



Trabalho de Conclusão de Curso

# **Algoritmo GRASP-VND para o problema da máxima biclique balanceada com peso no vértice**

Lucas Montenegro Andrade Assunção  
lmaa@ic.ufal.br

Orientador:  
Prof. Rian Gabriel Santos Pinheiro

Maceió, 24 de outubro de 2023

Lucas Montenegro Andrade Assunção

## **Algoritmo GRASP-VND para o problema da máxima biclique balanceada com peso no vértice**

Monografia apresentada como requisito parcial para  
obtenção do grau de Bacharel em Ciência da Com-  
putação do Instituto de Computação da Universidade  
Federal de Alagoas.

Orientador:

Prof. Rian Gabriel Santos Pinheiro

Maceió, 24 de outubro de 2023

**Catálogo na Fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 - 1767

A851a Assunção, Lucas Montenegro Andrade.

Algoritmo GRASP-VND para o problema da máxima biclique  
balanceada com peso no vértice / Lucas Montenegro Andrade  
Assunção. – 2023.  
40 f. : il.

Orientador: Rian Gabriel Santos Pinheiro.  
Monografia (Trabalho de conclusão de curso em Ciência da  
Computação) – Universidade Federal de Alagoas, Instituto de Computação.  
Maceió, 2023.

Bibliografia: f. 39-40.

1. Otimização em grafo. 2. Biclíques (Grafos). 3. Meta-heurística. I.  
Título.

CDU: 004.023

# Agradecimentos

Primeiramente agradeço aos meus pais por todo o suporte, educação e que me deram todo o apoio para enfrentar as dificuldades da vida.

Agradeço à minha família por ter sido o meu alicerce em várias situações que já passei e enfrentei.

Agradeço à Ana Beatriz, minha namorada, por sempre me tranquilizar e me apoiar em decisões difíceis, me fazendo uma pessoa mais madura e mais decidida.

Agradeço ao Prof. Rian, meu orientador, pelo suporte na construção desse trabalho, por todos os ensinamentos e conselhos, onde me fez crescer tanto na área científica quanto como pessoa.

À todos os professores do curso que compartilharam os seus conhecimentos, me fazendo um indivíduo mais preparado profissionalmente.

# Resumo

Este trabalho aborda o problema da "Máxima Biclique Balanceada com Peso nos Vértices". Nesse contexto, o objetivo é identificar, em um grafo não direcionado com atributos de peso associados a seus vértices, uma biclique (subgrafo bipartite completo) balanceada que maximize a soma dos pesos. Este problema é notoriamente complexo, caindo na classe  $\mathcal{NP}$ -difícil, e possui aplicações de grande relevância em diversas áreas.

Apresentamos uma abordagem baseada na meta-heurística GRASP, complementada por uma busca local VND com estruturas de vizinhança e uma técnica de redução adaptada ao problema em questão. Avaliamos a eficácia da nossa proposta usando instâncias amplamente reconhecidas na literatura, como as DIMACS e BHOSLIB. Os resultados obtidos indicam que nosso algoritmo supera o algoritmo exato (CPLEX) em termos de eficiência, sendo capaz de encontrar todas as 80 soluções ótimas conhecidas em um tempo computacional reduzido.

Por fim, nossa abordagem demonstrou excelentes resultados em instâncias mais desafiadoras, como as do conjunto KONECT, grafos de grande porte e esparsos. Nessas situações, a técnica de redução se mostrou altamente eficaz, eliminando até 99,78% dos vértices. Isso destaca a robustez e versatilidade da nossa metodologia na resolução do problema em diversos cenários, com implicações significativas nas aplicações práticas relacionadas.

**Palavras-chave:** Otimização em grafo, bicliques, meta-heurísticas.

# Abstract

This work addresses the problem of "Maximum Balanced Weighted Vertex Biclique". In this context, the objective is to identify, in an undirected graph with vertex-weight attributes, a balanced biclique (complete bipartite subgraph) that maximizes the sum of the weights. This problem is notably complex, falling into the  $\mathcal{NP}$ -hard class, and has significant relevance in various domains.

We present an approach based on the GRASP meta-heuristic, complemented by a VND local search with neighborhood structures and a reduction technique tailored to the problem at hand. We evaluated the effectiveness of our proposal using widely recognized instances in the literature, such as DIMACS and BHOSLIB. The results obtained indicate that our algorithm outperforms the exact algorithm (CPLEX) in terms of efficiency, being capable of finding all 80 known optimal solutions in a reduced computational time.

Finally, our approach demonstrated excellent results in more challenging instances, such as those from the KONECT dataset, encompassing large and sparse graphs. In these situations, the reduction technique proved highly effective, eliminating up to 99.78% of the vertices. This highlights the robustness and versatility of our methodology in solving the problem in various scenarios, with significant implications in related practical applications.

**Keywords:** Graph Optimization, bicliques, metaheuristics.

# Conteúdo

|                                                |           |
|------------------------------------------------|-----------|
| Lista de Figuras . . . . .                     | vii       |
| Lista de Tabelas . . . . .                     | viii      |
| Lista de Algoritmos . . . . .                  | ix        |
| <b>1 Introdução</b>                            | <b>1</b>  |
| 1.1 Biclique . . . . .                         | 2         |
| 1.1.1 Biclique com peso nos vértices . . . . . | 3         |
| 1.1.2 Entradas bipartidas e gerais . . . . .   | 4         |
| 1.1.3 Aplicações . . . . .                     | 4         |
| 1.2 Motivação . . . . .                        | 8         |
| 1.3 Objetivos . . . . .                        | 8         |
| 1.4 Contribuições . . . . .                    | 8         |
| 1.5 Estrutura do texto . . . . .               | 9         |
| <b>2 Trabalhos relacionados</b>                | <b>10</b> |
| 2.1 Trabalhos Teóricos . . . . .               | 10        |
| 2.2 Trabalhos Práticos . . . . .               | 11        |
| <b>3 Fundamentação teórica</b>                 | <b>13</b> |
| 3.1 Complexidade de problemas . . . . .        | 13        |
| 3.2 Programação Linear e Inteira . . . . .     | 14        |
| 3.3 Meta-heurísticas . . . . .                 | 15        |
| <b>4 Nossa proposta</b>                        | <b>16</b> |
| 4.1 Proposta . . . . .                         | 16        |
| 4.2 Algoritmo GRASP-VND . . . . .              | 17        |
| 4.3 Busca Local . . . . .                      | 18        |
| 4.3.1 Entradas Gerais . . . . .                | 19        |
| 4.3.2 Entradas Bipartidas . . . . .            | 19        |
| 4.4 Estruturas auxiliares . . . . .            | 20        |
| 4.5 Redução . . . . .                          | 20        |
| 4.6 Algoritmo da Redução . . . . .             | 22        |
| 4.7 Formulação Matemática . . . . .            | 23        |
| <b>5 Resultados e discussões</b>               | <b>24</b> |
| 5.1 Instâncias . . . . .                       | 24        |
| 5.2 Parâmetros utilizados . . . . .            | 27        |
| 5.3 Impacto do Reactive GRASP . . . . .        | 27        |
| 5.4 Impacto das Reduções . . . . .             | 30        |
| 5.5 Resultados Finais . . . . .                | 32        |

---

|          |                                              |           |
|----------|----------------------------------------------|-----------|
| 5.5.1    | Resultados para entradas gerais . . . . .    | 33        |
| 5.5.2    | Resultado para entradas bipartidas . . . . . | 35        |
| <b>6</b> | <b>Considerações Finais</b>                  | <b>37</b> |
|          | <b>References</b>                            | <b>39</b> |

# Lista de Figuras

|     |                                                                                             |    |
|-----|---------------------------------------------------------------------------------------------|----|
| 1.1 | Primeiro exemplo de grafo que não é uma biclique. . . . .                                   | 2  |
| 1.2 | Segundo exemplo de grafo que não é uma biclique. . . . .                                    | 3  |
| 1.3 | Exemplo de grafo biclique. . . . .                                                          | 4  |
| 1.4 | Biclique máxima ponderada. . . . .                                                          | 5  |
| 1.5 | Exemplo de biclique em Sistemas de Recomendação. . . . .                                    | 6  |
| 1.6 | Exemplo de biclique em Biologia Computacional. . . . .                                      | 7  |
| 4.1 | Exemplo do cálculo de $HI(v) = h$ . . . . .                                                 | 21 |
| 4.2 | Exemplo da estimativa do $UB(v)$ . . . . .                                                  | 22 |
| 5.1 | Curva de redução de vértices nas instâncias $\mathbb{G}(n, p)$ (não-bipartidas). . . . .    | 31 |
| 5.2 | Curva de redução de vértices nas instâncias $\mathbb{G}(n_1, n_2, p)$ (bipartidas). . . . . | 32 |

# Lista de Tabelas

|     |                                                                                    |    |
|-----|------------------------------------------------------------------------------------|----|
| 5.1 | Instâncias Utilizadas . . . . .                                                    | 25 |
| 5.2 | Instâncias Konect . . . . .                                                        | 26 |
| 5.3 | Comparativo do Algoritmo com e sem o Reactive para instâncias DIMACS . . . .       | 27 |
| 5.4 | Comparativo do Algoritmo com e sem o Reactive para instâncias BHOSLIB . . .        | 29 |
| 5.5 | Comparativo do Algoritmo com e sem o Reactive para instâncias KONECT . . .         | 29 |
| 5.6 | Comparativo do Algoritmo com redução e sem redução . . . . .                       | 31 |
| 5.7 | Resultados do GRASP <sub>VND</sub> para as instâncias do DIMACS e BHOSLIB. . . . . | 33 |
| 5.8 | Comparativo do Algoritmo com o Reactive e com o CPLEX para instâncias KONECT       | 35 |

# Lista de Algoritmos

|     |                                                |    |
|-----|------------------------------------------------|----|
| 4.1 | Pseudocódigo do GRASP <sub>VND</sub> . . . . . | 18 |
| 4.2 | Pseudocódigo da redução do GRASP. . . . .      | 23 |

# 1

## Introdução

O trabalho proposto está fundamentado em três principais áreas de estudo: Teoria dos Grafos, Pesquisa Operacional e Meta-heurísticas. A primeira delas, Teoria dos Grafos, é uma disciplina relativamente recente que se dedica à análise de estruturas de grafo. Essa área teve seu início no século XVIII, quando Leonhard Euler resolveu o famoso "Problema das Sete Pontes de Königsberg", um marco que pode ser considerado o surgimento da Teoria dos Grafos. De maneira fundamental, um grafo é definido da seguinte forma:

**Definição 1.** Um grafo  $G$  é composto por dois conjuntos,  $V$  e  $E$ , onde:

$V$  é o conjunto de vértices, representados como  $v \in V$ , onde  $v$  é um número natural ( $v \in \mathbb{N}$ ).

$E$  é o conjunto de arestas, representadas como  $e \in E$ , onde  $e$  é um par ordenado  $(v_1, v_2)$ , com  $v_1, v_2 \in V$  e  $v_1 \neq v_2$ .

A Teoria dos Grafos tem ganhado destaque devido às suas aplicações práticas, como a otimização de rotas em aplicativos como Uber e Waze, que utilizam modelos baseados em grafos.

A Pesquisa Operacional, por sua vez, aborda o desenvolvimento e aplicação de métodos matemáticos para maximizar ou minimizar aspectos em sistemas complexos. Por exemplo, uma empresa pode utilizar a Pesquisa Operacional para otimizar a produção, maximizando a eficiência e minimizando os custos. Embora essa área tenha surgido durante a primeira revolução industrial, ela continua a ser fundamental na gestão moderna, à medida que empresas operam em escala global, aproveitando avanços tecnológicos e conhecimento especializado.

A terceira área de destaque é a das Meta-Heurísticas, que consistem em algoritmos genéricos aplicáveis a diversos problemas, com o objetivo de otimizar a busca pela solução ideal em tempo reduzido em comparação a algoritmos exatos. Além disso, as Meta-Heurísticas visam manter o "GAP" (a diferença entre a solução de uma Meta-Heurística e a solução ótima) o menor possível. Essa área mantém uma forte relação com a Pesquisa Operacional, uma vez que busca otimizar processos de maneira eficaz, com a vantagem de ser computacionalmente mais eficiente, oferecendo soluções aproximadas ou iguais às soluções exatas.

## 1.1 Biclique

A Teoria dos Grafos desempenha um papel fundamental ao longo deste trabalho, e um tipo específico de grafo é central para o projeto em desenvolvimento: o grafo biclique.

Uma biclique, ou grafo bipartido completo, é uma categoria especial de grafo bipartido que apresenta características específicas para ser classificado como tal. Essas características incluem:

- I. Ser um grafo bipartido, o que significa que o grafo é dividido em duas partes, sendo cada parte um grafo independente.
- II. Ser um grafo completo, onde cada vértice de uma parte está conectado a todos os vértices da outra parte.

As Figuras 1.1, 1.2 e 1.3 exemplificam grafos com vértices em vermelho e azul, permitindo a determinação de se um grafo é uma biclique com base em suas propriedades. O primeiro exemplo é a Figura 1.1, que, embora seja um grafo bipartido, quebra a Propriedade II, pois um dos vértices azuis não está conectado a todos os vértices da parte vermelha.

Por outro lado, a Figura 1.2 possui arestas entre todos os vértices das duas partes (vermelhos e azuis), mas não é um grafo bipartido, pois inclui uma aresta entre dois vértices azuis, violando a Propriedade I. Finalmente, a Figura 1.3 representa uma biclique, pois atende a ambas as Propriedades I e II.

Os grafos bicliques têm diversas aplicações e são objeto de inúmeros estudos na literatura, como discutiremos posteriormente.

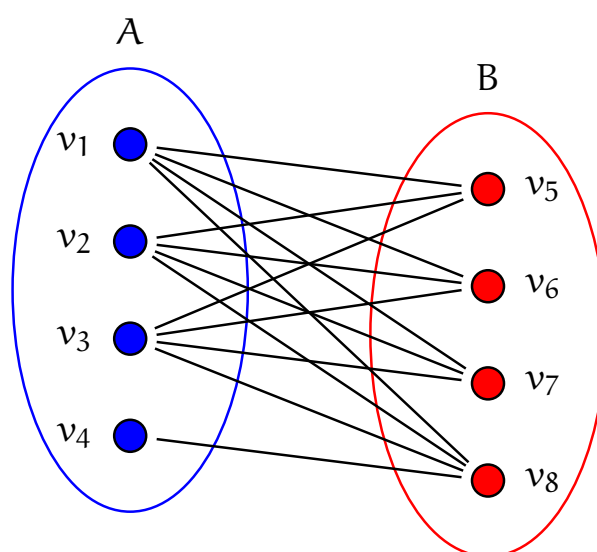


Figura 1.1: Primeiro exemplo de grafo que não é uma biclique.

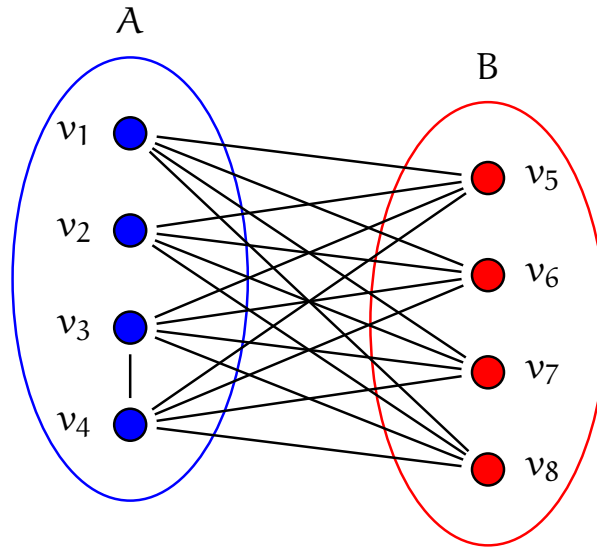


Figura 1.2: Segundo exemplo de grafo que não é uma biclique.

### 1.1.1 Biclique com peso nos vértices

Dado um grafo  $G = (V, E)$  de entrada, o problema da biclique (subgrafo bipartite completo) máxima balanceada tem como objetivo encontrar uma biclique balanceada (partes com o mesmo número de vértice) que tenha a cardinalidade maximizada. Uma generalização deste problema é a MÁXIMA BICLIQUE BALANCEADA COM PESO NOS VÉRTICES (PMBBPV), nesta variante, além do grafo de entrada, existe uma função de peso  $w : V \rightarrow \mathbb{R}$  que atribui a cada vértice  $v \in V$  um peso  $w(v)$ . Dessa forma, busca-se encontrar uma biclique balanceada em que a soma de todos os pesos dos vértices pertencentes a solução seja a maximizada. A Figura ?? mostra um grafo  $G$  de entrada para o PMBBPV com 8 vértices. Uma solução viável  $G' = (V', E')$  com peso 12 é apresentada na Figura 1.4a, note que  $G'$  possui 6 vértices os quais os seus vértices  $V'$  são particionados em  $V'_1$  e  $V'_2$  formados pelos vértices em azul tracejado e vermelho pontilhado, respectivamente.  $G'$  é uma biclique balanceada uma vez que: (i) para todo par de vértice  $v \in V'_1$  e  $u \in V'_2$  tem-se que  $vu \in E$ , (ii)  $V'_1$  e  $V'_2$  são conjuntos independentes e (iii)  $|V'_1| = |V'_2|$ . Finalmente, a Figura 1.4b apresenta a biclique  $G''$  com peso 20 que é a solução ótima do problema, mesmo tendo uma cardinalidade menor que  $G'$ .

Existe uma discussão sobre a complexidade de encontrar uma biclique em um grafo bipartido, já que existem diversas variantes de problemas relacionados.

Quando se trata de um problema balanceado, ou seja  $|V_1| = |V_2| = K$ , onde  $K$  é um valor inteiro, o problema se torna  $\mathcal{NP}$ -completo, como demonstrado em estudos anteriores (Garey & Johnson 1979, Feige & Kogan 2004, Peeters 2003). No entanto, quando a instância é bipartida, sem a necessidade de balanceamento, a solução ótima pode ser encontrada em tempo polinomial (Garey & Johnson 1979).

Dado que o problema em discussão é uma generalização da versão balanceada sem peso, ele também se enquadra na classe  $\mathcal{NP}$ -completo. Adiante, exploraremos várias aplicações

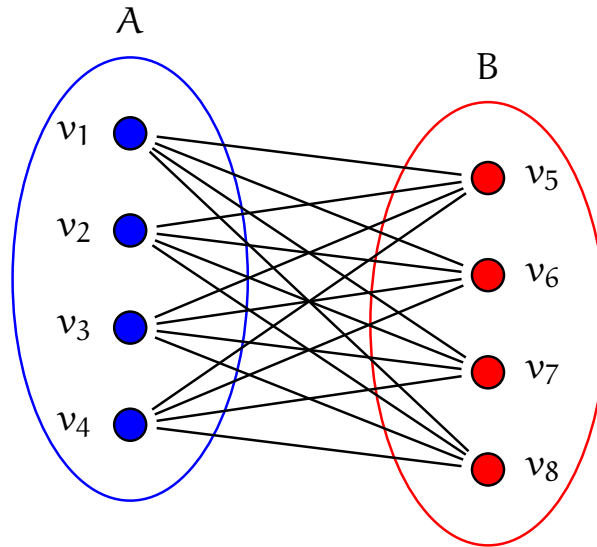


Figura 1.3: Exemplo de grafo biclique.

práticas desse problema, incluindo campos como biologia computacional e sistemas nanoeletrônicos (Wang et al. 2018), bem como sistemas de recomendação (Wang & Chen 2020), ressaltando sua relevância em diversas áreas de pesquisa e aplicação.

### 1.1.2 Entradas bipartidas e gerais

É fundamental ressaltar que este trabalho lida com duas categorias de entradas em grafos: entradas gerais e entradas bipartidas. A principal distinção entre elas reside no fato de que, nas entradas bipartidas, já temos uma divisão prévia em dois conjuntos de vértices: um pertencente à parte A e outro à parte B. Nesse cenário, os vértices de uma mesma parte não possuem conexões entre si. Por outro lado, nas entradas gerais, trabalhamos com um único conjunto de vértices e arestas, sem uma divisão explícita em partes.

Essa diferenciação tem implicações significativas, uma vez que, como será detalhado no Capítulo 4, requeremos abordagens distintas para lidar com esses dois tipos de entradas, incluindo soluções e estruturas diferentes.

### 1.1.3 Aplicações

Mencionamos anteriormente algumas aplicações relacionadas ao problema da Biclique Máxima Balanceada. Embora a maioria dessas aplicações esteja inicialmente associada a esse problema específico, é importante destacar que elas podem apresentar particularidades que levam a uma generalização do problema original. Abaixo, apresentamos alguns exemplos de aplicações no contexto do problema abordado neste trabalho:

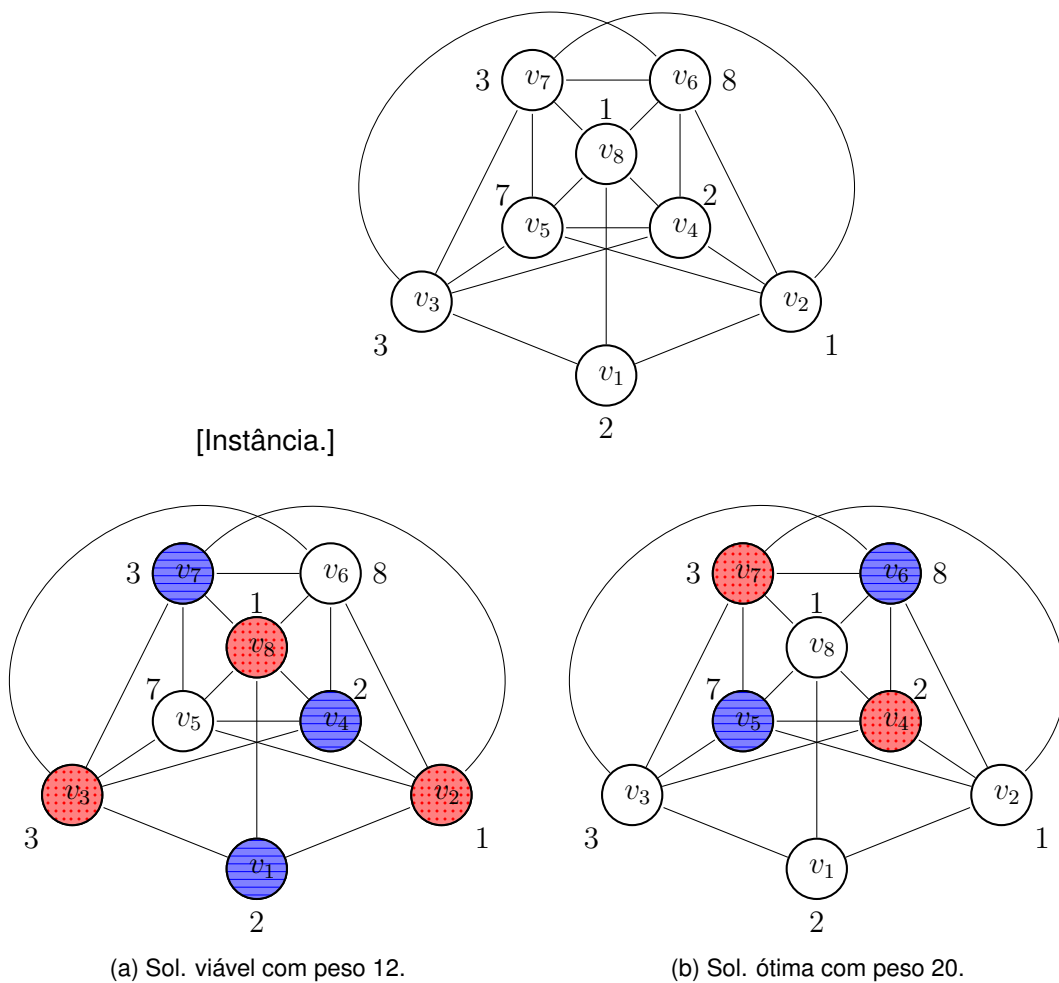


Figura 1.4: Biclique máxima ponderada.

## Sistemas de Recomendação

Uma das aplicações mais notáveis e em crescimento constante é de Sistemas de Recomendação.

Com a ascensão das redes sociais, como o Instagram, as plataformas de streaming, como Netflix e Spotify, e os gigantes do comércio eletrônico, como Amazon e Mercado Livre, tornou-se vital a implementação de ferramentas de recomendação capazes de proporcionar previsões precisas alinhadas com as preferências do usuário.

Neste contexto, as bicliques desempenham um papel central e fundamental. Como ilustrado na Figura 1.5, podemos representar usuários e filmes (ou outros itens) como vértices e estabelecer conexões entre eles quando os usuários avaliam positivamente os filmes. Encontrar a maior biclique máxima balanceada permite identificar grupos de usuários com preferências similares, possibilitando a recomendação de filmes, produtos ou músicas.

Conforme evidenciado anteriormente, esse exemplo pode ser facilmente adaptado a diversos outros contextos, como a relação entre usuários e itens em marketplaces ou a afinidade entre usuários e músicas.

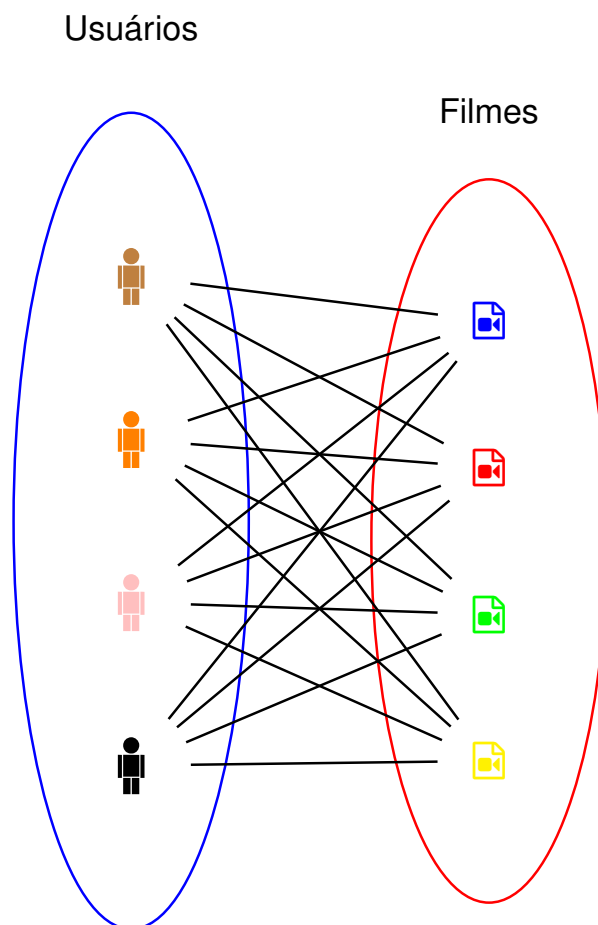


Figura 1.5: Exemplo de biclique em Sistemas de Recomendação.

## Biologia Computacional

Uma aplicação notável e essencial na vanguarda do desenvolvimento de novos medicamentos é encontrada na área de biologia computacional. Nesse contexto, é possível estabelecer conexões entre genes e doenças, criando uma base sólida para a pesquisa e produção de novos medicamentos com base nesses insights.

A representação típica desse problema é ilustrada na Figura 1.6. Os vértices são divididos em duas partes distintas: a parte relacionada aos genes e a parte referente às condições (representando doenças). Encontrar a maior biclique balanceada nos permite identificar um conjunto de genes e condições que compartilham características similares. Essa análise é de fundamental importância para estabelecer relações entre genes e doenças, que, por sua vez, serve como base para o desenvolvimento ou aprimoramento de fármacos.

Uma variante relevante dessa aplicação envolve a atribuição de pesos aos vértices, onde cada gene e condição são associados a um peso que reflete sua importância na análise. Quanto maior o peso atribuído, maior a relevância. Ao encontrar a maior biclique com pesos nos vértices, é possível estabelecer a relação mais precisa entre genes e condições, com base nos

pesos atribuídos, proporcionando uma análise mais refinada das conexões entre genes e doenças.

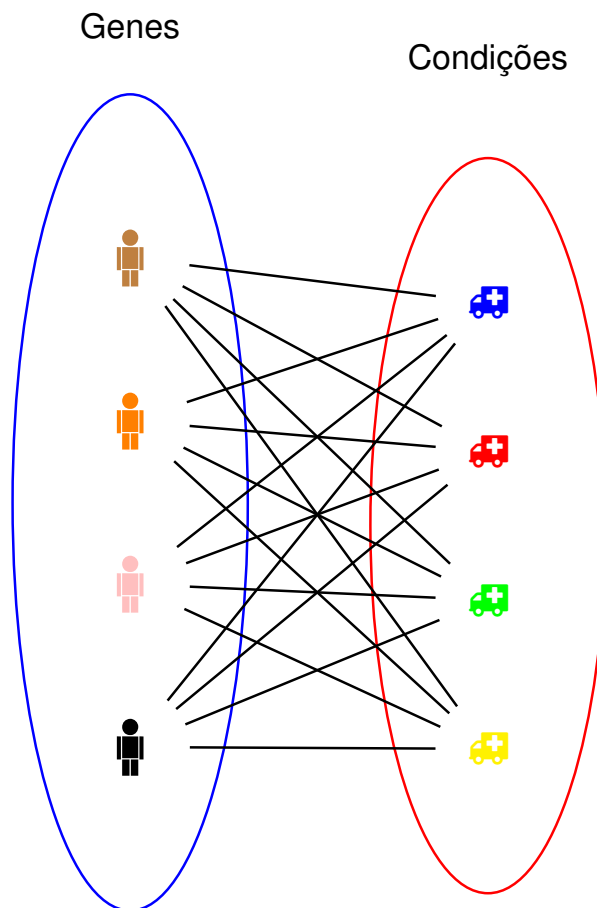


Figura 1.6: Exemplo de biclique em Biologia Computacional.

### Sistemas Nanoeletrônicos

Um desafio crucial que afeta diretamente o campo dos sistemas nanoeletrônicos diz respeito à tolerância defeituosa em interruptores de nanotecnologia, com o objetivo de identificar a maior barra livre de defeitos. Nesse contexto, uma barra é constituída por nanofios de entrada e saída, conectados por interruptores funcionais. O problema em questão pode ser representado como uma biclique, composta por duas partes distintas: os nanofios de entrada e os nanofios de saída.

Para solucionar o problema original, a abordagem consiste em encontrar a maior biclique possível, onde os interruptores sem defeitos são as arestas que conectam os nanofios. Para adaptar esse cenário ao problema tratado neste trabalho, basta introduzir a noção de prioridades nos vértices, atribuindo prioridades mais elevadas (ou pesos mais altos) aos nanofios considerados mais cruciais.

Essa adaptação permite uma análise refinada, levando em consideração a importância relativa dos componentes, e contribui para a resolução eficaz do problema em sistemas nanoeletrônicos.

## 1.2 Motivação

A motivação central deste trabalho reside na exploração e compreensão do problema da Biclique Máxima Balanceada com Peso nos Vértices, que se destaca por sua ampla aplicabilidade. Além disso, esse projeto visa ao desenvolvimento de habilidades pessoais nas áreas de iniciação científica, pesquisa operacional, teoria dos grafos e meta-heurísticas.

Um ponto relevante a ser destacado é a escassez de estudos abordando exatamente o mesmo problema, seja de forma exata ou heurística, na literatura existente. Embora haja inúmeros trabalhos relacionados a problemas semelhantes, uma investigação minuciosa revela que apenas um estudo ([Zhao et al. 2022](#)) se dedica a um problema bastante próximo, o da Biclique  $k$ -Máxima Balanceada com Peso nos Vértices. Nessa variante, existe um limite  $k$  para o tamanho da biclique. No entanto, não foi encontrado na literatura nenhum trabalho que aborde precisamente o problema tratado neste estudo, sem restrições ou limitações na solução, seja por meio de abordagens exatas ou meta-heurísticas.

## 1.3 Objetivos

O projeto tem como objetivo geral desenvolver soluções heurísticas para o problema da biclique balanceada máxima com peso nos vértices.

Além disso, o trabalho desenvolvido possui os seguintes objetivos específicos:

- I. Implementar uma meta-heurística para resolver o problema da biclique balanceada máxima com peso nos vértices.
- II. Desenvolver novos algoritmos de busca local, visando aprimorar ainda mais os resultados alcançados pela meta-heurística implementada.
- III. Estabelecer diretrizes para a redução das instâncias do problema, com o intuito de gerar um grafo reduzido que possa ser mais facilmente tratada por algoritmos exatos.

## 1.4 Contribuições

Neste estudo, desenvolvemos uma meta-heurística denominada Reactive GRASP para resolver o problema proposto, complementada pela aplicação do VND (Variable Neighborhood Descent) na etapa de busca local.

Adicionalmente, introduzimos estruturas de vizinhança específicas para aprimorar o desempenho da busca local, juntamente com a criação de uma regra de redução. Além disso, apresentamos um teorema que se mostra particularmente eficaz em cenários que envolvem instâncias esparsas do problema.

## 1.5 Estrutura do texto

A estrutura do texto segue com o Capítulo 2 que mostra alguns trabalhos relacionados junto de uma breve explicação sobre cada um e o Capítulo 3 discutindo sobre os conceitos envolvidos no trabalho, apresentando meta-heurísticas, complexidade de classes e programação linear. No Capítulo 4 temos a apresentação da nossa proposta com os algoritmos e estruturas utilizadas, além do teorema de redução desenvolvido.

Subsequentemente serão apresentados os resultados no Capítulo 5 com tabelas e gráficos, comparações de algoritmos e explicações de escolhas feitas durante o trabalho. Por fim, o Capítulo 6 contém a conclusão e as considerações finais.

# 2

## Trabalhos relacionados

Esse capítulo apresenta alguns dos principais trabalhos encontrados na literatura que estão relacionados de alguma forma ao problema abordado.

### 2.1 Trabalhos Teóricos

Alguns trabalhos, apesar de parte deles possuírem alguma implementação algorítmica, foram utilizados com um propósito teórico, seja para entender melhor sobre grafos bipartidos ou sobre as suas aplicações.

Em [Garey & Johnson \(1979\)](#) é feito um trabalho muito completo com provas das completudes de inúmeros problemas. Entre eles, se destacam alguns problemas que foram provados como a clique, clique *cover* e *maximum bipartite subgraph*. Nesse trabalho, é provado a  $\mathcal{NP}$ -completude do problema de encontrar uma biclique balanceada com um tamanho fixo e também prova que o problema de encontrar a biclique com o maior número de vértices, onde ela não é balanceada, pode ser resolvido em tempo polinomial. A partir de trabalhos como esse é possível ter uma base da dificuldade de implementação e da complexidade de um algoritmo para certos tipos de problemas, como geralmente acontece com grafos bipartidos.

[Peeters \(2003\)](#) cita o trabalho de [Garey & Johnson \(1979\)](#) falando sobre algumas provas já realizadas da biclique. Ele se propõe a provar a completude da *Maximum edge biclique problem (MBP)*, onde mostra que é  $\mathcal{NP}$ -completo. Ele cita também outros artigos que falam sobre o MBP e que confirmam a prova de completude realizada, além de mencionar que a sua versão com peso também é  $\mathcal{NP}$ -completo.

Já em [Feige & Kogan \(2004\)](#) é feito um estudo teórico onde são apresentados provas e teoremas da biclique balanceada máxima onde é feita a sua introdução e definição, além de apresentar a sua completude e aplicações. Outro ponto interessante de se ressaltar nesse trabalho é que também é feito o mesmo para o *Maximum Edge Biclique problem* que procura maximizar o número de arestas de uma biclique.

## 2.2 Trabalhos Práticos

Existem variações do problema abordado e uma delas é o problema da biclique máxima balanceada sem peso, onde várias soluções algorítmicas foram propostas na literatura. A maioria dos trabalhos detalhados abaixo remete a versão sem peso da biclique máxima balanceada, apenas o último artigo citado abaixo lida com um problema diferente e que se aproxima bastante do que está sendo tratado aqui.

No trabalho de [Zhou & Hao \(2017\)](#) apresenta uma heurística diferente que combina um procedimento de busca tabu baseado em restrições e duas técnicas de redução de grafo. Nos resultados, o algoritmo proposto teve uma eficácia em 30 instâncias de benchmarks clássicos aleatórios e em 25 instâncias do Koblenz Network Collection (KONECT) onde são grafos massivos, esparsos e de problemas reais.

Já em [Wang et al. \(2018\)](#) é proposto uma nova estrutura de busca local para o problema da biclique máxima balanceada onde quatro heurísticas foram desenvolvidas para melhorar seu desempenho. A estrutura criada alterna entre uma fase de extensão que adiciona pares de vértices e uma fase de reinício que remove pares de vértices da solução.

Foram criadas três heurísticas que selecionam os pares para adição e remoção. A primeira heurística é uma função gulosa de pontuação de previsão que seleciona os pares de vértices para adição fazendo uso da informação estrutural do problema. A segunda heurística é uma heurística de reinicialização auto-adaptativa que remove um número dinâmico de pares de vértices da solução candidata para permitir que a busca continue a partir de uma nova área de busca. A terceira é proposta para resolver grafos massivos e é chamada de heurística de perturbação *two-mode* em que é usado para selecionar pares de vértices que entrarão na solução e reduz a complexidade média para esta tarefa. Também foi desenvolvida uma regra de redução para diminuir a escala de instâncias massivas. Todas essas técnicas desenvolvidas levam a dois algoritmos de busca local eficientes para o problema abordado nesse artigo. Os resultados obtidos demonstraram que os algoritmos propostos funcionam bem com instâncias massivas de bilhões de arestas.

Em outra abordagem, [Zhou et al. \(2018\)](#) desenvolveram um algoritmo exato para o problema. Nesse trabalho foi proposto novas ideias para criar algoritmos exatos eficazes para o problema da biclique máxima balanceada em grafos bipartidos. Inicialmente é utilizado um procedimento de Upper Bound Propagation (UBP) para pré-calculer um *upper bound* de cada vértice. Em seguida, um algoritmo de Branch-and-Bound (B&B) é estendido para trabalhar com os resultados obtidos na fase anterior, onde foi calculado um *upper bound* de cada vértice. Com base no UBP também foi estudado um novo modelo de programação linear inteira. Além disso, foram introduzidas novas desigualdades válidas obtidas a partir dos limites superiores para restringir as formulações matemáticas. Nos experimentos realizados com grafos bipartidos aleatórios demonstraram a eficiência do algoritmo B&B estendido e das desigualdades válidas geradas.

Recentemente, [Li et al. \(2020\)](#) propuseram o algoritmo *general swap-based multiple neighborhood adaptive search (SBMNAS)* para a biclique máxima balanceada junto de uma regra de redução de grafo. Este algoritmo combina um operador *k-SWAP* que é usado na etapa de busca local e que funciona como uma regra adaptativa para a exploração da vizinhança e como uma estratégia de perturbação para garantir a diversificação da solução. O *SBMNAS* foi avaliado em 60 instâncias densas aleatórias e 25 instâncias Koblenz Network Collection (KONECT). Os resultados mostraram que o algoritmo proposto atinge todas as soluções mais conhecidas, exceto em uma, e encontra os melhores resultados conhecidos para 19 instâncias (novos limites inferiores).

Nesse ano, [Zhao et al. \(2022\)](#) propôs um problema um pouco diferente: encontrar a *k*-máxima biclique balanceada com peso nos vértices em grafos bipartidos. A principal diferença desse problema para o abordado aqui é justamente a limitação *k* que é imposta para o tamanho das partes da biclique.

Nesse artigo, dado um grafo bipartido ponderado *G* e uma restrição *k*, tem como objetivo encontrar a maior biclique *k*-máxima balanceada (*Max k BB*) com peso nos vértices. Para resolver esse problema é proposto uma abordagem baseada na enumeração de uma das partes da biclique *G* seguido de uma estrutura *Branch-and-Bound*. Para melhorar o desempenho foram elaboradas três estratégias de otimização para excluir ramificações inválidas. Além disso, foi utilizada uma estratégia de redução de grafo para reduzir o espaço de busca.

Experimentos extensivos foram conduzidos em 12 conjuntos de dados de grafos bipartidos que demonstraram a eficiência, eficácia e escalabilidade dos algoritmos propostos. Por fim, os resultados experimentais mostraram que os algoritmos podem resolver o problema de detecção da biclique máxima balanceada de forma eficiente.

# 3

## Fundamentação teórica

Este capítulo fornece a base conceitual para compreender as áreas e conceitos abordados neste trabalho. Ele inicia com uma introdução fundamental às classes de complexidade, aborda também a programação linear e, de maneira mais ampla, explora o campo das meta-heurísticas.

### 3.1 Complexidade de problemas

Vamos iniciar esta seção com a introdução ao conceito de classes de complexidade, um dos pilares fundamentais da teoria da computação. Este conceito é de importância crucial para compreender a dificuldade subjacente à busca de soluções para o problema abordado nesta pesquisa.

Para avaliar adequadamente a complexidade de um problema e, por conseguinte, a dificuldade inerente em resolvê-lo, é essencial categorizá-lo. As classes de complexidade oferecem um arcabouço teórico para essa classificação, agrupando problemas com base na complexidade de suas soluções mais eficientes dentro de cada classe. Além disso, é relevante destacar que existem distintas categorias de problemas, como problemas de decisão, geralmente mais simples de resolver, e problemas de otimização, que costumam ser mais desafiadores devido à sua complexidade temporal e algorítmica. No caso dos problemas de otimização, o algoritmo mais eficiente muitas vezes envolve múltiplas etapas e cálculos complexos.

Entre as classes de complexidade mais notáveis, destacam-se quatro:  $\mathcal{P}$ ,  $\mathcal{NP}$ ,  $\mathcal{NP}$ -completo e  $\mathcal{NP}$ -difícil. Problemas pertencentes à classe  $\mathcal{P}$  (Polinomial) podem ser resolvidos em tempo polinomial, ou seja, o tempo necessário para resolvê-los aumenta de forma razoável em relação ao tamanho da entrada. Por outro lado, a classe  $\mathcal{NP}$  (Polinomial Não-Determinístico) engloba problemas que permitem verificar se uma solução é válida em tempo polinomial. A principal diferença em relação à classe  $\mathcal{P}$  reside no fato de que encontrar a solução não é necessariamente fácil ou eficiente. Vale mencionar que uma das maiores dificuldades na teoria da complexidade

é provar que  $\mathcal{P}$  é igual a  $\mathcal{NP}$  ou não. Em outras palavras, se todo problema em  $\mathcal{NP}$  pode ser resolvido em tempo polinomial.

A classe  $\mathcal{NP}$ -completo abrange os problemas mais desafiadores que se encaixam na classe  $\mathcal{NP}$ , enquanto os problemas  $\mathcal{NP}$ -difícil, por outro lado, não são necessariamente limitados à classe  $\mathcal{NP}$ , podendo apresentar complexidades a nível de  $\mathcal{NP}$ -completo ou até mesmo superiores em relação aos problemas mais difíceis pertencentes à classe  $\mathcal{NP}$ . Em casos especiais, um problema  $\mathcal{NP}$ -difícil pode, de fato, ser classificado como  $\mathcal{NP}$ -completo.

Para mais informações sobre classes de complexidade, ver [Garey & Johnson \(1979\)](#) e [Papadimitriou \(1994\)](#).

## 3.2 Programação Linear e Inteira

A programação linear e inteira desempenham papéis significativos na área de Pesquisa Operacional (PO). Embora ambas sejam métodos exatos, elas se aplicam a categorias distintas de problemas. A programação linear lida com variáveis contínuas, como tempo e dinheiro, enquanto a programação inteira procura discretizar valores, tornando-a mais complexa devido à natureza discreta das variáveis.

No contexto deste estudo, optamos por utilizar a programação inteira para a busca de soluções ótimas e a usamos como ponto de comparação para a meta-heurística proposta. A escolha da programação inteira em vez da linear se justifica pela natureza do problema que estamos abordando: encontrar a maior biclique balanceada com peso nos vértices. Neste contexto, as variáveis representam a presença ou ausência de vértices na solução, sendo um valor booleano. Além disso, é importante ressaltar que a solução não admite "meio"vértice, apenas valores discretos. O livro [Wolsey \(1998\)](#) se estende mais sobre a programação inteira.

Um aspecto essencial da programação inteira reside na sua capacidade de simplificar a resolução de problemas complexos. Não é necessário desenvolver algoritmos intrincados para abordar desafios de alta complexidade; em vez disso, é crucial definir de maneira precisa a estrutura do problema, estabelecendo a função objetivo (indicando o que se deseja maximizar ou minimizar) juntamente com as restrições, que delineiam as limitações da solução. Como exemplo, podemos considerar a exclusão de um conjunto específico de vértices como resultado de uma restrição. Um exemplo da formulação da função objetivo e das restrições serão mostradas na seção 4.7.

No mercado, uma das tecnologias mais amplamente utilizadas para a aplicação de programação inteira é o CPLEX. Por meio dessa ferramenta, é possível definir a entrada, estabelecer a função objetivo e criar um conjunto de regras e restrições para gerar um algoritmo exato, simplificando significativamente a resolução de problemas complexos. O CPLEX, portanto, torna acessível a programação inteira e suas capacidades de otimização a uma ampla gama de aplicações práticas em diversas áreas, incluindo logística, economia, engenharia e muitas outras.

### 3.3 Meta-heurísticas

O último segmento da nossa base teórica trata das meta-heurísticas, que representam o foco central deste estudo, uma vez que a metodologia adotada está fundamentada na meta-heurística GRASP.

Em linhas gerais, as meta-heurísticas são algoritmos de otimização que não prometem uma solução exata, mas buscam se aproximar ao máximo dela. Embora não assegurem a obtenção da solução ótima (que corresponde ao resultado exato), essas abordagens têm a capacidade de gerar uma variedade de soluções, explorando diferentes vizinhanças em busca de ótimos locais e, ao mesmo tempo, direcionando-se em direção ao ótimo global (que não é conhecido). O fato de não garantirem a solução ótima, que normalmente demandaria um tempo computacional excessivo, torna essas técnicas particularmente atraentes, uma vez que frequentemente são capazes de encontrar soluções aceitáveis em prazos mais curtos. Isso tem contribuído para sua crescente popularidade e aplicação em uma ampla gama de problemas complexos de otimização.

Algumas das meta-heurísticas mais renomadas incluem o GRASP, ILS, busca tabu e algoritmos genéticos.

Meta-heurística como o GRASP possui duas fases, uma de construção de uma solução inicial a cada iteração e uma busca local nessa solução gerada. Já o ILS gera uma solução inicial e mantém a mesma solução ao longo das iterações, mas sempre com um mecanismo de perturbação para ter uma solução levemente diferente e procurar por outras vizinhanças. Ele também passa por uma busca local e reseta a sua solução após um número de iterações definidas por um parâmetro.

Um outro exemplo é a busca tabu, essa se caracteriza por bloquear soluções já encontradas anteriormente, colocando um valor tabu para vértices ou arestas e impedindo que entrem na solução até que esse valor seja modificado para possibilitar a sua entrada. Com essa estratégia, ela busca diversificar a sua vizinhança.

Algoritmos genéticos geram populações onde cada indivíduo é uma solução e possui estratégias de seleção dos melhores indivíduos da população a cada iteração, mutações para modificar levemente um indivíduo, crossover para gerar novos indivíduos baseado na combinação de dois indivíduos da população existente. A cada iteração, ocorre uma substituição da população pela nova população gerada, mantendo apenas os melhores indivíduos da população (parcela elite da população).

Na seção subsequente, dedicaremos uma análise mais aprofundada ao GRASP, detalhando suas características e os princípios de seu funcionamento. O [Resende & Ribeiro \(2019\)](#), [Talbi \(2009\)](#) e [Glover & Kochenberger \(2006\)](#) entram em mais detalhes sobre algumas meta-heurísticas citadas como o GRASP e mostram o funcionamento de diversas outras.



## Nossa proposta

Esse capítulo apresenta a proposta desenvolvida para o PMBBPV, onde será explicado a meta-heurística desenvolvida, a busca local (para entradas gerais e bipartidas) e o algoritmo de redução com pseudocódigo, definições e teoremas.

### 4.1 Proposta

Nesse trabalho é proposto uma meta-heurística Reactive GRASP com VND para o PMBBPV. A meta-heurística GRASP, proposta por [Feo & Resende \(1995\)](#), possui duas fases. A primeira um método guloso aleatório que constrói uma nova solução de acordo com um parâmetro de entrada  $\alpha$ . E a segunda fase, é executada uma busca local que refina a solução construída na etapa anterior. Dentre as meta-heurísticas existentes, o GRASP tem se mostrado uma das mais competitivas ([Resende & Ribeiro 2019](#)).

No GRASP<sub>VND</sub> proposto, a fase construtiva cria uma solução viável considerando o peso do vértice como critério guloso. O método é classificado como Reactive GRASP ([Prais & Ribeiro 2000](#)) pois é feito um ajuste automático do parâmetro  $\alpha$  no decorrer do algoritmos. Além disso, foi desenvolvido um mecanismo de redução de instância incorporado ao Reactive GRASP. Dessa forma, toda vez que um novo *best* é encontrado, o mecanismo de redução é ativado tentando reduzir o grafo a partir desse novo valor.

Também apresentamos uma aprimoramento no mecanismo de redução. Durante a fase de testes e experimentos conduzidos, observou-se que a estratégia de redução de grafos não era universalmente eficaz em todas as instâncias. Como resultado dessas observações, desenvolvemos uma condição que determina a ativação ou desativação do algoritmo de redução para uma instância específica. Essa condição é baseada na análise da quantidade de vértices e na densidade do grafo, conforme detalhado posteriormente nos resultados.

## 4.2 Algoritmo GRASP-VND

O Algoritmo 4.1 apresenta o pseudocódigo do  $\text{GRASP}_{\text{VND}}$  desenvolvido. Inicialmente, utilizando o construtivo do  $\text{GRASP}_{\text{VND}}$  é criada uma solução inicial, caso necessário, ocorre um balanceamento na solução seguido de uma busca local (linhas 3–6). Na linha 8, a variável  $x$  (inicialmente com valor  $\beta$ ) indicará por mais quantas iterações o algoritmo poderá encontrar a atual melhor solução (serve para que o programa não perca tempo tentando melhorar uma solução que supostamente já é ótimo global) e a linha 9 indica a quantidade de iterações que o parâmetro  $\alpha$  será reajustado com o `reactiveGrasp()`.

Nas linhas 2–11 o valor de  $\alpha$  é escolhido baseado em uma distribuição em que os valores entre 0 e 1 possuem uma probabilidade diferente de serem escolhidos de acordo com os resultados obtidos pelo Reactive GRASP.

O começo do *loop* do  $\text{GRASP}_{\text{VND}}$  é dado na linha 10 por um critério de parada. As linhas 12–15 repetem o mesmo processo das linhas 3–6. Após a busca local (VND), é verificado se a solução local ( $S'$ ) é melhor que a solução atual ( $S^*$ ) através da função `custo` que calcula o peso total das soluções (linha 16). Caso ela seja a melhor,  $S^*$  é atualizado (linha 17), o valor de  $x$  é reinicializado para o valor do parâmetro de encerramento  $\beta$  (linha 18) e a redução (linha 20) — utilizando a função `reduce` — é chamada caso a instância atinja as condições necessárias. Caso  $S' = S^*$  — utilizando a função `custo` para a verificação —, o valor de  $x$  é decrementado (linha 22).

Se  $y = 0$ , os parâmetros relacionados ao Reactive GRASP são atualizados de acordo com os valores de  $\alpha$  escolhidos e os resultados obtidos (linha 27). Finalmente, as linhas 24 e 29 representam a finalização do algoritmo retornando a melhor solução encontrada até o momento.

**Algoritmo 4.1** Pseudocódigo do GRASP<sub>VND</sub>**Input** : Grafo de entrada  $G = (V, E)$ , função de peso nas arestas  $w$ , parâmetros  $\alpha$  e  $\beta$ **Output**: Biclique balanceada  $S^*$ 

```

1 GRASPVND( $G, w, \alpha, \beta$ )
2    $\alpha = \text{chooseAlpha}()$ 
3    $S = \text{greedyRandomizedConstructive}(\alpha)$  // solução inicial
4   if  $S$  não é balanceada then
5      $\text{balanceBiclique}(S)$ 
6    $S = \text{LocalSearch}(S)$  // aplica VND
7    $S^* = S$  // melhor solução
8    $x = \beta$  // parâmetro de encerramento do algoritmo
9    $y = \text{alphaCalibration}$  // parâmetro de iterações do reactive grasp
10  while critério de parada do
11     $\alpha = \text{chooseAlpha}()$ 
12     $S' = \text{greedyRandomizedConstructive}(\alpha)$  // nova solução
13    if  $S'$  não é balanceada then
14       $\text{balanceBiclique}(S')$ 
15     $S' = \text{LocalSearch}(S')$  // aplica VND
16    if  $\text{custo}(S^*) < \text{custo}(S')$  then
17       $S^* = S'$  // atualiza a melhor solução
18       $x = \beta$  // reinicia o parâmetro de encerramento
19      if autoReduce then
20         $\text{reduce}(G, \text{custo}(S^*))$  // aplica a redução
21    else if  $\text{custo}(S^*) == \text{custo}(S')$  then
22       $x = x - 1$  // caso a solução seja a mesma
23    if  $x == 0$  then
24      return  $S^*$  // finaliza algoritmo ( $\beta$  iterações sem melhora)
25    if  $y == 0$  then
26      // atualização dos parâmetros do reactive grasp
27       $y = \text{alphaCalibration}$ 
28       $\text{reactiveGrasp}()$ 
29     $y = y - 1$ 
30  return  $S^*$ 

```

## 4.3 Busca Local

Após o método construtivo, inicia-se o método de busca local baseado no VND (Mladenovic & Hansen 1997). Tendo como entrada uma biclique inicial já balanceada, a busca local tenta melhorar a solução com a exploração do espaço de solução. Dado um conjunto de estruturas de vizinhança, o VND explora o espaço de solução trocando sistematicamente as estruturas de vizinhança. O método aceita apenas soluções que melhorem a solução corrente e, nesses

casos, o método volta à primeira estrutura de vizinhança e inicia uma nova busca.

Conforme mencionado na introdução, o algoritmo pode ser executado em dois tipos de entradas: as entradas gerais e as entradas bipartidas. Para viabilizar essa flexibilidade, além de adaptar o tratamento dos dados de entrada, desenvolvemos estruturas de vizinhança distintas, ajustadas às particularidades de cada tipo de entrada.

#### 4.3.1 Entradas Gerais

O método proposto para instâncias gerais utiliza três estruturas de vizinhança: “add”, “swap(1,1)” e o “swap(2,2)”, com mecanismo de *first-improvement* para a primeira e terceira estruturas e de *best-improvement* para a segunda.

A primeira estrutura é o add que tenta adicionar um par aleatório de vértices na solução. Inicialmente, é escolhido aleatoriamente o primeiro vértice livre (para a parte  $V_1$ ) e, a partir desse vértice, ocorre uma busca pelos seus vizinhos à fim de encontrar um vértice livre para a parte  $V_2$  que, juntos, apresentem uma melhora na solução. Caso o vértice escolhido aleatoriamente não encontre um vértice para a parte  $V_2$ , então será analisado o próximo vértice (entre os vértices livres para  $V_1$ ) e seus vizinhos. O processo acaba quando for encontrado o primeiro par que atenda às condições ou quando não há nenhum par.

A segunda estrutura é o swap(1,1). Esta estrutura irá realizar a melhor troca, ou seja, a que irá gerar o maior acréscimo no peso da solução, entre um vértice que está na solução e um vértice não-livre (não está na solução e não está disponível) desde que mantenha as propriedades de biclique balanceada.

A terceira estrutura é o swap(2,2). Esta estrutura irá realizar a primeira troca possível entre dois vértices que estão na solução e dois vértices que não podem entrar na solução por conta de um único vértice de forma que, após a troca, o peso da solução seja maior. Para realizar essa operação, selecionamos dois vértices que, inicialmente, não estão incluídos na solução, e verificamos se cada um deles possui apenas um vértice diferente que impede a sua inclusão na solução. Além disso, avaliamos se a diferença entre os pesos é positiva, ou seja, se o ganho de peso (representado pela diferença entre os pesos dos vértices que estão sendo incluídos e os que estão sendo removidos) é favorável. Caso ambas as condições sejam atendidas, procedemos com a troca.

#### 4.3.2 Entradas Bipartidas

O método proposto para instâncias bipartidas utiliza duas estruturas de vizinhança: “add” e “swap(1,1)”, com mecanismo de *first-improvement* para a primeira estrutura e de *best-improvement* para a segunda.

A primeira estrutura é a add e opera de maneira análoga àquela utilizada nas entradas gerais. Por sua vez, a segunda estrutura, conhecida como swap(1,1), também segue uma

lógica semelhante. No entanto, em vez de considerar o peso como critério principal para a troca, essa estrutura se baseia no grau do vértice.

## 4.4 Estruturas auxiliares

Foram criadas algumas estruturas auxiliares para facilitar a manipulação da solução, tanto na geração de uma solução inicial quanto nas estruturas de vizinhança durante a busca local. A primeira dessas estruturas é o vetor *Solution*, que é dividido em quatro partes: a primeira parte contém os vértices que estão na solução, a segunda parte contém os vértices que não estão na solução, mas estão disponíveis para entrar. A terceira parte contém os vértices que não podem entrar na solução, e a última parte contém os vértices removidos.

Outras estruturas incluem o *MU* e o *Tightness*. O *MU* contém os valores que cada vértice irá adicionar ou subtrair ao entrar na solução, enquanto o *Tightness* contém a quantidade de vizinhos que o vértice possui na solução.

Cada estrutura é dividida em duas partes: uma para a parte A e outra para a parte B. Cada vetor possui o tamanho correspondente à quantidade de vértices no grafo, sendo que, no *MU* e no *Tightness*, cada posição do vetor representa um vértice. Como cada vetor é dividido em duas partes (uma para cada componente), o que ele representa está relacionado àquela parte. Por exemplo, o *MU* de um vértice na parte A indica quanto aquele vértice irá contribuir, positiva ou negativamente, para a parte A ao entrar na solução.

## 4.5 Redução

Foi desenvolvida uma técnica de redução de grafo para a PMBBPV que funciona tanto para entradas bipartidas quanto para entradas não biparticionadas. A redução calcula um *upper bound* (UB) para todo vértice  $v \in V$  que estima o custo da biclique máxima que  $v$  pode pertencer. Este limite (bound) determina se o vértice pode ser removido ou não. O UB foi elaborado de forma que seu cálculo não fosse tão complexo a ponto de atrapalhar o desempenho da heurística. Posteriormente, percebeu-se que a redução funciona bem, principalmente, em grafos esparsos.

A redução implementada conseguiu remover uma grande quantidade de vértices e de arestas a partir de instâncias de grafos esparsos utilizadas em problemas reais, o que comprovou a sua eficiência.

Para explicar a técnica desenvolvida para a redução, serão apresentadas duas estruturas utilizadas: a primeira foi o *HI* (h-index) e a segunda estrutura é o *AS* (accumulated sum).

**Definição 2.** Dado um grafo  $G = (V, E)$ , em que  $v \in V$ ,  $N_G(v)$  é o conjunto de vizinhos de  $v$ ,  $u_i$  o  $i$ -ésimo vizinho de  $v$  ordenado de forma decrescente pelo grau e  $d(u_i)$  é o grau de  $u_i$ . O valor de *HI* de um vértice  $v$  é definido como um inteiro  $HI(v)$ , tal que  $HI(v) = \max\{i \in \mathbb{N} | d(u_i) \geq i\}$ .

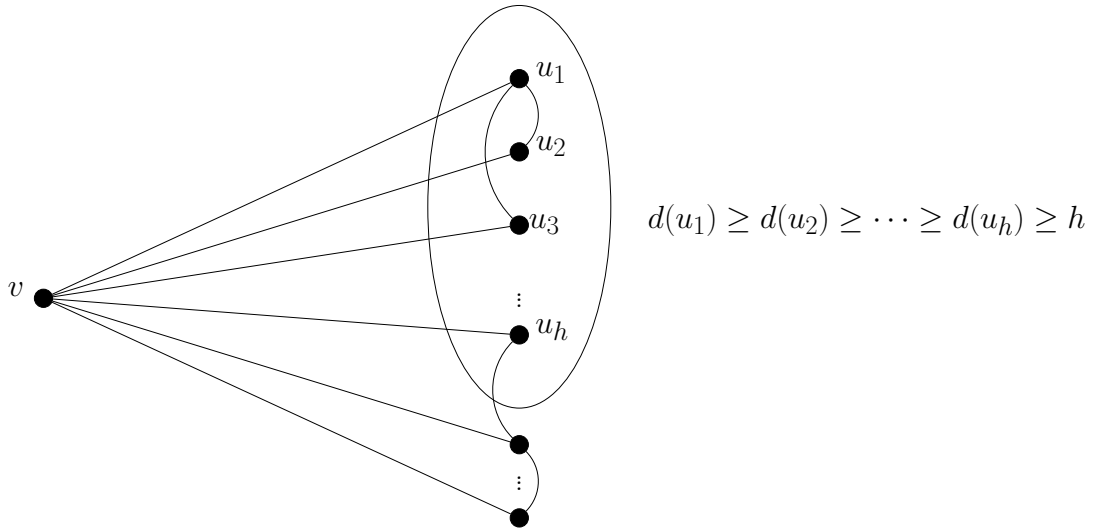


Figura 4.1: Exemplo do cálculo de  $HI(v) = h$ .

**Regra 1.** Dado um grafo  $G = (V, E)$  e um vértice  $v \in V$  com  $HI(v) = h$  temos que a maior biclique balanceada que  $v$  pode pertencer possui cardinalidade  $2h$ .

Com isso, podemos concluir que a estrutura HI pode calcular o tamanho máximo de uma possível biclique envolvendo o vértice  $v$ .

**Definição 3.** Dado um grafo  $G = (V, E)$ , onde  $v \in V$ ,  $w(\cdot)$  é uma função de peso e  $v_i$  é o  $i$ -ésimo vizinho de  $v$  ordenado de forma decrescente pelo peso  $w$ . Para cada  $v \in V$  e para cada  $1 \leq i \leq n$ , a soma acumulada dos  $i$ -vizinhos mais pesados de  $v$  é definido como:

$$AS(v, i) = \sum_{j=1}^{\min\{i, d(v)\}} w(v_j).$$

Note que, como os pesos de  $v_j$  estão ordenados de forma decrescente, para um vértice  $v \in V$ ,  $AS(v, i)$  representa soma acumulada dos pesos dos  $i$  vizinhos mais pesados de  $v$ .

**Definição 4.** Dado um grafo  $G = (V, E)$ ,  $v \in V$  com  $h = HI(v)$ , define-se:

$$UB(v) = AS(v, h) + \max_{x \in N(v)} \{AS(x, h)\}.$$

**Teorema 1.** Dado um grafo  $G = (V, E)$ , onde  $v \in V$ , o  $UB(v)$  sendo o upper bound do vértice  $v$  e  $w(B_{best})$  é o peso da maior biclique encontrada. Com isso, se  $UB(v) \leq w(B_{best})$ , então não existe uma biclique  $B$  tal que  $v \in B$  e o  $w(B) > w(B_{best})$ .

*Demonstração.* Seja  $v \in V$  tal que  $UB(v) \leq w(B_{best})$ . Vamos supor que existe uma biclique  $B$  tal que  $v \in B$  e o  $w(B) > w(B_{best})$ . Sem perda de generalidade, considere  $B_1$  e  $B_2$  as partes

da biclique  $B$  com  $v \in B_1$  e  $|B_1| = t$ . De acordo com a Regra 1, a cardinalidade de cada parte de  $B$  é menor ou igual que  $HI(v)$ , isto é  $t \leq h = HI(v)$ . Pela Definição 3, tem-se que  $w(B_2) \leq SA(v, t) \leq SA(v, h)$ . Analogamente,  $w(B_1) \leq \max_{x \in B_2} \{AS(x, t)\} \leq \max_{x \in N(v)} \{AS(x, h)\}$ . Logo  $w(B) \leq UB(v)$ , absurdo.  $\square$

**Regra 2** (Redução). *Dado um grafo  $G = (V, E)$ , em que  $v \in V$ , o  $UB(v)$  sendo o upper bound do vértice  $v$  e  $w(B_{best})$  é o peso da maior biclique encontrada. Se  $UB(v) \leq w(B_{best})$ , então remova  $v$  de  $G$ .*

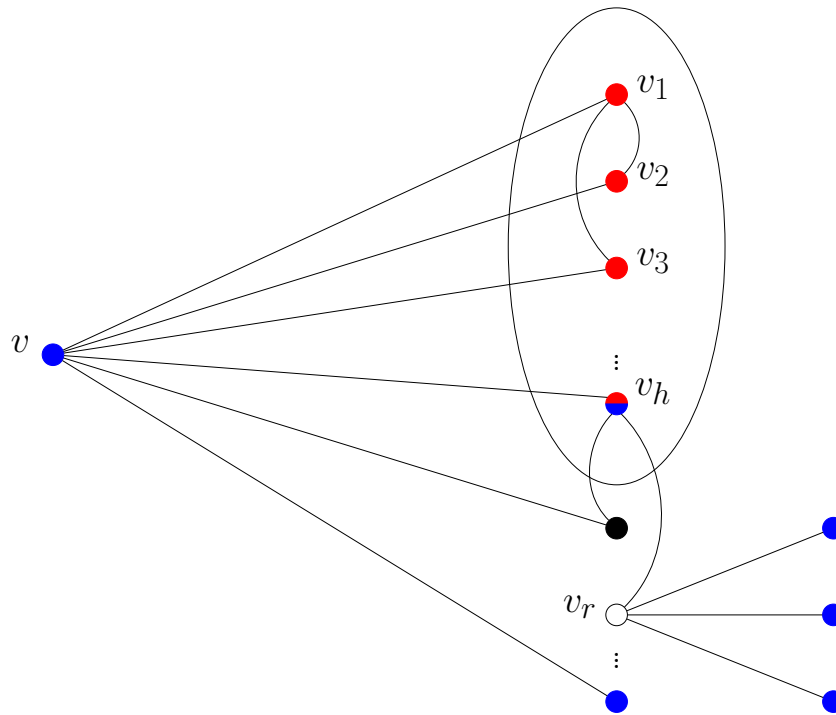


Figura 4.2: Exemplo da estimativa do  $UB(v)$ , em que os vértices em  $v_r$  é o vizinho de  $v$  que maximiza a soma acumulada. Os vértice azul são utilizados para o cálculo do *bound* na parte 1, e os vértice em vermelho são utilizados no cálculo da parte 2. Note que alguns vértices podem contribuir com as duas partes.

## 4.6 Algoritmo da Redução

O Algoritmo 4.2 descreve o pseudocódigo do algoritmo de redução. Inicialmente, a variável  $i$  é inicializada com o valor 1 (linha 31), esta variável indica a parte corrente da biclique. Em seguida, a criação da solução irá ocorrer enquanto houverem vértices existentes no grafo de uma parte (linha 32). Com isso, a linha 33 irá percorrer todos os vértices da parte selecionada e irá fazer um cálculo para prever a biclique de cada vértice. Caso a solução da biclique prevista seja pior que a solução atual (linha 34), então o vértice é removido do grafo (linha 35) e o contador de vértices removidos é acrescido (linha 36).

Ao final desse processo, a parte da bipartição é alterada (linha 37). Concluindo, na linha 39 a função `reduce` é chamada recursivamente caso algum vértice foi removido durante o processo.

**Algoritmo 4.2** Pseudocódigo da redução do GRASP.**Input** : Grafo  $G = (V, E)$  e  $best\_weight$  (peso da melhor solução)**Output**: Conjunto de vértices e arestas restantes de  $G$ 

```

30 reduce( $G = (V, E)$ ,  $best\_weight$ )
31    $i = 1$ 
32    $iter = 0$ 
33    $removed\_vertices = 0$ 
34   while  $iter \leq 2$  do
35     for  $u$  in vértices em  $V_i$  do
36       if  $predictWeight(u) \leq best\_weight$  then
37          $removeVertex(u)$  // remove o vértice  $u$  do grafo
38          $removed\_vertices = removed\_vertices + 1$ 
39        $i = (i \bmod 2) + 1$ 
40        $iter = iter + 1$ 
41   if  $removed\_vertices > 0$  then
42     return  $reduce(G, best\_weight)$  // chama de novo a redução

```

## 4.7 Formulação Matemática

Neste trabalho será proposta uma formulação matemática com o objetivo de comparar e avaliar a performance da heurística  $GRASP_{VND}$ . A formulação é definida como:

$$\max \sum_{v_i \in V} w_i x_i + \sum_{v_i \in V} w_i y_i \quad (4.1)$$

$$\text{s. a } x_i + y_i \leq 1 \quad \forall v_i \in V \quad (4.2)$$

$$x_i + x_j \leq 1 \text{ e } y_i + y_j \leq 1 \quad \forall v_i, v_j \in E \quad (4.3)$$

$$x_i + y_j \leq 1 \text{ e } y_i + x_j \leq 1 \quad \forall v_i, v_j \in E \quad (4.4)$$

$$\sum_{v_i \in V} x_i = \sum_{v_i \in V} y_i \quad (4.5)$$

$$x_i, y_i \in \{0, 1\} \quad \forall v_i \in V. \quad (4.6)$$

A formulação utiliza variáveis decisão binárias  $x_i$  e  $y_i$  para cada vértice  $v_i \in V$ . A variável  $x_i$  vale 1 se e somente se o vértice  $v_i$  pertencerá à parte  $V_1$  na solução final. Da mesma forma,  $y_i$  determina se  $v_i$  pertencerá à parte  $V_2$ . A função objetivo (4.1) contabiliza o peso da biclique. A restrição (4.2) determina que um vértice só pode pertencer a uma parte. A restrição (4.3) determina que vértices adjacentes não podem pertencer a mesma parte. A restrição (4.4) determina que vértices não-adjacentes não podem pertencer a partes distintas e a (4.5) força a biclique ser balanceada. Finalmente, a restrição (4.6) determina os domínios das variáveis.

# 5

## Resultados e discussões

Este capítulo apresenta os resultados obtidos através de uma série de testes em diversas instâncias bipartidas e não-bipartidas para uma análise mais profunda dos algoritmos apresentados e mostrar a sua eficiência.

Foi utilizada uma máquina Intel i7-10750H com 2.6 Ghz, 8 GB de memória RAM em um ambiente Windows 10 versão 21H2 onde foi utilizado o WSL 2 com Ubuntu 20.04.4. O código foi desenvolvido em C++ e compilado com o compilador GNU g++ 9.4.0. Além disso, um algoritmo exato foi implementado com o CPLEX 19.9.

### 5.1 Instâncias

Para a realização dos testes, foram selecionados diferentes tipos de instâncias para obter resultados diversos de forma que fosse possível chegar a uma conclusão para o algoritmo proposto. Um dos grupos escolhidos para teste foram as instâncias KONECT que são grandes, esparsas e obtidas de problemas reais.

As tabelas 5.1 e 5.2 mostram as instâncias DIMACS, BHOSLIB e KONECT utilizadas. A tabela 5.1 apresenta as instâncias DIMACS e BHOSLIB que tratam de grafos mais gerais, ou seja, não bipartidos e que são bastante comuns em testes na literatura. Já na tabela 5.2 temos um conjunto de instâncias KONECT que são grandes, esparsas e obtidas de problemas reais.

A coluna  $V$  indica a quantidade de vértices, a coluna  $E$  representa a quantidade de arestas e a coluna  $D$  mostra a densidade de cada instância.

Na maioria dos experimentos, o *GRASP* foi executado 10 vezes para cada instância, a fim de obter o melhor resultado *Best*, que representa o melhor desempenho obtido nas 10 execuções, a média das 10 execuções (se for um número decimal, o resultado é arredondado para baixo) e o tempo médio *Tempo(s)*, em segundos. A coluna *TTB* representa a média do tempo, em segundos, que o algoritmo leva para encontrar o valor *Best* em cada iteração daquela

instância. Se o *Best* não for encontrado na iteração específica, considera-se o valor limite de tempo naquela iteração específica ao calcular a média.

Os resultados obtidos utilizando o algoritmo exato para cada instância são apresentados na coluna *Ótimo* (indicando o valor exato da biclique) e na coluna *Tempo(s)*, em segundos. A coluna *Gap* mostra a diferença relativa entre o *Best* dos algoritmos comparados. Ao ser comparado com o exato, quanto mais próximo de 0%, melhor o resultado, indicando que a resposta da heurística está próxima do ótimo. Já para as instâncias que possuem '-', significa que o algoritmo exato não conseguiu encontrar a resposta dentro do tempo limite de 1 hora.

Tabela 5.1: Instâncias Utilizadas

| Instância                 | V    | E       | D      | Instância      | V    | E      | D      |
|---------------------------|------|---------|--------|----------------|------|--------|--------|
| <i>Instâncias DIMACS</i>  |      |         |        |                |      |        |        |
| C125-9                    | 125  | 6963    | 0.8985 | hamming6-2     | 64   | 1824   | 0.9048 |
| C250-9                    | 250  | 27984   | 0.8991 | hamming6-4     | 64   | 704    | 0.3492 |
| C500-9                    | 500  | 112332  | 0.9005 | hamming8-2     | 256  | 31616  | 0.9686 |
| C1000-9                   | 1000 | 450079  | 0.9011 | hamming8-4     | 256  | 20864  | 0.6392 |
| C2000-5                   | 2000 | 999836  | 0.5002 | p-hat300-3     | 300  | 33390  | 0.7445 |
| C2000-9                   | 2000 | 1799532 | 0.9002 | p-hat500-2     | 500  | 62946  | 0.5046 |
| C4000-5                   | 4000 | 4000268 | 0.5002 | p-hat500-3     | 500  | 93800  | 0.7519 |
| MANN-a9                   | 45   | 918     | 0.9273 | c-fat500-10    | 500  | 46627  | 0.3738 |
| keller4                   | 171  | 9435    | 0.6491 | hamming10-2    | 1024 | 518656 | 0.9902 |
| keller5                   | 776  | 225990  | 0.7515 | hamming10-4    | 1024 | 434176 | 0.8289 |
| keller6                   | 3361 | 4619898 | 0.8182 | p-hat1000-1    | 1000 | 122253 | 0.2448 |
| MANN-a27                  | 378  | 70551   | 0.9901 | p-hat1000-2    | 1000 | 244799 | 0.4901 |
| MANN-a45                  | 1035 | 533115  | 0.9963 | p-hat1000-3    | 1000 | 1000   | 0.0020 |
| MANN-a81                  | 3321 | 5506380 | 0.9988 | sanr200-0-7    | 200  | 13868  | 0.6969 |
| DSJC500-5                 | 500  | 125248  | 1.0000 | sanr200-0-9    | 200  | 17863  | 0.8976 |
| DSJC1000-5                | 1000 | 499652  | 1.0000 | sanr400-0-5    | 400  | 39984  | 0.5011 |
| brock200-1                | 200  | 14834   | 0.7454 | sanr400-0-7    | 400  | 55869  | 0.7001 |
| brock200-2                | 200  | 9876    | 0.4963 | johnson8-2-4   | 28   | 210    | 0.5556 |
| brock200-3                | 200  | 12048   | 0.6054 | san200-0-7-1   | 200  | 13930  | 0.7000 |
| brock200-4                | 200  | 13089   | 0.6577 | san200-0-7-2   | 200  | 13930  | 0.7000 |
| brock400-1                | 400  | 59723   | 0.7484 | san200-0-9-1   | 200  | 17910  | 0.9000 |
| brock400-2                | 400  | 59786   | 0.7492 | san200-0-9-2   | 200  | 17910  | 0.9000 |
| brock400-3                | 400  | 59681   | 0.7479 | san200-0-9-3   | 200  | 17910  | 0.9000 |
| brock400-4                | 400  | 59765   | 0.7489 | san400-0-5-1   | 400  | 39900  | 0.5000 |
| brock800-1                | 800  | 207505  | 0.6493 | san400-0-7-1   | 400  | 55860  | 0.7000 |
| brock800-2                | 800  | 208166  | 0.6513 | san400-0-7-2   | 400  | 55860  | 0.7000 |
| brock800-3                | 800  | 207333  | 0.6487 | san400-0-7-3   | 400  | 55860  | 0.7000 |
| brock800-4                | 800  | 207643  | 0.6497 | san400-0-9-1   | 400  | 71820  | 0.9000 |
| c-fat200-1                | 200  | 1534    | 0.0771 | johnson16-2-4  | 120  | 5460   | 0.7647 |
| c-fat200-2                | 200  | 3235    | 0.1626 | johnson32-2-4  | 496  | 107880 | 0.8788 |
| c-fat200-5                | 200  | 8473    | 0.4258 | gen200-p0-9-44 | 200  | 17910  | 0.9000 |
| c-fat500-1                | 500  | 4459    | 0.0357 | gen200-p0-9-55 | 200  | 17910  | 0.9000 |
| c-fat500-2                | 500  | 9139    | 0.0733 | gen400-p0-9-55 | 400  | 71820  | 0.9000 |
| c-fat500-5                | 500  | 23191   | 0.1859 | gen400-p0-9-65 | 400  | 71820  | 0.9000 |
| hamming6-2                | 64   | 1824    | 0.9048 | gen400-p0-9-75 | 400  | 71820  | 0.9000 |
| hamming6-4                | 64   | 704     | 0.3492 |                |      |        |        |
| <i>Instâncias BHOSLIB</i> |      |         |        |                |      |        |        |
| frb30-15-1                | 450  | 83198   | 0.8235 | frb45-21-4     | 945  | 387491 | 0.8687 |

*Continua na próxima página*

|            |     |        |        |             |      |         |        |
|------------|-----|--------|--------|-------------|------|---------|--------|
| frb30-15-2 | 450 | 83151  | 0.8231 | frb45-21-5  | 945  | 387461  | 0.8687 |
| frb30-15-3 | 450 | 83216  | 0.8237 | frb50-23-1  | 1150 | 580603  | 0.8788 |
| frb30-15-4 | 450 | 83194  | 0.8235 | frb50-23-2  | 1150 | 579824  | 0.8776 |
| frb30-15-5 | 450 | 83231  | 0.8239 | frb50-23-3  | 1150 | 579607  | 0.8773 |
| frb35-17-1 | 595 | 148859 | 0.8424 | frb50-23-4  | 1150 | 580417  | 0.8785 |
| frb35-17-2 | 595 | 148868 | 0.8424 | frb50-23-5  | 1150 | 580640  | 0.8789 |
| frb35-17-3 | 595 | 148784 | 0.8419 | frb53-24-1  | 1272 | 714129  | 0.8834 |
| frb35-17-4 | 595 | 148873 | 0.8424 | frb53-24-2  | 1272 | 714067  | 0.8834 |
| frb35-17-5 | 595 | 148572 | 0.8407 | frb53-24-3  | 1272 | 714229  | 0.8836 |
| frb40-19-1 | 760 | 247106 | 0.8568 | frb53-24-4  | 1272 | 714048  | 0.8833 |
| frb40-19-2 | 760 | 247157 | 0.8569 | frb53-24-5  | 1272 | 714130  | 0.8834 |
| frb40-19-3 | 760 | 247325 | 0.8575 | frb59-26-1  | 1534 | 1049256 | 0.8924 |
| frb40-19-4 | 760 | 246815 | 0.8557 | frb59-26-2  | 1534 | 1049648 | 0.8927 |
| frb40-19-5 | 760 | 246801 | 0.8557 | frb59-26-3  | 1534 | 1049729 | 0.8928 |
| frb45-21-1 | 945 | 386854 | 0.8673 | frb59-26-4  | 1534 | 1048800 | 0.8920 |
| frb45-21-2 | 945 | 387416 | 0.8686 | frb59-26-5  | 1534 | 1049829 | 0.8929 |
| frb45-21-3 | 945 | 387795 | 0.8694 | frb100-40-1 | 4000 | 7425226 | 0.9284 |
| frb45-21-4 | 945 | 387491 | 0.8687 |             |      |         |        |

Tabela 5.2: Instâncias Konect

| Instância                | V       | E        | D        |
|--------------------------|---------|----------|----------|
| actor-movie              | 854058  | 1470413  | 0.000004 |
| dbpedia-team             | 935627  | 1366466  | 0.000003 |
| bookcrossing_full-rating | 445801  | 1149739  | 0.000012 |
| dbpedia-writer           | 135569  | 144340   | 0.000016 |
| dplb-author              | 5425963 | 8649016  | 0.000001 |
| discogs_affiliation      | 2025594 | 14414659 | 0.000007 |
| dbpedia-genre            | 266717  | 463497   | 0.000013 |
| discogs_lgenre           | 270786  | 4147665  | 0.000113 |
| dbpedia-location         | 225498  | 293697   | 0.000012 |
| discogs_style            | 836545  | 10296105 | 0.000029 |
| dbpedia-producer         | 187677  | 207268   | 0.000012 |
| wiki-en-cat              | 2036440 | 3795796  | 0.000002 |
| dbpedia-recordlabel      | 186758  | 233286   | 0.000013 |
| youtube-groupmemberships | 59081   | 156969   | 0.000090 |
| dbpedia-starring         | 157184  | 281396   | 0.000023 |

Além disso, um grupo de instâncias aleatórias foram criadas utilizando o modelo  $G(n, p)$  — e  $G(n_1, n_2, p)$  para o caso bipartidos —, também conhecido como modelo binomial proposto por [Gilbert \(1959\)](#). Com ele, são construídos grafos  $G(V, E)$  em que  $|U| = n$  (para o caso bipartido  $G(V, U, E)$ , em que  $|V| = n_1$  e  $|U| = n_2$ ) e cada aresta de  $E$  é incluída no grafo com probabilidade  $p$ , independentemente das outras arestas. A quantidade de vértices foi fixada em 1000 (500 vértices em cada parte no caso bipartido) o valor de  $p$  foi variado de forma a perceber onde a redução funciona e onde para de funcionar.

Para cada instância, o peso dos vértices foi gerado da seguinte maneira:  $W(v) = (i + 1) \% 200$ , onde o peso do vértice  $v$  é calculado com base em seu índice  $i$ , somando 1 e aplicando o operador de módulo por 200.

## 5.2 Parâmetros utilizados

Alguns parâmetros foram utilizados na execução do algoritmo, incluindo o valor  $\beta$ . Se houver  $\beta$  iterações consecutivas com o resultado igual ao melhor encontrado *Best*, então a execução atual é encerrada. Outro parâmetro é o *alphaCalibration*, que determina a quantidade de iterações necessárias para calibrar os valores de alfa. O último parâmetro é o *edgeTime*, utilizado no cálculo do tempo que determinará se a redução será ativada ou não. É possível ver alguns desses parâmetros sendo utilizados no Algoritmo 4.1.

O IRACE, proposto por López-Ibáñez et al. (2016), foi empregado para ajustar os parâmetros, sendo fornecidas diferentes configurações para encontrar os melhores valores para cada um deles. Apesar de ter havido uma melhoria inicial ao utilizar o IRACE, não foi percebida uma mudança significativa ao modificar as configurações dos parâmetros em cada execução do IRACE.

## 5.3 Impacto do Reactive GRASP

Nessa subseção iremos ver o impacto do mecanismo "Reactive" nas instâncias DIMACS, BHOSLIB e KONECT para comparar o Best, TTB e as melhores médias entre os dois algoritmos.

A tabela 5.3 apresenta uma visão geral da comparação das instâncias DIMACS com e sem o algoritmo Reactive. Observamos que, em geral, a funcionalidade "Reactive" teve um resultado bem semelhante na qualidade das soluções encontradas, sendo consistentes em ambas as configurações. O algoritmo Reactive foi superior na média em relação ao não reativo, mas foi pior no Best e no TTB em relação ao algoritmo não reativo. Entretanto, a diferença foi muito pequena entre os dois, em certas instâncias um algoritmo é mais rápido que o outro e não deu para tirar muitas conclusões a partir desse teste.

Tabela 5.3: Comparativo do Algoritmo com e sem o Reactive para instâncias DIMACS

| Instância   | Com Reactive |       |         | Sem Reactive |       |         | GAP(%) |
|-------------|--------------|-------|---------|--------------|-------|---------|--------|
|             | Best         | Média | TTB     | Best         | Média | TTB     |        |
| C125-9      | 673          | 673   | 0.0044  | 673          | 673   | 0.0018  | 0.00   |
| hamming6-2  | 242          | 242   | 2.9410  | 242          | 242   | 2.121   | 0.00   |
| C250-9      | 1180         | 1180  | 0.0329  | 1180         | 1180  | 0.0201  | 0.00   |
| hamming6-4  | 455          | 455   | 2.0250  | 455          | 455   | 4.5120  | 0.00   |
| C500-9      | 1373         | 1373  | 0.2436  | 1373         | 1373  | 0.1231  | 0.00   |
| hamming8-2  | 786          | 786   | 1.6682  | 786          | 786   | 0.4816  | 0.00   |
| C1000-9     | 1669         | 1663  | 27.7367 | 1669         | 1657  | 39.6611 | 0.00   |
| hamming8-4  | 4048         | 4048  | 0.4936  | 4048         | 4048  | 0.9866  | 0.00   |
| C2000-5     | 2611         | 2560  | 56.9279 | 2797         | 2558  | 56.7973 | 6.64   |
| hamming10-2 | 798          | 798   | 0.0005  | 798          | 798   | 0.0010  | 0.00   |
| C2000-9     | 1844         | 1819  | 44.8895 | 1844         | 1826  | 39.8362 | 0.00   |
| hamming10-4 | 5684         | 5684  | 0.0000  | 5684         | 5684  | 1.6977  | 0.00   |
| C4000-5     | 2827         | 2759  | 54.3564 | 2915         | 2760  | 56.1606 | 3.01   |

*Continua na próxima página*

|                   |       |       |         |       |       |         |      |
|-------------------|-------|-------|---------|-------|-------|---------|------|
| p-hat300-3        | 1602  | 1602  | 0.0533  | 1602  | 1602  | 0.0294  | 0.00 |
| MANN-a9           | 255   | 255   | 4.0340  | 255   | 255   | 2.2681  | 0.00 |
| p-hat500-2        | 1766  | 1766  | 0.2035  | 1766  | 1766  | 0.3269  | 0.00 |
| MANN-a27          | 1173  | 1173  | 0.0028  | 1173  | 1173  | 0.0122  | 0.00 |
| p-hat500-3        | 1887  | 1887  | 0.7974  | 1887  | 1887  | 0.7926  | 0.00 |
| MANN-a45          | 1191  | 1191  | 0.0429  | 1191  | 1191  | 0.1298  | 0.00 |
| p-hat1000-1       | 2034  | 2033  | 41.4571 | 2034  | 2031  | 45.141  | 0.00 |
| MANN-a81          | 1194  | 1194  | 0.0078  | 1194  | 1194  | 1.2160  | 0.00 |
| p-hat1000-2       | 2140  | 2140  | 31.1509 | 2140  | 2136  | 33.9534 | 0.00 |
| keller4           | 2166  | 2166  | 0.0030  | 2166  | 2166  | 0.0023  | 0.00 |
| p-hat1000-3       | 265   | 265   | 0.3333  | -     | -     | -       | -    |
| keller5           | 5040  | 5026  | 31.7815 | 5040  | 5038  | 25.1542 | 0.00 |
| sanr200-0-7       | 1661  | 1661  | 0.0099  | 1661  | 1661  | 0.0222  | 0.00 |
| keller6           | 10510 | 10455 | 56.9683 | 10508 | 10322 | 56.6317 | 0.01 |
| sanr200-0-9       | 1231  | 1231  | 0.0243  | 1231  | 1231  | 0.0279  | 0.00 |
| DSJC500-5         | 2035  | 2035  | 0.3119  | 2035  | 2035  | 1.1040  | 0.00 |
| sanr400-0-5       | 2015  | 2015  | 2.1741  | 2015  | 2015  | 1.5816  | 0.00 |
| DSJC1000-5        | 2428  | 2401  | 44.4594 | 2428  | 2411  | 39.3505 | 0.00 |
| sanr400-0-7       | 1839  | 1839  | 0.2960  | 1839  | 1839  | 0.3845  | 0.00 |
| brock200-1        | 1542  | 1542  | 0.0500  | 1542  | 1542  | 0.0922  | 0.00 |
| johnson8-2-4      | 129   | 129   | 3.8210  | 129   | 129   | 3.4800  | 0.00 |
| brock200-2        | 1644  | 1644  | 0.0527  | 1644  | 1644  | 0.0574  | 0.00 |
| johnson16-2-4     | 1477  | 1477  | 3.8491  | 1477  | 1477  | 4.7470  | 0.00 |
| brock200-3        | 1681  | 1681  | 0.0221  | 1681  | 1681  | 0.0371  | 0.00 |
| johnson32-2-4     | 5050  | 5050  | 0.0735  | 5050  | 5050  | 0.1064  | 0.00 |
| brock200-4        | 1786  | 1786  | 0.0383  | 1786  | 1786  | 0.0162  | 0.00 |
| san200-0-7-1      | 1911  | 1911  | 0.0170  | 1911  | 1911  | 0.0181  | 0.00 |
| brock400-1        | 1788  | 1788  | 0.9813  | 1788  | 1788  | 0.8450  | 0.00 |
| san200-0-7-2      | 2764  | 2764  | 0.0051  | 2764  | 2764  | 0.0016  | 0.00 |
| brock400-2        | 1821  | 1821  | 0.1483  | 1821  | 1821  | 0.0732  | 0.00 |
| san200-0-9-1      | 1155  | 1155  | 0.0688  | 1155  | 1155  | 0.0733  | 0.00 |
| brock400-3        | 1952  | 1952  | 0.1470  | 1952  | 1952  | 0.1104  | 0.00 |
| san200-0-9-2      | 1362  | 1362  | 0.4744  | 1362  | 1362  | 1.5464  | 0.00 |
| brock400-4        | 1813  | 1813  | 1.5732  | 1813  | 1813  | 1.8549  | 0.00 |
| san200-0-9-3      | 1279  | 1279  | 0.0099  | 1279  | 1279  | 0.0120  | 0.00 |
| brock800-1        | 2357  | 2357  | 10.1452 | 2357  | 2357  | 9.0302  | 0.00 |
| san400-0-5-1      | 7274  | 7274  | 0.0013  | 7274  | 7274  | 0.0024  | 0.00 |
| brock800-2        | 2306  | 2306  | 10.8605 | 2306  | 2306  | 10.7538 | 0.00 |
| san400-0-7-1      | 2666  | 2666  | 0.0320  | 2666  | 2666  | 0.0142  | 0.00 |
| brock800-3        | 2282  | 2276  | 31.5915 | 2282  | 2282  | 25.1707 | 0.00 |
| san400-0-7-2      | 3520  | 3520  | 0.8599  | 3520  | 3520  | 0.0088  | 0.00 |
| brock800-4        | 2396  | 2393  | 20.4079 | 2396  | 2386  | 35.7791 | 0.00 |
| san400-0-7-3      | 4188  | 4188  | 0.0115  | 4188  | 4188  | 0.0091  | 0.00 |
| c-fat200-1        | 399   | 399   | 2.9551  | 399   | 399   | 0.7872  | 0.00 |
| san400-0-9-1      | 1427  | 1427  | 0.1427  | 1427  | 1427  | 0.0818  | 0.00 |
| c-fat200-2        | 399   | 399   | 1.4812  | 399   | 399   | 0.6432  | 0.00 |
| gen200-p0-9-44    | 1388  | 1388  | 0.0728  | 1388  | 1388  | 0.0321  | 0.00 |
| c-fat200-5        | 399   | 399   | 2.4761  | 399   | 399   | 2.4282  | 0.00 |
| gen200-p0-9-55    | 1278  | 1278  | 0.0109  | 1278  | 1278  | 0.9329  | 0.00 |
| c-fat500-1        | 399   | 399   | 1.2892  | 399   | 399   | 1.8304  | 0.00 |
| gen400-p0-9-55    | 2040  | 2040  | 0.0699  | 2040  | 2040  | 0.0292  | 0.00 |
| c-fat500-2        | 400   | 400   | 1.0793  | 400   | 400   | 0.6153  | 0.00 |
| gen400-p0-9-65    | 1811  | 1811  | 0.0206  | 1811  | 1811  | 0.0316  | 0.00 |
| c-fat500-5        | 399   | 399   | 0.6223  | 399   | 399   | 0.0003  | 0.00 |
| gen400-p0-9-75    | 1620  | 1620  | 0.2368  | 1620  | 1620  | 0.1658  | 0.00 |
| # Best            | 65    |       |         | 66    |       |         |      |
| # Melhores Médias |       | 62    |         |       | 61    |         |      |

Continua na próxima página

# Melhores TTB

31

36

A tabela 5.4 apresenta uma visão geral da comparação das instâncias BHOSLIB com e sem o algoritmo Reactive. Observamos que, em geral, a funcionalidade "Reactive" não teve um impacto significativo na qualidade das melhores soluções encontradas, sendo consistentes em ambas as configurações. No entanto, notamos variações no tempo para atingir a melhor solução (TTB). Em 14 instâncias a mais, o algoritmo com o Reactive foi mais rápido.

Tabela 5.4: Comparativo do Algoritmo com e sem o Reactive para instâncias BHOSLIB

| Instância         | Com Reactive |      |        | Sem Reactive |      |        | Instância | Com Reactive |       |       | Sem Reactive |       |       | GAP(%) |      |
|-------------------|--------------|------|--------|--------------|------|--------|-----------|--------------|-------|-------|--------------|-------|-------|--------|------|
|                   | Média        | Best | TTB    | Média        | Best | TTB    |           | Média        | Best  | TTB   | Média        | Best  | TTB   |        |      |
| frb30-15-1        | 5340         | 5340 | 0.0014 | 5340         | 5340 | 0.0010 | 0.00      | frb45-21-4   | 7728  | 7728  | 0.0033       | 7728  | 7728  | 0.0019 | 0.00 |
| frb30-15-2        | 5565         | 5565 | 0.9278 | 5565         | 5565 | 0.4351 | 0.00      | frb45-21-5   | 7707  | 7707  | 0.0020       | 7707  | 7707  | 0.0074 | 0.00 |
| frb30-15-3        | 5565         | 5565 | 0.0010 | 5565         | 5565 | 0.0009 | 0.00      | frb50-23-1   | 8441  | 8441  | 0.0141       | 8441  | 8441  | 0.0215 | 0.00 |
| frb30-15-4        | 5415         | 5415 | 0.0009 | 5415         | 5415 | 0.0010 | 0.00      | frb50-23-2   | 8441  | 8441  | 0.0049       | 8441  | 8441  | 0.0094 | 0.00 |
| frb30-15-5        | 5415         | 5415 | 0.0021 | 5415         | 5415 | 0.0021 | 0.00      | frb50-23-3   | 8234  | 8234  | 0.0070       | 8234  | 8234  | 0.0070 | 0.00 |
| frb35-17-1        | 6222         | 6222 | 0.0030 | 6222         | 6222 | 0.0018 | 0.00      | frb50-23-4   | 8441  | 8441  | 0.0030       | 8441  | 8441  | 0.0054 | 0.00 |
| frb35-17-2        | 6290         | 6290 | 0.0015 | 6290         | 6290 | 0.7600 | 0.00      | frb50-23-5   | 8441  | 8441  | 0.0046       | 8441  | 8441  | 0.0049 | 0.00 |
| frb35-17-3        | 6222         | 6222 | 0.0035 | 6222         | 6222 | 0.0091 | 0.00      | frb53-24-1   | 9048  | 9048  | 0.0045       | 9048  | 9048  | 0.0076 | 0.00 |
| frb35-17-4        | 6069         | 6069 | 0.0031 | 6069         | 6069 | 0.0018 | 0.00      | frb53-24-2   | 8856  | 8856  | 0.0062       | 8856  | 8856  | 0.0041 | 0.00 |
| frb35-17-5        | 6222         | 6222 | 0.0018 | 6222         | 6222 | 0.0030 | 0.00      | frb53-24-3   | 8856  | 8856  | 0.0024       | 8856  | 8856  | 0.0027 | 0.00 |
| frb40-19-1        | 6859         | 6859 | 0.0020 | 6859         | 6859 | 0.0025 | 0.00      | frb53-24-4   | 9048  | 9048  | 0.0078       | 9048  | 9048  | 0.0046 | 0.00 |
| frb40-19-2        | 7049         | 7049 | 0.0024 | 7049         | 7049 | 0.0037 | 0.00      | frb53-24-5   | 9048  | 9048  | 0.0048       | 9048  | 9048  | 0.0105 | 0.00 |
| frb40-19-3        | 7049         | 7049 | 0.0020 | 7049         | 7049 | 0.0054 | 0.00      | frb59-26-1   | 9594  | 9594  | 0.0113       | 9594  | 9594  | 0.0090 | 0.00 |
| frb40-19-4        | 7049         | 7049 | 0.0019 | 7049         | 7049 | 0.0028 | 0.00      | frb59-26-2   | 9594  | 9594  | 0.0124       | 9594  | 9594  | 0.0245 | 0.00 |
| frb40-19-5        | 7030         | 7030 | 0.0012 | 7030         | 7030 | 0.0034 | 0.00      | frb59-26-3   | 9438  | 9438  | 0.0254       | 9438  | 9438  | 0.0426 | 0.00 |
| frb45-21-1        | 7707         | 7707 | 0.0028 | 7707         | 7707 | 0.0045 | 0.00      | frb59-26-4   | 9594  | 9594  | 0.0144       | 9594  | 9594  | 0.0203 | 0.00 |
| frb45-21-2        | 7917         | 7917 | 0.0018 | 7917         | 7917 | 0.0073 | 0.00      | frb59-26-5   | 9594  | 9594  | 0.0099       | 9594  | 9594  | 0.0255 | 0.00 |
| frb45-21-3        | 7917         | 7917 | 0.0038 | 7917         | 7917 | 0.0095 | 0.00      | frb100-40-1  | 14440 | 14440 | 0.0025       | 14440 | 14440 | 0.0020 | 0.00 |
| # Best            | 36           |      |        | 36           |      |        |           |              |       |       |              |       |       |        |      |
| # Melhores Médias | 36           |      |        | 36           |      |        |           |              |       |       |              |       |       |        |      |
| # Melhores TTB    | 26           |      |        | 12           |      |        |           |              |       |       |              |       |       |        |      |

A tabela 5.5 mostra a comparação do algoritmo Reactive e sem Reactive nas instâncias KONECT. Em apenas uma instância o algoritmo sem o Reactive alcança o Best do Reactive, em todas as outras instâncias ele fica abaixo. Ele também perde em todas as instâncias em relação ao TTB e sobre as melhores médias.

Tabela 5.5: Comparativo do Algoritmo com e sem o Reactive para instâncias KONECT

| Instância                | Com Reactive |      |       | Sem Reactive |      |       | GAP    |
|--------------------------|--------------|------|-------|--------------|------|-------|--------|
|                          | Média        | Best | TTB   | Média        | Best | TTB   |        |
| actor-movie              | 1906         | 1916 | 29.98 | 640          | 1862 | 60.00 | 2.81%  |
| bookcrossing_full-rating | 2912         | 3237 | 59.99 | 271          | 271  | 60.00 | 91.62% |
| dblp-author              | 1374         | 1891 | 58.95 | 322          | 322  | 60.00 | 82.97% |
| dbpedia-genre            | 1324         | 1411 | 48.71 | 843          | 1146 | 60.00 | 18.78% |

*Continua na próxima página*

|                          |      |      |       |      |      |       |        |
|--------------------------|------|------|-------|------|------|-------|--------|
| dbpedia-location         | 1203 | 1211 | 16.84 | 676  | 1211 | 58.37 | 0.00%  |
| dbpedia-producer         | 1345 | 1345 | 0.52  | 45   | 45   | 60.00 | 96.65% |
| dbpedia-recordlabel      | 1312 | 1392 | 57.17 | 401  | 1310 | 60.00 | 5.89%  |
| dbpedia-starring         | 1710 | 1813 | 58.04 | 137  | 137  | 60.00 | 92.44% |
| dbpedia-team             | 1263 | 1444 | 59.79 | 735  | 1169 | 60.00 | 19.04% |
| dbpedia-writer           | 1611 | 1621 | 30.09 | 327  | 327  | 60.00 | 79.82% |
| discogs_afiliation       | 5196 | 5762 | 58.03 | 3822 | 5574 | 60.00 | 3.26%  |
| discogs_lgenre           | 2054 | 2616 | 59.60 | 1402 | 1680 | 60.00 | 35.77% |
| discogs_style            | 5486 | 5701 | 59.93 | 4011 | 4736 | 60.00 | 16.92% |
| wiki-en-cat              | 1872 | 2628 | 59.76 | 739  | 1627 | 60.00 | 38.08% |
| youtube-groupmemberships | 2743 | 2747 | 28.63 | 1209 | 2553 | 60.00 | 7.06%  |
| # Best                   |      | 15   |       |      | 1    |       |        |
| # Melhores Médias        | 15   |      |       | 0    |      |       |        |
| # Melhores TTB           |      |      | 15    |      |      | 0     |        |

## 5.4 Impacto das Reduções

Nessa subseção iremos ver o impacto do algoritmo de redução em instâncias geradas  $\mathbb{G}(n, p)$ , KONECT e as instâncias KONECT reduzidas que foram utilizadas para rodar o CPLEX e encontrar o ótimo em um conjunto menor de vértices e arestas. Instâncias  $\mathbb{G}(n, p)$  são geradas a partir de uma probabilidade  $p$  ter uma aresta entre dois vértices, com um total de 1000 vértices no grafo tanto para as instâncias gerais quanto bipartidas.

As Figuras 5.1 e 5.2 mostram o impacto da redução para instâncias não-bipartidas e bipartidas, respectivamente. Pode-se observar que para um grande intervalo de  $p$ , a redução conseguiu remover quase que 100% dos vértices, o que mostra que a solução encontrada pelo algoritmo é bastante próxima do ótimo. Também revela que, quanto mais esparso for o grafo (menor probabilidade de  $p$ ), maior é a chance da redução ir bem.

A Tabela 5.6 é o comparativo dos resultados do algoritmo desenvolvido para as instâncias KONECT com e sem a redução ativada. Lembrando que, por conta do mecanismo de auto-regulação pro algoritmo de redução, não serão todas as instâncias que irão rodar a redução no algoritmo com redução. A tabela foi dividida em 2 algoritmos — com auto-redução e sem redução, onde cada instância foi executada 30 vezes com tempo limite de 60 segundos, diferentemente dos outros testes onde cada instância teve 10 execuções.

No grupo das instâncias que rodaram a versão com auto-redução, possuímos 4 colunas: a média do melhor resultado das 30 execuções, o melhor resultado entre todas as execuções, a quantidade de vértices removidos em porcentagem (VR) e a quantidade de arestas removidas em porcentagem (AR). Já nas que rodaram sem redução, temos apenas a média e o melhor resultado. Por fim, temos o GAP que representa a diferença (em porcentagem) da coluna Best dos dois grupos.

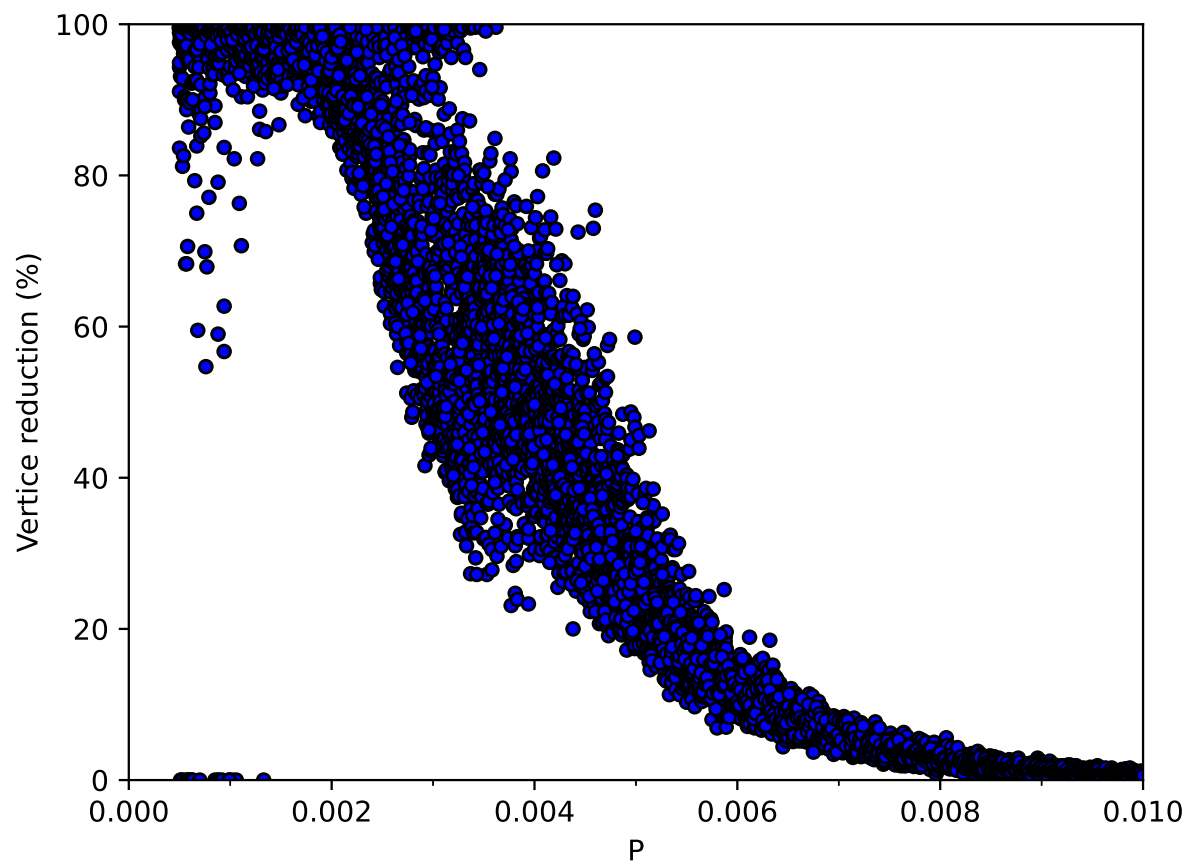


Figura 5.1: Gráfico que mostra a curva de redução de vértices nas instâncias  $\mathbb{G}(n, p)$  (não-bipartidas).

Tabela 5.6: Comparativo do Algoritmo com redução e sem redução

| Instância                | Com redução |      |       |       | Sem redução |      |       |
|--------------------------|-------------|------|-------|-------|-------------|------|-------|
|                          | Média       | Best | VR    | AR    | Média       | Best | GAP   |
| actor-movie              | 1906        | 1916 | 88.18 | 48.75 | 1406        | 1743 | 9.02% |
| bookcrossing_full-rating | 2912        | 3237 | 95.46 | 67.91 | 2804        | 3051 | 5.74% |
| dblp-author              | 1374        | 1891 | 0.00  | 0.00  | 1374        | 1891 | 0.00% |
| dbpedia-genre            | 1324        | 1411 | 9.07  | 9.14  | 1371        | 1411 | 0.00% |
| dbpedia-location         | 1203        | 1211 | 99.36 | 98.84 | 1180        | 1211 | 0.00% |
| dbpedia-producer         | 1345        | 1345 | 98.11 | 93.78 | 1331        | 1345 | 0.00% |
| dbpedia-recordlabel      | 1312        | 1392 | 98.73 | 96.33 | 1296        | 1310 | 5.89% |
| dbpedia-starring         | 1710        | 1813 | 94.75 | 90.71 | 1694        | 1806 | 0.38% |
| dbpedia-team             | 1263        | 1444 | 93.28 | 72.02 | 1213        | 1444 | 0.00% |
| dbpedia-writer           | 1611        | 1621 | 99.78 | 99.70 | 1491        | 1602 | 1.17% |
| discogs_affiliation      | 5196        | 5762 | 0.00  | 0.00  | 5196        | 5762 | 0.00% |
| discogs_lgenre           | 2054        | 2616 | 0.00  | 0.00  | 2054        | 2616 | 0.00% |
| discogs_style            | 5486        | 5701 | 0.00  | 0.00  | 5486        | 5701 | 0.00% |
| wiki-en-cat              | 1872        | 2628 | 0.00  | 0.00  | 1872        | 2628 | 0.00% |
| youtube-groupmemberships | 2743        | 2747 | 92.91 | 71.40 | 2596        | 2747 | 0.00% |
| # Best                   |             | 15   |       |       | 10          |      |       |
| # Melhores Médias        | 14          |      |       | 6     |             |      |       |

Ao analisar os resultados da Tabela 5.6, é possível perceber que o algoritmo com auto-

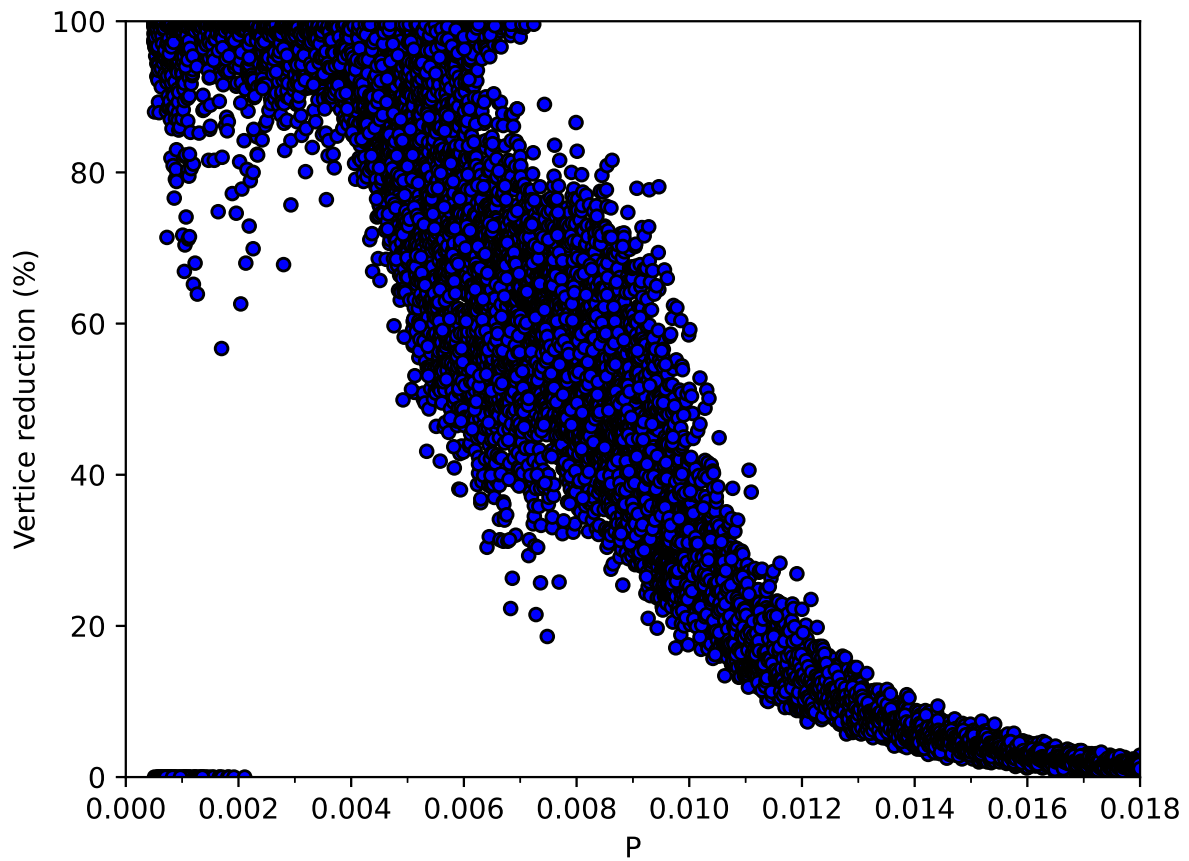


Figura 5.2: Gráfico que mostra a curva de redução de vértices nas instâncias  $\mathbb{G}(n_1, n_2, p)$  (bipartidas).

redução, quando não empata ele supera a versão sem redução ao observar o GAP. Em algumas instâncias, o resultado é idêntico porque o algoritmo de redução não atinge as condições necessárias para executar a redução (tem um certo limite para a quantidade de vértices e para a densidade). Mesmo assim, na maioria das instâncias em que é executado o algoritmo de redução, podemos perceber excelentes resultados com a remoção de mais de 90% dos vértices.

Além disso, um resultado muito interessante é a quantidade de vértices e arestas removidos que fazem com que se chegue mais rapidamente ao ótimo global e também permite reduzir o grafo para que, a partir da execução em um algoritmo heurístico, extraia o grafo reduzido e rode em um algoritmo exato para confirmar ou encontrar o ótimo global. Por esses motivos é destacado o algoritmo desenvolvido para a redução de grafo.

## 5.5 Resultados Finais

Ao final dos testes, utilizamos a versão do GRASP<sub>VND</sub> com o mecanismo de Reactive e o algoritmo de redução, junto das regras para ativar ou desativar a redução de grafo. Serão apresentadas as comparações dessa versão da meta-heurística proposta contra o CPLEX nas

instâncias gerais (DIMACS e BHOSLIB) e nas bipartidas (KONECT).

### 5.5.1 Resultados para entradas gerais

A Tabela 5.7 representa os resultados obtidos com os grupos de instâncias DIMACS e BHOSLIB.

Nas instâncias DIMACS, das 69 instâncias testadas, 45 obtiveram um GAP de 0%, chegando no resultado ótimo. Já com as instâncias BHOSLIB, das 36 instâncias testadas, 35 obtiveram um GAP de 0%. Para as demais instâncias, em ambas as tabelas, o CPLEX não encontrou o ótimo dentro do tempo limite. Em todas as instâncias com ótimo conhecido, o GRASP<sub>VND</sub> foi capaz de encontrar a solução ótima com um tempo muito abaixo do algoritmo exato.

Além disso, ele encontra a solução em um tempo mais rápido em 75 das 80 instâncias em que o algoritmo heurístico encontra o ótimo.

Tabela 5.7: Resultados do GRASP<sub>VND</sub> para as instâncias do DIMACS e BHOSLIB.

| Instância                | GRASP <sub>VND</sub> |       |         |         | CPLEX |           |
|--------------------------|----------------------|-------|---------|---------|-------|-----------|
|                          | Best                 | Média | TTB (s) | Gap (%) | Ótimo | Tempo (s) |
| <i>Instâncias DIMACS</i> |                      |       |         |         |       |           |
| C125-9                   | 673                  | 673   | 0.0044  | 0.00    | 673   | 4.82      |
| hamming6-2               | 242                  | 242   | 2.9410  | 0.00    | 242   | 0.06      |
| C250-9                   | 1180                 | 1180  | 0.0328  | 0.00    | 1180  | 10.83     |
| hamming6-4               | 455                  | 455   | 2.0250  | 0.00    | 455   | 2.13      |
| C500-9                   | 1373                 | 1373  | 0.2435  | 0.00    | 1373  | 606.53    |
| hamming8-2               | 786                  | 786   | 1.6682  | 0.00    | 786   | 0.87      |
| C1000-9                  | 1669                 | 1663  | 27.7367 | -       | -     | -         |
| hamming8-4               | 4048                 | 4048  | 0.4936  | 0.00    | 4048  | 285.47    |
| C2000-5                  | 2611                 | 2560  | 56.9279 | -       | -     | -         |
| hamming10-2              | 798                  | 798   | 0.0004  | 0.00    | 798   | 27.64     |
| C2000-9                  | 1844                 | 1819  | 44.8895 | -       | -     | -         |
| hamming10-4              | 5684                 | 5684  | 1.3522  | -       | -     | -         |
| C4000-5                  | 2827                 | 2759  | 54.3564 | -       | -     | -         |
| p-hat300-3               | 1602                 | 1602  | 0.0533  | 0.00    | 1602  | 2220.44   |
| MANN-a9                  | 255                  | 255   | 4.034   | 0.00    | 255   | 3.89      |
| p-hat500-2               | 1766                 | 1766  | 0.2034  | -       | -     | -         |
| MANN-a27                 | 1173                 | 1173  | 0.0027  | 0.00    | 1173  | 1.52      |
| p-hat500-3               | 1887                 | 1887  | 0.7973  | -       | -     | -         |
| MANN-a45                 | 1191                 | 1191  | 0.0429  | 0.00    | 1191  | 30.42     |
| p-hat1000-1              | 2034                 | 2033  | 41.4571 | -       | -     | -         |
| MANN-a81                 | 1194                 | 1194  | 0.0078  | 0.00    | 1194  | 369.21    |
| p-hat1000-2              | 2140                 | 2140  | 31.1509 | -       | -     | -         |
| keller4                  | 2166                 | 2166  | 0.0029  | 0.00    | 2166  | 221.53    |
| p-hat1000-3              | 265                  | 265   | 0.3333  | 0.00    | 265   | 65.79     |
| keller5                  | 5040                 | 5026  | 31.7815 | -       | -     | -         |
| sanr200-0-7              | 1661                 | 1661  | 0.0098  | 0.00    | 1661  | 1106.80   |
| keller6                  | 10510                | 10455 | 56.9683 | -       | -     | -         |
| sanr200-0-9              | 1231                 | 1231  | 0.0242  | 0.00    | 1231  | 80.64     |

*Continua na próxima página*

|                |      |      |         |      |      |         |
|----------------|------|------|---------|------|------|---------|
| DSJC500-5      | 2035 | 2035 | 0.3119  | -    | -    | -       |
| sanr400-0-5    | 2015 | 2015 | 2.1741  | -    | -    | -       |
| DSJC1000-5     | 2428 | 2401 | 44.4594 | -    | -    | -       |
| sanr400-0-7    | 1839 | 1839 | 0.2960  | -    | -    | -       |
| brock200-1     | 1542 | 1542 | 0.0500  | 0.00 | 1542 | 688.59  |
| johnson8-2-4   | 129  | 129  | 3.821   | 0.00 | 129  | 0.27    |
| brock200-2     | 1644 | 1644 | 0.0526  | 0.00 | 1644 | 3037.92 |
| johnson16-2-4  | 1477 | 1477 | 3.8490  | 0.00 | 1477 | 104.88  |
| brock200-3     | 1681 | 1681 | 0.0220  | 0.00 | 1681 | 1674.18 |
| johnson32-2-4  | 5050 | 5050 | 0.0735  | -    | -    | -       |
| brock200-4     | 1786 | 1786 | 0.0382  | 0.00 | 1786 | 1522.89 |
| san200-0-7-1   | 1911 | 1911 | 0.0169  | 0.00 | 1911 | 192.06  |
| brock400-1     | 1788 | 1788 | 0.9812  | -    | -    | -       |
| san200-0-7-2   | 2764 | 2764 | 0.0050  | 0.00 | 2764 | 308.64  |
| brock400-2     | 1821 | 1821 | 0.1483  | -    | -    | -       |
| san200-0-9-1   | 1155 | 1155 | 0.0688  | 0.00 | 1155 | 94.62   |
| brock400-3     | 1952 | 1952 | 0.1470  | -    | -    | -       |
| san200-0-9-2   | 1362 | 1362 | 0.4743  | 0.00 | 1362 | 30.71   |
| brock400-4     | 1813 | 1813 | 1.5731  | -    | -    | -       |
| san200-0-9-3   | 1279 | 1279 | 0.0098  | 0.00 | 1279 | 77.41   |
| brock800-1     | 2357 | 2357 | 10.1452 | -    | -    | -       |
| san400-0-5-1   | 7274 | 7274 | 0.0013  | 0.00 | 7274 | 20.99   |
| brock800-2     | 2306 | 2306 | 10.8605 | -    | -    | -       |
| san400-0-7-1   | 2666 | 2666 | 0.0320  | 0.00 | 2666 | 1165.40 |
| brock800-3     | 2282 | 2276 | 31.5915 | -    | -    | -       |
| san400-0-7-2   | 3520 | 3520 | 0.8598  | 0.00 | 3520 | 1266.08 |
| brock800-4     | 2396 | 2393 | 20.4079 | -    | -    | -       |
| san400-0-7-3   | 4188 | 4188 | 0.0114  | 0.00 | 4188 | 1828.57 |
| c-fat200-1     | 399  | 399  | 2.9550  | 0.00 | 399  | 3.20    |
| san400-0-9-1   | 1427 | 1427 | 0.0397  | 0.00 | 1427 | 83.60   |
| c-fat200-2     | 399  | 399  | 1.4811  | 0.00 | 399  | 3.36    |
| gen200-p0-9-44 | 1388 | 1388 | 0.0728  | 0.00 | 1388 | 117.18  |
| c-fat200-5     | 399  | 399  | 2.4761  | 0.00 | 399  | 0.70    |
| gen200-p0-9-55 | 1278 | 1278 | 0.0108  | 0.00 | 1278 | 55.71   |
| c-fat500-1     | 399  | 399  | 1.2891  | 0.00 | 399  | 23.44   |
| gen400-p0-9-55 | 2040 | 2040 | 0.0699  | 0.00 | 2040 | 124.25  |
| c-fat500-2     | 400  | 400  | 1.0792  | 0.00 | 400  | 6.17    |
| gen400-p0-9-65 | 1811 | 1811 | 0.0206  | 0.00 | 1811 | 167.69  |
| c-fat500-5     | 399  | 399  | 0.6222  | 0.00 | 399  | 7.53    |
| gen400-p0-9-75 | 1620 | 1620 | 0.2368  | 0.00 | 1620 | 177.69  |
| c-fat500-10    | 400  | 400  | 0.8083  | 0.00 | 400  | 1.42    |

*Instâncias BHOSLIB*

|            |      |      |        |      |      |        |
|------------|------|------|--------|------|------|--------|
| frb30-15-1 | 5340 | 5340 | 0.0014 | 0.00 | 5340 | 47.76  |
| frb45-21-4 | 7728 | 7728 | 0.0033 | 0.00 | 7728 | 291.71 |
| frb30-15-2 | 5565 | 5565 | 0.9278 | 0.00 | 5565 | 32.39  |
| frb45-21-5 | 7707 | 7707 | 0.0020 | 0.00 | 7707 | 222.76 |
| frb30-15-3 | 5565 | 5565 | 0.0010 | 0.00 | 5565 | 21.98  |
| frb50-23-1 | 8441 | 8441 | 0.0141 | 0.00 | 8441 | 378.73 |
| frb30-15-4 | 5415 | 5415 | 0.0009 | 0.00 | 5415 | 31.19  |
| frb50-23-2 | 8441 | 8441 | 0.0049 | 0.00 | 8441 | 429.18 |
| frb30-15-5 | 5415 | 5415 | 0.0021 | 0.00 | 5415 | 33.13  |
| frb50-23-3 | 8234 | 8234 | 0.0070 | 0.00 | 8234 | 617.89 |
| frb35-17-1 | 6222 | 6222 | 0.0030 | 0.00 | 6222 | 56.15  |
| frb50-23-4 | 8441 | 8441 | 0.0030 | 0.00 | 8441 | 338.90 |
| frb35-17-2 | 6290 | 6290 | 0.0015 | 0.00 | 6290 | 46.69  |
| frb50-23-5 | 8441 | 8441 | 0.0046 | 0.00 | 8441 | 461.89 |

*Continua na próxima página*

|                     |       |       |        |      |      |         |
|---------------------|-------|-------|--------|------|------|---------|
| frb35-17-3          | 6222  | 6222  | 0.0035 | 0.00 | 6222 | 57.59   |
| frb53-24-1          | 9048  | 9048  | 0.0045 | 0.00 | 9048 | 475.86  |
| frb35-17-4          | 6069  | 6069  | 0.0031 | 0.00 | 6069 | 88.53   |
| frb53-24-2          | 8856  | 8856  | 0.0062 | 0.00 | 8856 | 682.81  |
| frb35-17-5          | 6222  | 6222  | 0.0018 | 0.00 | 6222 | 67.86   |
| frb53-24-3          | 8856  | 8856  | 0.0024 | 0.00 | 8856 | 912.74  |
| frb40-19-1          | 6859  | 6859  | 0.0020 | 0.00 | 6859 | 190.16  |
| frb53-24-4          | 9048  | 9048  | 0.0078 | 0.00 | 9048 | 582.74  |
| frb40-19-2          | 7049  | 7049  | 0.0024 | 0.00 | 7049 | 105.66  |
| frb53-24-5          | 9048  | 9048  | 0.0048 | 0.00 | 9048 | 443.78  |
| frb40-19-3          | 7049  | 7049  | 0.0020 | 0.00 | 7049 | 107.76  |
| frb59-26-1          | 9594  | 9594  | 0.0113 | 0.00 | 9594 | 998.25  |
| frb40-19-4          | 7049  | 7049  | 0.0019 | 0.00 | 7049 | 97.27   |
| frb59-26-2          | 9594  | 9594  | 0.0124 | 0.00 | 9594 | 1007.03 |
| frb40-19-5          | 7030  | 7030  | 0.0012 | 0.00 | 7030 | 112.97  |
| frb59-26-3          | 9438  | 9438  | 0.0254 | 0.00 | 9438 | 2552.86 |
| frb45-21-1          | 7707  | 7707  | 0.0028 | 0.00 | 7707 | 342.83  |
| frb59-26-4          | 9594  | 9594  | 0.0144 | 0.00 | 9594 | 1009.55 |
| frb45-21-2          | 7917  | 7917  | 0.0018 | 0.00 | 7917 | 167.40  |
| frb59-26-5          | 9594  | 9594  | 0.0099 | 0.00 | 9594 | 1172.09 |
| frb45-21-3          | 7917  | 7917  | 0.0038 | 0.00 | 7917 | 165.84  |
| frb100-40-1         | 14440 | 14440 | 0.0025 | -    | -    | -       |
| # Encontrou o ótimo |       |       |        | 80   |      |         |
| # TTB               |       |       | 75     |      |      | 5       |

### 5.5.2 Resultado para entradas bipartidas

A Tabela 5.8 representa os resultados obtidos com o grupo de instâncias KONECT e o exato CPLEX, onde o algoritmo heurístico teve 30 execuções para cada instância. Vemos que em apenas uma instância o algoritmo exato conseguiu encontrar o ótimo, mesmo com as instâncias reduzidas geradas após o GRASP<sub>VND</sub>. Isso demonstra a dificuldade de encontrar uma solução nessas instâncias, o que a meta-heurística proposta foi capaz de encontrar, além da eficácia da redução em que foi possível reduzir uma instância a ponto de conseguir encontrar o seu ótimo no CPLEX.

Tabela 5.8: Comparativo do Algoritmo com o Reactive e com o CPLEX para instâncias KONECT

| Instância                | Com Reactive |      |       | CPLEX |           |       |
|--------------------------|--------------|------|-------|-------|-----------|-------|
|                          | Média        | Best | TTB   | Ótimo | Tempo (s) | GAP   |
| actor-movie              | 1906         | 1916 | 29.98 | -     | -         | -     |
| bookcrossing_full-rating | 2912         | 3237 | 59.99 | -     | -         | -     |
| dblp-author              | 1374         | 1891 | 58.95 | -     | -         | -     |
| dbpedia-genre            | 1324         | 1411 | 48.71 | -     | -         | -     |
| dbpedia-location         | 1203         | 1211 | 16.84 | -     | -         | -     |
| dbpedia-producer         | 1345         | 1345 | 0.52  | -     | -         | -     |
| dbpedia-recordlabel      | 1312         | 1392 | 57.17 | -     | -         | -     |
| dbpedia-starring         | 1710         | 1813 | 58.04 | -     | -         | -     |
| dbpedia-team             | 1263         | 1444 | 59.79 | -     | -         | -     |
| dbpedia-writer           | 1611         | 1621 | 30.09 | 1621  | 3.98      | 0.00% |
| discogs_afiliation       | 5196         | 5762 | 58.03 | -     | -         | -     |

*Continua na próxima página*

|                          |      |      |       |   |   |   |
|--------------------------|------|------|-------|---|---|---|
| discogs_lgenre           | 2054 | 2616 | 59.60 | - | - | - |
| discogs_style            | 5486 | 5701 | 59.93 | - | - | - |
| wiki-en-cat              | 1872 | 2628 | 59.76 | - | - | - |
| youtube-groupmemberships | 2743 | 2747 | 28.63 | - | - | - |
| # Chegou no ótimo        |      | 1    |       |   |   |   |

# 6

## Considerações Finais

Neste trabalho foi desenvolvido uma meta-heurística Reactive GRASP com o VND na busca local para tratar a PMBBPV, adicionado a um algoritmo de redução de grafo. Foram realizados estudos bibliográficos, mas não foram encontrados artigos relacionados ao mesmo tipo de problema. Além disso, formas eficientes de implementá-la foram estudadas, como o desenvolvimento de um GRASP com o mecanismo Reactive, uma técnica de redução e um VND com estruturas de vizinhança especializadas.

Por fim, foram feitos inúmeros testes que mostraram os diferentes comportamentos do algoritmo ao longo da sua evolução e com diferentes conjuntos de instâncias.

Começamos mostrando a evolução do algoritmo, primeiro com a comparação do GRASP Reativo e não reativo, onde tivemos melhores TTB nas instâncias BHOSLIB, uma enorme diferença na qualidade e na velocidade da solução para as instâncias KONECT onde o algoritmo reativo foi melhor em todas as instâncias, mas não conseguimos tirar tantas conclusões das instâncias DIMACS.

Após isso, tivemos a comparação das reduções onde é mostrado o comportamento do algoritmo de redução em instâncias GNP tanto para entradas gerais e bipartidas. Ficou claro que o mecanismo de redução funciona bem em instâncias esparsas e, por isso, utilizamos esse algoritmo nas instâncias KONECT. Tivemos uma boa redução de vértices e arestas na maioria das instâncias, acima de 90% em várias delas, além de gerarmos instâncias reduzidas baseado nos vértices e arestas não removidos. Isso possibilitou a utilização de um algoritmo exato CPLEX em instâncias grandes.

Finalmente, após a evolução da heurística, o algoritmo  $\text{GRASP}_{\text{VND}}$  foi comparado com o CPLEX, onde nas instâncias DIMACS, das 69 instâncias testadas, 65% delas obtiveram um GAP de 0% e nas instâncias BHOSLIB, das 36 instâncias testadas, apenas uma não obteve um GAP de 0%. Já para as demais instâncias, em ambas as tabelas, o CPLEX não encontrou o ótimo dentro do tempo limite. Além disso, das 80 instâncias que a heurística encontrou o ótimo, ela foi mais rápida que o CPLEX em 75 dessas instâncias.

O último resultado mostra a comparação das instâncias KONECT com a heurística e o resultado dessas instâncias reduzidas no CPLEX, onde conseguimos encontrar o resultado de uma instância e confirmamos o seu ótimo. Isso mostra que é bastante difícil encontrar uma solução nessas instâncias bem grandes, mas que a meta-heurística proposta foi capaz de encontrar. Também foi comprovado que o algoritmo de redução é de grande utilidade, pois foi possível reduzir uma instância a ponto de conseguir encontrar o seu ótimo no CPLEX.

Como trabalhos futuros, serão estudadas novas meta-heurísticas, estruturas de vizinhança e formas de acelerar a busca local.

## References

- Feige, U. & Kogan, S. (2004), Hardness of approximation of the balanced complete bipartite subgraph problem, Technical report.
- Feo, T. & Resende, M. (1995), 'Greedy randomized adaptive search procedures', *Journal of Global Optimization* **6**, 109–133.
- Garey, M. R. & Johnson, D. S. (1979), *Computers and intractability : a guide to the theory of NP-completeness*, San Francisco : W.H. Freeman.
- Gilbert, E. N. (1959), 'Random Graphs', *The Annals of Mathematical Statistics* **30**(4), 1141 – 1144.  
**URL:** <https://doi.org/10.1214/aoms/1177706098>
- Glover, F. & Kochenberger, G. (2006), *Handbook of Metaheuristics*, International Series in Operations Research & Management Science, Springer US.  
**URL:** <https://books.google.com.br/books?id=P-HpBwAAQBAJ>
- Li, M., Hao, J.-K. & Wu, Q. (2020), 'General swap-based multiple neighborhood adaptive search for the maximum balanced biclique problem', *Computers & Operations Research* **119**, 104922.
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M. & Stützle, T. (2016), 'The irace package: Iterated racing for automatic algorithm configuration', *Operations Research Perspectives* **3**, 43–58.  
**URL:** <https://www.sciencedirect.com/science/article/pii/S2214716015300270>
- Mladenovic, N. & Hansen, P. (1997), 'Variable neighborhood search', *Computers and Operations Research* **24**(11), 1097–1100.
- Papadimitriou, C. (1994), *Computational Complexity*, Theoretical computer science, Addison-Wesley.  
**URL:** <https://books.google.com.br/books?id=JogZAQAAIAAJ>

- Peeters, R. (2003), 'The maximum edge biclique problem is np-complete', *Discrete Applied Mathematics* **131**(3), 651 – 654.
- Prais, M. & Ribeiro, C. C. (2000), 'Reactive grasp: An application to a matrix decomposition problem in tdma traffic assignment', *INFORMS Journal on Computing* **12**(3), 164–176.  
**URL:** <https://doi.org/10.1287/ijoc.12.3.164.12639>
- Resende, M. G. C. & Ribeiro, C. C. (2019), *Greedy Randomized Adaptive Search Procedures: Advances and Extensions*, Springer International Publishing, Cham, pp. 169–220.
- Talbi, E. (2009), *Metaheuristics: From Design to Implementation*, Wiley Series on Parallel and Distributed Computing, Wiley.  
**URL:** <https://books.google.com.br/books?id=SIsa6zi5XV8C>
- Wang, H. & Chen, S. (2020), A bipartite graph-based recommender for crowdfunding with sparse data, in R. Haron, M. M. Husin & M. Murg, eds, 'Banking and Finance', IntechOpen, Rijeka, chapter 14.  
**URL:** <https://doi.org/10.5772/intechopen.92781>
- Wang, Y., Cai, S. & Yin, M. (2018), 'New heuristic approaches for maximum balanced biclique problem', *Information Sciences* **432**, 362 – 375.
- Wolsey, L. (1998), *Integer Programming*, Wiley Series in Discrete Mathematics and Optimization, Wiley.  
**URL:** <https://books.google.com.br/books?id=x7RvQgAACAAJ>
- Zhao, Y., Chen, Z., Chen, C., Wang, X., Lin, X. & Zhang, W. (2022), 'Finding the maximum k-balanced biclique on weighted bipartite graphs', *IEEE Transactions on Knowledge and Data Engineering* pp. 1–14.
- Zhou, Y. & Hao, J.-K. (2017), 'Combining tabu search and graph reduction to solve the maximum balanced biclique problem', *ArXiv* **abs/1705.07339**.
- Zhou, Y., Rossi, A. & Hao, J.-K. (2018), 'Towards effective exact methods for the maximum balanced biclique problem in bipartite graphs', *European Journal of Operational Research* **269**(3), 834 – 843.