UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS RUSSAS

CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

DAVI MAIA ANDRADE DA SILVA

APLICAÇÃO DE APRENDIZAGEM NÃO SUPERVISIONADA NO PROBLEMA DA DIVERSIDADE MÁXIMA EM GRAFOS

RUSSAS

2023

DAVI MAIA ANDRADE DA SILVA

APLICAÇÃO DE APRENDIZAGEM NÃO SUPERVISIONADA NO PROBLEMA DA DIVERSIDADE MÁXIMA EM GRAFOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: Prof. Dr. Pablo Luiz Braga Soares

RUSSAS

2023

DAVI MAIA ANDRADE DA SILVA

APLICAÇÃO DE APRENDIZAGEM NÃO SUPERVISIONADA NO PROBLEMA DA DIVERSIDADE MÁXIMA EM GRAFOS

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus Russas da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Pablo Luiz Braga Soares (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dra. Tatiane Fernandes Figueiredo
Universidade Federal do Ceará (UFC)

Prof. Dr. Eurinardo Rodrigues Costa
Universidade Federal do Ceará (UFC)

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S579a Silva, Davi Maia Andrade da.
Aplicação de aprendizagem não supervisionada no problema da diversidade máxima em grafos / Davi Maia Andrade da Silva. – 2023.
34 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2023.
Orientação: Prof. Dr. Pablo Luiz Braga Soares.

1. problema da diversidade máxima. 2. aprendizado de máquina. 3. modelos não supervisionado. 4. otimização combinatória. I. Título.

CDD 005

À minha Mãe, Valdenira, pelos seus conselhos e carinhos que sempre me deram forças pra seguir, mesmo diante das adversidades. Pai, seus puxões de orelha, seus incentivos e seus exemplos me motivaram muito para vivenciar este momento.

"Educação não transforma o mundo. Educação muda as pessoas. Pessoas transformam o mundo."

(Paulo Freire)

RESUMO

Este trabalho tem como foco à aplicação de modelos de aprendizado de máquina (*Machine Learning*) ao Problema da Diversidade Máxima (PDM) em grafos. O PDM na área de otimização combinatória, consiste em selecionar um subconjunto de q elementos de um conjunto de n elementos, de tal forma que a diversidade entre os seus elementos selecionados seja máxima. O modelo de aprendizagem de máquina utilizados neste trabalho é baseado em aprendizado não supervisionado, que tem como característica aprender por si só, encontrado significado ou padrões aos dados apresentados. Os resultados do modelo foram comparados com uma heurística e meta heurística da literatura que possua os melhores resultados para os grafos utilizados.

Palavras-chave: problema da diversidade máxima; aprendizado de máquina; modelos não supervisionado; otimização combinatória.

ABSTRACT

This study focuses on the application of machine learning models to the PDM in graphs. The PDM, in the field of combinatorial optimization, involves selecting a subset of q elements from a set of n elements in a way that maximizes diversity among the chosen elements. The machine learning model employed in this research is based on unsupervised learning, characterized by its ability to autonomously discover meaning or patterns within presented data. Comparative analysis was conducted between the model's outcomes and a heuristic as well as a meta-heuristic from the literature, known for delivering optimal results on the employed graphs.

Keywords: maximum diversity problem; machine learning; unsupervised models; combinatorial optimization.

LISTA DE FIGURAS

Figura 1 – Exemplo de Grafo 14
Figura 2 – Exemplo de Grafo Completo 15
Figura 3 – Exemplo de Grafo Direcionado 15
Figura 4 – Exemplo de Grafo Não Direcionado 16
Figura 5 – Exemplo de Grafo Ponderado 16
Figura 6 – Fluxo de lógica do algorítimo 18
Figura 7 – Exemplo de seleção de um subgrafo de tamanho 4 com maior diversidade . 19
Figura 8 – Exemplo de Grafo Ponderado 24
Figura 9 – Fluxo de lógica do algorítimo 26
Figura 10 – Relação tempo e tamanho do subgrafo para as instâncias SOM*b* de tamanho 100 28
Figura 11 – Relação tempo e tamanho do subgrafo para as instâncias SOM*b* de tamanho 200 28
Figura 12 – Relação tempo e tamanho do subgrafo para as instâncias SOM*b* de tamanho 300 29
Figura 13 – Comparativo entre as instâncias SOM*b* de tamanho 100, 200 e 300 29

LISTA DE TABELAS

Tabela 1 – Comparativo dos trabalhos 23
Tabela 2 – Resultados do algoritmo desenvolvido neste estudo 27
Tabela 3 – Comparação dos melhores valores 30

LISTA DE ABREVIATURAS E SIGLAS

ECLAT	*Equivalence Class Clustering and bottom-up Lattice Traversal*
PDM	Problema da Diversidade Máxima
SS	*Scatter search*
TRL	Técnica de Reformulação de Linearização
UBQP	*Unconstrained Binary Quadratic Program*

SUMÁRIO

1	**INTRODUÇÃO**	11
2	**OBJETIVOS**	13
2.1	**Objetivos Gerais**	13
2.2	**Objetivos Específicos**	13
3	**FUNDAMENTAÇÃO TEÓRICA**	14
3.1	**Conceitos Básicos sobre Grafos**	14
3.2	**Aprendizado de Máquina**	17
3.2.1	***Aprendizado Não Supervisionado***	17
3.3	**Problema da Diversidade Máxima(PDM)**	19
4	**TRABALHOS RELACIONADOS**	20
4.1	**Um algoritmo exato para o Problema da Diversidade Máxima**	20
4.2	**Usando Medidas Estatísticas e Aprendizado de Máquina para Redução de Grafos para Resolver Problemas de Cliques de Peso Máximo**	20
4.3	**Uma nota sobre abordagem heurística baseada na formulação UBQP do problema de máxima diversidade**	21
4.4	**Heurística híbrida para o problema de máxima diversidade**	21
4.5	**Comparativo dos trabalhos relacionados**	22
5	**METODOLOGIA**	24
5.1	**Coleta das Instâncias do Problema**	24
5.2	**Implementação e Execução**	25
5.3	**Coleta e Análise dos Resultados**	26
5.3.1	***Coleta***	27
5.3.2	***Análise dos Resultados***	27
6	**CONCLUSÃO**	31
6.1	**Considerações gerais**	31
6.2	**Trabalhos futuros**	31
	REFERÊNCIAS	32

1 INTRODUÇÃO

Diversidade significa variedade, pluralidade e diferença. É um substantivo feminino que caracteriza tudo que é diverso, que tem multiplicidade. Diversidade é a reunião de tudo aquilo que apresenta múltiplos aspectos e que se diferenciam entre si, por exemplo: diversidade cultural, diversidade biológica, diversidade étnica, linguística, religiosa e etc. Segundo Takane (2011), a depender do contexto, ela é importante para o pluralismo, heterogeneidade, tolerância mútua e sobrevivência de ideias. Sua carência pode ocasionar diminuição de bens intangíveis, prejuízo financeiro ou até extinção. Por exemplo, na engenharia genética, ela é indispensável para a sobrevivência ou extinção de uma espécie.

Duarte e Marti (2007) descrevem aplicações do Problema da Diversidade Máxima (PDM) em áreas como ecologia e engenharia genética. Na área da ecologia, numa aplicação envolvendo a preservação de sistemas ecológicos, as melhores condições de sobrevivência do sistema são alcançadas quando se maximiza a diversidade das espécies nele presentes. Na engenharia genética, alguns problemas de desenvolvimento de novas espécies podem ser formulados como um problema de PDM, onde se deseja controlar a diversidade do estoque de reprodução. Em Gallego *et al.* (2009) cita que tem aplicações em melhoramento de plantas, problemas sociais, preservação ecológica, controle de poluição, *design* de produto, investimento de capital, gerenciamento de força de trabalho, *design* de currículo e engenharia genética.

O PDM em grafos consiste em, dado um grafo ponderado $G = (V, E)$ com um conjunto de vértices $|V| = n$, um conjunto de arestas $|E| = m$ e pesos $c_{ij} \in \mathbb{R}, \quad \forall\{i, j\} \in E$, encontrar um subgrafo completo de tamanho $q < n$. Este subgrafo deve possuir uma diversidade, que é representada pela distância entre dois elementos $i, j \in G$, tal que a soma dos pesos das arestas seja a maior possível. Este problema pode ser aplicado na seleção da amostragem representativa de indivíduos com características diversas (país de origem, etnia, sexo e religião) (KUO *et al.*, 1993). Nesse caso, geralmente, deseja-se escolher um determinado número de vértices, que apresentam a maior diversidade possível entre si.

Na literatura existem alguns trabalhos que usam métodos exatos para resolver o problema, tais como Soares *et al.* (2018), Takane (2011) e KUO *et al.* (1993). No entanto pelo fato do PDM ser um problema NP-Difícil (KARP, 1975), estes e outros métodos exatos possuem limitação quanto a resolução de grafos com tamanhos moderados e grandes. Para essa classe de problemas difíceis, onde o PDM se encontra, os algoritmos heurísticos e meta-heurísticos são na maioria das vezes preferidos para encontrar soluções, possívelmente, quase ótimas em um

período de tempo aceitável. Em contrapartida, o aprendizado de máquina explora o estudo e a construção de algoritmos que podem aprender com seus erros e fazer previsões sobre dados.

Neste trabalho iremos usar o algoritmo *k-means* que é baseado na abordagem de aprendizado de máquina não supervisionado como alternativa para encontrar boas soluções para o PDM em grafos de tamanhos moderados e grandes.

Este trabalho está dividido em cinco capítulos. No capítulo dois, é exposto os objetivos que estão divididos em gerais e específicos. No capítulo três, é mostrada a fundamentação teórica, que descreve os conceitos chaves e relevante para um melhor entendimento do corrente trabalho. Já no capítulo quatro, é apresentado alguns trabalhos relacionados, no Capítulo cinco é apresentado a metodologia. Etapas, necessárias para realização desta pesquisa e o por fim no último capítulo, temos obtidos e as considerações finais.

2 OBJETIVOS

Neste capítulo é apresentado os objetivos almejados com o desenvolvimento deste trabalho. Na subsecção 2.1 são descritos os objetivos gerais e na subsecção 2.2 são descritos os objetivos específicos desta pesquisa.

2.1 Objetivos Gerais

Aplicar aprendizado de máquina não supervisionado no Problema de Diversidade Máxima e comparar os resultados obtidos com instâncias de tamanhos moderados que utilizam meta heurísticas para solucionar o PDM.

2.2 Objetivos Específicos

- Utilizar uma base de dados, já existente, composta de grafos ponderados de diversos tamanhos;
- Aplicar o modelo de aprendizado de máquina não supervisionado *k-means* na geração de *clusters* iniciais;
- Adequar os *clusters* ao tamanho do subgrafo esperado;
- Comparar os resultados do modelo desenvolvido com os métodos meta heurísticos existentes para o problema da diversidade máxima.

3 FUNDAMENTAÇÃO TEÓRICA

Nesta secção serão discutidos conceitos fundamentais para o entendimento da solução proposta neste trabalho. Nas secções 3.1, 3.2 e 3.3 encontra-se, respectivamente, uma explicação sobre Grafos, Aprendizado de Máquina e Problema da Diversidade Máxima.

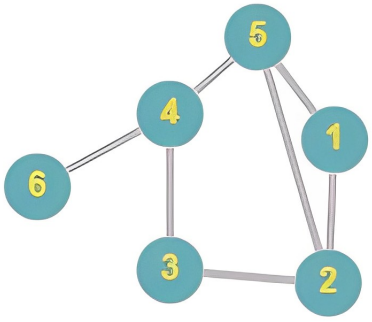
3.1 Conceitos Básicos sobre Grafos

Um grafo (simples) G consiste de um conjunto finito e não vazio $V(G)$ de objetos chamados vértices, juntamente com um conjunto $E(G)$ de pares não ordenados de vértices, os elementos de $E(G)$ são chamados de arestas. Podemos denotar $G = (V,E)$, onde $V = V(G)$ e $E = E(G)$.

Se $G = (V,E)$ é um grafo e u e v dois de seus vértices, diremos que u e v são adjacentes ou vizinhos se $\{u,v\} \in E$. Neste caso, dizemos ainda que a aresta $\{u,v\}$ incide nos vértices u e v. Podemos denotar por simplicidade a aresta $\{u,v\}$ por uv ou vu, sempre que não houver perigo de confusão. Se u e v não forem adjacentes, dizemos que são vértices não adjacentes de G.

Grafos são geralmente representados por diagramas, onde os elementos de V correspondem a pontos no plano e as arestas de G correspondem a arcos ligando os vértices correspondentes. Seu propósito sendo somente o de representar esquematicamente as relações de adjacência entre os vértices de G. Por exemplo, se $G = (V,E)$, tal que, $V = \{1,2,3,4,5,6\}$ e $E = \{\{1,2\},\{1,5\},\{2,3\},\{2,5\},\{3,4\},\{4,5\},\{4,6\}\}$, então G pode ser representado por qualquer diagrama, haja vista que mantenham as relações de adjacência, tal como na Figura 1.

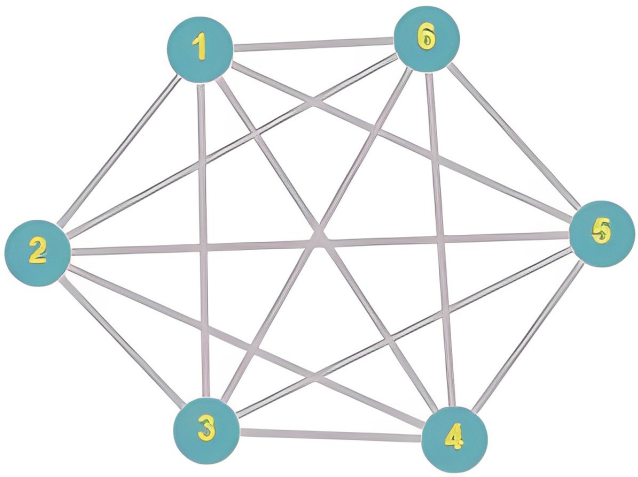
Figura 1 – Exemplo de Grafo



Fonte: Figura desenvolvida pelo autor

Um grafo completo é um grafo que possuí n vértices e todos conectados dois a dois por arestas. Ou seja, em outras palavras, um grafo completo é um grafo simples que contém o número máximo de arestas possíveis. Um grafo deste tipo com n vértices é denotado por K_n, em particular, K_n tem exatamente $n(n\text{ - }1)/2$ arestas. A Figura 2 mostra um desenho do grafo K_6 com 6 vértices e 15 arestas.

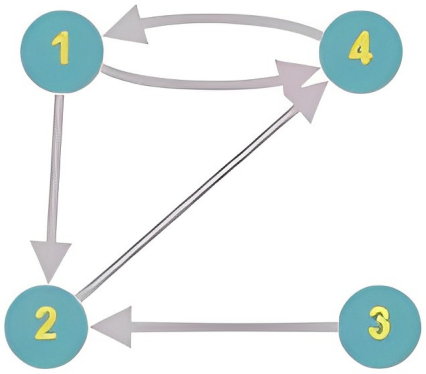
Figura 2 – Exemplo de Grafo Completo



Fonte: Figura desenvolvida pelo autor

Os digrafos são representados através de um diagrama onde os vértices são representados por pontos e cada aresta (v_i, v_j) é representada por uma linha ligando v_i a v_j com uma seta apontando para v_j. Na Figura 3 vemos a representação do digrafo $D = (V, A)$, onde $D = \{1,2,3,4\}$ e $A = \{1-4, 1-3, \ldots\}$. Por simplicidade podemos representar um digrafo não direcionado da mesmo que um grafos simples como na figura 3 do lado direito.

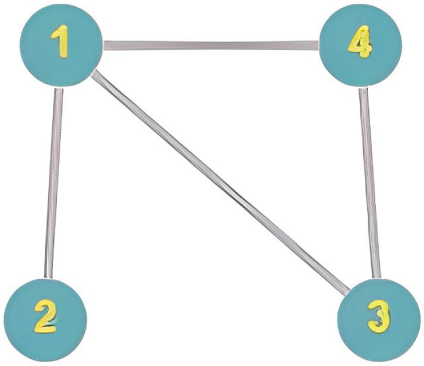
Figura 3 – Exemplo de Grafo Direcionado



Fonte: Figura desenvolvida pelo autor

Um grafo é não dirigido ou não direcionado,também conhecido por digrafo, se cada um de suas arestas é antiparalela a alguma outra. Para cada aresta $u - v$, o digrafo também tem a aresta $v - u$. Num digrafo não-dirigido, a relação de adjacência é simétrica, um vértice u é adjacente a um vértice v se e somente se v é adjacente ou vizinho a u. (O mesmo se aplica ao sinônimo "vizinho" de "adjacente"). Podemos observar na Figura 4 que não há indicação de caminho com uma seta, ou seja, existe uma relação de adjacência, sendo esse o não direcionado.

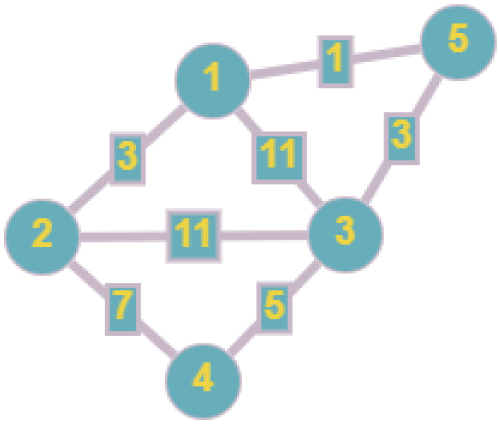
Figura 4 – Exemplo de Grafo Não Direcionado



Fonte: Figura desenvolvida pelo autor

Na Figura 5 pode ser visto um exemplo de grafo ponderado. Este tipo de grafo é definido como os grafos que possuem valoração, peso ou números reais nas suas arestas. De modo semelhante temos o digrafo ponderado.

Figura 5 – Exemplo de Grafo Ponderado



Fonte: Figura desenvolvida pelo autor

3.2 Aprendizado de Máquina

Samuel (1959) definiu aprendizado de máquina como o campo de estudo que dá aos computadores a habilidade de aprender sem ser explicitamente programado. Já Mitchell (1997) definiu afirmando que um programa de computador aprende pela experiência E em relação a algum tipo de tarefa T e alguma medida de desempenho P se o seu desempenho em T, conforme medido por P, melhora com a experiência E.

Desta forma, podemos definir Aprendizado de Máquina ou *Machine Learning* como a utilização de algoritmos com a finalidade de extrair informações de dados brutos e representá-los por meio de algum tipo de modelo matemático. Este modelo é então usado para fazer inferências, ou predições, a partir de novos conjuntos de dados. Modelos de aprendizado de máquina buscam, em geral, descobrir padrões ou fórmulas matemáticas que expliquem o relacionamento entre os dados, e estuda formas de automatização de tarefas inteligentes que seriam difíceis ou até mesmo impossíveis de serem realizadas manualmente por seres humanos (ESCOVEDO, 2020).

Os sistemas de Aprendizado de Máquina podem ser classificados de acordo com a quantidade e o tipo de supervisão que recebem durante o treinamento. Existem quatro categorias principais: aprendizado supervisionado, aprendizado não supervisionado, aprendizado semi-supervisionado e aprendizado por reforço.

3.2.1 Aprendizado Não Supervisionado

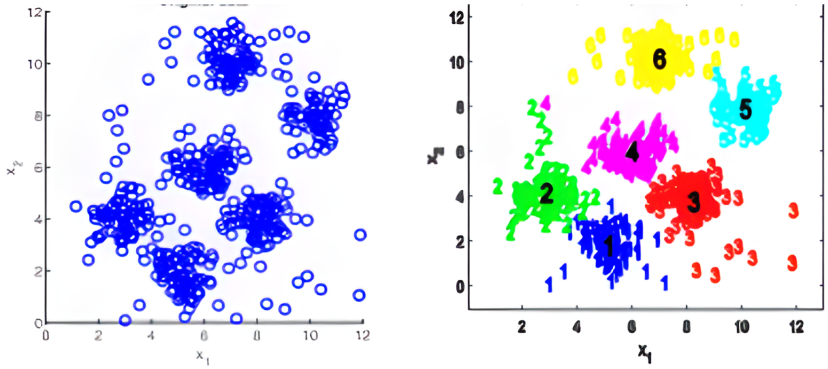
O aprendizado não supervisionado, os dados de treinamento são não rotulados e dessa forma, o modelo tenta aprender sem um professor. É usado para identificar novos padrões e detectar anomalias. O modelo tenta dar sentido aos dados por conta própria, encontrando recursos e padrões. Para isso, existem algumas técnicas que utilizam regras de associação e clusterização. Essas técnicas podem ser aplicadas na fase de pré-processamento (ou mineração de dados), para se encontrar anomalias nos dados (os *outliers*), realizar redução de dimensionalidade em *features*, etc. Elas podem ser aplicadas também sobre amostras de um problema durante um processo de agrupamento de instâncias (VICERI, 2020).

Existe alguns tipos de métodos de aprendizado não supervisionado que serão citados e definidos a seguir: Primeiramente, a Rede de Kohonen ou Mapas Auto Organizáveis que tem como princípio a aprendizagem competitiva, simulando processos específicos do cérebro humano na aprendizagem por respostas sensoriais (HAYKIN, 2001). Este algoritmo organiza os

neurônios em vizinhanças locais. Então, de acordo com Braga *et al.* (2007), cada vez que um novo padrão é apresentado à rede, os neurônios competem entre si para ver quem gera a melhor saída. Escolhido o neurônio vencedor e seus vizinhos dentro de um raio ou área de vizinhança atualizam seus pesos. Uma observação relevante durante o treinamento, a taxa de aprendizagem e o raio de vizinhança são decrementados a medida que o algoritmo vai sendo executado.

Por seguinte, pode ser citado o método *k-means* que tem o objetivo de criar partições de uma população $n-dimensional$ em k grupos em uma dada base de dados. O algoritmo *k-means* utiliza um parâmetro de entrada k, que determina a quantidade de *clusters*. Segundo Sousa e Esmin (2011), após a execução do *k-means*, pretende-se obter uma alta similaridade dos elementos de um grupo e baixa similaridade entres os *clusters* criados pelo algoritmo. Assim como pode ser visto na Figura 6 (a) apresenta um conjunto de 50 pontos e na Figura 6 (b) está posto como fica após a aplicação do *k-means* quando aplicado com a entrada de $k = 6$ *clusters*.

Figura 6 – Fluxo de lógica do algorítimo



Fonte: Sinaga e Yang (2020)

Finalmente descrevemos o *Equivalence Class Clustering and bottom-up Lattice Traversal* (ECLAT), algoritmo que usa uma abordagem de busca em profundidade. Isso significa que o ECLAT realiza a busca de forma vertical em todo o conjunto de dados. Ele começa no nó raiz, depois vai um nível de profundidade e continua até atingir a primeira nota terminal. Digamos que o nó terminal esteja no nível X. Um nó terminal inicial é alcançado, o algoritmo então dá um passo para trás e atinge o nível $(X-1)$ e continua até encontrar um nó terminal novamente.

3.3 Problema da Diversidade Máxima(PDM)

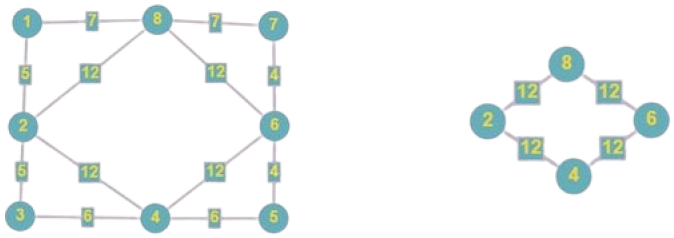
O objetivo do PDM é selecionar, a partir de um conjunto $|N|$ com um tamanho n, um subconjunto $Q \subseteq N$ com q elementos que possuam a maior diversidade entre si. Uma medida de diversidade entre dois elementos pode ser representada pela distância entre eles. Desta forma, para definir o problema, considera-se o conjunto de índices $N = \{1, 2, \ldots, n\}$, e:

- t é o número de atributos que caracterizam os elementos de N;
- a_{ik}, o valor do atributo do elemento i, onde i, N e $k \in \{1, \ldots, t\}$
- $d(i, j)$, o índice de diversidade entre dois elementos distintos, i e j que é calculado pelos, respectivos, valores de atributos $(a_{i1}, a_{i2}, \ldots, a_{it})$ e $(a_{j1}, a_{j2}, \ldots, a_{jt})$.

Segundo Glover *et al.* (1998), o PDM pode ser aplicado nas mais diversas áreas, tais como: no ramo da genética animal e vegetal, quando temos o objetivo de obter novas variedades através da reprodução de indivíduos distintos de forma controlada; no equilíbrio ambiental; no desenho de produto; no gerenciamento de força de trabalho e engenharia genética.

O PDM em grafos consiste em dado um digrafo ponderado $G = (V, E)$ com $|V| = n$, $|E| = m$ e pesos $c_{ij} \in \mathbb{R}, para todo \forall \{i, j\} \in E$, encontrar um subgrafo completo, considerando que as arestas não exibidas possuem a valoração zero, de tamanho $q < n$. Este subgrafo deve possuir uma diversidade entre os elementos selecionados tal que a soma dos pesos das arestas seja a maior possível. A Figura 7 mostra dois grafos ponderados. O grafo do lado esquerdo possui 8 vértices e 12 arestas, e o grafo que está na direita é um subgrafo de tamanho 4, que gera a maior diversidade de valor 48. Importante mencionar que o grafo B é considerado completo ao adicionar as arestas $(4, 8)$ e $(2, 6)$ com peso zero.

Figura 7 – Exemplo de seleção de um subgrafo de tamanho 4 com maior diversidade



Fonte: Figura desenvolvida pelo autor

4 TRABALHOS RELACIONADOS

Neste capítulo será apresentado artigos disponíveis na literatura que abordam técnicas que utilizamos para no glsPDM em grafos e trabalhos que desenvolveram métodos para solucionar o glsPDM em grafos.

4.1 Um algoritmo exato para o Problema da Diversidade Máxima

O trabalho de Takane (2011) apresenta a construção de um método exato, que é baseado na Técnica de Reformulação de Linearização (TRL) e na aplicação do método de Decomposição de *Benders* Revisado que possui um pré-processamento. Inicialmente são feitas três formulações matemática para o problema. Logo após é realizada a aplicação do TRL sobre estas equações, gerando um resultado não tão satisfatório. Assim foi aplicado o método de Decomposição de *Benders* Revisado com pré-processamento. Após todo esse processo de formulação foi feito o método exato para a solução do PDM.

Por fim, revelaram os resultados computacionais tendo como base de dado o tempo e o GAP de otimalidade. As soluções apresentadas mostram que esse algoritmo é capaz de resolver problemas de até 150 elementos com 15 selecionados com menos de 1 hora de processamento, demonstrando ser competitivo frente aos métodos propostos na literatura para solução do problema.

4.2 Usando Medidas Estatísticas e Aprendizado de Máquina para Redução de Grafos para Resolver Problemas de Cliques de Peso Máximo

Foi esclarecido no artigo Yuan *et al.* (2021) o uso de medidas estáticas e aprendizado de máquina para redução do problema de clique de peso máximo, removendo vértices de uma instância do problema que são improváveis de fazerem parte da solução ótima.

Primeiramente usaram medidas estatísticas calculadas a partir de amostragem estocástica de soluções viáveis para avaliar a qualidade de cada decisão. Em seguida desenvolveram uma abordagem de aprendizado de máquina para redução de problemas, usando um modelo de aprendizado supervisionado pronto para treinar instâncias de problemas fáceis nas quais pertencem a solução ótima. Desta forma sendo possível prever melhor as variáveis de decisão que fazem parte da solução ótima para um determinado problema difícil.

Por fim, concluíram que o conhecimento aprendido com grafos fáceis é útil para

reduzir o tamanho do problema. Foi feita também, uma avaliação das técnicas de reduções aplicadas no problema usando experimentos de simulação,no qual foi possível mostrar a eficácia nos métodos de soluções existentes através do aumento de desempenho.

4.3 Uma nota sobre abordagem heurística baseada na formulação UBQP do problema de máxima diversidade

A principal contribuição do artigo de Alidaee e Wang (2016) é fornecer um conjunto de abordagens de diversificação altamente eficazes para o PDM que podem ser implementadas dentro de meta-heurísticas. Com base na formulação *Unconstrained Binary Quadratic Program* (UBQP) do problema, foi fornecido otimalidade local para procedimentos de melhoria de *r-flip*. Uma nova abordagem de diversificação baseada na sequência para implementar essas regras de melhoria é apresentada.

Primeiro forneceram vários resultados teóricos para a otimização local da formulação UBQP do PDM. Em seguida, inidicaram uma regra de otimalidade local *r-flip* para o problema. Uma implementação simples e eficaz de uma regra de *2-flip* para melhoria da busca local é aplicada. Por seguinte, uma regra simples de uma inversão e uma regra de duas inversões que geram soluções ótimas locais são fornecidas. A estratégia de diversificação, que é baseada na sequência de implementação da regra de melhoria de *2-flip*, é apresentada. Em seguida, expõe a implementação combinada da diversificação com uma estratégia *multi-start* dentro de uma simples busca.

Para conclusão, apresentaram um conjunto altamente eficaz de abordagens de diversificação baseadas no procedimento de sequenciamento. Foram consideradas quatro versões de implementação das abordagens combinadas com três estratégias *multi-start* dentro de uma simples busca tabu. Foi mostrado um extenso experimento computacional em 140 problemas de *benchmark* que estão disponíveis na *Internet*. Por fim, os resultados computacionais mostraram que os procedimentos foram altamente eficazes, os métodos produziram novas melhores soluções para vinte e dois dos grandes problemas.

4.4 Heurística híbrida para o problema de máxima diversidade

No trabalho do Gallego *et al.* (2009) foi esclarecido que a *Scatter search* (SS) ou busca por dispersão é uma estrutura meta heurística que explora espaços de solução por

desenvolvimento de um conjunto de pontos de referências. A busca começa com a aplicação de um método de geração de diversificação que resulta em uma população de pontos a partir da qual um subconjunto é selecionado como o conjunto de referência inicial. A evolução do conjunto de referência é induzida pela aplicação de quatro métodos adicionais: geração de subconjunto, combinação, melhoria e atualização.

O objetivo deste experimento é testar a aplicação seletiva do método de melhoria. Como a execução do método de melhoria é computacionalmente cara, aplicá-lo a todas as soluções pode impedir que a pesquisa visite soluções adicionais durante o tempo de busca alocado, testaram a ideia de aplicar seletivamente o método de melhoria a um subconjunto das soluções que são visitadas durante a busca. Uma vez que estas soluções foram adicionadas ao conjunto de referência inicial, então o método de melhoria é aplicado para apenas estas soluções. Da mesma forma, após a aplicação do método de combinação, o método de melhoria não é aplicado a todas as soluções resultantes. Em vez disso, o melhor é selecionado e o método de melhoria é aplicado apenas a este subconjunto.

Por fim, julgam que o trabalho é o primeiro a testar vários procedimentos hibridizados dentro da estrutura de busca por dispersão. Acreditam que o aumento de desempenho obtido pelo uso de mecanismos de memória simples (alguns baseados em informações recentes e outros baseados em frequência) dentro de um projeto de pesquisa de dispersão é uma lição valiosa para futuras implementações.

4.5 Comparativo dos trabalhos relacionados

A Tabela 1 mostra um comparativo entre os trabalhos relacionados e este trabalho utilizando as seguintes medidas: se o trabalho usa aprendizado de máquina; se é aplicado em grafos; se o trabalho usa aprendizado de máquina; se é aplicado ao PDM; se o método é exato ou de solução aproximada.

Tabela 1 – Comparativo dos trabalhos

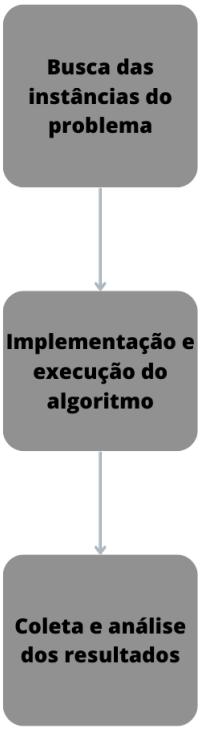
	Aprendizado de Máquina	Aplicado em Grafos	Aplicado ao PDM	Método
Takane (2011)	Não	Não	Sim	Exato
Yuan *et al.* (2021)	Sim	Sim	Não	Heurística
Alidaee e Wang (2016)	Não	Não	Sim	Heurística
Gallego *et al.* (2009)	Não	Sim	Sim	Heurística
Este Trabalho	Sim	Sim	Sim	Heurística

Fonte: Elaborada pelo autor.

5 METODOLOGIA

Esta seção descreve as etapas necessárias para realização desta pesquisa como pode ser observado na Figura 8.

Figura 8 – Exemplo de Grafo Ponderado



Fonte: Figura desenvolvida pelo autor

5.1 Coleta das Instâncias do Problema

Neste trabalho foram utilizadas as instâncias MDPLIB, mostradas na Tabela **??**, a compilação de um conjunto abrangente de instâncias de referência representativas das coleções usadas anteriormente para experimentos computacionais no PDM. A MDPLIB contém 315 instâncias e os valores mais conhecidos e os maiores são: SOM-b, GKD-c, MDG-a, MDG-b e MDG-c. Entre estes utilizamos um destes conjuntos, onde os tamanhos das instâncias são:

- SOM-b, tais que para $n = 100$, $q \in \{10, 20, 30, 40\}$; para $n = 200$, $q \in \{20, 40, 60, 80\}$; para $n = 300$, $q \in \{30, 60, 90, 120\}$; para $n = 400$, $q \in \{40, 80, 120, 160\}$; e para $n = 500$, $q \in \{50, 100, 150, 200\}$.

As instâncias selecionadas para experimentos neste trabalho, bem como o número de vértices e o

tamanho do subgrafo q, podem serem vistas na Tabela 2.

5.2 Implementação e Execução

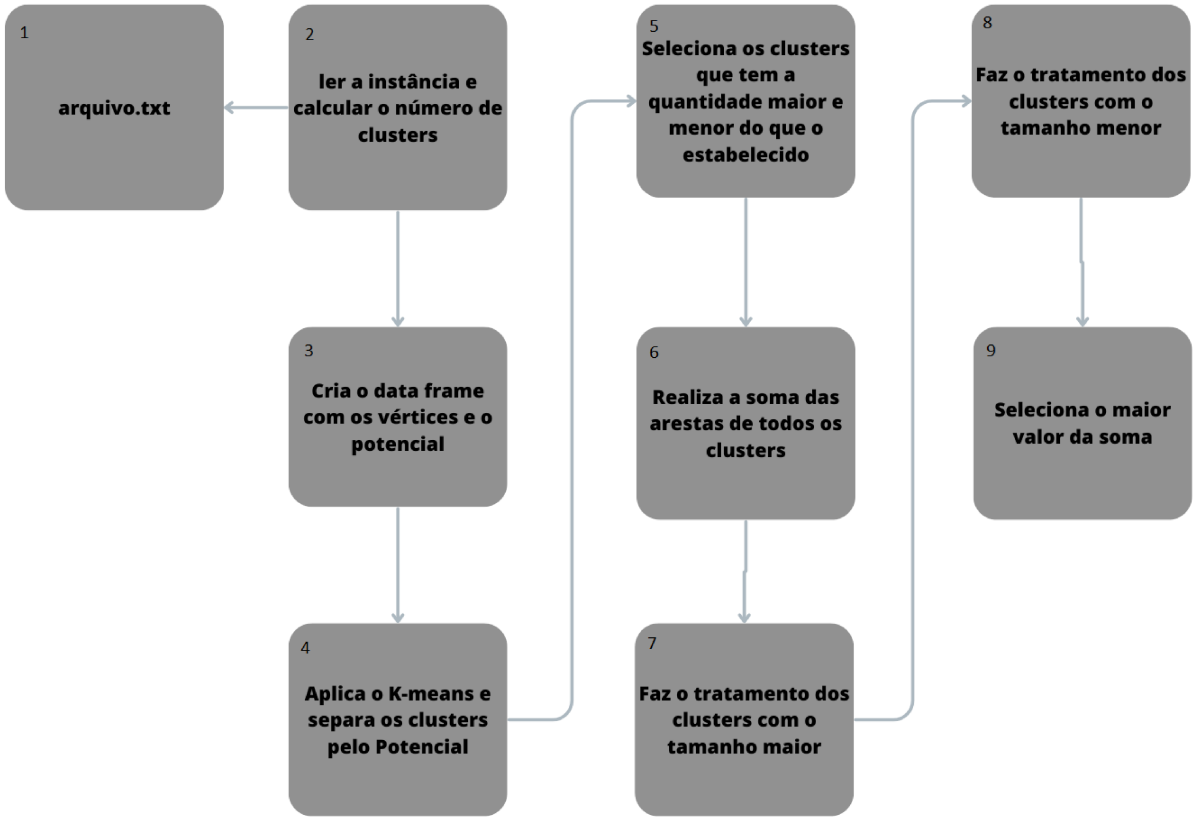
A implementação e execução do algoritmo foi feita na plataforma *Colab* e foi utilizada a linguagem de programação *Python*.

Na implementação do código foi seguida uma ordem de desenvolvimento, onde esta gerou uma lógica que proporcionou um conjunto de funções/passos, como pode ser visto no fluxo da Figura 9.

1. O aquivo ".txt", a matriz, que é enviado para o algorítmo.
2. Realiza a leitura onde armazena o valor da primeira linha que esta indica a quantidade de vértice e o tamanho do subgrafo desejado, logo após faz o calculo do números de *clusters*, que obtemos dividindo a quantidade de vértices da instância pelo tamanho do subgrafo.
3. Posteriormente criamos o *Data Frame* no qual mostra os vértices e seus respectivos potenciais, que é a soma dos valores de todas arestas ligadas ao determinado vértice.
4. Aplicamos o *K-means* no *Data Frame* e este se encarrega de fazer o agrupamento de acordo com os potenciais apresentados, tendo assim os primeiros subgrafos.
5. Em seguida, é feito uma seleção destes subgrafos, separando aqueles que tem um tamanho maior e ou menor do que o estabelecido na instância.
6. Depois é feita a soma da diversidade de cada subgrafo, onde é calculada a soma dos valores das arestas que pertencem a cada subgrafo. Geramos uma combinação de todos os vértices do subgrafo para saber quais as arestas pertencem àquele subgrafo, por conseguinte, abrimos o arquivo da instância para identificar os valor de todas as estas arestas. O valor de cada subgrafo é adicionado a uma lista.
7. Logo depois foi desenvolvido um tratamento para os subgrafos que possui o tamanho do vértice maior que o estabelecido. O algoritmo retira um vértice e faz a verificação da sua diversidade, aquele que mantiver maior a soma dos valores de suas arestas será o retirado do subgrafo e posto em uma lista. Isso é feito até os subgrafos ficarem com o tamanho solicitado.
8. Na sequência, nos subgrafos que possuem o tamanho menor do que o setado pela instância, o algoritmo adiciona os vértices que foram retirados dos subgrafos que eram maiores e estão armazenados em uma lista. De forma que sempre pegam os primeiros da lista.
9. Por fim, refaz a soma da diversidades dos subgrafos que foram tratados, atualiza a lista

que contém os valores desta soma e chamamos um método que seleciona o maior valor da lista e assim obtemos o valor mais diverso.

Figura 9 – Fluxo de lógica do algorítimo



Fonte: Figura desenvolvida pelo autor

A execução do algoritmo ocorreu na plataforma *Colab* do *Google* com *Phyton 3* e a máquina virtual possuía 13,55GB de RAM e 131,9GB de Disco. As instâncias que foram seleciondas para serem utilizadas podem serem vistas na Tabela **??**. Esta descreve a quantidade de vértice de cada instância juntamente ao tamanho do subgrafo e o valor da sua maior diversidade que foi obtido pelos métodos:

- G_SS : Algoritmo de busca de dispersão com estruturas de memória. Gallego *et al.* (2009).

5.3 Coleta e Análise dos Resultados

Dada e execução do algoritmo com as instâncias selecionadas, os dados foram coletados e armazenados em tabelas. No entanto foram construídas três tabelas nas quais possuem quinze linhas. Os resultados encontrados nas execuções estão na tabela 2 e podem serem vistos na Subseção 5.3.1. Por fim, na Subseção 5.3.2, é posto a análise detalhada dos

resultados.

5.3.1 Coleta

Na Tabela 2 é possível ver as informações coletadas na execução das instância no algoritmo desenvolvido nesta monografia, onde é observável o nome da instância, a quantidade de vértices(n), o tamanho do subgrafo(q), o valor obtido e o tempo de execução.

Tabela 2 – Resultados do algoritmo desenvolvido neste estudo

Nome da Instância	n	q	Valor obtido	Melhor Valor Método G_SS	Tempo de execução
SOMb_1	100	10	242	333	24s
SOMb_2	100	20	875	1195	60s
SOMb_3	100	30	2016	2457	240s
SOMb_4	100	40	3661	4142	720s
SOMb_5	200	20	924	1247	720s
SOMb_6	200	40	3614	4450	3300s
SOMb_7	200	60	8949	9437	6300s
SOMb_8	200	80	15117	16225	12600s
SOMb_9	300	30	2069	2694	3000s
SOMb_{10}	300	60	8173	9689	9600s
SOMb_{11}	300	90	19125	20743	14400s
SOMb_{12}	300	120	34133	35881	21600s
SOMb_{13}	400	40	3760	4658	25200s

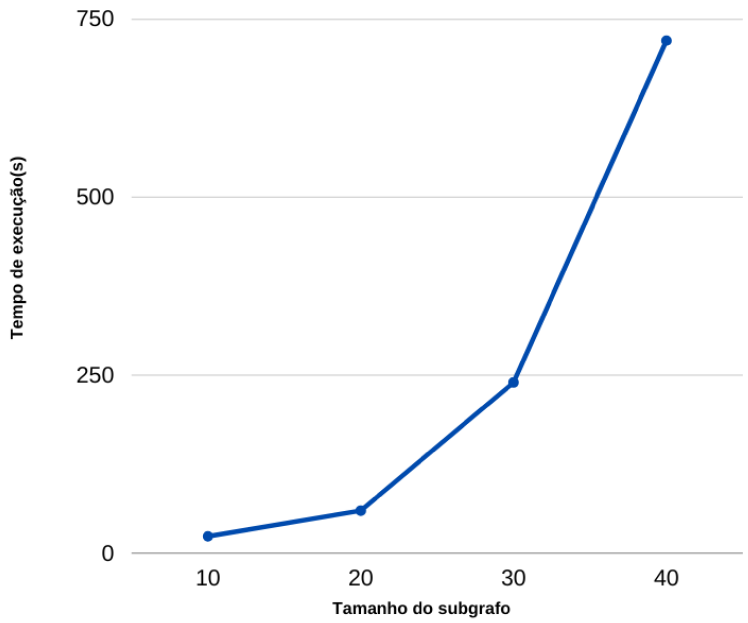
Fonte: Elaborada pelo autor.

5.3.2 Análise dos Resultados

Analisando todas as instâncias em conjunto, o algoritmo desenvolvido que aplica aprendizado de máquina foi sempre inferior quando se trata do valor da diversidade máxima e na maior parte das instâncias, também é menos efetivo. Então as soluções obtidas pelo método G_SS que utiliza meta heurísticas, se mostraram mais efetivas de um modo geral.

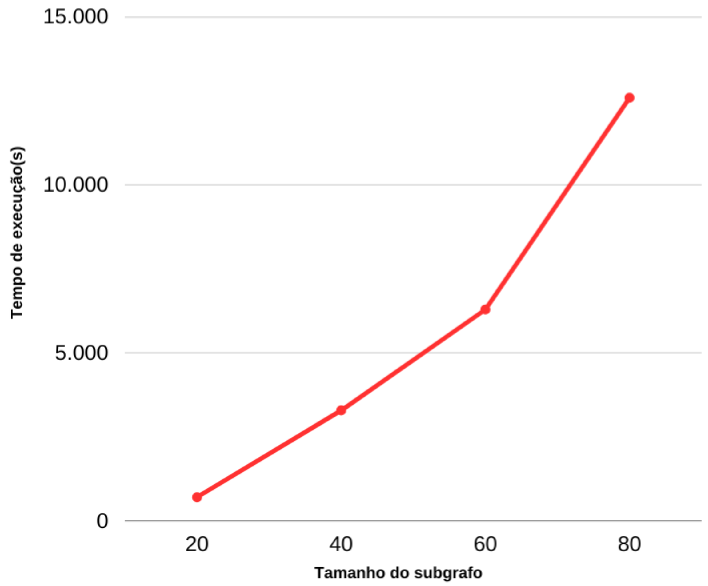
A seguir será visto algumas figuras onde é feito um comparativo de tempo em segundos, eixo y, e o tamanho do subgrafo (q), eixo x, para os diferentes tamanhos nas instâncias de SOMb. É notório o aumento do tempo de acordo com o tamanho da instância e o tamanho do subgrafo. A figura 13 é um comparativo entre os três tamanhos de instância e o seus respectivos tamanhos de subgrafo. Na Figura 10 com a linha azul vemos a como se comportou a instância SOM$_b$ para n = 100, já na Figura 11 com a linha vermelha temos os dados da instância para n = 200 e por fim, a linha verde, que está na Figura 12 e ilustra as relação para n = 300.

Figura 10 – Relação tempo e tamanho do subgrafo para as instâncias SOM*b* de tamanho 100



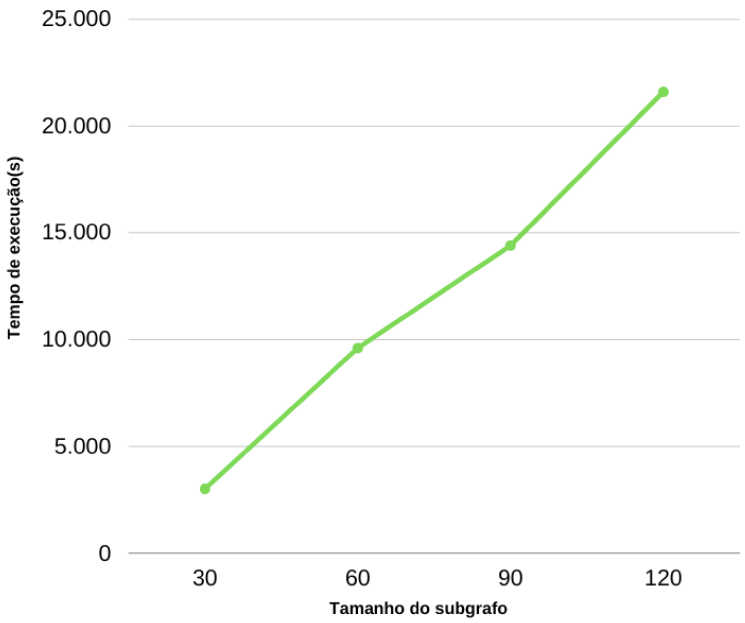
Fonte: Figura desenvolvida pelo autor

Figura 11 – Relação tempo e tamanho do subgrafo para as instâncias SOM*b* de tamanho 200



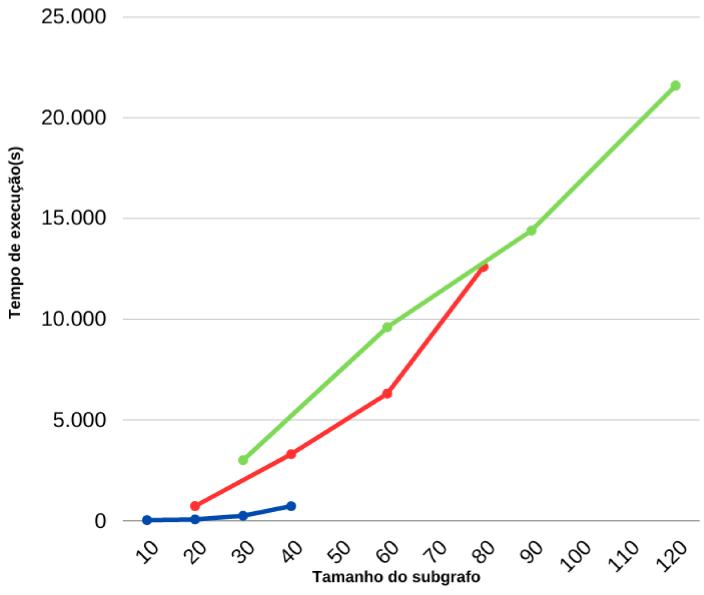
Fonte: Figura desenvolvida pelo autor

Figura 12 – Relação tempo e tamanho do subgrafo para as instâncias SOM*b* de tamanho 300



Fonte: Figura desenvolvida pelo autor

Figura 13 – Comparativo entre as instâncias SOM*b* de tamanho 100, 200 e 300



Fonte: Figura desenvolvida pelo autor

Na tabela 3 é mostrada uma coluna "Valor Obtido", onde tem os melhores valores encontrado pelo algoritmo desenvolvido neste trabalho para determinada instância, outra "Melhor Valor", que é o valor encontrado no método que utiliza meta heurísticas e por final, a "GAP", onde o valor mostrado é o calculo da porcentagem de diferença ou desvio entre os valores obtidos e os melhores valores. Vemos que há uma média de diferença entre estes de 16,6%. Além disso, vemos na Tabela 2 que as instâncias de até no máximo duzentos vértices, aproximadamente, possuem um tempo de execução inferior, assumindo que todas as instâncias selecionadas tem um tempo de execução de duas horas no método G_SS, e que essa média de diferença do GAP vai para cerca de 18,3%. Isso mostra que em instâncias menores, menos complexas e o tamanho do subgrafo seja menor, o algoritmo desenvolvido tem uma discrepância maior dos valores das somas das arestas e o tempo de execução bem inferior.

Tabela 3 – Comparação dos melhores valores

Nome da Instância	n	q	Valor Obtido	Melhor Valor Método G_SS	GAP
SOMb_1	100	10	242	333	27,3%
SOMb_2	100	20	875	1195	26,7%
SOMb_3	100	30	2016	2457	17,9%
SOMb_4	100	40	3661	4142	11,6%
SOMb_5	200	20	924	1247	25,9%
SOMb_6	200	40	3614	4450	18,7%
SOMb_7	200	60	8349	9437	11,5%
SOMb_8	200	80	15117	16225	6,8%
SOMb_9	300	30	2069	2694	23,1%
SOMb_{10}	300	60	8173	9689	15,6%
SOMb_{11}	300	90	19125	20743	7,8%
SOMb_{12}	300	120	34133	35881	4,8%
SOMb_{13}	400	40	3760	4658	19,2%
Média					16,6%

Fonte: Elaborada pelo autor.

6 CONCLUSÃO

Este capítulo visa expor considerações finais acerca deste trabalho e melhorias que podem ser abordadas em trabalhos futuros. Na Subseção 6.1 estão as considerações e na Subseção 6.2 mostra alguns pontos para futuros trabalhos.

6.1 Considerações gerais

Neste trabalho, buscou-se aplicar aprendizado de máquina não supervisionado, por meio do algoritmo de clusterização *k-means*, no Problema da Diversidade Máxima(PDM) e comparar o resultado obtido com resultados já existentes que foram solucionados por métodos que utilizam meta heurísticas. Acerca deste objetivo, foi possível extrair algumas informações.

- A complexidade do algoritmo que foi aplicado a aprendizagem de máquina é maior do que o do método G_SS, sendo aproximadamente O(n^3).
- O tempo de execução do algoritmo desenvolvido nesta monografia, em pequenas instâncias, é bem mais rápido quando comparado com o método G_SS.
- Em todos os casos o valor da diversidade dos métodos que utilizaram meta heurísticas foram melhores, ou seja, mais diverso.
- Nas instâncias maiores e mais complexas, apesar do algoritmo desenvolvido neste estudo obter um resultado inferior de diversidade, o GAP nestas instâncias foram menores.

6.2 Trabalhos futuros

Considerando os resultados encontrados neste trabalho, em trabalhos futuros é esperado que seja desenvolvido um algoritmo de aprendizagem não supervisionada, que tenha uma menor complexidade, assim sendo capaz de reduzir o tempo de execução. Podendo também mudar a forma no qual é escolhido os vértices para fazer os tratamentos dos subgrafos quando tem mais ou menos vértices do que o pedido na instância, isso pode melhorar o valor da soma da diversidade.

Por fim, deve ser aplicado no PDM algum algoritmo de aprendizagem supervisionada para fazer a comparação dos resultados entre os métodos que solucionaram utilizando meta heurísticas e aprendizagem de máquina não supervisionada.

REFERÊNCIAS

ALIDAEE, B.; WANG, H. A note on heuristic approach based on ubqp formulation of the maximum diversity problem. **Journal of the Operational Research Society**, IBM, 2016.

BRAGA, A. de P.; LUDERMIR, T. B.; CARVALHO, A. C. P. de L. F. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: LTC Editora, 2007. v. 2.

DUARTE, A.; MARTI, R. Tabu search and grasp for the maximum diversity problem. **European Journal of Operational Research**, v. 178, p. 71–84, 02 2007.

ESCOVEDO, T. **Machine Learning: Conceitos e Modelos — Parte I: Aprendizado Supervisionado**. 2020. Disponível em: <https://tatianaesc.medium.com/machine-learning-conceitos-e-modelos-f0373bf4f445>.

GALLEGO, M.; DUARTE, A.; LAGUNA, M.; MARTÍ, R. Hybrid heuristics for the maximum diversity problem. **Computational Optimization and Applications**, Springer, v. 44, p. 411–426, 2009.

GLOVER, F.; KUO, C.; DHIR, K. S. Heuristic algorithms for the maximum diversity problem. **Decision Sciences**, Journal of Information and Optimization Sciences, v. 19, n. 1, p. 109—-132, 1998.

HAYKIN, S. **Redes Neurais: Princípios e Prática**. [S.l.]: Bookman Editora, 2001. ISBN 9788577800865.

KARP, R. M. On the computational complexity of combinatorial problems. **Networks**, Wiley Online Library, v. 5, n. 1, p. 45–68, 1975.

KUO; GLOVER, F. C.-C.; DHIR, K. S. Analyzing and modeling the maximum diversity problem by zero-one programming. **Decision Sciences**, Decision Sciences Institute, v. 24, n. 6, p. 1171—-1185, 1993.

MITCHELL, T. M. **Meachine Learning.** [S.l.]: McGraw-Hill Science/Engineering/Math, 1997.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of research and development**, IBM, v. 3, n. 3, p. 535–554, 1959.

SINAGA, K. P.; YANG, M.-S. Unsupervised k-means clustering algorithm. **IEEE access**, IEEE, v. 8, p. 80716–80727, 2020.

SOARES, P. L. B.; NETO, M. B. C.; REBOUÇAS, D. N. **Exact method for maximum diversity problem**. [S.l.], 2018.

SOUSA, G. H. A.; ESMIN, A. A. A. Algoritmo de enxame de partículas híbrido aplicado a clusterização de dados. Universidade Federal de Lavras (UFLA), 2011.

TAKANE, B. **Um algoritmo exato para o problema da diversidade máxima**. 58 p. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, Belo Horizonte, 2011.

VICERI. **As classificações dos algoritmos de Machine Learning**. 2020. Disponível em: <https://viceri.com.br/insights/as-classificacoes-dos-algoritmos-de-machine-learning/>.

YUAN, S.; LI, X.; ERNST, A. Using statistical measures and machine learning for graph reduction to solve maximum weight clique problems. **IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE**, v. 43, 2021.