UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Tális Breda

# ESTUDO DA FAMÍLIA DE EMPARELHAMENTO EM GRAFOS

Florianópolis
2024

Tális Breda

# ESTUDO DA FAMÍLIA DE EMPARELHAMENTO EM GRAFOS

Trabalho de Conclusão do Curso de Graduação em Ciências da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciências da Computação.
Orientador: Prof. Dr. Rafael de Santiago

Florianópolis
2024

# RESUMO

Problemas de emparelhamento em grafos, definidos como uma seleção de conjuntos de arestas sem vértices em comum, possuem ampla relevância em áreas da computação, como visão computacional, e também fora dela, como na biologia. São bastante importantes em tarefas que envolvem correspondência de elementos, como a comparação e correspondência de elementos em conjuntos distintos, por exemplo em análise de dados estruturados. Devido à complexidade computacional frequentemente NP-completa desses problemas, são necessárias abordagens que conciliem precisão e eficiência. Considerando a quantidade de algoritmos e métodos presentes na literatura, torna-se difícil encontrar a maneira mais eficiente de resolver um problema específico. Este trabalho busca levantar, comparar e classificar métodos computacionais, destacando suas aplicações e limitações, com o objetivo de fornecer insights úteis para pesquisadores e profissionais na escolha de soluções eficazes para cenários práticos.

**Palavras-chave:** Emparelhamento em grafos, grafos bipartidos, visão computacional

# ABSTRACT

Graph matching problems, defined as the selection of edge sets with no common vertices, are highly relevant in fields of computer science such as computer vision, as well as in areas outside of it, like biology. They are particularly important for tasks involving element matching, such as comparing and matching elements in distinct sets, for example, in structured data analysis. Due to the often NP-complete computational complexity of these problems, approaches that balance precision and efficiency are necessary. Given the large number of algorithms and methods in the literature, it is challenging to determine the most efficient way to solve a specific problem. This work aims to survey, compare, and classify computational methods, highlighting their applications and limitations, with the goal of providing useful insights for researchers and professionals in selecting effective solutions for practical scenarios.


**Keywords:** Graph matching, bipartite graphs, computer vision

# Conteúdo

# 1 Introdução

Um emparelhamento em um grafo não-dirigido é definido como um conjunto de arestas sem pontas em comum. Em outras palavras, um emparelhamento é um conjunto $M$ de arestas que satisfaz a seguinte propriedade: o grau de cada vértice no subconjunto $M$ é no máximo 1 (FEOFILOFF, 2019). Os principais problemas relacionados a emparelhamentos geralmente envolvem grafos bipartidos, aqueles que podem ser particionados em dois conjuntos independentes $U$ e $V$, onde todas as arestas conectam vértices de $U$ com vértices de $V$.

Os problemas de emparelhamento têm aplicações significativas em várias áreas, como biologia computacional, onde podem ser empregados na diferenciação e classificação de estruturas proteicas (TAYLOR, 2002); redes sociais, para identificação de comunidades ou grupos (FAN, 2012); e visão computacional, para correspondências entre elementos de conjuntos distintos, como pontos em imagens ou vértices em malhas tridimensionais (HALLER et al., 2022).

Resolver problemas de emparelhamento em grafos é um desafio devido à sua complexidade computacional, frequentemente NP-completa, exigindo métodos inovadores para equilibrar precisão e eficiência. Estratégias clássicas, como o Problema de Atribuição Quadrática (QAP), modelam o emparelhamento como uma questão de otimização combinatória, empregando relaxações espectrais e heurísticas para encontrar soluções aproximadas de forma eficiente (YAN; YANG; HANCOCK, 2020). Por outro lado, avanços em transporte ótimo duplamente estocástico, como o algoritmo GOAT, têm demonstrado melhorias em robustez e velocidade, especialmente em grafos maiores (SAAD-ELDIN et al., 2021).

Estudos como esses ilustram a importância de investigar e comparar métodos computacionais que abordem a complexidade inerente ao problema de emparelhamento em grafos, contribuindo para aplicações como aprendizado de máquina e visão computacional. Essa abordagem torna-se essencial para desenvolver soluções adaptadas a diferentes cenários e com impacto direto na melhoria de algoritmos existentes.

Diante desse cenário, este trabalho visa sistematizar os principais problemas e métodos computacionais associados ao emparelhamento em grafos. Ele busca oferecer uma comparação detalhada entre suas aplicações e limitações, contribuindo para o entendimento aprofundado das ferramentas disponíveis e identificando oportunidades para o desenvolvimento de novas soluções. Essa abordagem torna-se ainda mais relevante à luz da importância prática desses algoritmos para resolver problemas reais em grande escala, conforme evidenciado por estudos recentes (SUSSMAN et al., 2021).

## 1.1 Objetivos

### 1.1.1 Objetivo geral

Este trabalho tem como objetivo principal levantar, organizar e comparar problemas de emparelhamento em grafos para diversos casos de uso, além de comparar diferentes métodos computacionais presentes na literatura para a solução desses problemas.

### 1.1.2 Objetivos específicos

- Mapear as variações e características de diferentes problemas de emparelhamento presentes na literatura.
- Investigar e classificar métodos computacionais presentes na literatura de acordo com suas vantagens, limitações e casos de uso.
- Criar um ambiente de testes unificado a fim de reduzir variabilidade nos resultados e proporcionar comparações consistentes
- Implementar e testar os algoritmos descritos, a fim de comparar desempenho e eficiência em diferentes situações

- Organizar os resultados em um formato que facilite o acesso e a compreensão dos tópicos abordados por pesquisadores e profissionais.

### 1.2 Delimitação do estudo

Os problemas de emparelhamento em grafos podem ser classificados de diversas maneiras, como sendo em grafos bipartidos ou não-bipartidos, ponderados ou não-ponderados, direcionados ou não-direcionados, problemas em multigrafos, problemas em hipergrafos, entre outros. Este trabalho focará em problemas que envolvem grafos simples, isto é, excluindo multigrafos e hipergrafos.

## 2 Fundamentação teórica

### 2.1 Definições sobre grafos

Um **grafo** $G$ é uma uma estrutura definida por um par $G = (V, E)$, consistindo de um conjunto finito $V$ de elementos chamados **vértices** (ou **nós** ou **pontos**) e um conjunto (ou família) $E$ de pares não ordenados de vértices, chamados de **arestas** (JUNGNICKEL, 2008). O conjunto V é o conjunto de vértices de $G(VG)$ e $E$ é a família de arestas de $G(EG)$.

No contexto deste projeto, assumimos que as arestas $e = u, v$ são pares não ordenados de vértices distintos. Se uma aresta $e$ conecta os vértices $a$ e $b$, diz-se que $a$ e $b$ são **adjacentes** ou **vizinhos**, e que a aresta $e$ é **incidente** a $a$ e $b$ (SCHRIJVER, 2004).

O conceito de grafo é amplamente utilizado como uma representação abstrata concisa para estruturas complexas e para codificar relações binárias entre um conjunto de objetos.

#### 2.1.1 Grafo direcionado

Se as relações entre os vértices forem assimétricas, utilizamos um **grafo direcionado** ou **dígrafo**, $D = (V, A)$. Neste caso, $A$ é uma família de pares ordenados de vértices, chamados **arcos** (ou arestas direcionadas). Para um arco $e = (u, v)$, $u$ é chamado o vértice inicial ou **cauda** (tail), e $v$ é o vértice final ou **cabeça** (head) (MANBER, 1989)

#### 2.1.2 Grafo ponderado

Um grafo $G = (V, E)$ é chamado **grafo ponderado** (ou com pesos) se uma **função de peso** (ou função de custo, ou função de comprimento) $w : E \to R$ está associada às arestas. Formalmente: $G = (V, E, w)$, onde $w : E \to R$. Geralmente, o peso $w(e)$ de uma aresta $e$ é um valor real não negativo que representa o custo, comprimento ou capacidade associada àquela aresta (ROSEN; KREHER; STINSON, 1998).

Grafos ponderados podem ser representados por matrizes de adjacência de valor real $A$, onde a entrada $A_{ij}$ é o peso $w_{ij}$ da aresta (ZASLAVSKIY; VERT, 2009)

#### 2.1.3 Grafo bipartido

Um grafo bipartido é um grafo não direcionado que pode ser facilmente colorido com apenas duas cores (MANBER, 1989)

Formalmente, um grafo $G = (V, E)$ é chamado **grafo bipartido** se o conjunto de vértices $V$ puder ser particionado em dois subconjuntos disjuntos $V_1$ e $V_2$, chamados calsses de cor, tais que:

$$V = V_1 \cup V_2 \quad e \quad V_1 \cap V_2 = \emptyset \tag{1}$$

e todas as arestas $e \in E$ conectam um vértice em $V_1$ a um vértice em $V_2$. Ou seja, não existem arestas que conectem vértices dentro do mesmo subconjunto (DASGUPTA CHRISTOS H. PAPADIMITRIOU, 2006).

Observações:

- Um grafo é bipartido se e somente se não contém ciclos de comprimento ímpar (SCHRIJVER, 2004)
- O grafo bipartido completo $K_{m,n}$ possui $m$ vértices em $V_1$ e $n$ vértices em $V_2$, e contém todas as arestas possíveis entre $V_1$ e $V_2$ (JUNGNICKEL, 2008)

## 2.2 Problema Geral do Emparelhamento

O problema de **matching** (emparelhamento) em grafos é um problema fundamental na otimização combinatória

Um **matching** M em um grafo não-direcionado $G = (V, E)$ é um subconjunto de arestas $M \subseteq E$ tal que nenhum par de arestas em $M$ compartilha um vértice comum (KLEINBERG; TARDOS, 2005). Em outras palavras, cada nó aparece em no máximo uma aresta de $M$.

- Um vértice é chamado **coberto** (*matched*) se for incidente a uma aresta em $M$; caso contrário, é **descoberto** (*unmatched* ou *exposed*) (CORMEN CHARLES E. LEISERSON, 2009)
- Um **matching de cardinalidade máxima** (Maximum Matching) é um matching com o maior número possível de arestas (CORMEN CHARLES E. LEISERSON, 2009). A cardinalidade máxima de um matching é denotada por $v(G)$ (SCHRIJVER, 2004)
- Um **matching perfeito** é um matching que cobre todos os vértices do grafo (KLEINBERG; TARDOS, 2005)
- O **problema de matching ponderado** (Weighted Matching Problem) envolve encontrar um matching para o qual a soma dos pesos das arestas é máxima. Em um grafo ponderado $G = (V, E, w)$, busca-se um $M \subseteq E$ que maximize $w(M)$ (MANDULAK SAYAN GHOSH; SLOTA, 2024)

### 2.2.1 Matching em grafos bipartidos

O problema de **Matching Bipartido** é o caso clássico de encontrar um matching de cardinalidade máxima em um grafo bipartido $G = (V, E)$, onde $V = X \cup Y$ (KLEINBERG; TARDOS, 2005)

- O matching em grafos bipartidos pode modelar situações de atribuição, como associar empregos ($X$) a máquinas ($Y$), ou professores ($X$) a cursos ($Y$), onde uma aresta indica uma capacidade de atribuição (KLEINBERG; TARDOS, 2005)
- O problema de matching ponderado em grafos bipartidos é equivalente ao **problema de atribuição (LAWLER, 2011)**, que historicamente motivou o desenvolvimento do **método Húngaro** (SCHRIJVER, 2004)

### 2.2.2 Formulação geral (emparelhamento e otimização)

Em um contexto mais amplo, o emparelhamento de grafos pode ser formalizado como um problema de otimização que busca maximizar a compatibilidade entre dois grafos $G$ e $G'$ (CAETANO JULIAN J. MCAULEY; SMOLA, 2009).

- O problema de graph matching é frequentemente abordado como um **problema de atribuição quadrática (QAP)**. Essa formulação busca maximizar uma função objetivo que combina termos de compatibilidade unária (nó-nó, $c_{ii'}$) e compatibilidade par a par (aresta-aresta, $d_{ii'jj'}$), sujeito a restrições de atribuição binária ($y_{ii'} \in 0, 1$). O termo quadrático codifica a preservação das relações (arestas) entre os nós (CAETANO JULIAN J. MCAULEY; SMOLA, 2009)

- Para grafos bipartidos, a determinação de um matching máximo pode ser resolvida de maneira eficiente e está intimamente ligada a problemas de *network flow* (fluxo em redes) (KLEINBERG; TARDOS, 2005)

# 3 Problemas de Emparelhamento

## 3.1 Introdução sobre taxonomia dos problemas

## 3.2 Emparelhamento em grafos bipartidos

### 3.2.1 Emparelhamento bipartido

### 3.2.2 Emparelhamento de cardinalidade máxima

Dado um grafo bipartido, o problema da cardinalidade máxima busca encontrar o emparelhamento que contém o maior número possível de arestas. Ele pode ser reduzido ao problema de fluxo máximo em uma rede da seguinte maneira:

Construimos um grafo direcionado a partir do grafo bipartido original, adicionando um vértice fonte $s$ e um vértice sumidouro $t$. Conectamos o vértice fonte $s$ a todos os vértices do primeiro conjunto de partição com arestas de capacidade 1, e conectamos todos os vértices do segundo conjunto de partição ao vértice sumidouro $t$ com arestas de capacidade 1. As arestas entre os dois conjuntos de partição recebem capacidade infinita. Seguindo o teorema da integralidade **CITAR FONTES**, qualquer fluxo máximo encontrado nesta rede será integral, correspondendo a um emparelhamento no grafo bipartido original.

Aplicando um algoritmo de fluxo máximo, como o algoritmo de Edmonds-Karp ou o algoritmo de Dinic, podemos determinar o fluxo máximo da fonte ao sumidouro. O conjunto de arestas que transportam fluxo na solução do fluxo máximo forma o emparelhamento de cardinalidade máxima no grafo bipartido original. **CITAR FONTES**

### 3.2.3 Problema de atribuição (Assignment Problem)

**Talvez chamar de Problema de emparelhamento ponderado?**

O problema de atribuição consiste em encontrar uma combinação ótima de atribuições entre dois conjuntos disjuntos, minimizando o custo total associado a essas atribuições. Exemplo: Considere $N$ trabalhadores e $N$ tarefas, onde cada trabalhador pode ser designado a exatamente uma tarefa, e cada tarefa deve ser atribuída a exatamente um trabalhador. O custo de atribuir o trabalhador i à tarefa j é representado por uma matriz de custos $C = [c_{ij}]$ (LAWLER, 2011). O objetivo é encontrar um conjunto de atribuições que minimize o custo total.

É possível modelar esse problema como um problema de emparelhamento de custo mínimo em um grafo bipartido ponderado, da seguinte forma:

Construimos um grafo bipartido com $N$ vértices em cada conjunto de partição, onde cada vértice do primeiro conjunto representa um homem e cada vértice do segundo conjunto representa um trabalho. Cada aresta entre um vértice do primeiro conjunto e um vértice do segundo conjunto é ponderada com o custo $c_{ij}$ de atribuir o homem i ao trabalho j. Cria-se um novo vértice fonte conectado a todos os vértices do primeiro conjunto com arestas de peso zero, e um novo vértice sumidouro conectado a todos os vértices do segundo conjunto com arestas de peso zero. O objetivo é encontrar um fluxo máximo de custo mínimo neste grafo modificado, que corresponde a um emparelhamento de custo mínimo no grafo bipartido original (LAWLER, 2011). Assim, reduzimos o problema de emparelhamento em um grafo bipartido ponderado ao problema de fluxo de custo mínimo, que pode ser resolvido eficientemente usando algoritmos de fluxo de custo mínimo, como o algoritmo de ciclo negativo **CITAR FONTES**

### 3.2.4 Problema de atribuição gargalo

O problema de atribuição gargalo, também conhecido como **Problema de Emparelhamento Min-Max** ou **Bottleneck Assignment problem**, é uma variação do problema de atribuição tradicional. O objetivo deste problema é encontrar, em um grafo bipartido ponderado, um emparelhamento de cardinalidade máxima (maior número possível de arestas) no qual o mínimo dos pesos das arestas do emparelhamento seja maximizado (LAWLER, 2011).

Por exemplo. Considere $N$ trabalhadores e $N$ estações de trabalho. $w_{ij}$ representa a eficiência do trabalhador i na estação de trabalho j. A eficiência total da produção é limitada pela eficiência do trabalhador menos eficiente. O objetivo é atribuir trabalhadores às estações de trabalho de forma a maximizar a eficiência mínima entre todas as atribuições. Para resolver este problema, podemos utilizar o método de threshold (LAWLER, 2011).

### 3.2.5 Problema do emparelhamento estável (Stable Marriage Problem)

### 3.2.6 Problema de atribuição quadrática (QAP)

### 3.2.7 Problema de atribuição linear (LAP)

# Referências


CAETANO JULIAN J. MCAULEY, L. C. Q. V. L. T. S.; SMOLA, A. J. Learning graph matching. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, IEEE Computer Society, United States, v. 31, n. X, December 2009. ISSN 0162-8828.

CORMEN CHARLES E. LEISERSON, R. L. R. C. S. T. H. *Introduction to Algorithms*. 3rd. ed. Cambdrige, MA: The MIT Press, 2009. ISBN 978-0-262-53305-8.

DASGUPTA CHRISTOS H. PAPADIMITRIOU, U. V. V. S. *Algorithms*. 1st. ed. New York, NY: McGraw-Hill Education, 2006. ISBN 978-0073523408.

FAN, W. Graph pattern matching revised for social network analysis. In: *Proceedings of the 15th International Conference on Database Theory*. New York, NY, USA: Association for Computing Machinery, 2012. (ICDT ’12), p. 8–21. ISBN 9781450307918. Disponível em: ⟨https://doi.org/10.1145/2274576.2274578⟩.

FEOFILOFF, P. *Emparelhamento em Grafos Bipartidos*. 2019. IME-USP. Disponível em: ⟨https://www.ime.usp.br/~pf/algoritmos_para_grafos/aulas/matching-bipartite.html⟩.

HALLER, S. et al. *A Comparative Study of Graph Matching Algorithms in Computer Vision*. 2022. Disponível em: ⟨https://arxiv.org/abs/2207.00291⟩.

JUNGNICKEL, D. *Graphs, Networks and Algorithms*. 3rd. ed. Berlin, Germany: Springer, 2008. ISBN 978-354727798.

KLEINBERG, J.; TARDOS Éva. *Algorithm Design*. 1st. ed. Boston, MA: Pearson, 2005. ISBN 978-0321295354.

LAWLER, E. L. *Combinatorial Optimization: Networks and Matroids*. New York, NY: Dover Publications, 2011. Reprint edition. ISBN 978-0486414539.

MANBER, U. *Introduction to Algorithms: A Creative Approach*. 1st. ed. Reading, MA: Addison-Wesley, 1989. ISBN 978-0201120370.

MANDULAK SAYAN GHOSH, S. M. F. M. H. M.; SLOTA, G. Efficient weighted graph matching on gpus. *SC24*, IEEE Computer Society, United States, November 2024. ISSN 0162-8828.

ROSEN, K. H.; KREHER, D. L.; STINSON, D. R. *Discrete Mathematics and Its Applications*. 1st. ed. Boca Raton, FL: CRC Press, 1998. ISBN 978-0849339882.

SAAD-ELDIN, A. et al. *Graph Matching via Optimal Transport*. 2021. Disponível em: ⟨https://arxiv.org/abs/2111.05366⟩.

SCHRIJVER, A. *Combinatorial Optimization: Polyhedra and Efficiency*. 1st. ed. Berlin, Germany: Springer, 2004. ISBN 3540204563.

SUSSMAN, D. L. et al. *Overview of Graph Matching Challenges and Approaches*. 2021. Acessado em: 26 nov. 2024. Disponível em: ⟨https://www.ll.mit.edu/⟩.

TAYLOR, W. R. Protein structure comparison using bipartite graph matching and its application to protein structure classification. *Molecular and Cellular Proteomics*, American Society for Biochemistry and Molecular Biology, United States, v. 1, n. 4, April 2002. ISSN 1535-9476.

YAN, J.; YANG, S.; HANCOCK, E. Learning for graph matching and related combinatorial optimization problems. In: BESSIERE, C. (Ed.). *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. International Joint Conferences on Artificial Intelligence Organization, 2020. p. 4988–4996. Survey track. Disponível em: ⟨https://doi.org/10.24963/ijcai.2020/694⟩.

ZASLAVSKIY, F. B. M.; VERT, J.-P. A path following algorithm for the graph matching problem. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, IEEE Computer Society, United States, v. 31, n. 12, December 2009. ISSN 0162-8828.