



**UNIVERSIDADE FEDERAL DO CEARÁ  
CENTRO DE CIÊNCIAS  
DEPARTAMENTO DE ESTATÍSTICA E MATEMÁTICA APLICADA  
CURSO DE GRADUAÇÃO EM MATEMÁTICA INDUSTRIAL**

**DARIO FILIPE DA SILVA COSTA**

**APLICAÇÃO DE HEURÍSTICAS AO K-MIS**

**FORTALEZA  
2025**

DARIO FILIPE DA SILVA COSTA

APLICAÇÃO DE HEURÍSTICAS AO K-MIS

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Matemática Industrial  
do Centro de Ciências da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Matemática Industrial.

Orientador: Prof. Dr. Carlos Diego Rodrigues.

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

S579a Silva Costa, Dario Filipe da.

Aplicação de Heurísticas ao k-MIS / Dario Filipe da Silva Costa. – 2025.  
56 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Ciências,  
Curso de Matemática Industrial, Fortaleza, 2025.

Orientação: Prof. Dr. Carlos Diego Rodrigues.

1. Heurísticas. 2. Máxima Interseção de k-Subconjuntos. 3. Colônia de Formigas. 4. Greedy Randomized  
Adaptive Search Procedure. 5. Vizinhança Variada. I. Título.

CDD 510

---

DARIO FILIPE DA SILVA COSTA

**APLICAÇÃO DE HEURÍSTICAS AO K-MIS**

Trabalho de Conclusão de Curso apresentado ao  
Curso de Graduação em Matemática Industrial  
do Centro de Ciências da Universidade Federal  
do Ceará, como requisito parcial à obtenção do  
grau de bacharel em Matemática Industrial.

Aprovada em: 25/07/2025

**BANCA EXAMINADORA**

---

Prof. Dr. Carlos Diego Rodrigues (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Wladimir Araújo Tavares  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Michael Ferreira de Souza  
Universidade Federal do Ceará (UFC)

Aos meus pais, Maria Luzimar e Jose Maria.

## **AGRADECIMENTOS**

Aos meus pais, Maria Luzimar e Jose Maria, cujo suporte tornou possível.

Aos meus irmãos, Davi, Dennis, Daniel e Debora, por viabilizarem minha permanência na graduação.

Ao Prof. Dr. Carlos Diego Rodrigues, por reconhecer meu potencial nesta área e me motivar a realizar este trabalho, orientando nos momentos cruciais.

Aos Professores Doutores Wladimir Araújo Tavares e Michael Ferreira de Souza, por integrarem a banca examinadora deste trabalho, contribuindo com suas expertises para o aprimoramento da versão final.

A Deus, por permitir que meu pai esteja comigo neste momento, apesar das muitas dificuldades advindas da idade.

## RESUMO

O problema da Máxima Interseção de k-Subconjuntos (kMIS),  $\mathcal{NP}$ -difícil e sem algoritmo  $\alpha$ -aproximado, possui aplicações na área de privacidade de dados e redes sociais. Há abordagens exatas e meta-heurísticas indicadas na literatura, mas, como se trata de um problema desafiador, existe espaço para avanços, pois as abordagens exatas tomam um tempo proibitivo, em instâncias maiores, e as meta-heurísticas parecem sempre poder evoluir. O kMIS consiste em escolher  $k$  subconjuntos distintos  $S_{i_1}, \dots, S_{i_k} \subset \mathcal{R}$  pertencentes a  $\mathcal{L} = \{S_1, \dots, S_n\}$ , de modo que a interseção  $|S_{i_1} \cap \dots \cap S_{i_k}|$  seja máxima. As Meta-Heurísticas Greedy Randomized Adaptive Search Procedure (GRASP) e Colônia de Formigas (ANT) obtêm relativo sucesso na prática, sobretudo quando combinadas com intensificações como a Vizinhança Variada (VND) ou Busca Tabu (TS). Neste trabalho, são propostas modificações ao ANT e ao VND, em busca de obter melhores resultados. E, em especial, é proposta uma metodologia simplificada para a comparação entre as heurísticas, padronizando o tempo de execução em função do tamanho do conjunto  $\mathcal{L}$  de cada instância, pois comumente, na literatura, os testes computacionais são realizados com discrepância de tempo de execução, fazendo a comparação ocorrer tanto no quesito qualidade da solução, quanto tempo. Portanto, fixar o tempo possibilita focar apenas na qualidade da solução, tornando mais claros os resultados do experimento. Os procedimentos heurísticos são organizados em duas componentes: (i) a meta-heurística principal, responsável pela construção das soluções, e (ii) o método de intensificação, acionado para aprimorá-las. Este estudo realizou experimentos computacionais com nove combinações distintas entre os três métodos principais, GRASP, ANT e ANT Modificado, e três técnicas de intensificação, TS, VND e VND Modificado.

**Palavras-chave:** Heurísticas; Máxima Interseção de k-Subconjuntos; Colônia de Formigas; Greedy Randomized Adaptive Search Procedure; Vizinhança Variada.

## ABSTRACT

The Maximum k-Subset Intersection Problem (kMIS),  $\mathcal{NP}$ -hard and with no  $\alpha$ -approximation algorithm, finds applications in data privacy and social networks. There are exact and meta-heuristic approaches in the literature, but as a challenging problem, remains room for advances, because exact methods become prohibitive on large instances, and meta-heuristics always seem able to take another step. The kMIS consists of choosing  $k$  distinct subsets  $S_{i_1}, \dots, S_{i_k} \subset \mathcal{R}$  from  $\mathcal{L} = \{S_1, \dots, S_n\}$  so that the intersection  $|S_{i_1} \cap \dots \cap S_{i_k}|$  is maximum. The metaheuristics Greedy Randomized Adaptive Search Procedure (GRASP) and Ant Colony System (ANT) achieve relative success in practice, especially when combined with intensification techniques like Variable Neighborhood Descent (VND) or Tabu Search (TS). This work proposes modifications to ANT and VND to obtain better results. Additionally, proposes a simplified methodology for comparing heuristics, by standardizing execution time according to the size of  $\mathcal{L}$  for each instance, since computational experiments in the literature are often conducted with time discrepancies, making comparisons happen in both solution quality and time. Therefore, by fixing the time and focusing solely on solution quality, we reduce the complexity of the comparison and clarify the experimental results. The heuristic procedures are organized into two components: (i) the main meta-heuristic, responsible for solution construction, and (ii) the intensification method, called to enhance them. This study reports computational experiments with nine distinct combinations among the three main methods, GRASP, ANT and Modified ANT, and the three intensification techniques, TS, VND and Modified VND.

**Keywords:** Heuristics; Maximum k-Subset Intersection; Ant Colony System; Greedy Randomized Adaptive Search Procedure; Variable Neighborhood Descent.

## LISTA DE TABELAS

Tabela 1 – Pacientes fictícios . . . . .	13
Tabela 2 – Matriz Binária $M$ . . . . .	13
Tabela 3 – Classes das instâncias . . . . .	32
Tabela 4 – Instâncias viáveis por classe . . . . .	32
Tabela 5 – Instâncias viáveis por tamanho de $\mathcal{L}$ . . . . .	33
Tabela 6 – Tempo de execução por tamanho de $\mathcal{L}$ . . . . .	34
Tabela 7 – Parâmetros testados na heurística ANT_VND . . . . .	35
Tabela 8 – Parâmetros testados na heurística GRASP_TS . . . . .	35
Tabela 9 – Os 10 maiores scores dos parâmetros do ANT_VND . . . . .	42
Tabela 10 – Os 10 maiores scores dos parâmetros do GRASP_TS . . . . .	42
Tabela 11 – Estatísticas da redução por Classe . . . . .	43
Tabela 12 – Tamanho das classes antes e após a redução . . . . .	44
Tabela 13 – Estatísticas das Heurísticas nas Instâncias Base . . . . .	44
Tabela 14 – Score das Heurísticas nas Instâncias Base por Classe . . . . .	45
Tabela 15 – Estatísticas das Heurísticas nas Instâncias Reduzidas . . . . .	46
Tabela 16 – Score das Heurísticas nas Instâncias Reduzidas por Classe . . . . .	47
Tabela 17 – Score das Heurísticas nas Instâncias Reduzidas por Classe Real . . . . .	47

## LISTA DE ALGORITMOS

Algoritmo 1 – Heurística Gulosa básica (HG) . . . . .	19
Algoritmo 2 – Greedy Randomized Adaptive Search Procedure (GRASP) . . . . .	20
Algoritmo 3 – Construção Randômica e Gulosa (CRG) . . . . .	20
Algoritmo 4 – Colônia de Formigas (ANT) . . . . .	23
Algoritmo 5 – Busca Local (LS) . . . . .	25
Algoritmo 6 – Busca Tabu (TS) . . . . .	26
Algoritmo 7 – Variable Neighborhood Descent (VND) . . . . .	28
Algoritmo 8 – Colônia de Formigas modificada (ANT') . . . . .	39
Algoritmo 9 – VND modificado (VND') . . . . .	40

## **LISTA DE ABREVIATURAS E SIGLAS**

ANT	Colônia de Formigas
GRASP	Greedy Randomized Adaptive Search Procedure
HG	Heurística Gulosa básica
kMIS	Máxima Interseção de k-Subconjuntos
LS	Busca Local
MEB	Maximum Edge Biclique
TS	Busca Tabu
VND	Descida em Vizinhança Variada

## LISTA DE SÍMBOLOS

$\mathcal{L}$	Conjunto dos subconjuntos
$\mathcal{R}$	Conjunto de todos os elementos
$S_i$	Subconjunto de $\mathcal{R}$ , do rótulo $i$ , pertencente a $\mathcal{L}$
$k$	Número de subconjuntos pertencentes a $\mathcal{L}$ que devem ser escolhidos
$\mathcal{L}'$	Uma solução, completa ou parcial, $\mathcal{L}' \subseteq \mathcal{L}$ com $ \mathcal{L}'  \leq k$ .
$I$	Uma instância do kMIS, com $I = (\mathcal{L}, \mathcal{R}, k)$

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>12</b>
<b>2</b>	<b>REVISÃO DA LITERATURA . . . . .</b>	<b>16</b>
<b>2.1</b>	<b>Definições e notações . . . . .</b>	<b>16</b>
<b>2.2</b>	<b>Meta-Heurísticas principais . . . . .</b>	<b>18</b>
<b>2.2.1</b>	<b><i>Heurísticas Gulosas (HG e HG Estendida)</i> . . . . .</b>	<b>18</b>
<b>2.2.2</b>	<b><i>Greedy Randomized Adaptive Search Procedure (GRASP)</i> . . . . .</b>	<b>19</b>
<b>2.2.3</b>	<b><i>Colônia de Formigas (ANT)</i> . . . . .</b>	<b>21</b>
<b>2.3</b>	<b>Intensificações . . . . .</b>	<b>24</b>
<b>2.3.1</b>	<b><i>Busca Local (LS) e Busca Tabu (TS)</i> . . . . .</b>	<b>24</b>
<b>2.3.2</b>	<b><i>Variable Neighborhood Descent (VND)</i> . . . . .</b>	<b>27</b>
<b>2.4</b>	<b>Redução das Instâncias . . . . .</b>	<b>28</b>
<b>3</b>	<b>METODOLOGIA . . . . .</b>	<b>31</b>
<b>3.1</b>	<b>Instâncias e critérios de avaliação . . . . .</b>	<b>31</b>
<b>3.1.1</b>	<b><i>Instâncias</i> . . . . .</b>	<b>31</b>
<b>3.1.2</b>	<b><i>Critérios de Avaliação</i> . . . . .</b>	<b>33</b>
<b>3.1.3</b>	<b><i>Experimento preliminar dos parâmetros</i> . . . . .</b>	<b>34</b>
<b>3.2</b>	<b>Representação do problema . . . . .</b>	<b>36</b>
<b>3.2.1</b>	<b><i>Bit Paralelismo</i> . . . . .</b>	<b>36</b>
<b>3.2.2</b>	<b><i>Representação prática das instâncias e soluções</i> . . . . .</b>	<b>37</b>
<b>3.3</b>	<b>Propostas do autor . . . . .</b>	<b>38</b>
<b>3.3.1</b>	<b><i>Modificação do ANT</i> . . . . .</b>	<b>38</b>
<b>3.3.2</b>	<b><i>Modificação do VND</i> . . . . .</b>	<b>39</b>
<b>4</b>	<b>RESULTADOS . . . . .</b>	<b>41</b>
<b>4.1</b>	<b>Seleção dos parâmetros . . . . .</b>	<b>41</b>
<b>4.2</b>	<b>Redução das Instâncias . . . . .</b>	<b>42</b>
<b>4.3</b>	<b>Experimento nas Instâncias Base . . . . .</b>	<b>44</b>
<b>4.4</b>	<b>Experimento nas Instâncias Reduzidas . . . . .</b>	<b>45</b>
<b>5</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS . . . . .</b>	<b>48</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>50</b>
	<b>APÊNDICE A –Agrupamentos por classe . . . . .</b>	<b>51</b>

## 1 INTRODUÇÃO

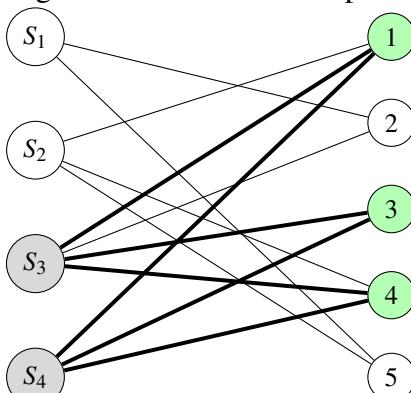
Considere uma rede social especializada em criar comunidades, essa rede indica usuários que tem interesse em comum. Suponha que a rede deseja indicar as  $k$  pessoas com o máximo de interesses em comum, para formar uma comunidade, todos juntos. Escolher essas pessoas mais compatíveis parece uma tarefa simples à primeira vista, mas se mostra um problema combinatório complicado.

Modelando um usuário como um conjunto  $S_u$ , que tem todos os interesses que ele indicou na rede. O objetivo, então, é selecionar  $k$  usuários de modo que ao fazer a interseção  $S_{u_1} \cap S_{u_2} \cap \dots \cap S_{u_k}$  dos interesses, ela seja a máxima. Esse é um exemplo do Problema da Máxima Interseção de  $k$ -Subconjuntos (kMIS).

Definido como: Dado um conjunto  $\mathcal{L} = \{S_1, \dots, S_n\}$  com  $n$  subconjuntos de um conjunto finito  $\mathcal{R} = \{e_1, \dots, e_m\}$  e um inteiro positivo  $k$ , o problema consiste em selecionar exatamente  $k$  dos subconjuntos pertencentes a  $\mathcal{L}$ , como  $\{S_{i_1}, \dots, S_{i_k}\}$ , de forma que a cardinalidade da interseção dos escolhidos  $|S_{i_1} \cap \dots \cap S_{i_k}|$  seja máxima.

Por exemplo, seja  $\mathcal{R} = \{1, 2, 3, 4, 5\}$ ,  $\mathcal{L} = \{S_1, S_2, S_3, S_4\}$ , onde  $S_1 = \{2, 5\}$ ,  $S_2 = \{1, 4, 5\}$ ,  $S_3 = \{1, 2, 3, 4\}$  e  $S_4 = \{1, 3, 4\}$ , e  $k = 2$ . Uma tentativa de solução válida pode ser definida como  $L' = \{S_1, S_3\}$ , que tem como resultado  $|S_1 \cap S_3| = |\{2\}| = 1$ . Neste exemplo, é fácil perceber que há escolhas com tamanho de interseção maior, logo,  $L'$  não é uma solução ótima. Para instâncias de pequeno porte, é possível obter a solução ótima ao enumerar todas as combinações, neste caso, tem como resultado:  $|S_1 \cap S_2| = 1$ ,  $|S_1 \cap S_3| = 1$ ,  $|S_1 \cap S_4| = 0$ ,  $|S_2 \cap S_3| = 2$ ,  $|S_2 \cap S_4| = 2$ ,  $|S_3 \cap S_4| = 3$ . E, portanto,  $L'' = \{S_3, S_4\}$  é uma solução ótima, com  $|S_3 \cap S_4| = |\{1, 3, 4\}| = 3$ . A Figura 1 ilustra o grafo bipartido que representa essa instância exemplo, com sua solução ótima  $L''$  destacada.

Figura 1 – Instância exemplo



Fonte: Próprio autor

O kMIS foi inicialmente proposto e demonstrado  $\mathcal{NP}$ -difícil por (VINTERBO, 2002), o seu contexto de aplicação consistia em selecionar dados hospitalares que pudessem ser expostos sem pôr em risco a privacidade dos pacientes. Esses dados divulgados precisariam estar em outros pacientes ao mesmo tempo, para tornar difícil identificar de quem veio. Assim o hospital conseguiria divulgar relatórios sem causar prejuízo.

Para exemplificar essa aplicação do kMIS em privacidade de dados, mesmo que de maneira simplificada, são apresentados na Tabela 1 dados fictícios de pacientes. O objetivo é identificar quais atributos do paciente com  $id=1$  podem ser expostos de forma segura. Considerando que para serem classificados como seguros, esses atributos devem estar presentes, ao mesmo tempo, em pelo menos um outro paciente. Tais atributos devem ser indiscerníveis dos correspondentes na linha comparada.

A matriz binária  $M$  desses atributos é construída, na qual cada valor de atributo distinto possui uma coluna correspondente, com bit 1 apenas nas linhas em que esse valor ocorre. A matriz  $M$ , neste exemplo, é apresentada na Tabela 2.

Tabela 1 – Pacientes fictícios

id	idade	sintoma	peso
1	25	1	62
2	35	2	75
3	18	1	62
4	25	3	75

Fonte: Próprio autor

Tabela 2 – Matriz Binária  $M$

		idade			sintoma			peso	
		18	25	35	1	2	3	62	75
$i$	$j$	1	2	3	4	5	6	7	8
		1	0	1	0	1	0	0	1
1	0	0	1	0	0	1	0	0	1
2	1	0	0	0	1	0	0	1	0
3	0	1	0	0	0	0	1	0	1
4	0	0	1	0	0	0	1	0	1

Fonte: Próprio autor

Sendo então  $\mathcal{R} = \{1, 2, 3, 4, 5, 6, 7, 8\}$  e cada subconjunto  $S_i = \{j \in \mathcal{R} : M_{ij} = 1\}$ , partindo da solução parcial  $L' = \{S_1\}$ , esta instância do kMIS pode ser resolvida, sendo obtidos os  $k - 1$  subconjuntos com a maior interseção com  $S_1$ , e, assim, serem identificados os atributos seguros, presentes na interseção, e os que devem ser apagados, ausentes.

Neste exemplo com  $k = 2$ , a maior interseção é formada entre  $S_1$  e  $S_3$ , nas colunas de índice 4 e 7, portanto, os atributos de sintoma e peso do paciente de  $id=1$  são considerados seguros, anônimos, os demais atributos devem ser ocultados, neste caso, a idade.

A ocultação na matriz  $M$  pode ser realizada pela atribuição de 1 a todas as colunas daquele atributo, dessa forma, o valor original não é identificado e passa a ser considerado na interseção com todos. Na tabela real, inserir um caractere específico, como “-”, é suficiente.

Ao realizar esse procedimento para cada paciente, obtém-se uma tabela em que cada linha é indiscernível, ou confundível, com ao menos uma outra linha.

Esse foi um breve exemplo, simplificado, de aplicação do kMIS em privacidade de dados.

Apesar da simples definição, o problema kMIS foi demonstrado  $\mathcal{NP}$ -difícil e sem algoritmo  $\frac{1}{N^{\epsilon'}}$ -aproximado por (XAVIER, 2012), sendo  $N$  o tamanho da entrada e  $\epsilon'$  dependente apenas de uma constante. Na demonstração, o autor apresenta uma redução do problema Maximum Edge Biclique (MEB), já provado  $\mathcal{NP}$ -difícil, para o kMIS e em seguida uma função de tempo polinomial que leva soluções ótimas do kMIS para o MEB correspondente, portanto, caso o kMIS possa ser resolvido ou ter solução  $\frac{1}{N^{\epsilon'}}$ -aproximada da ótima em tempo polinomial, o MEB também teria, algo impossível, a menos que  $\mathcal{NP} = \mathcal{P}$ .

E em (BOGUE *et al.*, 2014) o kMIS foi utilizado para obter soluções de instâncias baseadas em dados reais de uma Rádio FM, onde a rádio possuía uma lista de bandas ( $\mathcal{L}$ ) e os ouvintes ( $\mathcal{R}$ ) que se consideravam fãs de algumas bandas ao mesmo tempo, se um ouvinte  $e_v$  é fã da banda  $S_u$ , então  $e_v \in S_u$ . A solução é um grupo de  $k$  bandas que tem o máximo de fãs em comum.

Neste trabalho, os procedimentos heurísticos são organizados em duas componentes:

1. A heurística principal, chamada diretamente, responsável pela construção de soluções diversas e acionamento da intensificação para algumas ou todas elas;
2. O método de intensificação, acionado para aprimorar uma solução de entrada, retornando sempre uma solução de qualidade igual ou superior.

Duas das principais meta-heurísticas propostas na literatura para o problema são abordadas, sendo elas: Colônia de Formigas (ANT) com Vizinhança Variada (VND), proposta por (DIAS *et al.*, 2022); Greedy Randomized Adaptive Search Procedure (GRASP) com Busca Tabu (TS), proposta por (CASADO *et al.*, 2022).

Modificações são propostas para o ANT e para a intensificação VND, sendo denominadas ANT' e VND'. Ao final, são apresentados os resultados dos experimentos computacionais realizados com cada combinação das três heurísticas principais, GRASP, ANT, e ANT', com as três intensificações, TS, VND e VND', totalizando nove combinações distintas.

Os experimentos computacionais foram conduzidos com uma metodologia levemente distinta da observada na literatura. O tempo de execução foi fixado em função do tamanho de  $\mathcal{L}$  da instância, com o intuito de simplificar a comparação entre as heurísticas. Assim, a

comparação passa a ocorrer apenas sobre a qualidade das soluções encontradas.

Dos objetivos deste trabalho:

1) Objetivo Geral:

Este trabalho tem como objetivo geral propor uma metodologia de comparação entre as heurísticas abordadas, e, também, modificações para melhorar a qualidade das soluções obtidas por elas.

2) Objetivo Específico:

- Propor modificações que avancem a qualidade das heurísticas aplicadas ao kMIS que tiveram sucesso na literatura;
- Propor uma metodologia de avaliação das heurísticas quando aplicadas ao kMIS;
- Fornecer material descritivo que facilite a compreensão de como implementar na prática as estruturas usadas para resolver o kMIS;
- Comparar as heurísticas de relevância da literatura com as modificadas propostas neste trabalho.

Neste trabalho, no Capítulo 2 são apresentados dois procedimentos meta-heurísticos e três de intensificação que foram aplicados ao kMIS com destaque na literatura, e, ao final, um procedimento de redução de instâncias. No Capítulo 3 são apresentadas as metodologias de criação de instâncias para os experimentos e metodologia de avaliação de resultados, esta, sendo diferente das utilizadas na literatura, fazendo uma padronização do tempo de execução para a comparação entre os procedimentos, e ao final duas propostas de modificações em heurísticas. No Capítulo 4 os resultados computacionais são apresentados, para a seleção de parâmetros, redução das instâncias e comparação entre heurísticas. No Capítulo 5 as conclusões e trabalhos futuros propostos.

## 2 REVISÃO DA LITERATURA

Contém a apresentação dos conceitos utilizados durante todo o trabalho. Os procedimentos heurísticos são organizados em duas componentes:

1. A heurística principal, chamada diretamente, responsável pela construção de soluções diversas e acionamento da intensificação para algumas ou todas elas;
2. O método de intensificação, acionado para aprimorar uma solução de entrada.

Na Seção 2.1 são apresentadas algumas definições importantes, conjuntos e notações utilizadas durante todo o trabalho. Na Seção 2.2 são apresentadas duas Heurísticas Gulosas e duas Meta-Heurísticas, que tiveram destaque de sucesso aplicadas ao kMIS na literatura. Na Seção 2.3 são apresentados três procedimentos de intensificação, que podem ser combinados as meta-heurísticas principais. Na Seção 2.4 um procedimento de redução para o problema.

### 2.1 Definições e notações

Nesta seção, são apresentadas algumas das definições, notações e nomenclaturas utilizadas durante todo o trabalho. Além da importância para o entendimento do texto, estas notações podem ser de grande valor para a compreensão dos códigos fontes deste trabalho, muitas variáveis são nomeadas com a mesma notação como base.

De início, o próprio problema: Como definido em (BOGUE *et al.*, 2013), dado um conjunto  $\mathcal{L} = \{S_1, \dots, S_n\}$  com  $n$  subconjuntos de um conjunto finito  $\mathcal{R} = \{e_1, \dots, e_m\}$  e um inteiro positivo  $k$ , o problema kMIS consiste em selecionar exatamente  $k$  dos subconjuntos pertencentes a  $\mathcal{L}$ , como  $\{S_{i_1}, \dots, S_{i_k}\}$ , de forma que a cardinalidade da interseção dos escolhidos  $|S_{i_1} \cap \dots \cap S_{i_k}|$  seja máxima. Alguns exemplos simples podem ser encontrados no Capítulo 1.

As definições a seguir:

1. Temos os Conjuntos Fundamentais:

- a) **Definição 2.1.1** *Conjunto  $\mathcal{R}$ :*

Universo de elementos  $e_i$ , que modelam algo que se deseja maximizar na interseção dos subconjuntos,  $\mathcal{R} = \{e_1, \dots, e_m\}$ .

- b) **Definição 2.1.2** *Um subconjunto  $S_i$  qualquer:*

Contém todos ou alguns dos elementos  $\{e_t\}$ ,  $t \in \{1, \dots, m\}$ , portanto,  $S_i \subseteq \mathcal{R}$ .

- c) **Definição 2.1.3** *Conjunto  $\mathcal{L}$ :*

Universo de subconjuntos  $S_i$  que podem ser selecionados para compor uma resposta,

$$\mathcal{L} = \{S_1, \dots, S_n\}.$$

2. Instância, soluções e funções de avaliação:

d) **Definição 2.1.4** *Instância I:*

É uma tripla definida tal qual  $I = (\mathcal{L}, \mathcal{R}, k)$ , onde  $k$  é o número de subconjuntos  $S_i$ , distintos, a serem escolhidos.

e) **Definição 2.1.5** *Solução completa ou parcial  $L'$ :*

Um conjunto  $L' = \{S_{k_1}, \dots, S_{k_k}\}$ , onde, caso  $|L'| = k$ , é uma solução completa; se  $|L'| < k$ , ainda é parcial.

f) **Definição 2.1.6** *Função de avaliação  $f(L')$ :*

É o valor da cardinalidade da interseção dos subconjuntos em  $L'$ . Definida como:

$$f(L') = \left| \bigcap_{S_i \in L'} S_i \right|.$$

g) **Definição 2.1.7** *Função de avaliação parcial  $g(L', S_i)$ :*

Carrega a mesma ideia de  $f$ , mas aplicada a uma solução parcial  $L'$  e a um subconjunto  $S_i$ , candidato a ser incluído no conjunto, portanto:

$$g(L', S_i) = \left| \bigcap (L' \cup \{S_i\}) \right|.$$

3. Representação em grafo:

h) **Definição 2.1.8** *Grafo representante  $G$ :*

Um grafo simples e bipartido  $G = (\mathcal{L} \cup \mathcal{R}, E)$  representa  $I$  se

$$\forall u \in \{1, \dots, n\}, \forall v \in \{1, \dots, m\}, (S_u, e_v) \in E \Leftrightarrow e_v \in S_u.$$

Ou seja, a aresta existe se o subconjunto  $S_u$  contém  $\{e_v\}$  e não existe caso contrário.

4. Conjuntos auxiliares:

i) **Definição 2.1.9** *Lista de Candidatos  $CL$ :*

$CL = \mathcal{L} \setminus L'$ , subconjuntos ainda disponíveis para entrar em  $L'$ .

j) **Definição 2.1.10** *Lista de Remoção  $EL$ :*

$EL = L'$ , ou seja, os subconjuntos que podem ser removidos da solução.

5. Vizinhanças e Intensificação:

k) **Definição 2.1.11** *Vizinhança de uma solução  $L'$ :*

Definida uma operação de modificação (por exemplo, trocar  $S_i \in EL$  por  $S_j \in CL$ ), a vizinhança é o conjunto de todas as soluções obtidas com uma única modificação.

1) **Definição 2.1.12 *Intensificação*:**

Processo que, a partir de uma solução de partida, explora sua vizinhança (e/ou outras estratégias) buscando soluções de qualidade superior.

## 2.2 Meta-Heurísticas principais

Nesta seção são inicialmente apresentadas duas heurísticas gulosas, que são a base de pensamento para como vários dos outros procedimentos foram elaborados. Posteriormente, apresentadas as duas meta-heurísticas principais, que são chamadas diretamente nos testes computacionais. Estas constroem as soluções de maneiras diversas, e chamam uma intensificação para ser aplicada as suas soluções.

A combinação de uma heurística principal com uma intensificação específica é referida como a junção das abreviações separadas por um “\_”. Sendo, por exemplo, o GRASP combinado a Busca Tabu referido como GRASP\_TS.

### 2.2.1 *Heurísticas Gulosas (HG e HG Estendida)*

Uma heurística gulosa é caracterizada como um método construtivo em que, a cada etapa, é selecionada a alternativa de maior valor da etapa atual. Impactos futuros dessa escolha não são considerados. Tem como vantagem a eficiência computacional, por conta da simplicidade consegue obter uma solução rapidamente. Costuma ter bons resultados, mas não garante obtenção do ótimo (CORMEN *et al.*, 2022).

Em (BOGUE *et al.*, 2013) é apresentado um procedimento guloso para o kMIS. Consiste em um processo de construção que escolhe sempre o subconjunto que gera a maior interseção na etapa, ou seja, escolhe o  $S_j$  de máximo valor de  $g(L', S_j)$ , até que  $|L'| = k$ . Sendo a primeira escolha o maior subconjunto. Dessa forma, o subconjunto  $S = \arg \max_{S \in \mathcal{L}} |S|$  sempre é presente na solução, pois é a primeira escolha. No Algoritmo 1 o pseudocódigo do procedimento.

O único argumento é  $I = (\mathcal{L}, \mathcal{R}, k)$ , a instância. Na Linha 1,  $\mathbf{B}$  é inicializado com todos os elementos e durante a construção, no laço, armazena o conjunto  $\cap(L')$ , para velocidade no cálculo de  $g(L', S) = |\cap(L' \cup \{S\})| = |\mathbf{B} \cap S|$ .

Este procedimento consegue ser um bom ponto de partida, mas uma escolha totalmente gulosa dessa forma, no kMIS, leva com frequência a ótimos locais e até a não encontrar solução de valor maior que 0, mesmo em instâncias com solução.

---

**Algoritmo 1:** Heurística Gulosa básica (HG)
 

---

**Entrada :**  $I = (\mathcal{L}, \mathcal{R}, k)$

**1**  $\mathbf{B} \leftarrow \mathcal{R}$   
**2**  $L' \leftarrow \emptyset$   
**3**  $CL \leftarrow \mathcal{L}$   
**4** **Enquanto**  $|L'| < k$  **faça:**  
**5**    $S_{\max} \leftarrow \arg \max_{S \in CL} |\mathbf{B} \cap S|$   
**6**    $L' \leftarrow L' \cup \{S_{\max}\}$   
**7**    $CL \leftarrow CL \setminus \{S_{\max}\}$   
**8**    $\mathbf{B} \leftarrow \mathbf{B} \cap S_{\max}$   
**Saída**   :  $L'$

---

Para aumentar um pouco o espaço de busca, já que a Heurística Gulosa básica (HG) executa rapidamente, foi proposto em (COSTA *et al.*, 2018) tentar todos os subconjuntos  $S_i \in \mathcal{L}, i \in \{1, \dots, n\}$  como o conjunto inicial, e obter os outros  $k - 1$  executando igual a HG, retornando a melhor das  $n = |\mathcal{L}|$  soluções geradas. Para evitar cálculo desnecessário, é possível abortar a construção de soluções parciais que já estejam com valor menor que a melhor obtida até o momento. Essa implementação da HG com início em todos os subconjuntos será referida como Heurística Gulosa Estendida (HG\_Estendida), e será usada no processo de redução das instâncias.

### 2.2.2 Greedy Randomized Adaptive Search Procedure (GRASP)

Inicialmente proposto por (FEO; RESENDE, 1989) para o problema de cobertura de conjuntos, o Greedy Randomized Adaptive Search Procedure (GRASP) teve destaque por também ter bons resultados em diversos outros problemas de otimização combinatória, além da sua facilidade de paralelização e flexibilidade (CARVALHO *et al.*, 2020). Ele consiste em repetir duas etapas principais: primeira, a construção de uma solução de maneira Randômica, Gulosa e Adaptativa (sim, tudo ao mesmo tempo); segunda, uma intensificação dessa solução, comumente uma busca em alguma estrutura de vizinhança, mas podendo ser heurísticas mais complexas.

Em (CASADO *et al.*, 2022) é apresentada uma versão do GRASP aplicada ao kMIS. O processo de construção do GRASP é dito **Adaptativo**, pois sempre que um subconjunto  $S_k$  qualquer é selecionado e passa a fazer parte da solução parcial,  $L' \leftarrow L' \cup \{S_k\}$ , todos os outros subconjuntos tem seus valores de avaliação, ou *atração*, atualizados para o novo estado de  $L'$ , por exemplo, a partir do momento que  $S_k$  entra, o valor de  $g(L', S_1)$  pode ter mudado. Dito **Guloso**, pois a função  $g(L', S_i)$  é usada para favorecer subconjuntos que geram uma próxima etapa de

maior valor. E **Randômico**, pois o processo não escolhe o  $S_{j_{\max}} = \arg \max_{S_i \in CL} g(L', S_i)$ , na verdade, permite que subconjuntos de valor de  $g$  menor tenham chance de serem selecionados, assim, garantindo diversidade nas soluções.

No Algoritmo 2 o GRASP citado.

---

**Algoritmo 2:** Greedy Randomized Adaptive Search Procedure (GRASP)

---

**Entrada :**  $I = (\mathcal{L}, \mathcal{R}, k), \alpha, \Delta$

- 1  $L_b \leftarrow \emptyset$
- 2 **Para**  $i \in 1, \dots, \Delta$  **faça:**
- 3     $L' \leftarrow \text{Construção}(I, \alpha)$  // Etapa 1
- 4     $L'' \leftarrow \text{Intensificação}(L')$  // Etapa 2
- 5    **Se**  $f(L'') > f(L_b)$  **então:**
- 6       $L_b \leftarrow L''$

**Saída :**  $L_b$

---

Os três parâmetros são  $I = (\mathcal{L}, \mathcal{R}, k)$ , a instância,  $\alpha \in [0, 1]$ , controle de aleatoriedade da construção (próximo de 0, mais randômico, próximo de 1, menos), e  $\Delta \in \mathbb{Z}_+$ , o número de repetições. No laço de repetição principal (Linha 2) são construídas e intensificadas  $\Delta$  soluções e, ao final, a melhor é retornada.

Para a primeira etapa, uma abordagem clássica usada no original (FEO; RESENDE, 1989) seria criar uma Lista de Candidatos Restrita,  $RCL \subseteq CL$ , com apenas os subconjuntos que ultrapassem um valor mínimo  $\mu = \alpha \cdot g(S_{j_{\max}})$  e, então, selecionar aleatoriamente desta lista menor de subconjuntos, que são mais atrativos. Porém, também é possível inverter a ordem, definir um tamanho  $\lambda_{RCL} = \max(1, \lfloor \alpha \cdot |CL| + 0.5 \rfloor)$  para a  $RCL$ , selecionar randomicamente os  $\lambda_{RCL}$  subconjuntos e, então, pegar o de maior valor de  $g(L', S)$ . A segunda teve mais sucesso em (CASADO *et al.*, 2022), no Algoritmo 3 é apresentado o pseudocódigo da Construção Randômica e Gulosa (CRG) utilizada.

---

**Algoritmo 3:** Construção Randômica e Gulosa (CRG)

---

**Entrada :**  $I = (\mathcal{L}, \mathcal{R}, k), \alpha$

- 1  $S_{init} \leftarrow \text{Random}(\mathcal{L})$
- 2  $L' \leftarrow \{S_{init}\}$
- 3  $CL \leftarrow \mathcal{L} \setminus \{S_{init}\}$
- 4 **Enquanto**  $|L'| < k$  **faça:**
- 5     $RCL \leftarrow \text{SelectRandom}(CL, \max(1, \lfloor \alpha \cdot |CL| + 0.5 \rfloor))$
- 6     $S_{max} \leftarrow \arg \max_{S \in CL} g(S)$
- 7     $L' \leftarrow L' \cup \{S_{max}\}$
- 8     $CL \leftarrow CL \setminus \{S_{max}\}$

**Saída :**  $L'$

---

Os dois parâmetros são  $I = (\mathcal{L}, \mathcal{R}, k)$ , a instância, e  $\alpha \in [0, 1]$ , controle de aleatoriedade da construção. Na Linha 6 a avaliação  $g(S)$  consiste na própria avaliação  $g(L', S) = |\cap(L' \cup \{S\})|$ . A implementação real tem muitos outros detalhes, como, por exemplo, a  $CL$  pode ser uma lista de boolianos, se um subconjunto está ou não disponível, ou que é importante manter o conjunto  $B = \cap L'$ , para que a verificação da interseção com os próximos subconjuntos candidatos seja mais rápida. Esses detalhes, da implementação real, ficam mais claros na Seção 3.2, onde a representação prática do problema é explicada.

A segunda etapa consiste em utilizar uma das intensificações descritas na Seção 2.3, para obter uma solução igual ou melhor que a construída.

### 2.2.3 Colônia de Formigas (ANT)

A Meta-Heurística Colônia de Formigas traz inspiração do comportamento das mesmas, depositando feromônios para criar rotas de interesse, uma forma de comunicação indireta que move centenas delas para seus objetivos em comum. Em (DIAS *et al.*, 2022) é apresentado um ANT\_VND para o kMIS, com resultados competitivos.

No ANT de 2022, é montado o grafo simples e completo  $G_a = (V_a, E_a)$ , onde as formigas “andam”, com  $V_a = \mathcal{L}$  e  $E_a$  todas as arestas possíveis com os vértices  $V_a$ . Toda aresta tem um valor de feromônio  $\tau$  associado, que é iniciado com uma constante  $r_0 = 1/f(HG(I))$ , onde  $I$  é a instância em questão, HG a Heurística Gulosa Básica (Subseção 2.2.1) e  $f$  a função de avaliação da solução retornada. No caso excepcional em que  $f(HG(I))$  retorne 0, foi escolhido o valor 5 como padrão, iniciando com 0.2 os feromônios.

Toda formiga  $t$  inicia seu caminho, de vértices colecionados em  $A_t \subseteq V_a$ , no vértice  $S_t$ . Portanto, inicialmente  $A_t = \{S_t\}$ . Então, percorre outros  $k - 1$  vértices em  $V_a$ . Cada caminho de formiga é uma solução do kMIS. Para construir esse caminho, dado que uma formiga  $t$  visitou o vértice  $S_i$  por último, cada vértice  $S_j \in V_a \setminus A_t$  tem uma probabilidade  $p_{ij}^t$  de ser escolhido.

Essa probabilidade  $p_{ij}^t$  é da seguinte forma. Um  $r \in [0, 1]$  é sorteado, e se  $r$  for menor ou igual a um parâmetro  $q_0 \in [0, 1]$ , o  $S_j$  que maximiza um certo valor de atratividade é escolhido (Regra da Proporcionalidade Pseudoaleatória (ACS)). Por outro lado, se  $r$  for maior que  $q_0$ , então, um sorteio proporcional é realizado (Regra da Proporcionalidade Aleatória (AS)). A atratividade, aqui, é produto de um fator guloso  $\eta_j^t$  com o feromônio  $\tau_{ij}$  da aresta  $(S_i, S_j) \in E_a$ .

Nas Equações 2.1, 2.2 e 2.3 as definições da Regra ACS, Regra AS e  $\eta_j^t$ . Sendo  $\alpha$  e  $\beta$  parâmetros de entrada que controlam a relação do  $\tau_{ij}$  e  $\eta_j^t$ .

Se  $r \leq q_0$ , na Regra da Proporcionalidade Pseudoaleatória (ACS), o  $S_j$  escolhido é:

$$S_j = \begin{cases} \arg \max_{S_k \in V_a \setminus A_t} \left\{ \tau_{ik} \cdot [\eta_k^t]^\beta \right\} & , r \leq q_0 \\ \text{Usa sorteio R.P.A. Equação 2.2 , C. C.} & \end{cases} \quad (2.1)$$

Caso  $r > q_0$ , pela Regra da Proporcionalidade Aleatória (AS), as probabilidades de cada  $S_j$  são dadas por:

$$p_{ij}^t = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_j^t]^\beta}{\sum_{S_k \in V_a \setminus A_t} \left\{ [\tau_{ik}]^\alpha \cdot [\eta_k^t]^\beta \right\}} & , S_j \in V_a \setminus A_t \\ 0 & , \text{C. C.} \end{cases} \quad (2.2)$$

E o fator heurístico guloso é dado por:

$$\eta_j^t = \frac{1}{|\cap(A_t) - \cap(A_t \cup \{S_j\})| + 1} \quad (2.3)$$

O valor de  $\eta_j^t$  representa o impacto negativo de adicionar  $S_j$  ao caminho da formiga  $t$ , refletindo a perda de elementos na interseção. Se não houver perda,  $\eta_j^t = 1$ , e quanto maior for a redução da interseção ao incluir  $S_j$ , menor será seu valor, diminuindo, assim, a probabilidade de  $S_j$  ser escolhido.

Ao final de cada iteração (sem atualização local durante as construções) cada aresta tem seu  $\tau_{ij}$  atualizado da seguinte forma:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta_{ij} \quad (2.4)$$

Onde  $\rho$  é o parâmetro que controla o quanto do feromônio guardado vai ser reduzido e o quanto do valor novo ( $\Delta_{ij}$ ) adicionado na atualização, chamado de parâmetro de evaporação.

Sendo  $Q_{ij} = \{A_k : S_i \in A_k \wedge S_j \in A_k, \}$ , o conjunto dos  $A_k$  para cada formiga  $k$  que passou pela aresta  $(S_i, S_j)$  **na iteração atual**, então:

$$\Delta_{ij} = \begin{cases} \frac{\sum_{A_k \in Q_{ij}} f(A_k)}{f(L_b) \cdot |Q_{ij}|} & , |Q_{ij}| > 0 \\ 0 & , C.C. \end{cases} \quad (2.5)$$

Onde  $L_b$  é a melhor solução obtida até então. Portanto,  $\Delta_{ij}$  é a média dos valores das soluções que tiveram  $S_i$  e  $S_j$ , na iteração atual, dividida pelo valor da melhor solução obtida.

Após definir os mecanismos de construção dos caminhos e atualização dos feromônios, outra estratégia implementada em (DIAS *et al.*, 2022) é o controle das formigas ativas. Duas estratégias são aplicadas de forma alternada a cada iteração: a primeira é a seleção aleatória das  $q_{\text{tdformigas}}$  formigas ativas, que criará seus caminhos na iteração; a segunda é escolher as  $q_{\text{tdformigas}}$  com as melhores médias de valor das soluções já geradas.

Para que a segunda estratégia de escolha de formigas ativas seja eficaz, todas as formigas são mantidas ativas e seus caminhos são construídos até que se atinja o número de iterações  $G_{\text{reativo}}$ . Desse modo, garante-se maior estabilidade às médias dos valores das soluções. A partir da iteração  $G_{\text{reativo}} + 1$ , alternam-se as duas estratégias, sendo as  $q_{\text{tdformigas}}$  formigas ativas escolhidas ora por sorteio aleatório, ora com base na média. Deve ser observado que  $G_{\text{reativo}} < \text{iterMax}$ , sendo  $\text{iterMax}$  o parâmetro que controla o número total de iterações.

No Algoritmo 4 encontra-se o pseudocódigo do Colônia de Formigas descrito.

---

**Algoritmo 4:** Colônia de Formigas (ANT)

---

**Entrada :**  $I = (\mathcal{L}, \mathcal{R}, k), \alpha, \beta, \rho, q_0, q_{\text{tdformigas}}, G_{\text{reativo}}, \text{iterMax}, \tau_0$

- 1  $M[t] \leftarrow (0, 0)$  // Estatísticas das Formigas
- 2  $V_a, E_a \leftarrow \text{constroe\_}G_a(I); L_b \leftarrow \emptyset; \tau[i, j] \leftarrow \tau_0; A[t] \leftarrow \emptyset$
- 3 **Para**  $ite \in \{1, \dots, \text{iterMax}\}$  **faça:**
- 4      $A_{\text{ativas}} \leftarrow \text{EscolheFormigasAtivas}(ite, G_{\text{reativo}}, M, V_a)$
- 5     **Para**  $t \in A_{\text{ativas}}$  **faça:**
- 6          $A[t] \leftarrow \{S_t\}$
- 7          $i \leftarrow t$
- 8         **Enquanto**  $|A[t]| < k$  **faça:**
- 9              $r \leftarrow \text{random}(0, 1)$
- 10             **Se**  $r \leq q_0$  **então:**
- 11                  $j \leftarrow \text{SelecaoACS}(i, \tau, A[t], V_a, \alpha, \beta)$
- 12             **Se**  $\text{não}$ :
- 13                  $j \leftarrow \text{SorteioAS}(i, \tau, A[t], V_a, \alpha, \beta)$
- 14              $A[t] \leftarrow A[t] \cup \{S_j\}$
- 15              $i \leftarrow j$
- 16          $M[t] \leftarrow M[t] + (f(A[t]), 1)$
- 17          $L_{ite} \leftarrow \text{Intensifica\_a\_Melhor}(A, A_{\text{ativas}})$  // Opcional, mas recomendado
- 18         **Se**  $f(L_{ite}) > f(L_b)$  **então:**
- 19              $L_b \leftarrow L_{ite}$
- 20          $\tau \leftarrow \text{Atualiza\_tau}(\tau, \rho, A, A_{\text{ativas}}, L_b)$  // O poder de um pseudocódigo
- Saída :**  $L_b$

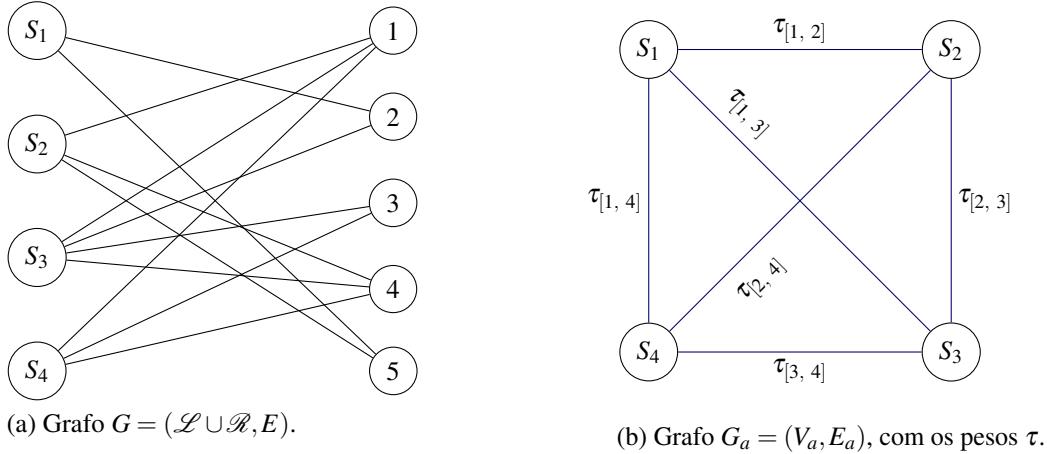
---

Os 9 argumentos são, I, a instância,  $\alpha$ , o expoente do feromônio na atratividade,  $\beta$ , o expoente do fator heurístico guloso,  $\rho$ , a evaporação,  $q_0$ , a probabilidade da escolha determinística (ACS) ser usada,  $q_{\text{tdformigas}}$ , o número de formigas ativas após as iterações iniciais,

$G_{reativo}$ , o número de iterações iniciais com todas as formigas ativas,  $iterMax$ , a quantidade de iterações, e por fim  $\tau_0$ , o valor inicial dos Feromônios.

Para facilitar o entendimento deste procedimento, considere a instância exemplo que foi apresentada no Capítulo 1, de Introdução. Temos seu grafo  $G$  e  $G_a$  representados na Figura 2. O  $G_a$  sendo onde as formigas constroem suas soluções, destacadas as arestas com seus feromônios.

Figura 2 – Grafo  $G$  e  $G_a$  da Instância Exemplo



Fonte: Próprio autor

Passamos a ter, em  $G_a$ , um grafo que relaciona diretamente os próprios subconjuntos. Na etapa de construção, cada formiga ativa  $t$  começa no vértice  $S_t$  e caminha para  $k - 1$  outros vértices, seguindo as Regras ACS ou AS, definidas nas Equações 2.1 e 2.2.

### 2.3 Intensificações

Nesta seção, são apresentados três procedimentos de intensificação. Cada um deles parte de uma solução previamente obtida, referida como  $L'$ , e aplica estratégias de trocas, remoções e outras operações para gerar uma nova solução de qualidade superior.

Os procedimentos considerados são:

- Busca Local (LS) e Busca Tabu (TS), propostas por (CASADO *et al.*, 2022);
- Descida em Vizinhança Variada (VND), proposta por (DIAS *et al.*, 2022).

#### 2.3.1 Busca Local (LS) e Busca Tabu (TS)

Seja a operação  $Swap$  definida como  $Swap(L', S_i, S_j) = (L' \setminus \{S_i\}) \cup \{S_j\}$ , uma nova solução alcançada pela troca de  $S_i$  por  $S_j$ . Uma vizinhança  $N(L')$  pode ser definida por todas as

soluções possíveis a partir de um Swap de  $L'$ .

Em (CASADO *et al.*, 2022) são definidos dois procedimentos de Busca nessa vizinhança. Uma Busca Local (LS) e uma Busca Tabu (TS).

A Busca Local, consiste em realizar a operação de *Swap* entre todo  $S_i \in L'$  e  $S_j \in \mathcal{L} \setminus L'$ , caso encontre uma melhora, repete o processo para a nova solução. Caso não encontre, retorna a mesma solução de entrada. No Algoritmo 5 o pseudocódigo do procedimento.

---

**Algoritmo 5:** Busca Local (LS)

---

```

Entrada :  $I = (\mathcal{L}, \mathcal{R}, k), L'$ 
1 Melhorou  $\leftarrow$  Verdadeiro
2 Enquanto Melhorou faça:
3   Melhorou  $\leftarrow$  Falso
4   Para  $S_i \in RandomOrder(L')$  faça:
5      $\mathbf{B}' \leftarrow \bigcap_{S_t \in L' \setminus \{S_i\}} S_t$ 
6     Para  $S_j \in RandomOrder(\mathcal{L} \setminus L')$  faça:
7        $\mathbf{B}'' \leftarrow \mathbf{B}' \cap S_j$ 
8       Se  $|\mathbf{B}''| > f(L')$  então:
9          $L' \leftarrow Swap(L', S_i, S_j)$ 
10        Melhorou  $\leftarrow$  Verdadeiro
11   go to 2
Saída   : $L'$ 

```

---

Os dois parâmetros são  $I = (\mathcal{L}, \mathcal{R}, k)$ , a instância, e  $L'$ , a solução que terá a vizinhança explorada. Na Linha 5 é armazenado o conjunto da interseção com  $S_i$  retirado (é importante que esse cálculo seja eficiente, recomendado paralelismo de bits, Subseção 3.2.1) e então os  $S_j$  são tentados no lugar, sendo necessário realizar apenas mais uma interseção com o  $\mathbf{B}'$  para cada  $S_j$ . Vale ressaltar que, como a primeira melhora já reinicia o processo, é importante percorrer a Lista de Remoção (EL) e a Lista de Candidatos (CL) de maneira aleatória (*RandomOrder*) para não favorecer nenhuma ordem criada, por qualquer motivo, nesses conjuntos.

Assim como a Busca Local 5, a Busca Tabu (TS) realiza movimentos (o *Swap* citado) em uma vizinhança da solução inicial, mas tenta melhorar a maneira de explorar esse espaço. Para isso, a principal ideia é manter alguns subconjuntos que entraram na solução como intocáveis, por isso “Tabu”, algo que fica marcado na memória de não ser mexido por um período (PITSOULIS; RESENDE, 2002). Subconjuntos marcados dessa forma são chamados de **Tabu Ativos**.

Fazendo uso de Memória de Curta Duração (*Short Term Memory, STM*) nos ele-

mentos recém adicionados. Eles são colocados como Tabu Ativos, restringindo a saída deles da solução por um tempo determinado pelo tamanho da  $STM$ , definido por um parâmetro  $\tau$ . Quando a  $STM$  está no tamanho máximo, ela retira a condição de Tabu Ativo do subconjunto mais antigo na lista, este passando a poder ser removido da solução a partir desse momento.

Além disso, um ponto diferente nesta TS em relação a LS, é que caso nenhuma solução visitada seja superior que a melhor já armazenada, o movimento de menor perda de qualidade da iteração é feito, em uma solução secundaria. Ou seja, a exploração continua mesmo que, para isso, faça trocas que levem para uma solução de qualidade igual ou até inferior que a melhor obtida. Neste trabalho, essa é a única intensificação que faz esse tipo de exploração, passando por soluções de menor qualidade, as outras só efetivam o movimento ao encontrar uma melhora.

Como a busca continua mesmo passando por soluções de qualidade inferior, o critério de parada também se distingue, passando a ser um certo número  $\gamma$  de iterações sem encontrar um avanço na melhor solução armazenada. No Algoritmo 6 o pseudocódigo desse procedimento.

---

**Algoritmo 6:** Busca Tabu (TS)

---

```

Entrada :  $I = (\mathcal{L}, \mathcal{R}, k), L', \tau, \gamma$ 
1  $L_b \leftarrow L'; STM \leftarrow \emptyset; \Gamma \leftarrow 0$ 
2 Enquanto  $\Gamma < \gamma$  faça:
3   Melhorou  $\leftarrow$  Falso
4    $\langle S_i^b, S_j^b, f^b \rangle \leftarrow \langle -1, -1, -1 \rangle$ 
5   Para  $S_i \in RandomOrder(L' \setminus STM)$  faça:
6      $B' \leftarrow \bigcap_{S_t \in L' \setminus \{S_i\}} S_t$ 
7     Para  $S_j \in RandomOrder(\mathcal{L} \setminus L')$  faça:
8        $B'' \leftarrow B' \cap S_j$ 
9       Se  $|B''| > f(L_b)$  então:
10       $L' \leftarrow Swap(L', S_i, S_j)$                                 // Melhora
11       $Melhorou \leftarrow$  Verdadeiro
12       $\Gamma \leftarrow 0$ 
13       $L_b \leftarrow L'$ 
14       $MarkTabu(STM, S_j, \tau)$ 
15      go to 2
16      Se  $\text{não } Melhorou \wedge |B''| > f^b$  então:
17         $\langle S_i^b, S_j^b, f^b \rangle \leftarrow \langle S_i, S_j, |B''| \rangle$           // Att o mov de menor perda
18      Se  $\text{não } Melhorou$  então:
19         $L' \leftarrow Swap(L', S_i^b, S_j^b)$                                 // Movimento de menor perda
20         $\Gamma \leftarrow \Gamma + 1$ 
21         $MarkTabu(STM, S_j^b, \tau)$ 


---


Saída :  $L_b$ 

```

---

Os quatro parâmetros são  $I$ , a instância,  $L'$ , a solução que vai ter sua vizinhança explorada,  $\tau \in [0, 1]$ , o tamanho da lista  $STM$  em relação a  $k$ , e  $\gamma \in \mathbb{Z}_+$ , o número de iterações sem melhora até a parada. Em comparação a Busca Local (LS) as grandes diferenças são, na Linha 5 onde o subconjunto que pode ser retirado  $S_i$  percorre aleatoriamente  $L' \setminus STM$  e não toda a  $L'$ . E na Linha 19 onde uma troca por uma solução de qualidade mais baixa acontece. E claro, todo o controle da  $STM$ . Quando a  $STM$  atinge o tamanho máximo e um novo subconjunto  $S_j$  é marcado como Tabu Ativo, o subconjunto mais antigo na  $STM$  tem seu status de Tabu Ativo removido, passando a fazer parte da Lista de Remoção Restrita ( $L' \setminus STM$ ).

A Busca Tabu foi comparada a Busca Local em (CASADO *et al.*, 2022), tendo-se verificado que a TS supera a LS quando combinadas ao GRASP. Em virtude desse resultado, um GRASP\_LS não foi incluído nos testes finais deste trabalho. Em experimentos preliminares constatou-se, como na referência, a LS com desempenho consideravelmente inferior aos demais métodos. Essa exclusão não compromete, portanto, os resultados do estudo e evita combinações excessivas entre heurísticas principais e intensificações.

A combinação GRASP\_TS obteve bons resultados na referência, sendo o algoritmo de destaque para o kMIS na literatura até o ano de 2022, com a proposta do ANT\_VND de (DIAS *et al.*, 2022) com resultados competitivos. Aqui a TS vai ser experimentada como intensificação na Colônia de Formigas (ANT), também.

### 2.3.2 *Variable Neighborhood Descent (VND)*

Os outros dois procedimentos utilizados como intensificação, Tabu Search (Subseção 2.3.1) e Local Search (Algoritmo 5), buscam obter soluções melhores trocando apenas um subconjunto por vez, utilizam a mesma estrutura de vizinhança, apenas com uma pequena restrição na TS. A proposta do Descida em Vizinhança Variada (VND) de (DIAS *et al.*, 2022) modifica isso, explora de uma forma diferente, em múltiplas estruturas de vizinhanças.

É proposto o seguinte procedimento: Remover aleatoriamente  $t = \max(3; \lfloor 0.3 \cdot k \rfloor)$  subconjuntos de  $L'$ , nas instâncias com  $k > 3$ , e remontar a solução com a ideia gulosa do HG (Subseção 2.2.1), que consiste em escolher para fazer parte da solução os subconjuntos que geram a maior interseção na etapa atual, repetir esse processo até a solução ficar com o tamanho  $k$ .

O processo de remover  $t$  subconjuntos é repetido três vezes, sem permitir remoções repetidas, mas livre para que subconjuntos removidos voltem para a solução. No Algoritmo 7 o

pseudocódigo dessa proposta.

---

**Algoritmo 7:** Variable Neighborhood Descent (VND)

---

```

Entrada :  $I = (\mathcal{L}, \mathcal{R}, k), L'$ 
1  $L_b \leftarrow L'$ 
2  $rep \leftarrow 3$ 
3  $t \leftarrow \max(3, 0.3 \cdot k)$ 
4  $\Theta \leftarrow \emptyset$  // Subconjuntos já sorteados
5 Enquanto  $rep > 0$  faça:
6    $rep \leftarrow rep - 1$ 
7    $EL \leftarrow escolhe\_aleatorio(t, (L_b \setminus \Theta))$ 
8    $\Theta \leftarrow \Theta \cup EL$ 
9    $L' \leftarrow remove(L_b, EL)$  // Copia, sem modificar  $L_b$ 
10   $L' \leftarrow remonta_guloso(L', I)$ 
11  Se  $f(L') > f(L_b)$  então:
12     $L_b \leftarrow L'$ 
Saída :  $L_b$ 

```

---

Os dois argumentos são  $I$ , a instância, e  $L'$ , a solução que terá sua vizinhança explorada.

Essa exploração bem rápida, montando 3 soluções, apresentou bons resultados no artigo de referência como intensificação de uma Colônia de Formigas. Aqui vai ser utilizada como intensificação de uma versão do GRASP, chamada GRASP\_VND e do mesmo ANT.

## 2.4 Redução das Instâncias

Em (BOGUE *et al.*, 2014) é apresentado um procedimento de Redução das Instâncias do kMIS, algo que busca facilitar a sua resolução, afinal, quanto menos vértices, mais fácil encontrar a combinação de melhor valor. Reduzir o conjunto de soluções possíveis é bem-vindo.

Sendo  $G = (\mathcal{L} \cup \mathcal{R}, E)$  o grafo que representa a instância  $I = (\mathcal{L}, \mathcal{R}, k)$ , e  $\lambda = f(HG\_Estendida(I))$ , o valor da solução da Heurística Gulosa Estendida (Subseção 2.2.1) aplicada a  $I$ . O  $\lambda$  é um limite inferior para o valor da melhor solução do kMIS, com ele é possível guiar a remoção de alguns vértices de  $G$  que estão apenas em soluções de valor abaixo de  $\lambda$ .

Sejam dois conjuntos  $\mathbf{L}^*$  e  $\mathbf{R}^*$  da forma:

$$\mathbf{L}^*(S_u, \lambda) = \{S_v \in \mathcal{L} \setminus \{S_u\} : |N_G(S_u) \cap N_G(S_v)| < \lambda\} \quad (2.6)$$

$$\mathbf{R}^*(e_v, k) = \{e_u \in \mathcal{R} \setminus \{e_v\} : |N_G(e_v) \cap N_G(e_u)| < k\} \quad (2.7)$$

A vizinhança  $N_G(S_u)$ , de um  $S_u \in \mathcal{L}$  qualquer, são todos os elementos  $e_k \in \mathcal{R}$  que também pertencem ao subconjunto  $e_k \in S_u$ . Já a vizinhança  $N_G(e_v)$  de um  $e_v \in \mathcal{R}$  são todos os subconjuntos  $S_i \in \mathcal{L}$  aos quais  $e_v$  pertence.

Para um  $S_u$  qualquer, o conjunto  $\mathbf{L}^*(S_u, \lambda)$  contém todos os  $S_v \in \mathcal{L} \setminus \{S_u\}$  que ao formar par com  $S_u$  já passam a ter a interseção abaixo de  $\lambda$ , ou seja, formam pares ruins com ele, que já limitam a solução para um valor abaixo do limite inferior. E para um  $e_v$  qualquer, o conjunto  $\mathbf{R}^*(e_v, k)$  contém todos os  $e_u \in \mathcal{R} \setminus \{e_v\}$  que ao formar par com  $e_v$  não pertencem, ao mesmo tempo, a  $k$  subconjuntos, ou seja, também são pares ruins, pois juntos não podem ser elementos de qualquer solução.

Se um  $S_u$  tem o conjunto  $\mathbf{L}^*(S_u, \lambda)$  tal que  $|\mathcal{L}| - |\mathbf{L}^*(S_u, \lambda)| < k$ , ele pode ser removido de  $G$ , pois esse  $S_u$  não tem pares suficiente fora de  $\mathbf{L}^*(S_u, \lambda)$  para formar uma solução com chance de ter interseção maior que  $\lambda$ . Ele precisaria de ao menos um  $S_k \in \mathbf{L}^*(S_u, \lambda)$  junto a ele para completar uma solução, portanto seria inferior a  $\lambda$ .

Se um  $e_v$  tem o conjunto  $\mathbf{R}^*(e_v, k)$  tal que  $|\mathcal{R}| - |\mathbf{R}^*(e_v, k)| < \lambda$ , então  $e_v$  pode ser removido, pois não forma bons pares com ao menos  $\lambda$  elementos, que estejam ao mesmo tempo em  $k$  subconjuntos. Logo  $e_v$  só pode ser elemento de interseções de tamanho menor que  $\lambda$ .

As regras de redução ficam:

1. Enquanto conseguir remover vértices repita **2** e **3**.
2.  $\forall S_u \in \mathcal{L}$ , se  $|\mathcal{L}| - |\mathbf{L}^*(S_u, \lambda)| < k$  remova  $S_u$ .
3.  $\forall e_v \in \mathcal{R}$ , se  $|\mathcal{R}| - |\mathbf{R}^*(e_v, k)| < \lambda$  remova  $e_v$ .

Esse procedimento foi realizado nas Instâncias deste trabalho, mas as Instâncias originais, chamadas de “Base”, não foram descartadas, com o objetivo de comparar todos os métodos implementados aplicados nas Instâncias Base e Reduzidas, para conseguir analisar o quanto a redução melhora a eficiência na busca.

Vale ressaltar que uma versão aprimorada dessa Redução é apresentada em (COSTA, 2021), onde o autor foca nos conjuntos inversos a  $\mathbf{R}^*$  e  $\mathbf{L}^*$ . Observando os subconjuntos que são bons pares com um  $S_u$ , definido por:

$$\overline{\mathbf{L}^*}(S_u, \lambda) = \mathcal{L} \setminus (\mathbf{L}^*(S_u, \lambda) \cup \{S_u\})$$

É possível checar se cada elemento da vizinhança  $N_G(S_u) \subseteq \mathcal{R}$  está em pelo menos  $k - 1$  dos subconjuntos que formam bons pares com  $S_u$ . Se não estiver, esse elemento não é capaz de fazer parte da interseção de uma solução com valor melhor ou igual a  $\lambda$  que tenha  $S_u$ , portanto, pode ser desconsiderado como um elemento de  $S_u$  para efeitos da redução. Ao desconsiderar todos os elementos de  $S_u$  que não estão em ao menos  $k - 1$  dos subconjuntos em  $\overline{\mathbf{L}^*}(S_u, \lambda)$ , restam apenas os elementos que realmente têm potencial para estar na interseção de uma solução que tenha  $S_u$  e valor igual ou maior que  $\lambda$ . O conjunto desses elementos é definido por:

$$C_{\mathcal{L}}(S_u) = \{e_v \in N_G(S_u) : |\overline{\mathbf{L}^*}(S_u, \lambda) \cap N_G(e_v)| \geq k - 1\}$$

Caso o tamanho desse conjunto,  $C_{\mathcal{L}}(S_u)$ , seja menor que  $\lambda$ , o  $S_u$  pode ser removido.

A ideia análoga, observando um dado elemento  $e_v$  e seu conjunto  $\overline{\mathbf{R}^*}(e_v, k)$ , também acontece. Sendo  $\overline{\mathbf{R}^*}$  definido como:

$$\overline{\mathbf{R}^*}(e_v, k) = \mathcal{R} \setminus (\mathbf{R}^*(e_v, k) \cup \{e_v\})$$

E é usada para remover elementos, caso o conjunto  $C_{\mathcal{R}}(e_v)$  tenha tamanho menor que  $k$ . Sendo  $C_{\mathcal{R}}(e_v)$  definido como:

$$C_{\mathcal{R}}(e_v) = \{S_u \in N_G(e_v) : |\overline{\mathbf{R}^*}(e_v, k) \cap N_G(S_u)| \geq \lambda - 1\}$$

A Redução de (COSTA, 2021), focando nos conjuntos dos bons pares, conseguiu remover 26% mais vértices que a proposta anterior, de 2014, nas instâncias do seu experimento. Porém, em razão do maior foco na comparação das heurísticas neste trabalho, o aprimoramento do processo de redução não foi implementado, apenas mencionado nesta seção, tendo sido aplicada a redução mais simples, definida em (BOGUE *et al.*, 2014).

### 3 METODOLOGIA

Detalhes da implementação real, como: representação na linguagem escolhida; escolha das instâncias usadas; estatísticas de comparação entre métodos deste trabalho. Além de fornecer a metodologia rigorosamente, para garantir a reproduzibilidade da pesquisa. É neste segmento que são ressaltadas as diferenças no emprego dos métodos em relação aos materiais de referência, pequenas modificações na tentativa de obter melhores resultados.

A Seção 3.1 inicia com a descrição do processo de geração das instâncias e, posteriormente, detalha os critérios utilizados para avaliar as heurísticas e definir seus parâmetros. Uma metodologia diferente da literatura é proposta, com os tempos de execução padronizados. Na Seção 3.2 são detalhadas as estruturas implementadas na prática, na linguagem de programação escolhida. E o mais empolgante, na Seção 3.3 são apresentadas propostas de modificações para a Meta-Heurística ANT e para a intensificação VND.

#### 3.1 Instâncias e critérios de avaliação

Nesta seção, são inicialmente detalhadas as instâncias utilizadas, e, em seguida, os critérios de avaliação da qualidade das soluções geradas nos testes computacionais. Por fim, é apresentado o processo de definição dos parâmetros. Os critérios de avaliação estabelecidos são empregados tanto na análise comparativa dos parâmetros de uma mesma heurística, quanto na comparação entre heurísticas distintas.

##### 3.1.1 *Instâncias*

Foram geradas 540 instâncias aleatórias, construídas da seguinte forma: Definido um tamanho  $n$  para o conjunto  $\mathcal{L}$ , são gerados 3 tamanhos  $m$  para o conjunto  $\mathcal{R}$ ,  $m_{0.8} = \lfloor n \cdot 0.8 + 0.5 \rfloor$ ,  $m_1 = n$  e  $m_{1.25} = \lfloor n \cdot 1.25 + 0.5 \rfloor$ , de modo que, quando com tamanhos diferentes, a relação do menor conjunto sobre o maior é aproximadamente 80%, não é exatamente, pois o inteiro mais próximo é selecionado. Para cada par  $(n, m_{0.8})$ ,  $(n, m_1)$  e  $(n, m_{1.25})$  são geradas duas instâncias por classe.

As classes são 9, definidas em (BOGUE *et al.*, 2013), com relação a densidade de arestas  $p$  e ao número de subconjuntos a serem escolhidos  $k$ . Sendo  $E_K$  o conjunto de todas as arestas possíveis no grafo bipartido de vértices  $\mathcal{L} \cup \mathcal{R}$  que representa a instância sendo construída (Def. 2.1.8),  $|E_K| = |\mathcal{L}| \cdot |\mathcal{R}|$ , temos que  $p$  é classificado como baixo se  $p \in [0.1, 0.3] \cdot |E_K|$ ,

médio se  $p \in [0.4, 0.6] \cdot |E_K|$  e alto se  $p \in [0.7, 0.9] \cdot |E_K|$ . Já  $k$  é classificado como baixo se  $k \in [0.1, 0.3] \cdot |\mathcal{L}|$ , médio se  $k \in [0.4, 0.6] \cdot |\mathcal{L}|$  e alto se  $k \in [0.7, 0.9] \cdot |\mathcal{L}|$ . Na Tabela 3 uma melhor visualização das classes em relação a  $p$  e  $k$ .

Tabela 3 – Classes das instâncias

$p \backslash k$	baixo	médio	alto
baixo	C1	C2	C3
médio	C4	C5	C6
alto	C7	C8	C9

Fonte: BOGUE *et al.* 2013

Em cada classe,  $p$  e  $k$  são sorteados dentro da sua respectiva faixa, e com  $p$  da instância definido as  $\lfloor p \cdot |\mathcal{L}| \cdot |\mathcal{R}| + 0.5 \rfloor$  arestas são selecionadas aleatoriamente para fazer parte da instância. Cada aresta  $(S_u, e_v) \rightarrow e_v \in S_u$ .

Resumindo, para cada tamanho de  $\mathcal{L}$  em  $\{40, 60, 80, 100, 140, 180, 200, 240, 280, 300\}$  são geradas  $(3 \text{ variações de } |\mathcal{R}|) \cdot (9 \text{ classes}) \cdot (2 \text{ repetições}) = 54$  instâncias.

Das 540, como nos trabalhos anteriores, todas da classe C3 foram inviáveis, não possuem solução de valor superior a 0, e quase todas das C2 e C6 também. As instâncias inviáveis foram retiradas de todos os processos. Ao todo foram 340 viáveis, distribuídas como mostram as Tabelas 4 e 5.

Tabela 4 – Instâncias viáveis por classe

Classe	Qtd.	(%)
C1	51	85.0
C2	2	3.3
C3	0	0.0
C4	<b>60</b>	<b>100.0</b>
C5	54	90.0
C6	2	3.3
C7	<b>60</b>	<b>100.0</b>
C8	<b>60</b>	<b>100.0</b>
C9	51	85.0

Fonte: Próprio autor

As instâncias antes de passarem pelo processo de redução de (BOGUE *et al.*, 2014) são referidas como Instâncias Base, já depois da redução como Instâncias Reduzidas. Neste trabalho são analisados os resultados das heurísticas aplicadas a ambos os conjuntos de instâncias.

Tabela 5 – Instâncias viáveis por tamanho de  $\mathcal{L}$ 

$ \mathcal{L} $	Qtd.	(%)
40	<b>38</b>	<b>70.3</b>
60	35	64.8
80	37	68.5
100	33	61.1
140	31	57.4
180	35	64.8
200	33	61.1
240	33	61.1
280	33	61.1
300	32	59.2

Fonte: Próprio autor

### 3.1.2 Critérios de Avaliação

Boa parte da metodologia de avaliação usada em (DIAS *et al.*, 2022), com exceção do que diz respeito a tempo de execução, é seguida. Em cada teste, as heurísticas são executadas 10 vezes por instância, são armazenados o menor valor de solução, o maior e a média das repetições por instância. Com esses dados, são geradas 6 estatísticas para avaliação de qualidade: Média das mínimas ( $\bar{X}_{min}$ ); média das máximas ( $\bar{X}_{max}$ ); média das médias ( $\bar{X}_{avg}$ ); contagem de vezes que teve a melhor mínima de uma instância ( $\Sigma_{min}$ ); contagem de vezes que teve a melhor máxima de uma instância ( $\Sigma_{max}$ ); contagem de vezes que teve a melhor média de uma instância ( $\Sigma_{avg}$ ).

Então, as heurísticas são ranqueadas com base em uma função (score  $\Phi$ ) dessas estatísticas, valores entre 0 e 100. Seja  $\lambda^i = (\bar{X}_{min}^i, \bar{X}_{max}^i, \bar{X}_{avg}^i, \Sigma_{min}^i, \Sigma_{max}^i, \Sigma_{avg}^i)$  as estatísticas de uma heurística  $i$ , então, na Equação 3.1 é definida a função de pontuação:

$$\begin{aligned} \Phi(\lambda^i) = & 15 \cdot \left( \frac{\bar{X}_{min}^i - \min(\bar{X}_{min})}{\max(\bar{X}_{min}) - \min(\bar{X}_{min})} \right) + 15 \cdot \left( \frac{\bar{X}_{max}^i - \min(\bar{X}_{max})}{\max(\bar{X}_{max}) - \min(\bar{X}_{max})} \right) \\ & + 20 \cdot \left( \frac{\bar{X}_{avg}^i - \min(\bar{X}_{avg})}{\max(\bar{X}_{avg}) - \min(\bar{X}_{avg})} \right) + 15 \cdot \left( \frac{\Sigma_{min}^i - \min(\Sigma_{min})}{\max(\Sigma_{min}) - \min(\Sigma_{min})} \right) \\ & + 15 \cdot \left( \frac{\Sigma_{max}^i - \min(\Sigma_{max})}{\max(\Sigma_{max}) - \min(\Sigma_{max})} \right) + 20 \cdot \left( \frac{\Sigma_{avg}^i - \min(\Sigma_{avg})}{\max(\Sigma_{avg}) - \min(\Sigma_{avg})} \right) \end{aligned} \quad (3.1)$$

É dado um maior peso para a qualidade da média. Média das médias  $\bar{X}_{avg}$  e contagem de melhor média  $\Sigma_{avg}$  representam juntas 40% do valor de score.

Essa mesma metodologia de avaliação foi usada para ranquear os parâmetros de cada heurística, no teste de parâmetros, a diferença é que foram executados em 34 instâncias (10%

das viáveis), sorteadas aleatoriamente. Os parâmetros que tiveram melhor score foram fixados nas heurísticas para o teste de confronto entre heurísticas.

Quanto ao quesito tempo de execução, essa é a maior diferença das referências anteriores para este trabalho, no que diz respeito a metodologia de avaliação. O tempo foi padronizado com base no tamanho de  $\mathcal{L}$  da instância.

Todas as heurísticas executam seus laços de repetição principal até um tempo limite  $t_{\text{lim}}$  ser ultrapassado, que foi fixado em 10 segundos para instâncias de maior número de subconjuntos (instâncias de  $|\mathcal{L}| = 300$  executam por 10 segundos) e o tempo limite reduz linearmente conforme o tamanho de  $\mathcal{L}$  reduz. Ou seja, dada uma instância  $I = (\mathcal{L}, \mathcal{R}, k)$  o tempo de execução é dado por  $t_{\text{lim}}(I) = |\mathcal{L}| \cdot \left(\frac{10}{300}\right)$  segundos. Para total clareza, a Tabela 6 mostra o  $t_{\text{lim}}$  de cada tamanho de  $\mathcal{L}$ .

Tabela 6 – Tempo de execução por tamanho de  $\mathcal{L}$

$ \mathcal{L} $	$t_{\text{lim}}$ (segundos)
40	1.33
60	2.00
80	2.66
100	3.33
140	4.66
180	6.00
200	6.66
240	8.00
280	9.33
300	10.0

Fonte: Próprio autor

Quando o tempo  $t_{\text{lim}}$  é ultrapassado as construções de soluções param, mas caso uma solução já tenha sido construída, na iteração atual, é permitido uma última intensificação (contudo, restrita para apenas 1 iteração do seu laço principal) e comparação antes de ser devolvida a melhor solução. Por conta disto, ocorre de uma execução tomar um tempo mais longo que o  $t_{\text{lim}}$  (no máximo 160 milissegundos e em média 30 milissegundos a mais), mas como essa última intensificação ocorre em todas as heurísticas esse tempo extra é ignorado para qualquer tipo de avaliação.

### 3.1.3 Experimento preliminar dos parâmetros

Utilizando os critérios de avaliação definidos na Subseção 3.1.2, as heurísticas são executadas 10 vezes em cada uma das 34 Instâncias Base (10% das viáveis, antes de passarem pela redução), escolhidas aleatoriamente, para cada combinação dos parâmetros. Sendo 81

combinações nos parâmetros do ANT\_VND e 27 nos do GRASP\_TS.

Os parâmetros da ANT (2.2.3) com intensificação VND (2.3.2) foram testados nos conjuntos mostrados na Tabela 7.

Tabela 7 – Parâmetros testados na heurística ANT\_VND

Parâmetro	Conjunto	Descrição
$\beta$	{0.6, 0.8, 1.2}	Expoente do fator guloso na construção.
$\rho$	{0.1, 0.3, 0.5}	Taxa de evaporação do feromônio.
$q_0$	{0.7, 0.8, 0.9}	Probabilidade de escolha determinística (ACS).
qtd <sub>formigas</sub>	{0.1, 0.2, 0.3}	Quantidade de formigas ativas ( $ \mathcal{L}  \cdot \text{qtd}_{\text{formigas}}$ ) após $G_{\text{reativo}}$ iterações.

Fonte: Próprio autor

Os parâmetros  $\alpha$  e  $G_{\text{reativo}}$  foram fixados em 1 e 5, respectivamente. Dessa forma, a relação entre  $\tau$  e o fator guloso  $\eta$  passou a ser controlada exclusivamente pelo expoente  $\beta$ , do fator guloso. O valor de  $G_{\text{reativo}}$  foi fixado em 5, conforme recomendado pelos autores em (DIAS *et al.*, 2022). No que tange ao VND, o número de subconjuntos removidos da solução em cada tentativa, denotado por  $t$ , foi definido como  $t = \max(3, \lfloor 0.3 \cdot k \rfloor)$ , se  $k > 3$ , e  $k - 1$ , caso contrário.

Já os parâmetros do GRASP (2.2.2) com intensificação TS (2.3.1) foram testados nos conjuntos mostrados na Tabela 8.

Tabela 8 – Parâmetros testados na heurística GRASP\_TS

Parâmetro	Conjunto	Descrição
$\alpha$	{0.2, 0.5, 0.8}	Grau de “gulosidade” na construção da solução.
$\tau$	{0.1, 0.3, 0.5}	Tamanho $(\tau \cdot k)$ da Lista Tabu usada.
$\gamma$	{5, 10, 15}	Número de iterações sem melhora até a parada da TS.

Fonte: Próprio autor

Esses conjuntos de parâmetros foram selecionados de acordo com as indicações dos autores originais, mas pequenas variações foram introduzidas para que as heurísticas apresentassem melhor desempenho e uma comparação justa fosse obtida na nova metodologia de avaliação com tempo fixo, em vez de com número fixo de iterações, como ocorreu nos trabalhos anteriores.

Os parâmetros mais bem pontuados na  $\Phi$  do ANT ( $\beta, \rho, q_0, \text{qtd}_{\text{formigas}}$ ), no GRASP ( $\alpha$ ) e na intensificação TS ( $\tau, \gamma$ ) foram fixados em todas as combinações de heurísticas geradas. Desse modo, o  $\alpha$  destacado no experimento do GRASP\_TS foi utilizado no GRASP\_VND, e os parâmetros evidenciados no ANT\_VND e GRASP\_TS foram combinados no ANT\_TS. Idealmente, cada combinação de heurística principal e intensificação deveria ter seus parâmetros

revisados individualmente, entretanto, o tempo necessário para conduzir o experimento dessa maneira não estava disponível.

Com o intuito de minimizar os impactos decorrentes de eventuais escolhas inadequadas de parâmetros, as comparações propostas no Capítulo 5 foram conduzidas com cada componente isolada e a outra mantida fixa. Por exemplo, na comparação das heurísticas principais, o ANT\_TS foi comparado apenas ao GRASP\_TS e ao ANT'\_TS, sendo a segunda componente mantida fixa. Já na comparação das intensificações, o GRASP\_VND foi comparado apenas ao GRASP\_VND' e ao GRASP\_TS, sendo a primeira componente mantida fixa. Dessa forma, as avaliações foram realizadas de maneira mais justa, com a componente não analisada sendo mantida estável.

### 3.2 Representação do problema

Detalhes importantes da representação prática do kMIS e suas soluções, para conseguir eficiência nas operações realizadas nas heurísticas aplicadas.

#### 3.2.1 Bit Paralelismo

Conforme observado por (COSTA *et al.*, 2018), o uso de bit paralelismo é fundamental para a eficiência da operação de interseção entre os conjuntos. Como o objetivo no kMIS é maximizar a interseção, essa operação é repetida inúmeras vezes para a avaliação da qualidade das soluções obtidas.

Para realizar o Bit Paralelismo, é necessária a representação do conjunto como um vetor de bits, onde cada bit indica a presença (1) ou ausência (0) do elemento representado por aquela posição  $i$  específica do vetor. Com essa representação, é possível utilizar um operador de comparação bit a bit entre dois conjuntos. Sua vantagem é que compara, até o tamanho da palavra (32 ou 64 bits), em uma única operação do processador, sendo necessárias consideravelmente menos operações para a obtenção da interseção.

Em Python 3.11.9 o operador `&`, na forma  $(int_1) \& (int_2)$ , retorna um inteiro  $int_3$  que tem os bits como (1), na sua representação binária, apenas nas posições que eram (1) em ambos os inteiros passados, operação lógica AND ( $\wedge$ ) entre os bits de mesma posição. Estes  $int$  do Python são representados pelo *PyLongObject*, um vetor de inteiros mais tradicionais `uint32_t` e `uint16_t`, restritos às bases  $2^{30}$  e  $2^{15}$ , nos sistemas 64 e 32 bits, respectivamente. Essa restrição

é imposta para evitar *overflow* nas operações entre *int*, especialmente na multiplicação. Portanto, a operação `&` em Python atinge alta eficiência, usando o operador de comparação bit a bit do processador em cada par de inteiros tradicionais (FOUNDATION, 2024).

Em C++ haveria a vantagem da possibilidade de uso do tamanho de palavra completo para essa operação, pois `&` não causa *overflow*, pelo contrário, sempre resulta em um *int* menor. Porém, o projeto deste TCC começou por meio de um celular e Python era mais acessível pelo Google Colab.

Recapitulando, em Python representando cada subconjunto  $S_i \in \mathcal{L}$  como um *int*  $\mathbf{B}_i$  onde a *posição*  $j$  da sua forma binária é (1) se, e somente se, o subconjunto  $S_i$  tem como elemento o  $e_j \in \mathcal{R}$ . Assim, o operador é utilizado para obter a interseção dos subconjuntos. Basta realizar a operação `&` entre os dois inteiros que representam os subconjuntos  $S_{i_1}$  e  $S_{i_2}$ ,  $(\mathbf{B}_{i_1}) \& (\mathbf{B}_{i_2})$ , que o inteiro resultante  $\mathbf{B}_{i_1 \cap i_2}$  representa nos seus bits exatamente o conjunto  $S_{i_1} \cap S_{i_2}$ .

O termo “*posição*” é ambíguo, mas intencional, não importa se a forma binária do inteiro é encarada como um vetor de bits, “da esquerda para a direita”, ou pelas casas binárias, “direita para a esquerda”. Na implementação deste trabalho, a forma binária do inteiro é utilizada como representação na ordem das casas, portanto, a menor casa  $2^0$  representa o elemento  $e_1 \in \mathcal{R}$  e assim por diante, a casa  $2^{v-1}$  representa  $e_v$ .

### 3.2.2 Representação prática das instâncias e soluções

Com o paralelismo de bits em mente (Subseção 3.2.1), uma instância é representada em Python 3.11.9 por uma Classe (KMIS). Nela são armazenados: Dois inteiros, `tamR` e `k`, que guardam  $|\mathcal{R}|$  e  $k$ , respectivamente; uma lista `L` (de tamanho  $n = |\mathcal{L}|$ ) de inteiros, onde cada inteiro  $\mathbf{B}_i$ ,  $i \in \{0, \dots, n-1\}$  representa em seus bits o subconjunto  $S_{i+1}$  associado.

Com essa estrutura simples, todas as informações da instância são mantidas de fácil acesso. Do conjunto  $\mathcal{R}$ , ter apenas sua cardinalidade é suficiente, pois, uma vez conhecida, cada inteiro em  $\{0, 1, \dots, |\mathcal{R}| - 1\}$  representa implicitamente os elementos de  $\mathcal{R}$ .

Uma solução  $L'$ , completa ou parcial, é representada pela Classe (SOLUCAO). Nessa classe, são definidas duas listas, `L_linha`, de tamanho  $k$ , e `emL`, de tamanho  $|\mathcal{L}|$ . E um inteiro, `lenL_linha`. As listas representam a mesma solução de duas maneiras: A primeira `L_linha` é mais tradicional, contém os índices  $i_0, i_1, \dots, i_{\text{lenL\_linha}}$ , todos distintos e pertencentes a  $\{0, 1, \dots, |\mathcal{L}| - 1\}$ , indicando os subconjuntos associados a cada índice  $i_k + 1$ ; já a segunda `emL`,

é uma lista de *booleanos* com verdadeiro nos índices que estão na solução e falso caso contrário.

O inteiro `lenL_linha` é utilizado para controlar o tamanho efetivo da solução, permitindo soluções parciais sem ser necessário modificar o tamanho real da lista `L_linha`, são consideradas as `lenL_linha` primeiras posições da lista.

A segunda lista, `emL`, é muito utilizada na obtenção dos conjuntos CL (Lista de Candidatos) e EL (Lista de Remoção). Para verificar a inclusão ou não de um  $S_i$  qualquer na solução, basta conferir se `emL[i]` é verdadeiro ou falso, retirando a necessidade de buscar na `L_linha`, ou manter uma CL e EL a todo momento. Essa forma de estruturar a solução é baseada no código fonte de (COSTA *et al.*, 2018).

Durante o desenvolvimento deste trabalho, também foi experimentada a representação com apenas `L_linha`, e usando CL e EL como listas de inteiros separadas, mas o desempenho no quesito tempo era por volta de 25% mais lento, nas menores diferenças, e chegando a ser cerca de 300% mais lento na HG (Subseção 2.2.1), em experimentos preliminares. Então logo a representação citada acima foi adotada.

### 3.3 Propostas do autor

Além da metodologia de avaliação diferente, nesta seção são apresentadas as modificações implantadas no ANT e VND, são referidos como ANT' e VND', visando melhores resultados.

#### 3.3.1 Modificação do ANT

No ANT de (DIAS *et al.*, 2022), o feromônio  $\tau_{ij}$  usado como guia por uma formiga  $t$ , durante a construção do seu caminho, é sempre partindo do último vértice  $S_i$  adicionado a  $A_t$  para o vértice  $S_j \in V_a \setminus A_t$ . Contudo, no kMIS, o último vértice adicionado não determina o “quão boa” é a relação com o  $S_j$ , mas sim o conjunto  $A_t$  inteiro, pois a interseção é feita entre todos. Se qualquer  $S_r \in A_t \setminus \{S_i\}$  tiver um feromônio péssimo com o  $S_j$  e limitar muito a interseção, isso é refletido apenas no fator heurístico guloso  $\eta_j^t$ . Além disso, realizar cinco iterações com todas as formigas ativas, no início, gera um custo de tempo que reduz consideravelmente o número de intensificações total que podem ser realizadas, pois é feita em uma solução por iteração.

São propostas duas modificações para o ANT':

- 1) Quebrar essa ligação do feromônio  $\tau$  com o último vértice que entrou. No lugar de usar

como referência o  $S_i$  adicionado por último, um  $S_q$  é sorteado randomicamente de  $A_t$ , em cada uma das  $k - 1$  etapa da construção, e o feromônio usado, em todos os cálculos para a escolha do próximo  $S_j$ , passa a ser  $\tau_{qj}$ .

- 2) Reduzir e fixar  $G_{reativo}$  para 1, com o objetivo de favorecer mais intensificações em um mesmo tempo. A função de escolha das formigas ativas passa a trabalhar, inicialmente, com o valor da solução obtida na primeira iteração e durante as próximas iterações essas médias se ajustam. Mesmo que uma formiga  $p$  seja prejudicada por ter tido uma solução muito ruim na iteração 1, ela ainda tem chance de ser escolhida nas iterações onde as formigas ativas são totalmente randômicas.

O pseudocódigo presente no Algoritmo 8 mostra as diferenças do  $\text{ANT}'$  para o  $\text{ANT}$ .

---

**Algoritmo 8:** Colônia de Formigas modificada ( $\text{ANT}'$ )

---

**Entrada :**  $I = (\mathcal{L}, \mathcal{R}, k), \alpha, \beta, \rho, q_0, \text{qtd}_{\text{formigas}}, \text{iterMax}, \tau_0$

```

1 :                                     // Igual o  $\text{ANT}$  original
2 Para  $ite \in \{1, \dots, \text{iterMax}\}$  faça:
3    $A_{\text{ativas}} \leftarrow \text{EscolheFormigasAtivas}(ite, 1, M, V_a)$ 
4   Para  $t \in A_{\text{ativas}}$  faça:
5      $A[t] \leftarrow \{S_t\}$ 
6     Enquanto  $|A[t]| < k$  faça:
7        $i \leftarrow \text{escolha\_randomica\_nos\_indices}(A[t])$ 
8       :
9       :
10      :
Saída   :  $L_b$ 

```

---

Com essas modificações na construção, agora a formiga escolhe o próximo subconjunto partindo de qualquer um dos vértices já presentes na sua solução, portanto, não forma uma estrutura de caminho, na verdade, se o percurso da formiga for acompanhado passo a passo, é obtida uma estrutura de árvore.

### 3.3.2 *Modificação do VND*

No VND de 2022 (Subseção 2.3.2) ocorrem 3 tentativas de reconstrução, em grupos de 30% da solução distintos e retorna. Na busca por explorar com mais insistência, duas modificações ocorrem no  $\text{VND}'$ .

No  $\text{VND}'$ :

- 1) Caso encontre uma solução melhor, a solução base passa a ser a nova e todo o processo reinicia, reiniciando os contadores de repetição e a lista de remoções já tentadas  $\Theta$ .
- 2) Um laço mais externo é colocado, repetindo todo o processo do início ao menos 5 vezes, mantendo a atualização da melhor solução. Imaginemos um caso em que o VND não melhore a solução e que temos a mãos os 3 grupos ( $S_1, S_2$  e  $S_3$ ) de 30%, dos subconjuntos da solução que foram sorteados para remoção em cada etapa. Esses 3 grupos não se comunicam, mas uma melhor solução poderia, quem sabe, ser obtida removendo 40% dos subconjuntos presentes em  $S_1$  e 60% presentes em  $S_3$ , por exemplo. A ideia, então, é dar chance a mais combinações de remoção.

O Algoritmo 9 mostra o pseudocódigo dessas diferenças do VND'.

---

**Algoritmo 9:** VND modificado (VND')

---

**Entrada :**  $I = (\mathcal{L}, \mathcal{R}, k), L'$

```

1 :                                         // Igual o VND original
2 insistencia ← 5
3 Enquanto insistencia > 0 faça:
4   |   rep ← 3
5   |    $\Theta$  ←  $\emptyset$                                 //  $\Theta$  é esvaziado
6   |   Enquanto rep > 0 faça:
7   |   |   :
8   |   |   Se  $f(L') > f(L_b)$  então:
9   |   |   |    $L_b \leftarrow L'$ 
10  |   |   |   insistencia ← 5
11  |   |   |   go to 3
12  |   |   insistencia ← insistencia - 1
Saída :  $L_b$ 

```

---

O VND' é utilizado como intensificação do GRASP, ANT e ANT', sendo referidos como GRASP\_VND', ANT\_VND' e ANT'\_VND'.

## 4 RESULTADOS

Este capítulo contém a apresentação dos resultados obtidos em todos os experimentos computacionais. Na Seção 4.1 os resultados do processo de escolha dos parâmetros de cada heurística. Na Seção 4.2 as instâncias obtidas com o processo de Redução. E nas seções 4.3 e 4.4 os resultados das heurísticas executadas nas Instâncias Base e Instâncias Reduzidas, respectivamente.

O experimento ocorreu utilizando um computador AMD Ryzen 5 5600G, a 4.20 GHz, 16 GBytes de memória RAM, no sistema operacional Windows 11 e com as implementações na linguagem Python (3.11.9). Cada chamada de uma heurística foi liberada em um processo diferente, os processos foram alocados pelo próprio sistema operacional em 10 núcleos lógicos do processador. As chamadas ocorreram sempre alternando entre Heurísticas (ou parâmetros, no teste de parâmetros) para que fossem alocadas randomicamente e, assim, que qualquer efeito de redução de performance, por qualquer motivo que seja, tenha impactado todas igualmente, caso tenha ocorrido.

Todos os códigos fonte e instâncias, além das tabelas completas com o resultado de cada chamada de heurística, estão disponíveis no repositório do GitHub em <https://github.com/dariocostta/Monografia-KMIS>, para precisão, caso haja alterações no repositório, a versão deste trabalho fica disponível no commit 9d8cc8c590f9b5dee6a8d99e4df2139d783aa3be.

### 4.1 Seleção dos parâmetros

Ao realizar o procedimento descrito na Subseção 3.1.3, foram avaliados os valores da função score  $\Phi$  das 81 combinações de parâmetros do ANT\_VND e das 27 combinações do GRASP\_TS.

Para 81 combinações de parâmetros do ANT\_VND, na Tabela 9 são apresentadas as seis estatísticas e o score das dez com melhor desempenho.

Podemos observar que os parâmetros mais estáveis foram o  $q_0$ , sendo 0.9 em todas, e  $\beta$ , estando sempre nos dois maiores valores, com favorecimento maior para o 1.2. Já  $\rho$  e  $qtd_{formigas}$  variaram muito, algo que dificulta decidir o melhor parâmetro, mas a metodologia foi seguida e os parâmetros com maior score ( $\alpha = 1.2, \rho = 0.1, q_0 = 0.9, qtd_{formigas} = 0.1$ ) foram fixados para o teste final. Relembrando que a quantidade de formigas ativas é dada por  $\max(2, qtd_{formigas} \cdot |\mathcal{L}|)$ .

Tabela 9 – Os 10 maiores scores dos parâmetros do ANT\_VND

$(\beta, \rho, q_0, \text{qtd}_{\text{formigas}})$	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
(1.2, 0.1, 0.9, 0.1)	7.97	8.25	8.56	28	19	28	99.82
(0.8, 0.1, 0.9, 0.1)	7.97	8.24	8.47	28	18	26	95.80
(1.2, 0.3, 0.9, 0.1)	7.97	8.23	8.53	28	16	27	94.45
(1.2, 0.5, 0.9, 0.3)	7.91	8.25	8.53	26	17	27	94.27
(1.2, 0.3, 0.9, 0.3)	7.94	8.23	8.53	27	16	27	93.52
(1.2, 0.5, 0.9, 0.1)	7.88	8.25	8.50	25	17	26	92.28
(1.2, 0.1, 0.9, 0.2)	7.94	8.26	8.44	27	17	24	92.13
(1.2, 0.3, 0.9, 0.2)	7.91	8.24	8.50	26	15	26	90.45
(0.8, 0.1, 0.9, 0.2)	7.88	8.18	8.53	25	16	26	90.34
(0.8, 0.3, 0.9, 0.1)	7.88	8.21	8.53	25	15	26	89.39

Fonte: Próprio autor

Para as 27 combinações de parâmetros do GRASP\_TS, na Tabela 10 são apresentadas todas as seis estatísticas e o score das dez com o melhor score.

Tabela 10 – Os 10 maiores scores dos parâmetros do GRASP\_TS

$(\alpha, \tau, \gamma)$	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
(0.8, 0.3, 5)	8.18	8.41	8.74	32	22	31	94.63
(0.8, 0.3, 10)	8.15	8.39	8.76	31	19	32	88.87
(0.8, 0.1, 5)	8.15	8.41	8.74	31	19	30	87.12
(0.8, 0.1, 10)	8.21	8.38	8.71	33	17	30	85.18
(0.5, 0.1, 15)	8.15	8.37	8.74	31	19	30	84.01
(0.5, 0.3, 10)	8.15	8.38	8.74	31	18	30	83.24
(0.8, 0.1, 15)	8.15	8.39	8.71	31	17	30	81.45
(0.8, 0.3, 15)	8.18	8.37	8.68	32	18	29	81.02
(0.8, 0.5, 5)	8.18	8.41	8.59	32	21	26	80.13
(0.5, 0.5, 15)	8.12	8.34	8.76	30	17	31	79.01

Fonte: Próprio autor

Para o GRASP\_TS, o parâmetro  $\alpha$  foi o mais conclusivo, sendo os quatro melhores iguais a 0.8, a Construção Randômica e Gulosa ficou o mais gulosa disponível no conjunto. Já  $\tau$  e  $\gamma$  variaram mais, porém se mantendo nos dois menores valores dos conjuntos testados. Os valores ( $\alpha = 0.8, \tau = 0.3, \gamma = 5$ ) foram fixados para o teste de comparação entre heurísticas.

## 4.2 Redução das Instâncias

Ao realizar o procedimento de redução descrito na Seção 2.4, um novo grupo de instâncias foi criado, referido como Instâncias Reduzidas. Em média foram necessários 1.27 segundos de processo para 37.54 remoções de vértices. A Tabela 11 mostra as estatísticas da redução em mais detalhe, nas instâncias agrupadas por classe.

Podemos observar que as classes C1, C5 e C9 são as mais demoradas para reduzir, com um grande número de vértices retirados. As classes C2 e C6 tiveram proporcionalmente

Tabela 11 – Estatísticas da redução por Classe

Classe	Qtd. original		Tempo (segundos)		Vértices removidos			
	I	$ \mathcal{L}  +  \mathcal{R} $	Média	Máximo	Qtd.	Média	Proporção (%)	Total
C1	51	16687	1.73	11.76	<b>85.49</b>	26.1	<b>4360</b>	
C2	2	224	0.05	0.09	76.00	<b>67.9</b>	152	
C4	60	19602	0.30	4.88	4.22	1.3	253	
C5	54	17012	<b>2.92</b>	<b>24.89</b>	78.59	24.9	4244	
C6	2	170	0.02	0.02	47.50	55.9	95	
C7	60	19602	0.19	0.74	0.45	0.1	27	
C8	60	19602	0.53	6.11	5.25	1.6	315	
C9	51	15770	2.41	19.74	65.04	21.0	3317	

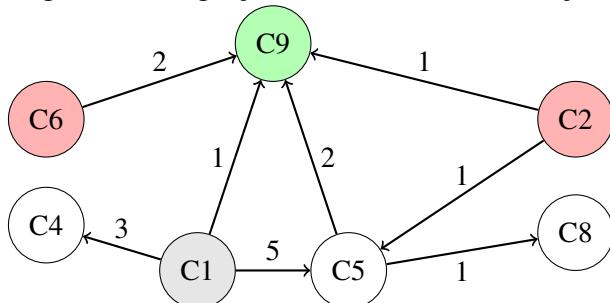
Fonte: Próprio autor

mais vértices retirados, mas eram apenas duas instâncias em cada. As classes C4, C7 e C8 quase não se modificam.

Houve uma migração de classes, entenda como migração a mudança da classe pertencente, principalmente da classe C1 ( $p$  e  $k$  baixos) para a C5 ( $p$  e  $k$  médios), 5 instâncias da C1 passaram a ser da C5 após redução. Algo esperado, já que são removidos os vértices de  $\mathcal{R}$  com vizinhanças menores que  $k$  e de  $\mathcal{L}$  menores que o valor da HG\_Estendida (Subseção 2.2.1), tornando o grafo final mais denso e com um  $k$  proporcionalmente maior em relação a  $|\mathcal{L}|$ .

A Figura 3 apresenta um grafo onde cada arco representa a mudança da classe do vértice de origem para a classe do vértice de chegada, os pesos são a quantidade de instâncias que tiveram essa mudança. Destacados estão C9, a classe que mais aumentou na quantidade de instâncias, C1, a que mais reduziu, e as classes C2 e C6, que passaram a não ter representantes nas Instâncias Reduzidas, zeraram como a C3.

Figura 3 – Migração de Classe com a redução



Fonte: Próprio autor

Após a redução, observou-se que uma densidade  $p_i$  poderia ficar entre os intervalos  $[0.1, 0.3] \cdot |E_K| < p_i < [0.4, 0.6] \cdot |E_K|$ , ficando sem classificação. Para que todas as instâncias reduzidas sejam atribuídas a alguma classe, foram expandidos os limites dos intervalos baixo, médio e alto para abranger todo  $[0, 1]$ . Passando a ser baixo em  $[0, 0.35]$ , médio em  $(0.35, 0.65]$

e alto em  $(0.65, 1]$ , multiplicado pelo respectivo fator  $|\mathcal{L}|$  ou  $|E_K|$ .

O tamanho final de cada Classe é apresentado na Tabela 12.

Tabela 12 – Tamanho das classes antes e após a redução

Classe	Qtd. Antes	Qtd. Depois
C1	51	42
C2	2	0
C3	0	0
C4	60	63
C5	54	57
C6	2	0
C7	60	60
C8	60	61
C9	51	57

Fonte: Próprio autor

As classes C4 e C8 ultrapassaram os 60, que era o máximo no processo de criação definido na Subseção 3.1.1.

### 4.3 Experimento nas Instâncias Base

Os resultados gerais do experimento nas Instâncias Base, sem agrupamento, são expostos na Tabela 13.

A Heurística Gulosa Estendida foi adicionada (HG\_Est), mesmo ela dando a mesma resposta nas 10 execuções por instância, como uma base, ela executa mais rápido que todas as outras, e apesar de ter boas mínimas, suas máximas são bem ruins, reforçando a noção de que escolhas totalmente gulosas resultam em respostas presas a ótimos locais.

Tabela 13 – Estatísticas das Heurísticas nas Instâncias Base

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
GRASP_VND'	<b>5.98</b>	<b>6.13</b>	<b>6.27</b>	<b>303</b>	<b>276</b>	<b>320</b>	<b>100.00</b>
GRASP_VND	5.95	6.10	6.25	291	259	313	84.31
GRASP_TS	5.94	6.07	6.23	292	242	305	75.46
ANT'_VND'	5.84	6.03	6.21	255	211	301	44.57
HG_Est	5.93	5.93	5.93	288	237	235	32.85
ANT'_TS	5.80	6.00	6.21	243	197	300	32.22
ANT_VND'	5.79	5.99	6.17	244	206	288	29.32
ANT_TS	5.79	5.98	6.18	242	201	292	27.49
ANT'_VND	5.81	5.99	6.14	247	201	280	26.90
ANT_VND	5.78	5.96	6.15	240	203	287	22.71

Fonte: Próprio autor

Ressaltando que um score de 22 não significa que a ANT\_VND teve resultados

horríveis, na verdade, em  $\bar{X}_{\text{avg}}$  do menor para o maior varia 0.2005, ou seja, todos os Algoritmos apresentam soluções de qualidade bem próxima. Como o Score, aqui utilizado, é um ponderamento das estatísticas normalizadas na escala Min-Max, fica um pouco extravagante, mas o objetivo é ressaltar as diferenças.

Nas Instâncias Base o GRASP\_VND' teve destaque inquestionável, se saiu melhor em todas as propriedades, na verdade, todos os GRASP. Indicando que quando o tempo de execução do ANT e GRASP são igualados, permitindo que ambos executem por um tempo fixo, o GRASP tem uma boa vantagem.

Analizando os resultados das Instâncias Base agrupados por classes, na Tabela 14, observando apenas a coluna do Score, pois precisariam ser 8 tabelas para visualizar todas as Estatísticas (as tabelas de cada Heurística por Classe estão em Apêndice A).

Tabela 14 – Score das Heurísticas nas Instâncias Base por Classe

Heurística	C1	C2	C4	C5	C6	C7	C8	C9	M. P.
GRASP_VND'	<b>95.71</b>	<b>100.00</b>	92.75	95.38	<b>100.00</b>	<b>100.00</b>	<b>95.90</b>	<b>100.00</b>	<b>96.62</b>
GRASP_VND	69.35	<b>100.00</b>	<b>98.33</b>	<b>96.96</b>	<b>100.00</b>	55.69	94.97	97.28	85.51
GRASP_TS	60.92	<b>100.00</b>	86.64	89.87	<b>100.00</b>	65.36	68.50	68.86	73.83
HG_Est	69.35	<b>100.00</b>	40.38	78.10	<b>100.00</b>	6.75	42.94	35.00	45.13
ANT'_VND'	3.87	<b>100.00</b>	36.16	13.95	<b>100.00</b>	61.30	48.13	74.66	40.86
ANT'_TS	37.14	<b>100.00</b>	25.43	12.24	<b>100.00</b>	57.27	21.81	49.20	34.51
ANT_VND'	10.37	<b>100.00</b>	37.03	28.13	<b>100.00</b>	31.68	39.57	36.64	31.80
ANT'_VND	12.30	<b>100.00</b>	34.58	0.49	<b>100.00</b>	33.87	33.21	51.98	28.84
ANT_TS	10.37	<b>100.00</b>	34.92	14.68	<b>100.00</b>	34.92	33.61	35.90	28.70
ANT_VND	14.29	<b>100.00</b>	38.78	12.00	<b>100.00</b>	24.16	24.84	49.29	28.11

Fonte: Próprio autor

Nota: M.P: Média ponderada pelo tamanho da Classe.

As classes C2 e C6 possuem apenas duas instâncias cada, nas Instâncias Base, e todos as heurísticas encontraram o mesmo valor de solução. Observar-se que o GRASP\_VND apresenta uma certa vantagem nas classes C4 e C5, em relação ao GRASP\_VND', mas nas restantes teve uma pontuação bem abaixo.

#### 4.4 Experimento nas Instâncias Reduzidas

Cada instância rodou com o tempo limite seguindo a função definida como  $t_{\text{lim}}(I) = |\mathcal{L}| \cdot (\frac{10}{300})$ , como proposto na Subseção 3.1.2. As estatísticas das Heurísticas nas Instâncias Reduzidas, sem agrupamentos, na Tabela 15.

Ao observar que as propriedades quantitativas da HG\_Est reduziram nas Instâncias Reduzidas, uma preocupação de existir algum erro surgiu, porém, ao remover vértices  $e_v \in \mathcal{R}$

Tabela 15 – Estatísticas das Heurísticas nas Instâncias Reduzidas

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
GRASP_VND'	<b>6.01</b>	<b>6.13</b>	<b>6.26</b>	<b>316</b>	<b>279</b>	<b>320</b>	<b>100.00</b>
GRASP_VND	5.95	6.11	<b>6.26</b>	296	265	318	82.70
GRASP_TS	5.96	6.09	6.23	298	251	308	75.14
ANT'_VND'	5.86	6.05	6.21	268	222	302	45.89
ANT'_TS	5.85	6.03	6.20	263	214	301	38.74
ANT_VND'	5.85	6.02	6.19	264	220	299	38.70
ANT'_VND	5.83	6.01	6.19	259	216	298	34.40
ANT_TS	5.82	6.00	6.18	256	215	297	31.02
ANT_VND	5.82	5.99	6.16	256	217	289	28.57
HG_Est	5.91	5.91	5.91	285	233	231	20.14

Fonte: Próprio autor

que o  $\lambda$  da solução obtida pela própria HG\_Est indicou, estes vértices não estavam na interseção da solução, mas ao retira-los, uma escolha gulosa intermediaria do HG\_Est pode mudar e isso fazer tomar um caminho para uma solução pior do que a qual obtinha na instância completa. Contudo, isso não é problema algum, as outras heurísticas melhoraram bastante de desempenho nas instâncias reduzidas.

Mesmo com muitas instâncias agora tendo menos tempo para executar, pois o  $|\mathcal{L}|$  reduziu, os valores de mínima  $\bar{X}_{\min}$  e média  $\bar{X}_{\text{avg}}$  tiveram uma melhora, ficando ainda mais próximos entre si, ignorando a HG\_Est, o tamanho do intervalo de diferença entre as  $\bar{X}_{\text{avg}}$  passou a ser 0.1435, uma redução de 30% em relação a Execução nas Instâncias Base.

O processo de redução é bem rápido, em média 1.2 segundos por instância, e faz com que todas as heurísticas superem os valores da Heurística Gulosa Estendida, nas médias e máximas, com muito mais facilidade. Quanto a posição no “ranking” de score a única mudança foi que ANT'\_VND subiu de posição e ultrapassou ANT\_TS nas instâncias reduzidas, mas por uma margem muito pequena.

A Tabela 16 apresenta o Score de cada heurística por classe, sendo agrupado pela classe original das instâncias para facilitar comparação. Os resultados completos no Apêndice A.

Agrupando por classe o GRASP\_VND passou a liderar, por pequena margem, mas no geral posições relativas parecidas. O mais interessante é ver, disponível nos resultados completos em Apêndice A, que a Classe C1 passou a ter quase as mesmas respostas para todas as heurísticas, o que dificultou muito a função  $\Phi$  escolhida ter um resultado relevante, na verdade, ficou bem instável para variações muito pequenas, um ponto negativo da escala Min-Max.

Na Tabela 17 o agrupamento das Instâncias Reduzidas nas suas Classes reais, que passaram a fazer parte depois do processo de Redução.

Tabela 16 – Score das Heurísticas nas Instâncias Reduzidas por Classe

Heurística	C1	C2	C4	C5	C6	C7	C8	C9	M. P.
GRASP_VND	<b>100.00</b>	<b>100.00</b>	<b>97.86</b>	<b>100.00</b>	<b>100.00</b>	55.18	87.55	<b>94.86</b>	<b>88.74</b>
GRASP_VND'	30.00	<b>100.00</b>	90.34	97.70	<b>100.00</b>	<b>100.00</b>	<b>98.67</b>	86.32	85.14
GRASP_TS	30.00	<b>100.00</b>	89.66	95.08	<b>100.00</b>	69.10	53.24	69.23	68.57
ANT'_VND'	70.00	<b>100.00</b>	29.21	25.61	<b>100.00</b>	65.80	34.69	75.31	49.93
ANT_TS	70.00	<b>100.00</b>	38.09	34.14	<b>100.00</b>	31.70	32.75	50.05	42.70
ANT_VND'	30.00	<b>100.00</b>	42.56	43.18	<b>100.00</b>	41.51	31.82	61.55	42.22
ANT'_TS	30.00	<b>100.00</b>	25.45	37.51	<b>100.00</b>	48.76	36.49	68.26	41.41
ANT'_VND	30.00	<b>100.00</b>	18.95	31.28	<b>100.00</b>	37.44	40.56	80.74	39.87
ANT_VND	30.00	<b>100.00</b>	30.08	36.81	<b>100.00</b>	25.21	21.99	85.99	38.06
HG_Est	30.00	<b>100.00</b>	40.08	46.50	<b>100.00</b>	7.55	31.02	0.00	26.94

Fonte: Próprio autor

Nota: M.P: Média ponderada pelo tamanho da Classe.

Tabela 17 – Score das Heurísticas nas Instâncias Reduzidas por Classe Real

Heurística	C1	C4	C5	C7	C8	C9	M. P.
GRASP_VND	<b>100.00</b>	<b>97.86</b>	<b>100.00</b>	55.18	87.55	<b>94.86</b>	<b>88.60</b>
GRASP_VND'	30.00	90.34	97.70	<b>100.00</b>	<b>98.67</b>	86.32	86.64
GRASP_TS	30.00	89.66	95.08	69.10	53.24	69.23	69.61
ANT'_VND'	70.00	29.21	25.61	65.80	34.69	75.31	48.81
ANT_VND'	30.00	42.56	43.18	41.51	31.82	61.55	42.18
ANT'_TS	30.00	25.45	37.51	48.76	36.49	68.26	41.31
ANT_TS	70.00	38.09	34.14	31.70	32.75	50.05	41.29
ANT'_VND	30.00	18.95	31.28	37.44	40.56	80.74	39.88
ANT_VND	30.00	30.08	36.81	25.21	21.99	85.99	38.26
HG_Est	30.00	40.08	46.50	7.55	31.02	0.00	25.83

Fonte: Próprio autor

Nota: M.P: Média ponderada pelo tamanho da Classe.

Todas as colunas das Classes se mantiveram com os mesmos valores em relação a Tabela 16, o que mudou foi a Média Ponderada, pois os tamanhos das classes mudaram, além da retirada das colunas C2 e C6, pois passaram a não ter instâncias. As colunas das classes só se mantiveram com o exato mesmo score, pois todas as 16 instâncias que migraram obtêm o mesmo valor de solução para qualquer uma das heurísticas aplicadas.

## 5 CONCLUSÕES E TRABALHOS FUTUROS

Com base nos resultados obtidos nos experimentos computacionais, verifica-se que o procedimento de Redução (BOGUE *et al.*, 2014) é recomendado e eficiente. De fácil implementação, permite que, em um tempo reduzido, sejam obtidas soluções melhores nas médias e mínimas. As máximas não sofreram uma mudança tão expressiva, mas também positiva.

Das 340 instâncias, somando 108669 vértices, foi possível remover ao todo 12763, equivalente a 11.7% do total. Entender o processo de migração das classes, de antes para depois da redução, também é relevante. As 16 que migraram foram em direção a classes com densidade ou proporção de  $k$ , em relação a  $\mathcal{L}$ , mais alta. As Classes C2 e C6 deixaram de apresentar representantes, como a C3. Dado que essas classes já tinham baixa viabilidade no próprio processo de criação, seria interessante investigar, em estudos futuros, se a ausência de representantes após a redução é um padrão recorrente ou apenas pontual.

Quanto às intensificações propostas, os resultados indicam que o VND' entrega soluções superiores em comparação com as demais abordagens. Ficando à frente do VND padrão em 13 das 15 comparações, quando aplicados como intensificação de uma mesma heurística. E a frente do Busca Tabu em 14 das 15 comparações, nas mesmas condições.

Todas as vezes que o VND' teve Score menor que outra intensificação, aplicada a mesma heurística, ocorreram ao agrupar os resultados das Instâncias Reduzidas por Classe, classe original e classe após redução, onde a função Score escolhida teve muita instabilidade no agrupamento da Classe 1. Dificuldade comentada nas limitações do trabalho.

A modificação ANT' ficou à frente da ANT em 14 das 15 comparações, quando usando a mesma intensificação. Ficando com score menor apenas nas Instâncias Reduzidas por Classe, onde a ANT'\_TS pontuou menos que a ANT\_TS, mas uma diferença de 1.3 pontos da média ponderada dos Scores, apenas.

Na visão geral, quando os tempos de execução são padronizados e todas executam seus laços principais até um tempo  $t_{lim}$ , o GRASP se destaca a frente de forma quase indiscutível, obtendo melhores resultados independentemente da intensificação utilizada. Provavelmente, isso ocorre devido à carga de informações que o ANT precisa atualizar a cada iteração, tanto as informações de cada formiga, quanto de cada aresta, para manter o histórico, o que limita o número de soluções em que a intensificação pode ser aplicada.

Percebo como limitações deste trabalho:

- 1) A linguagem: Acredito que a linguagem possa ter impactado negativamente o desempenho do ANT, uma vez que sua execução envolve a manipulação de um número maior de objetos, o que pode dificultar a atuação do interpretador da linguagem e limitar otimizações internas do próprio Python.
- 2) O tempo limite definido: O tempo limite escolhido, de 10 segundos para a maior instância, deveria ser variado, analisar o comportamento das heurísticas com mais ou menos tempo disponível tornaria as conclusões mais robustas.
- 3) A função Score: Quando o intervalo é muito pequeno nos agrupamentos por classe, como ocorre na classe C1 das Instâncias Reduzidas, a melhora de 1 ponto em uma estatística de contagem aumenta 70 pontos no Score. Essa variação brusca, dando muito mais pontos para uma diferença bem pequena, deve ser corrigida. Talvez a escala Min-Max não tenha sido a melhor escolha, ou poderia retirar essas classes que são tecnicamente um empate, as estatísticas de média só mudam na terceira casa.

Como possibilidade de Trabalhos Futuros, listo:

- 1) A elaboração de uma função Score que consiga lidar com os intervalos curtos entre os valores das soluções e ainda assim fornece informação sobre qual heurística performou melhor.
- 2) Utilizando uma intensificação ágil, como o VND base, obter os resultados realizando a intensificação em todos os Caminhos gerados por cada formiga;
- 3) Propor uma melhor função limitante de tempo de execução  $t_{lim}$ , que utilize o tamanho de  $\mathcal{L}$  e  $k$  juntos, que facilite a comparação entre as heurísticas. E variar esse tempo limite, para analisar o comportamento com diferentes limitantes;
- 4) Paralelismo no GRASP, Colônia de Formigas e VND'. O GRASP e o VND' podem ser totalmente paralelizados, no final apenas comparando para retornar a melhor solução. Já o ANT precisa atualizar os Feromônios, mas ainda é possível paralelizar as construções de caminhos.
- 5) A realização destes testes, com essa metodologia, em uma linguagem de baixo nível, como C ou Java.

## REFERÊNCIAS

BOGUE, E. T.; SOUZA, C. C. de; XAVIER, E. C.; FREIRE, A. da S. O problema da máxima interseção de k-subconjuntos. **ANAIS DO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL**, Natal/RN., 2013.

BOGUE, E. T.; SOUZA, C. C. de; XAVIER, E. C.; FREIRE, A. S. An integer programming formulation for the maximum k-subset intersection problem. Em: FOUILHOUX, P.; GOUVEIA, L. E. N.; MAHJOUB, A. R.; PASCHOS, V. T. (Ed.). **Combinatorial Optimization**. Cham: Springer International Publishing, 2014. p. 87–99.

CARVALHO, R. L. de; ALMEIDA, T. da S.; ROCHA, M. L. **GRASP**. 2020. Em: *Introdução às Metaheurísticas*. ABEC, Palmas, TO, p. 68–74.

CASADO, A.; PÉREZ-PELÓ, S.; SÁNCHEZ-ORO, J.; DUARTE, A. A grasp algorithm with tabu search improvement for solving the maximum intersection of k-subsets problem. **Journal of Heuristics**, v. 28, n. 1, p. 121–146, Feb 2022. ISSN 1572-9397.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Greedy algorithms. Em: **Introduction to Algorithms**. 4th. ed. Cambridge, Massachusetts: The MIT Press, 2022. cap. 15. ISBN 9780262367509.

COSTA, J. R. de F. **Abordagens Heurísticas e Exatas para o Problema da Máxima Interseção de k-Subconjuntos**. Quixadá, CE: Universidade Federal do Ceará, 2021. Monografia, Bacharelado em Engenharia da Computação.

COSTA, J. R. de F.; DIAS, F. C. S.; TAVARES, W. A. Um algoritmo vns para o problema da máxima interseção de k subconjuntos. **ANAIS DO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL**, Rio de Janeiro., 2018.

DIAS, F. C. S.; COSTA, J. R. de F.; TAVARES, W. A. Algoritmo colônia de formigas com uma busca vnd para o problema da máxima interseção de k-subconjuntos. **ANAIS DO SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL**, Campinas : Galoá, 2022.

FEO, T. A.; RESENDE, M. G. A probabilistic heuristic for a computationally difficult set covering problem. **Operations Research Letters**, v. 8, n. 2, p. 67–71, 1989. ISSN 0167-6377.

FOUNDATION, P. S. **CPython – Implementação do arquivo longobject.c (versão 3.11)**. 2024. <https://github.com/python/cpython/blob/3.11/Objects/longobject.c>. Estrutura PyLongObject e operações bit a bit em Python.

PITSOULIS, L.; RESENDE, M. Tabu search. Em: PARDALOS, P.; RESENDE, M. (Ed.). **Handbook of Applied Optimization**. New York: Oxford University Press, 2002. p. 194–209. ISBN 9780195125948.

VINTERBO, S. A. **A Note on the Hardness of the k-Ambiguity Problem**. Boston, MA, USA: Technical Report, Harvard Medical School, 2002.

XAVIER, E. C. A note on a maximum k-subset intersection problem. **Information Processing Letters**, v. 112, n. 12, p. 471–472, 2012. ISSN 0020-0190.

## APÊNDICE A – AGRUPAMENTOS POR CLASSE

### 1) HEURÍSTICAS NAS INSTÂNCIAS BASE

Classe C1 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	1.0980	1.1569	<b>1.2353</b>	45	44	<b>51</b>	37.14
ANT'_VND	1.0980	1.1725	1.2157	45	44	50	12.30
ANT'_VND'	1.0784	1.1686	1.2157	44	43	50	3.87
ANT_TS	1.0980	1.1667	1.2157	45	44	50	10.37
ANT_VND	1.1176	1.1569	1.2157	46	45	50	14.29
ANT_VND'	1.0980	1.1667	1.2157	45	44	50	10.37
GRASP_TS	1.1961	1.2118	1.2157	50	49	50	60.92
GRASP_VND	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	<b>50</b>	50	69.35
GRASP_VND'	1.1961	<b>1.2176</b>	<b>1.2353</b>	50	<b>50</b>	<b>51</b>	<b>95.71</b>
HG_Est	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	<b>50</b>	50	69.35

Fonte: Próprio autor

Classe C2 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT'_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT'_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
HG_Est	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>

Fonte: Próprio autor

Classe C4 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	3.2000	3.4367	3.6833	41	28	56	25.43
ANT'_VND	3.2667	3.4550	3.6333	45	32	53	34.58
ANT'_VND'	3.3000	3.4417	3.6167	47	33	52	36.16
ANT_TS	3.3000	3.4450	3.6000	47	33	51	34.92
ANT_VND	3.3000	3.4417	3.6333	47	34	53	38.78
ANT_VND'	3.2833	3.4567	3.6167	46	34	52	37.03
GRASP_TS	3.4333	3.5383	<b>3.7333</b>	55	44	<b>59</b>	86.64
GRASP_VND	<b>3.4500</b>	<b>3.5900</b>	3.7167	<b>56</b>	<b>49</b>	58	<b>98.33</b>
GRASP_VND'	<b>3.4500</b>	3.5817	3.7000	<b>56</b>	46	57	92.75
HG_Est	3.4333	3.4333	3.4333	55	41	41	40.38

Fonte: Próprio autor

Classe C5 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	0.9630	1.0833	1.2037	40	38	51	12.24
ANT'_VND	0.9630	1.0852	1.1667	40	38	49	0.49
ANT'_VND'	0.9630	1.0963	1.2037	40	38	51	13.95
ANT_TS	0.9630	1.1019	1.2037	40	38	51	14.68
ANT_VND	0.9630	1.0815	1.2037	40	38	51	12.00
ANT_VND'	1.0000	1.1000	1.2222	42	40	52	28.13
GRASP_TS	1.1667	1.2148	<b>1.2593</b>	51	49	<b>54</b>	89.87
GRASP_VND	1.1852	1.2278	<b>1.2593</b>	52	<b>51</b>	<b>54</b>	<b>96.96</b>
GRASP_VND'	1.1667	<b>1.2333</b>	<b>1.2593</b>	51	<b>51</b>	<b>54</b>	95.38
HG_Est	<b>1.2037</b>	1.2037	1.2037	<b>53</b>	<b>51</b>	51	78.10

Fonte: Próprio autor

Classe C6 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT'_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT'_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
HG_Est	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>

Fonte: Próprio autor

Classe C7 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	21.3667	21.6633	22.1167	40	26	44	57.27
ANT'_VND	21.2333	21.5683	21.8667	34	22	33	33.87
ANT'_VND'	21.3500	21.7350	22.1833	39	27	47	61.30
ANT_TS	21.1500	21.5483	22.0167	31	23	41	34.92
ANT_VND	21.1167	21.4383	21.8667	29	23	35	24.16
ANT_VND'	21.1500	21.5483	21.9667	30	23	37	31.68
GRASP_TS	21.4500	21.7467	22.1000	45	27	41	65.36
GRASP_VND	21.3333	21.7183	22.1167	37	26	44	55.69
GRASP_VND'	<b>21.5667</b>	<b>21.8900</b>	<b>22.2833</b>	<b>50</b>	<b>46</b>	<b>52</b>	<b>100.00</b>
HG_Est	21.2333	21.2333	21.2333	33	21	19	6.75

Fonte: Próprio autor

Classe C8 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	4.6333	4.9300	5.1667	33	22	47	21.81
ANT'_VND	4.7167	4.9450	5.1167	38	26	44	33.21
ANT'_VND'	4.7333	4.9767	5.2000	39	28	49	48.13
ANT_TS	4.7000	4.9383	5.1667	37	25	47	33.61
ANT_VND	4.6667	4.9267	5.1500	35	23	46	24.84
ANT_VND'	4.7167	4.9533	5.1667	38	27	47	39.57
GRASP_TS	4.8333	5.0167	5.2000	45	32	49	68.50
GRASP_VND	4.8833	5.0733	<b>5.2833</b>	47	<b>39</b>	<b>54</b>	94.97
GRASP_VND'	4.8833	<b>5.0850</b>	5.2667	48	<b>39</b>	53	<b>95.90</b>
HG_Est	<b>4.9167</b>	4.9167	4.9167	<b>49</b>	33	33	42.94

Fonte: Próprio autor

Classe C9 das Instâncias Base por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	2.1176	2.2627	2.3725	40	35	47	49.20
ANT'_VND	2.1373	2.2451	2.3725	41	35	47	51.98
ANT'_VND'	2.1569	2.2863	2.3922	42	38	48	74.66
ANT_TS	2.0784	2.2431	2.3922	38	34	48	35.90
ANT_VND	2.0980	2.2569	2.3922	39	36	48	49.29
ANT_VND'	2.0980	2.2490	2.3529	39	34	46	36.64
GRASP_TS	2.1569	2.2667	2.3922	42	37	48	68.86
GRASP_VND	<b>2.1961</b>	2.3137	<b>2.4118</b>	<b>44</b>	<b>40</b>	<b>49</b>	97.28
GRASP_VND'	<b>2.1961</b>	<b>2.3353</b>	<b>2.4118</b>	<b>44</b>	<b>40</b>	<b>49</b>	<b>100.00</b>
HG_Est	2.1765	2.1765	2.1765	43	37	37	35.00

Fonte: Próprio autor

## 2) HEURÍSTICAS NAS INSTÂNCIAS REDUZIDAS

Classe C1 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	50	50	30.00
ANT'_VND	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	50	50	30.00
ANT'_VND'	<b>1.2157</b>	1.2176	<b>1.2353</b>	<b>51</b>	50	<b>51</b>	70.00
ANT_TS	<b>1.2157</b>	1.2176	<b>1.2353</b>	<b>51</b>	50	<b>51</b>	70.00
ANT_VND	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	50	50	30.00
ANT_VND'	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	50	50	30.00
GRASP_TS	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	50	50	30.00
GRASP_VND	<b>1.2157</b>	<b>1.2196</b>	<b>1.2353</b>	<b>51</b>	<b>51</b>	<b>51</b>	<b>100.00</b>
GRASP_VND'	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	50	50	30.00
HG_Est	<b>1.2157</b>	1.2157	1.2157	<b>51</b>	50	50	30.00

Fonte: Próprio autor

Classe C2 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
ANT'_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
ANT'_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
ANT_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
ANT_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
ANT_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
GRASP_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
GRASP_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
GRASP_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>
HG_Est	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	<b>2</b>	<b>2</b>	<b>2</b>	<b>100.00</b>

Fonte: Próprio autor

Classe C4 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	3.2333	3.4417	3.6333	43	32	53	25.45
ANT'_VND	3.2167	3.4317	3.6333	42	29	53	18.95
ANT'_VND'	3.2500	3.4767	3.6000	44	32	51	29.21
ANT_TS	3.2833	3.4567	3.6667	46	33	55	38.09
ANT_VND	3.2833	3.4400	3.6167	46	32	52	30.08
ANT_VND'	3.3333	3.4667	3.6000	49	36	51	42.56
GRASP_TS	<b>3.4500</b>	3.5550	3.7167	<b>56</b>	45	58	89.66
GRASP_VND	3.4333	<b>3.5783</b>	<b>3.7500</b>	55	<b>49</b>	<b>60</b>	<b>97.86</b>
GRASP_VND'	3.4333	3.5683	3.7167	55	46	58	90.34
HG_Est	3.4333	3.4333	3.4333	55	41	41	40.08

Fonte: Próprio autor

Classe C5 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	0.9815	1.1519	<b>1.2593</b>	42	39	<b>54</b>	37.51
ANT'_VND	1.0000	1.1296	1.2407	43	40	53	31.28
ANT'_VND'	0.9630	1.1389	1.2407	41	38	53	25.61
ANT_TS	0.9815	1.1593	1.2407	42	40	53	34.14
ANT_VND	1.0000	1.1537	1.2407	43	41	53	36.81
ANT_VND'	1.0185	1.1426	<b>1.2593</b>	44	41	<b>54</b>	43.18
GRASP_TS	<b>1.2037</b>	1.2315	<b>1.2593</b>	<b>54</b>	51	<b>54</b>	95.08
GRASP_VND	<b>1.2037</b>	<b>1.2444</b>	<b>1.2593</b>	<b>54</b>	<b>53</b>	<b>54</b>	<b>100.00</b>
GRASP_VND'	<b>1.2037</b>	1.2389	<b>1.2593</b>	<b>54</b>	52	<b>54</b>	97.70
HG_Est	1.1667	1.1667	1.1667	52	49	49	46.50

Fonte: Próprio autor

Classe C6 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT'_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT'_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
ANT_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_TS	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_VND	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
GRASP_VND'	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>
HG_Est	<b>1.0000</b>	<b>1.0000</b>	<b>1.0000</b>	2	2	2	<b>100.00</b>

Fonte: Próprio autor

Classe C7 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	21.3167	21.6600	22.0167	38	23	43	48.76
ANT'_VND	21.2000	21.5767	21.9833	33	22	40	37.44
ANT'_VND'	21.3833	21.7417	22.1500	42	30	49	65.80
ANT_TS	21.1333	21.5117	21.9167	30	23	40	31.70
ANT_VND	21.0833	21.4483	21.9333	27	22	38	25.21
ANT_VND'	21.1833	21.5917	22.0500	32	23	46	41.51
GRASP_TS	21.4333	21.7800	22.1333	44	31	47	69.10
GRASP_VND	21.3333	21.7300	22.1000	38	25	45	55.18
GRASP_VND'	<b>21.6167</b>	<b>21.8883</b>	<b>22.2167</b>	<b>54</b>	<b>47</b>	<b>53</b>	<b>100.00</b>
HG_Est	21.2333	21.2333	21.2333	33	22	21	7.55

Fonte: Próprio autor

Classe C8 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	4.7500	4.9583	5.2167	41	28	49	36.49
ANT'_VND	4.7500	4.9683	5.2167	41	30	49	40.56
ANT'_VND'	4.7500	4.9650	5.1667	41	29	46	34.69
ANT_TS	4.7667	4.9500	5.1333	42	29	44	32.75
ANT_VND	4.7167	4.9400	5.1333	39	27	44	21.99
ANT_VND'	4.7500	4.9517	5.1667	41	28	46	31.82
GRASP_TS	4.8333	5.0067	5.1833	45	31	47	53.24
GRASP_VND	4.8833	5.0683	<b>5.2833</b>	49	39	<b>53</b>	87.55
GRASP_VND'	<b>4.9500</b>	<b>5.1000</b>	5.2667	<b>52</b>	<b>40</b>	52	<b>98.67</b>
HG_Est	4.9000	4.9000	4.9000	49	32	31	31.02

Fonte: Próprio autor

Classe C9 das Instâncias Reduzidas por Heurística

Heurística	$\bar{X}_{\min}$	$\bar{X}_{\text{avg}}$	$\bar{X}_{\max}$	$\Sigma_{\min}$	$\Sigma_{\text{avg}}$	$\Sigma_{\max}$	$\Phi$
ANT'_TS	2.1961	2.3137	2.3922	44	38	48	68.26
ANT'_VND	2.2157	2.3000	2.4118	45	41	49	80.74
ANT'_VND'	2.2157	2.3098	2.3922	45	39	48	75.31
ANT_TS	2.1373	2.2902	2.4314	41	36	50	50.05
ANT_VND	<b>2.2353</b>	2.3235	2.3922	<b>46</b>	41	48	85.99
ANT_VND'	2.1765	2.2941	2.3922	43	38	48	61.55
GRASP_TS	2.1961	2.2980	2.3922	44	39	48	69.23
GRASP_VND	2.2157	<b>2.3490</b>	<b>2.4510</b>	45	<b>44</b>	<b>51</b>	<b>94.86</b>
GRASP_VND'	<b>2.2353</b>	2.3333	2.4118	<b>46</b>	40	49	86.32
HG_Est	2.0980	2.0980	2.0980	41	35	35	0.00

Fonte: Próprio autor