares e de fácil interpretação, sofrem com sobreajuste em árvores muito profundas [Quinlan, 1993]. Para mitigar isso, a Floresta Randômica (RF) combina múltiplas DT, construídas com amostras e subconjuntos de features aleatórios, agregando previsões por votação (classificação) ou média (regressão) [Ho, 1995].

As Redes Neurais Artificiais (ANN) são modelos que simulam o processamento cerebral através de neurônios interconectados em camadas, ajustando pesos durante o treinamento para minimizar erros entre saídas previstas e reais [Goodfellow et al., 2016]. Sua variante mais simples, a Rede Neural *Feed-forward* (FNN), processa informações unidirecionalmente (da entrada à saída, via camadas ocultas), aplicando-se a tarefas de classificação e regressão sem realimentação de ciclos [Rosenblatt, 1958].

2.4. Técnicas para pré-processamento dos dados

O método **SMOTE** [Chawla et al., 2002] balanceia classes gerando amostras sintéticas da classe minoritária via interpolação entre vizinhos próximos (*k-NN*), evitando duplicação e melhorando a generalização. Contrariamente, o **RandomUnderSampler** reduz a classe majoritária por amostragem aleatória, eficaz em cenários com desbalanceamento extremo, porém arriscando perda de dados e variabilidade [He et al., 2008]. Assim, enquanto o SMOTE cria novas instâncias ao longo de linhas entre pontos existentes, o *RandomUnderSampler* elimina exemplos majoritários, exigindo ponderação entre ganho de equilíbrio e preservação de informações críticas.

Complementando essas técnicas, a Análise de Componentes Principais, do inglês (PCA) simplifica dados multidimensionais ao projetá-los em componentes principais não correlacionados, reduzindo dimensionalidade sem perda significativa de informação. Essa abordagem é crítica em conjuntos com alta dimensionalidade, equilibrando complexidade e preservação de padrões essenciais para a modelagem [Jolliffe e Cadima, 2016].

3. Metodologia

Esta seção descreve a metodologia para aplicação de métodos de aprendizado supervisionado em instâncias do Problema da Diversidade Máxima. A Seção 3.1 detalha a coleta de 315 instâncias da MDPLIB e a obtenção de soluções ótimas via métodos exatos e metaheurísticas. A Seção 3.2 aborda o pré-processamento, incluindo a organização em dataframes e extração de características estruturais dos grafos. A Seção 3.3 explica o treinamento de modelos como Regressão Logística e Redes Neurais, adaptados às relações lineares e não lineares dos dados, além do ajuste de previsões via ranqueamento de probabilidades. A Seção 3.4 discute o tratamento de discrepâncias dimensionais com PCA, e a Seção 3.5 apresenta as ferramentas computacionais e ambientes utilizados.

3.1. Coleta de Dados

Foram selecionadas 315 instâncias da MDPLIB, biblioteca de referência para o problema da diversidade máxima, extraídas dos trabalhos de [Glover et al., 1998], [Silva et al., 2004], [Martí et al., 2010] e [Duarte e Martí, 2007]. As instâncias são caracterizadas por ‘n’, o número total de vértices no grafo, e ‘m’, o tamanho do subconjunto a ser selecionado.

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

O conjunto SOM (Silva, Ochi e Martins) inclui 70 matrizes matrizes com valores discretos entre 0 e 9, gerados por distribuição uniforme: 50 instâncias *SOM-a* [Martí et al., 2010], com tamanhos $n \in \{25, 50, 100, 125, 150\}$ e $m \in \{2, 5, 7, 10, 12, 15, 30, 37, 45\}$, e 20 instâncias *SOM-b* [Silva et al., 2004], com $n \in \{100, 200, 300, 400, 500\}$ e $m \in \{10, 20, 30, 40, 50, 100, 150, 200\}$.

O conjunto **GKD (Glover, Kuo e Dhir)** contém 145 matrizes baseadas em distâncias euclidianas de pontos aleatórios (coordenadas entre 0 e 10): 75 instâncias *GKD-a* [Glover et al., 1998] ($n \in \{10, 15, 30\}$, $m \in \{2, 3, 4, 6, 8, 9, 12, 18, 24\}$), 50 instâncias *GKD-b* [Martí et al., 2010] com $n \in \{25, 50, 100, 125, 150\}$ e $m \in \{2, 5, 7, 10, 12, 15, 30, 37, 45\}$, e 20 instâncias *GKD-c* [Duarte e Martí, 2007] com $n = 500$ e $m = 50$.

O conjunto **MDG (Martí, Duarte e Gallego)** compreende 100 matrizes com valores reais uniformes (0-10), divididas em: 40 instâncias *MDG-a* [Duarte e Martí, 2007] e 40 *MDG-b* [Duarte e Martí, 2007; Gallego et al., 2009], ambas com $n \in \{500, 2000\}$ e $m \in \{50, 200\}$, além de 20 instâncias *MDG-c* ($n = 3000$, $m \in \{300, 400, 500, 600\}$), as maiores da biblioteca, similares às de Palubeckis [2007]. As variáveis n (vértices) e m (tamanho do conjunto de máxima diversidade) variam conforme a complexidade de cada subconjunto.

O método exato de [Soares, 2018] foi aplicado para calcular as alocações ótimas de vértices para as instâncias de menor porte. As soluções ótimas obtidas serviram como rótulos de "verdade fundamental" (ground truth) para o treinamento supervisionado. Contudo, dado que métodos exatos não escalam para instâncias grandes, para os casos de maior escala, adotou-se a meta-heurística Têmpera Simulada (SA) [Kirkpatrick et al., 1983] para gerar soluções de alta qualidade que serviram como referência.

A meta-heurística da Têmpera Simulada é baseada no processo de recozimento metálico. A cada iteração, um movimento aleatório (troca de vértices) é gerado: soluções melhores são aceitas imediatamente, enquanto piores podem ser aceitas probabilisticamente via distribuição de Boltzmann, controlada por um parâmetro de temperatura inicial (igual à maior distância) reduzido gradualmente. A SA garantiu soluções de alta qualidade em instâncias complexas.

3.2. Preparação dos Dados e Engenharia de Características

A etapa de preparação de dados é crucial para o sucesso de modelos de aprendizado de máquina. Para este trabalho, o conjunto de dados foi estruturado de forma que cada **linha representasse um único vértice** v_i de um grafo de uma instância do problema. Essa abordagem permite que o modelo aprenda a classificar vértices individualmente com base em suas características locais e relacionais.

Para cada vértice, foi extraído um vetor de características para capturar suas propriedades estruturais. As seguintes métricas foram calculadas para cada vértice v_i:

- **Grau Ponderado:** A soma dos pesos de todas as arestas incidentes ao vértice v_i.
- **Peso Médio das Arestas:** A média dos pesos das arestas conectadas a v_i.
- **Desvio Padrão dos Pesos:** O desvio padrão dos pesos das arestas de v_i, medindo a variabilidade de suas conexões.
- **Peso Máximo e Mínimo das Arestas:** O maior e o menor peso entre as arestas de v_i.
- **Soma dos Graus dos Vizinhos:** A soma dos graus ponderados de todos os vértices adjacentes a v_i, servindo como uma medida de conectividade de segunda ordem.

Inicialmente, características mais complexas, como centralidade de intermediação e coeficiente de aglomeração, foram consideradas. Contudo, a extração dessas métricas em grafos completos, como os do MDP, possui um custo computacional proibitivo. Uma vez que o objetivo é

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

desenvolver um modelo que sirva como uma heurística rápida, optou-se por um conjunto de características mais simples e computacionalmente eficientes, garantindo um equilíbrio entre poder preditivo e viabilidade.

Finalmente, foi adicionada uma coluna alvo denominada "Conjunto", contendo a variável binária x_i para cada vértice v_i, indicando se ele pertence à solução de referência (ótima ou da metaheurística): $x_i = 1$ se v_i pertence ao conjunto de máxima diversidade, e $x_i = 0$ caso contrário.

3.3. Treinamento dos Modelos Supervisionados

O treinamento supervisionado foi realizado utilizando um conjunto de dados consolidado, que agrupa os dataframes de vértices de todas as instâncias de treino. O objetivo é prever a variável alvo x_i com base no vetor de características de cada vértice. A escolha dos modelos, como Regressão Logística, Redes Neurais *Feed-Forward*, Árvores de Decisão, *Random Forest* e KNN, foi diretamente vinculada às características dos dados. Essa combinação visa explorar múltiplas perspectivas dos dados, equilibrando eficiência computacional e adaptação às nuances estruturais identificadas nas instâncias de grafos.

A Regressão Logística foi priorizada por sua eficiência em problemas binários e capacidade de capturar a leve separabilidade linear induzida pelas *features*, como grau ponderado, peso médio e máximo das Arestas. Já as Redes Neurais complementaram essa abordagem, lidando com não linearidades residuais que modelos puramente lineares não capturariam. As Árvores de Decisão e *Random Forest* foram escolhidas para modelar interações complexas entre variáveis, por exemplo, soma dos graus dos vizinhos e desvio padrão dos pesos, segmentando os dados em regiões homogêneas. Por fim, o kNN aproveitou padrões locais de similaridade entre vértices, classificando com base na proximidade direta, sendo uma estratégia crítica para problemas dependentes de contextos espaciais.

Para garantir a consistência dos resultados e evitar sobreajuste, foi utilizada uma metodologia de validação cruzada com 5 dobras (5-fold cross-validation) durante a fase de treinamento. Essa técnica permitiu uma avaliação honesta do desempenho de generalização dos modelos e auxiliou no ajuste fino de hiperparâmetros.

Adicionalmente, como o tamanho do conjunto de máxima diversidade nas instâncias é arbitrário (por volta de 5%–40% do grafo), a acurácia dos modelos não garante a alocação exata do número de vértices. Para contornar essa limitação, implementou-se um mecanismo de ranqueamento de probabilidades: as distribuições geradas pelos modelos são ordenadas, e ajustes são feitos com base nesse ranking. Se o modelo prevê menos vértices que o necessário, adicionam-se os de maior probabilidade; se prevê mais, realiza-se uma poda removendo os de menor probabilidade. Esse processo assegura o tamanho correto do conjunto, sem descartar as previsões originais. Ressalta-se que os modelos já produzem naturalmente as distribuições de probabilidade, que são armazenadas e utilizadas diretamente nos ajustes.

3.4. Tratamento de Instâncias com Diferentes Dimensões

A aplicação de modelos de aprendizado de máquina em múltiplos grafos de tamanhos variados apresenta um desafio de dimensionalidade. A abordagem adotada neste trabalho contorna esse problema diretamente através da **engenharia de características**. Ao transformar cada vértice de qualquer grafo em um vetor de características de tamanho fixo (neste caso, 6 características), cria-se um *dataset* uniforme onde cada amostra (vértice) tem a mesma dimensão, independentemente do tamanho do grafo original.

Adicionalmente, investigou-se o uso do PCA como uma técnica de redução de dimensionalidade sobre o espaço de características já extraído. O PCA foi testado para projetar os dados em um número menor de componentes (de 1 a 500). Observou-se que com apenas 1 componente

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

havia perda de desempenho, mas resultados consistentes foram obtidos com 10 a 500 componentes. A escolha de 10 componentes representou um equilíbrio ideal entre eficiência e preservação da informação, correspondendo a uma variância explicada acumulada de aproximadamente 85% (valor a ser confirmado experimentalmente). Esta abordagem se mostrou superior em termos de consumo de memória e consistência em comparação a uma alternativa ingênua de preenchimento de matrizes de adjacência.

3.5. Ferramentas e Implementação

O desenvolvimento e implementação dos métodos descritos foram realizados utilizando a linguagem de programação *Python*. As principais bibliotecas incluem Pandas e NumPy para manipulação eficiente de dados e cálculos numéricos, aproveitando operações vetorizadas e estruturas como *dataframes*. O *Scikit-Learn* facilitou a experimentação com algoritmos de aprendizado supervisionado e não supervisionado, além de etapas críticas como pré-processamento e validação de modelos. Para redes neurais, o *PyTorch* foi escolhido por sua flexibilidade na construção de arquiteturas dinâmicas e suporte a ajustes em tempo real, essencial para modelos complexos.

Na etapa de análise, *Matplotlib* e *Seaborn* garantiram visualizações claras e estatisticamente informativas, enquanto o *NetworkX* permitiu a modelagem e extração de métricas estruturais de grafos. Todo o processo foi integrado no *Jupyter Notebooks*, ambiente que combinou execução modular, documentação interativa e visualização imediata de resultados, otimizando a iteração entre desenvolvimento, teste e refinamento das soluções propostas.

Para garantir a reprodutibilidade, os modelos da biblioteca Scikit-Learn (Regressão Logística, k-NN, Árvore de Decisão, Floresta Randômica) foram inicializados com seus parâmetros padrão, que serviram como uma linha de base. Posteriormente, os hiperparâmetros mais influentes (e.g., o número de vizinhos para k-NN, a profundidade da árvore para DT) foram otimizados utilizando a validação cruzada mencionada na Seção 3.3. Para a Rede Neural implementada com PyTorch, a arquitetura base consistiu em duas camadas ocultas com 64 e 32 neurônios, respectivamente, utilizando a função de ativação ReLU.

4. Análise Descritiva dos Dados

Antes da avaliação final, foi realizada uma análise estatística descritiva das 255 instâncias utilizadas no treinamento para compreender a estrutura e a distribuição dos dados. A Tabela 1 resume as principais estatísticas das características extraídas.

Tabela 1: Estatísticas descritivas das Instâncias

Características	mean	std	min	25%	50%	75%	max
Quantidade de Vértices	1.96×10^2	1.50×10^2	0.0	6.2×10^1	1.63×10^2	3.23×10^2	4.99×10^2
Grau (Ponderado)	5.75×10^4	1.01×10^5	7.5×10^1	2.38×10^3	5.72×10^3	1.81×10^4	2.75×10^5
Peso Médio das Arestas	1.29×10^2	1.99×10^2	2.76	4.87	1.22×10^1	1.61×10^2	5.51×10^2
Desvio Padrão dos Pesos	6.68×10^1	1.16×10^2	1.56	2.76	2.90	2.39×10^1	3.10×10^2
Peso Máximo das Arestas	2.46×10^2	3.98×10^2	7.0	9.96	1.81×10^1	2.09×10^2	1.00×10^3
Peso Mínimo das Arestas	1.12×10^1	2.83×10^1	0.0	0.0	6.40	5.53	1.73×10^2
Soma dos Graus dos Vizinhos	2.81×10^7	5.06×10^7	2.49×10^3	1.12×10^6	1.25×10^6	3.21×10^6	1.25×10^8

Como pode ser observado na Tabela 1, as características extraídas das 255 instâncias da MDPLib apresentam uma variedade considerável, com médias e desvios padrão que indicam uma ampla gama de estruturas de grafos. Embora os grafos da MDPLib, em sua forma original, possuam uma distribuição uniforme e constante nas suas conexões e pesos, a extração e escolha das características (como o Grau (Ponderado), Peso Médio das Arestas, Desvio Padrão dos Pesos, Peso Máximo das Arestas, Peso Mínimo das Arestas, Soma dos Graus dos Vizinhos) permitiu induzir uma leve linearidade nos dados. Isso foi suficiente para que os modelos, especialmente a regressão logística, pudessem atuar de forma eficiente.

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

Ao observar a tabela, vemos que, apesar de uma grande variação no número de vértices (com uma média de 196 vértices e um desvio padrão de 150) e na conectividade das instâncias (grau ponderado médio de 57.500, com um desvio padrão de 101.000), a estrutura geral dos grafos não é excessivamente complexa. Esse tipo de estrutura favoreceu a adaptação dos modelos, uma vez que as características extraídas permitiram identificar padrões de conectividade e peso com um nível de simplicidade suficiente para os modelos de aprendizado, evitando o risco de sobreajuste.

A regressão logística, em particular, se destacou nesse cenário. Embora o problema da diversidade máxima em grafos envolva relações complexas entre os vértices e suas conexões, a linearidade induzida pelas características extraídas fez com que a regressão logística fosse eficaz na modelagem da probabilidade de um vértice pertencer ao conjunto de máxima diversidade. A regressão logística é um modelo particularmente eficaz para problemas de classificação binária com fronteiras de decisão lineares, e a relação entre as características extraídas e a alocação de vértices foi suficientemente linear para permitir que o modelo gerasse boas previsões.

A simplicidade das características extraídas não só favoreceu o bom desempenho da regressão logística, mas também contribuiu para o desempenho dos modelos em diferentes instâncias de treinamento. A análise dos resultados mostrou que a proximidade das soluções em relação ao valor ótimo aumentava à medida que mais instâncias eram incluídas no treinamento, refletindo a capacidade da regressão logística de se adaptar de forma eficaz a dados adicionais sem sofrer com viés de aprendizado.

5. Resultados

Esta seção apresenta os resultados obtidos a partir da aplicação de modelos de aprendizado de máquina ao problema da diversidade máxima em grafos. O objetivo principal foi avaliar a capacidade dos modelos em alocar corretamente os vértices em um conjunto de máxima diversidade e, com isso, alcançar um valor de diversidade máxima próximo ao valor ótimo. A seguir, são apresentados os resultados dos testes realizados, as métricas utilizadas para avaliação, e uma análise crítica sobre o desempenho dos modelos.

5.1. Análise da Curva de Aprendizagem com Instâncias Aleatórias

Para avaliar o impacto do volume de dados no desempenho dos modelos, foi conduzido um experimento incremental. Os modelos foram treinados com subconjuntos crescentes das 255 instâncias disponíveis (grafos de 10 a 500 vértices), e seu desempenho foi medido em conjuntos de teste correspondentes. Três configurações foram testadas: Teste 1 (100 instâncias de treino), Teste 2 (150) e Teste 3 (200). Este estudo da curva de aprendizagem é secundário ao objetivo principal, mas serve para demonstrar a capacidade de generalização dos modelos à medida que mais dados são fornecidos.

A aparente contradição entre a baixa acurácia de classificação e a alta qualidade da solução final é um dos resultados mais importantes deste trabalho. Isso ocorre porque o modelo de aprendizado de máquina não atua como um classificador perfeito, mas sim como um **gerador de heurística**, aprendendo uma função de pontuação (scoring function) para os vértices. O sucesso da abordagem reside no *mecanismo de ranqueamento de probabilidades* (Seção 3.3). Para obter uma boa solução, não é necessário que o modelo classifique corretamente cada vértice com um limiar fixo (e.g., 0.5). Em vez disso, é suficiente que ele atribua probabilidades mais altas aos vértices que pertencem a uma boa solução em detrimento daqueles que não pertencem. A ordenação dessas probabilidades permite selecionar os ‘m’ vértices mais promissores, uma tarefa consideravelmente mais fácil do que a classificação binária exata. Portanto, a acurácia de classificação é um “proxy” inadequado para a qualidade da decisão final neste contexto de “prever e otimizar”.

Conforme as Tabelas 2, 3 e 4, apesar da acurácia dos modelos variar entre 0,45 e 0,60, os valores de diversidade máxima obtidos mostraram-se promissores. A métrica de acurácia, iso-

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

ladamente, não reflete a qualidade das soluções, reforçando que a capacidade dos modelos em maximizar a diversidade constitui um indicador mais relevante.

Tabela 2: Treinamento com 100 Instâncias

Instâncias de Teste	Valor Objetivo (Metaheurística)	Árvores de Decisão		Redes Neurais		Floresta Randômica		Regressão Logística		KNN	
		Acurácia	Valor	Acurácia	Valor	Acurácia	Valor	Acurácia	Valor	Acurácia	Valor
SOM-a-2-(n25 e m2)	9	0.45	6	0.40	5	0.50	6	0.45	6	0.52	6
SOM-a-5-(n25 e m2)	9	0.60	7	0.60	8	0.60	8	0.60	8	0.62	7
SOM-a-7-(n25 e m7)	150	0.50	105	0.60	90	0.62	93	0.60	90	0.62	93
SOM-a-9-(n25 e m7)	147	0.50	88	0.58	85	0.60	88	0.62	91	0.62	92
SOM-a-25-(n100 e m10)	337	0.52	202	0.62	216	0.63	212	0.65	219	0.62	210
SOM-a-27-(n100 e m30)	2477	0.54	1730	0.63	1550	0.64	1587	0.65	1600	0.64	1580
SOM-a-32-(n125 e m12)	472	0.51	241	0.62	292	0.61	288	0.63	297	0.63	297
SOM-a-37-(n125 e m37)	3648	0.53	1932	0.63	2290	0.64	2334	0.65	2377	0.62	2268
SOM-a-42-(n150 e m15)	719	0.52	374	0.62	446	0.61	440	0.64	460	0.62	446
SOM-a-46-(n150 e m45)	5379	0.58	3120	0.64	3443	0.63	3389	0.64	3453	0.62	3332
SOM-b-6-(n200 e m40)	4432	0.53	2350	0.63	2791	0.62	2747	0.64	2857	0.62	2747
SOM-b-8-(n200 e m60)	16161	0.55	8884	0.62	10015	0.61	9860	0.63	10134	0.62	10034
SOM-b-9-(n300 e m30)	2676	0.50	1338	0.58	1553	0.60	1606	0.61	1635	0.62	1650
SOM-b-12-(n300 e m120)	35745	0.53	18946	0.60	21447	0.61	21756	0.63	22461	0.62	22161
SOM-b-14-(n400 e m80)	16824	0.54	9070	0.62	10442	0.63	10674	0.64	10876	0.62	10442
SOM-b-15-(n400 e m120)	36106	0.55	19858	0.63	22786	0.64	23190	0.65	23469	0.62	22342
SOM-b-16-(n400 e m160)	62217	0.56	34842	0.63	39197	0.62	38574	0.64	39859	0.62	38573
SOM-b-17-(n500 e m50)	7074	0.54	3819	0.61	4315	0.62	4384	0.64	4510	0.62	4386
SOM-b-18-(n500 e m100)	26062	0.56	14594	0.63	16459	0.62	16175	0.64	16740	0.62	16175
SOM-b-19-(n500 e m150)	56308	0.55	31069	0.61	34368	0.62	34910	0.64	35957	0.62	34910

A análise dos resultados revela uma tendência clara: à medida que mais instâncias eram incluídas no treinamento, a proximidade dos valores previstos em relação à solução ótima aumentava significativamente. Por exemplo, no Teste 1, com 100 instâncias de treinamento, a proximidade com a solução ótima variou entre 70% e 75%. No Teste 2, com 150 instâncias, essa proximidade subiu para 82% a 87%, e no Teste 3, com 200 instâncias, o desempenho dos modelos melhorou ainda mais, alcançando uma proximidade de 93% a 97%. Em alguns casos, os modelos conseguiram atingir até 100% da solução ótima, demonstrando que, com a adição de mais dados, o modelo se torna progressivamente mais capaz de identificar soluções ótimas ou muito próximas delas.

Tabela 3: Treinamento com 150 Instâncias

Instâncias de Teste	Valor Objetivo (Metaheurística)	Árvores de Decisão		Redes Neurais		Floresta Randômica		Regressão Logística		KNN	
		Acurácia	Valor	Acurácia	Valor	Acurácia	Valor	Acurácia	Valor	Acurácia	Valor
SOM-a-2-(n25 e m2)	7	0.4536	5	0.3945	6	0.4936	5	0.5205	6	0.5205	5
SOM-a-5-(n25 e m2)	7	0.6045	6	0.6110	5	0.6048	6	0.6264	6	0.6264	5
SOM-a-7-(n25 e m7)	129	0.5016	108	0.5958	109	0.6100	106	0.6271	110	0.6271	111
SOM-a-9-(n25 e m7)	125	0.5082	107	0.5870	107	0.6052	104	0.6258	109	0.6258	105
SOM-a-25-(n100 e m10)	294	0.5279	248	0.6111	256	0.6375	251	0.6279	255	0.6279	246
SOM-a-27-(n100 e m30)	2032	0.5388	1746	0.6202	1699	0.6425	1698	0.6281	1770	0.6281	1745
SOM-a-32-(n125 e m12)	406	0.5102	353	0.6232	346	0.6100	347	0.6243	364	0.6243	351
SOM-a-37-(n125 e m37)	3128	0.5242	2597	0.6395	2727	0.6507	2731	0.6090	2713	0.6090	2742
SOM-a-42-(n150 e m15)	625	0.5228	543	0.6211	531	0.6212	536	0.6124	553	0.6124	537
SOM-a-46-(n150 e m45)	4723	0.5761	3988	0.6440	4101	0.6297	3996	0.6085	4033	0.6085	4066
SOM-b-6-(n200 e m40)	3838	0.5251	3328	0.6300	3317	0.6311	3148	0.6242	3379	0.6242	3204
SOM-b-8-(n200 e m60)	13854	0.5476	12182	0.6218	12067	0.6034	11966	0.6313	12354	0.6313	11983
SOM-b-9-(n300 e m30)	2288	0.5059	1876	0.5786	2000	0.5881	1893	0.6252	1945	0.6252	1881
SOM-b-12-(n300 e m120)	30950	0.5258	26583	0.6116	26788	0.6161	26538	0.6310	26915	0.6310	25385
SOM-b-14-(n400 e m80)	14325	0.5397	12388	0.6192	12314	0.6222	12029	0.6231	12539	0.6231	11845
SOM-b-15-(n400 e m120)	30916	0.5411	26266	0.6345	26336	0.6467	26167	0.6280	26841	0.6280	26754
SOM-b-16-(n400 e m160)	51418	0.5499	44585	0.6390	44308	0.6221	45143	0.6078	44825	0.6078	44065
SOM-b-17-(n500 e m50)	6198	0.5359	5256	0.5983	5154	0.6131	5439	0.6126	5542	0.6126	5178
SOM-b-18-(n500 e m100)	22660	0.5496	19564	0.6394	19739	0.6185	18800	0.6202	20378	0.6202	19095
SOM-b-19-(n500 e m150)	48948	0.5485	43033	0.6158	42175	0.6134	42343	0.6315	43767	0.6315	41782

Esse aumento contínuo na proximidade com a solução ótima esperada reforça a ideia de que os modelos não sofreram viés de aprendizado. Em vez disso, o aumento no número de instâncias no conjunto de treinamento parece ter fortalecido a capacidade dos modelos em aprender e genera-

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

lizar melhor sobre as instâncias de teste. Em particular, a tendência de melhoria ao longo dos testes sugere que, ao contrário de ocorrer um sobreajuste (overfitting), os modelos se beneficiaram com mais dados, atingindo níveis mais altos de precisão.

Vale destacar que, em certos casos, o modelo alcançou a solução ótima mesmo com distribuições distintas de vértices. Por exemplo, em alguns grafos, selecionar pares como 1 e 5 ou 2 e 8 resultou em diversidade máxima, ainda que os vértices fossem alocados diferentemente. Isso sugere que, mesmo ao "errar"a escolha exata de vértices, o modelo soube identificar corretamente o conjunto com máxima diversidade, evidenciando sua robustez.

Tabela 4: Treinamento com 200 Instâncias

Instâncias de Teste	Valor Objetivo (Metaheurística)	Árvores de Decisão		Redes Neurais		Floresta Randômica		Regressão Logística		KNN	
		Acurácia	Valor	Acurácia	Valor	Acurácia	Valor	Acurácia	Valor	Acurácia	Valor
SOM-a-2-(n25 e m2)	6	0.4536	5	0.3945	5	0.4936	5	0.5205	5	0.5205	5
SOM-a-5-(n25 e m2)	6	0.6045	5	0.6110	5	0.6048	5	0.6264	5	0.6264	5
SOM-a-7-(n25 e m7)	114	0.5016	101	0.5958	99	0.6100	102	0.6271	108	0.6271	103
SOM-a-9-(n25 e m7)	111	0.5082	99	0.5870	101	0.6052	97	0.6258	105	0.6258	99
SOM-a-25-(n100 e m10)	258	0.5279	234	0.6111	226	0.6375	231	0.6279	255	0.6279	228
SOM-a-27-(n100 e m30)	1861	0.5388	1669	0.6202	1648	0.6425	1640	0.6281	1811	0.6281	1639
SOM-a-32-(n125 e m12)	366	0.5102	317	0.6232	319	0.6100	318	0.6243	359	0.6243	320
SOM-a-37-(n125 e m37)	2834	0.5242	2502	0.6395	2462	0.6507	2511	0.6090	2734	0.6090	2522
SOM-a-42-(n150 e m15)	548	0.5228	499	0.6211	504	0.6212	491	0.6124	542	0.6124	485
SOM-a-46-(n150 e m45)	4269	0.5761	3856	0.6440	3746	0.6297	3845	0.6085	4116	0.6085	3779
SOM-b-6-(n200 e m40)	3396	0.5251	2938	0.6300	3008	0.6311	2995	0.6242	3207	0.6242	3055
SOM-b-8-(n200 e m60)	12580	0.5476	10973	0.6218	11556	0.6034	10854	0.6313	11894	0.6313	11105
SOM-b-9-(n300 e m30)	2101	0.5059	1861	0.5786	1915	0.5881	1910	0.6252	1987	0.6252	1847
SOM-b-12-(n300 e m120)	26738	0.5258	24480	0.6116	23762	0.6161	23978	0.6310	25742	0.6310	24036
SOM-b-14-(n400 e m80)	12728	0.5397	11262	0.6192	11298	0.6222	11379	0.6231	12467	0.6231	11362
SOM-b-15-(n400 e m120)	26923	0.5411	23368	0.6345	24678	0.6467	23421	0.6280	25364	0.6280	24513
SOM-b-16-(n400 e m160)	44420	0.5499	38484	0.6390	39209	0.6221	39743	0.6078	41798	0.6078	38234
SOM-b-17-(n500 e m50)	5463	0.5359	4729	0.5983	4709	0.6131	4755	0.6126	5381	0.6126	4942
SOM-b-18-(n500 e m100)	20426	0.5496	17944	0.6394	18155	0.6185	18043	0.6202	19339	0.6202	18674
SOM-b-19-(n500 e m150)	42976	0.5485	37192	0.6158	39155	0.6134	38035	0.6315	41101	0.6315	37234

5.2. Análise de Generalização em Instâncias de Grande Escala

Após os testes com as instâncias menores, os modelos foram aplicados nas instâncias grandes da MDPLib (40 instâncias de 2000 vértices e 20 de 3000 vértices) para avaliar a capacidade de generalização.

Os modelos foram treinados com o conjunto completo de 255 instâncias (vértices de 10 a 500). Surpreendentemente, os resultados nas instâncias maiores foram ainda mais promissores, com a proximidade da solução de referência alcançando um valor mínimo de 91%, conforme mostrado nas Tabelas 5 e 6. Mesmo nestes cenários, a acurácia de classificação no conjunto de teste permaneceu em níveis moderados (entre 55% e 65%), reforçando a conclusão de que a ordenação de probabilidades é o fator chave para o desempenho.

Tabela 5: Resultados para Instâncias de Tamanho 2000

Instância de Teste	$\|V\|$	$\|S\|$	Valor Objetivo (Metaheurística)	Valor Alcançado (Modelos)	Proximidade em Porcentagem
MDG-a_21_n2000_m200	2000	200	113162	103067.95	91.08
MDG-a_22_n2000_m200	2000	200	112962	103156.90	91.32
MDG-a_23_n2000_m200	2000	200	113080	104746.00	92.63
MDG-a_24_n2000_m200	2000	200	113001	104299.92	92.30
MDG-a_25_n2000_m200	2000	200	113265	103580.84	91.45
MDG-a_26_n2000_m200	2000	200	113112	104900.07	92.74

Continua na próxima página

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

Continuação da Tabela 5

Instância de Teste	$\|V\|$	$\|S\|$	Valor Objetivo (Metaheurística)	Valor Alcançado (Modelos)	Proximidade em Porcentagem
MDG-a_27_n2000_m200	2000	200	113046	104137.98	92.12
MDG-a_28_n2000_m200	2000	200	113218	104069.99	91.92
MDG-a_29_n2000_m200	2000	200	113066	103692.83	91.71
MDG-a_30_n2000_m200	2000	200	113149	104798.60	92.62
MDG-a_31_n2000_m200	2000	200	113006	105005.18	92.92
MDG-a_32_n2000_m200	2000	200	113117	104814.21	92.66
MDG-a_33_n2000_m200	2000	200	113096	103562.01	91.57
MDG-a_34_n2000_m200	2000	200	113183	102973.89	90.98
MDG-a_35_n2000_m200	2000	200	113080	103242.04	91.30
MDG-a_36_n2000_m200	2000	200	113173	104005.99	91.90
MDG-a_37_n2000_m200	2000	200	113045	104566.63	92.50
MDG-a_38_n2000_m200	2000	200	113205	103605.22	91.52
MDG-a_39_n2000_m200	2000	200	113058	103504.60	91.55
MDG-a_40_n2000_m200	2000	200	113115	104190.23	92.11

Os resultados indicam que os modelos, quando adequadamente treinados, conseguem lidar de forma eficaz com instâncias de tamanhos maiores, com um desempenho excelente nas instâncias de 2000 e 3000 vértices. A proximidade com a solução ótima, que foi consistentemente acima de 91%, reforça a capacidade de generalização mesmo em problemas mais complexos e de maior escala.

Tabela 6: Resultados para Instâncias de Tamanho 3000

Instância de Teste	$\|V\|$	$\|S\|$	Valor Objetivo (Metaheurística)	Valor Alcançado (Modelos)	Proximidade em Porcentagem
MDG-c_1_n3000_m300	3000	300	24704176	22498093.08	91.07
MDG-c_2_n3000_m300	3000	300	24701619	22478473.29	91.00
MDG-c_3_n3000_m300	3000	300	24698742	22522782.83	91.19
MDG-c_4_n3000_m300	3000	300	24693815	22486187.94	91.06
MDG-c_5_n3000_m300	3000	300	24678847	22462686.54	91.02
MDG-c_6_n3000_m400	3000	400	43135740	39956635.96	92.63
MDG-c_7_n3000_m400	3000	400	43165190	39953699.86	92.56
MDG-c_8_n3000_m400	3000	400	43133072	39915344.83	92.54
MDG-c_9_n3000_m400	3000	400	43121905	39930884.03	92.60
MDG-c_10_n3000_m400	3000	400	43147020	39919622.90	92.52
MDG-c_11_n3000_m500	3000	500	66625758	62608224.79	93.97
MDG-c_12_n3000_m500	3000	500	66593447	62411378.53	93.72
MDG-c_13_n3000_m500	3000	500	66604075	62441320.31	93.75
MDG-c_14_n3000_m500	3000	500	66595684	62366858.07	93.65
MDG-c_15_n3000_m500	3000	500	66642407	62363964.47	93.58
MDG-c_16_n3000_m600	3000	600	95136120	90874021.82	95.52

Continua na próxima página

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

Continuação da Tabela 6 — Resultados para Instâncias de Tamanho 3000

Instância de Teste	$\|V\|$	$\|S\|$	Valor Objetivo (Metaheurística)	Valor Alcançado (Modelos)	Proximidade em Porcentagem
MDG-c_17_n3000_m600	3000	600	95096043	90873778.69	95.56
MDG-c_18_n3000_m600	3000	600	95045765	90692668.96	95.42
MDG-c_19_n3000_m600	3000	600	95132938	90928062.14	95.58
MDG-c_20_n3000_m600	3000	600	95070981	90944900.42	95.66

A análise das instâncias de tamanho 3000, por exemplo, revela que, à medida que a complexidade do grafo aumenta, os modelos continuam a apresentar alta proximidade com os valores ótimos, alcançando até 95.66% de proximidade nas instâncias com 600 vértices de conjunto. Tais resultados demonstram que o treinamento realizado com dados de tamanho menor e moderado (instâncias de 10 a 500 vértices) pode ser generalizado para problemas com instâncias significativamente maiores (2000 e 3000 vértices), garantindo um bom desempenho.

6. Conclusão

Este estudo demonstrou a viabilidade de métodos de aprendizado de máquina na resolução aproximada do Problema da Diversidade Máxima (MDP). Os modelos supervisionados alcançaram soluções com mais de 90% de proximidade aos valores de referência de uma meta-heurística de alta qualidade, especialmente em grafos de grande escala. A ênfase na maximização da diversidade, em detrimento de métricas de acurácia de classificação, revelou-se uma abordagem mais eficaz, pois a capacidade do modelo de ordenar vértices por probabilidade foi mais importante do que a sua precisão de classificação absoluta.

Como perspectivas futuras, destaca-se a exploração de arquiteturas especializadas, como redes neurais convolucionais e *Graph Neural Networks* (GNNs), capazes de capturar relações estruturais complexas em grafos. A integração dinâmica de técnicas de otimização e ajuste adaptativo de hiperparâmetros pode refinar ainda mais a qualidade das soluções, enquanto a inclusão de grafos com distribuições variadas de pesos testará a robustez dos modelos em cenários desafiadores.

Por fim, a combinação entre aprendizado de máquina e métodos de otimização tradicionais surge como uma fronteira promissora, potencializando eficiência computacional e precisão. Essa sinergia não apenas avança a resolução prática do MDP, mas também contribui para o desenvolvimento teórico de abordagens híbridas aplicáveis a problemas combinatórios de alta complexidade.

Referências

Altman, N. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Besanko, D., Perry, M., e Spady, R. (1990). The economics of product differentiation. *International Journal of Industrial Organization*, 8:431–447.

Blum, C. e Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308.

Carrabs, F., Cerulli, R., e Toth, P. (2022). New approaches for solving the maximum diversity problem in large graphs. *Journal of Heuristics*, 28(3):231–256.

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

Chawla, N. V., Bowyer, K. W., Hall, L. O., e Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Cover, T. M. e Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Cox, D. R. (1958). The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–242.

Duarte, A. e Martí, R. (2007). Tabu search and grasp for the maximum diversity problem. *European Journal of Operational Research*, 178(1):71–84.

Duarte, A., Martí, R., Sánchez-Oro, J., e Glover, F. (2015). Metaheuristics for the maximum diversity problem. *European Journal of Operational Research*, 240(1):24–36.

Erkut, E. e Neuman, S. (1991). The discrete p-dispersion problem. *European Journal of Operational Research*, 46:48–60.

Faceli, K., Lorena, A. C., Gama, J., e Carvalho, A. C. P. d. L. F. d. (2011). *Inteligência artificial: uma abordagem de aprendizado de máquina*. LTC.

Gallego, M., Duarte, A., Laguna, M., e Martí, R. (2009). Hybrid heuristics for the maximum diversity problem. *Computational Optimization and Applications*, 44(3):411–426.

Garey, M. R. e Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company.

Ghyczy, T. V. (1985). Product diversity and proliferation as a new mode of competing in the motor car. *International Journal of Vehicle Design*, 6:423–425.

Glover, F., Kuo, C., e Dhir, K. (1998). Heuristic algorithms for the maximum diversity problem. *Journal of Information and Optimization Sciences*, 19(1):109–132.

Gonzalez, J. A., Hernández-Pérez, H., e Lozano, M. (2020). A review on maximum diversity problems. *European Journal of Operational Research*, 287(2):401–420.

Goodfellow, I., Bengio, Y., e Courville, A. (2016). *Deep Learning*. MIT Press.

Gu, S. e Yang, Y. (2020). A deep learning algorithm for the max-cut problem based on pointer network structure with supervised learning and reinforcement learning strategies. *Mathematics*, 8(2). ISSN 2227-7390. URL `https://www.mdpi.com/2227-7390/8/2/298`.

He, H., Garcia, E. A., e Chen, S. (2008). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Ho, T. K. (1995). Random decision forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, volume 1, p. 278–282. IEEE.

Jolliffe, I. T. e Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

Kirkpatrick, S., Gelatt Jr, C. D., e Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598):671–680.

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br

Kuo, C. C., Glover, F., e Dhir, K. S. (1993). Analyzing and modeling the maximum diversity problem by zero-one programming. *Decision Sciences*, 24(6):1171–1185.

Lopez, F., Montero, J., e Gallego, M. (2019). An efficient grasp algorithm for solving maximum diversity problems. *Computers & Operations Research*, 101:201–211.

Martí, R., Gallego, M., e Duarte, A. (2010). A branch and bound algorithm for the maximum diversity problem. *European Journal of Operational Research*, 200(1):36–44.

Palubeckis, G. (2007). Iterated tabu search for the maximum diversity problem. *Applied Mathematics and Computation*, 189(1):371–383.

Parreno, F., Mortes, J., e Martí, R. (2024). Measuring diversity: A review and an empirical analysis. *European Journal of Operational Research*.

Porter, R. et al. (1975). *The Genetics of Livestock Improvement*. Academic Press.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1(1):81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Shi, Y. e Zhang, Y. (2022). The neural network methods for solving traveling salesman problem. *Procedia Computer Science*, 199:681–686. ISSN 1877-0509. The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 & 2021): Developing Global Digital Economy after COVID-19.

Silva, G., Ochi, L., e Martins, S. (2004). Experimental comparison of greedy randomized adaptive search procedures for the maximum diversity problem. In *Experimental and Efficient Algorithms*, p. 498–512. Springer.

Soares, P. L. B. (2018). *Métodos exatos para problemas quadráticos binários*. PhD thesis, Universidade Federal de Ceará.

DOI: 10.59254/sbpo-2025-212180
https://proceedings.science/p/212180?lang=pt-br