



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS RUSSAS
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

HUMBERTO CAVALCANTE DAMASCENO

CONSTRUÇÃO E COMPARAÇÃO DE HEURÍSTICAS: ESTUDOS DE CASO
APLICADOS AO PROBLEMA DA DIVERSIDADE MÁXIMA

RUSSAS

2020

HUMBERTO CAVALCANTE DAMASCENO

CONSTRUÇÃO E COMPARAÇÃO DE HEURÍSTICAS: ESTUDOS DE CASO APLICADOS
AO PROBLEMA DA DIVERSIDADE MÁXIMA

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

Orientador: Prof. Dr. Pablo Luiz Braga
Soares

RUSSAS

2020

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

D162c Damasceno, Humberto Cavalcante.
Construção e comparação de heurísticas: Estudos de caso aplicados ao Problema da Diversidade Máxima / Humberto Cavalcante Damasceno. – 2020.
55 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2020.
Orientação: Prof. Dr. Pablo Luiz Braga Soares.

1. Problema da Diversidade Máxima. 2. Heurísticas. 3. Estudos de Caso. 4. Algoritmo Probabilístico. 5. Algoritmo Guloso. I. Título.

CDD 005.1

HUMBERTO CAVALCANTE DAMASCENO

CONSTRUÇÃO E COMPARAÇÃO DE HEURÍSTICAS: ESTUDOS DE CASO APLICADOS
AO PROBLEMA DA DIVERSIDADE MÁXIMA

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

Aprovada em:

BANCA EXAMINADORA

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

Prof. Dr. Marcio Costa Santos
Universidade Federal do Ceará (UFC)

Profa. Ms. Tatiane Fernandes Figueiredo
Universidade Federal do Ceará (UFC)

Dedico este trabalho a toda minha família, em especial a minha mãe Francisca Betânia Cavalcante Damasceno, meu pai Hotebeculos Nogueira Damasceno, meu irmão Nilberto Cavalcante Damasceno, minha avó Maria Cleonice e meu avô Francisco Chagas Sobrinho.

AGRADECIMENTOS

Agradeço a Deus pelo dom da vida, e pela oportunidade que me destes para que eu pudesse seguir o meu caminho com determinação.

Ao Prof. Dr. Pablo Luiz Braga Soares por além de me orientar neste trabalho de conclusão de curso, ter me auxiliado em grandes oportunidades acadêmicas, me alertado quando necessário, e acima de tudo, ser meu amigo.

Aos meus pais, irmão, avós, tios e primos, por todo o amor, confiança e apoio para com a minha pessoa.

Aos amigos que estão comigo desde antes da universidade, em especial Gilberto Lima, Gilson Lima, Hévila Souza, Iago Bezerra, Johnny Marcos, Leiliane Santos, Leonardo Andrade, Marcos Renê, Raul Bezerra, Tálisson Cavalcante e William Chaves, que a muito tempo me acompanham e me apoiam em todas as minhas aventuras de vida.

Aos amigos que a Universidade Federal do Ceará me proporcionou conhecer, em especial Aureliano Neto, Beatriz Oliveira, Carla Kelvia, Emanuel Félix, Gabriel Nogueira, Ismael Jefferson, Ismael Moreira, João Denílson, João Lauro, José Evamberto, Marcílio Feitosa, Marlo Henrique, Rayane Araújo, Reinaldo Linhares, Susana Moreira, Thiago Firmiano e Williana Leite, por estarem sempre comigo durante o árduo período da graduação.

A toda a galera do busão que sai de Morada Nova rumo a Russas todos os dias, pelos momentos descontraídos e pelo companheirismo compartilhado durante esses quatro anos e milhares de quilômetros viajados durante a graduação.

A todos os professores e membros do grupo de pesquisa NEMO, pelo modo descontraído e ao mesmo tempo sério de estudar e encarar as dificuldades, e por todo o conhecimento compartilhado.

A todos os professores, servidores e colaboradores da Universidade Federal do Ceará - Campus Russas, por me proporcionarem o conhecimento acadêmico, por me fazerem evoluir como ser humano e por me prepararem para a vida após a universidade.

Ao Doutorando em Engenharia Elétrica, Ednardo Moreira Rodrigues, e seu assistente, Alan Batista de Oliveira, aluno de graduação em Engenharia Elétrica, pela adequação do *template* utilizado neste trabalho para que o mesmo ficasse de acordo com as normas da biblioteca da Universidade Federal do Ceará (UFC).

“Nunca se esqueça de quem você é, porque é certo que o mundo não se esquecerá. Faça disso sua força. Assim, não poderá ser nunca a sua fraqueza. Arme-se com esta lembrança, e ela nunca poderá ser usada para lhe magoar.”

(Tyrion Lannister, Game of Thrones)

RESUMO

O Problema da Diversidade Máxima (PDM) consiste em determinar, a partir de um conjunto V com n elementos, um subconjunto S de V com m ($m < n$) elementos, de tal forma que a diversidade (que é calculada através da soma das arestas do grafo) entre os pares ordenados $\{x, y\}$ de S seja a maior possível. Este trabalho apresenta implementações de dois algoritmos heurísticos que usam o método de Busca Local (BL) em suas estruturas. A primeira é baseada em probabilidades como critério de seleção e a segunda é fundamentada sobre o princípio guloso. As implementações são testadas em um conjunto de instâncias *benchmark* da literatura e em instâncias geradas por meio de dois estudos de caso que foram introduzidos na Universidade Federal do Ceará — Campus Russas. No melhor do nosso conhecimento, estamos entre os primeiros a realizar uma abordagem comparativa entre algoritmos heurísticos e um método exato acerca do PDM. Os resultados dos experimentos computacionais salientam que as heurísticas desenvolvidas são capazes de resolver o PDM e obter resultados tão bons quanto os obtidos pelo método exato de Soares *et al.* (2018) com as instâncias dos estudos de caso. Em relação as instâncias da literatura, os experimentos evidenciam que os algoritmos desenvolvidos obtiveram, na maioria dos testes realizados, resultados tão bons ou melhores que os atingidos pela Heurística de Soares (HS), que foi usada para fins comparativos.

Palavras-chave: Problema da Diversidade Máxima. Heurísticas. Estudos de Caso. Algoritmo Probabilístico. Algoritmo Guloso.

ABSTRACT

The Maximum Diversity Problem (PDM) consists of determining, from a set V with n elements, a subset S of V with m ($m < n$) elements, in such a way that the diversity (which is calculated by adding the edges of the graph) between the ordered pairs $\{x, y\}$ of S is as large as possible. This work presents implementations of two heuristic algorithms that use the Local Search (BL) method in their structures. The first is based on probabilities as a selection criterion and the second is based on the greedy principle. The implementations are tested on *benchmark* instances in the literature and on instances generated by two case studies that were introduced at the Federal University of Ceará - Campus Russas. To the best of our knowledge, we are among the first to carry out a comparative approach between heuristic algorithms and an exact method about PDM. The results of the computational experiments highlight that the developed heuristics are able to solve the PDM and obtain results as good as those obtained by the exact method of Soares *et al.* (2018) with the instances of the case studies. In relation to the instances of the literature, the experiments show that the developed algorithms obtained, in most tests performed, results as good or better than those achieved by the Soares Heuristic (HS), which was used for comparative purposes.

Keywords: Maximum Diversity Problem. Heuristics. Exact Methods. Probabilistic Algorithm. Greedy Algorithm.

LISTA DE FIGURAS

Figura 1 – Representação de grafo não orientado, subgrafo e clique.	20
Figura 2 – Fluxograma da metodologia de pesquisa.	28
Figura 3 – Comparação do tempo de execução do primeiro Estudo de Caso.	41
Figura 4 – Comparação do tempo de execução do segundo Estudo de Caso.	43
Figura 5 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares <i>et al.</i> (2018) com as instâncias intermediárias do grupo GKD.	46
Figura 6 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares <i>et al.</i> (2018) com as instâncias robustas do grupo GKD.	47
Figura 7 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares <i>et al.</i> (2018) com as instâncias do grupo MDG com $n = 500$ e $m = 50$	47
Figura 8 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares <i>et al.</i> (2018) com as instâncias do grupo MDG com $n = 2000$ e $m = 200$	48
Figura 9 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares <i>et al.</i> (2018) com as instâncias do grupo SOM.	48

LISTA DE TABELAS

Tabela 1 – Comparação desta pesquisa em relação ao estado da arte.	27
Tabela 2 – Execução da instância do primeiro estudo de caso.	41
Tabela 3 – Execução da instância do segundo estudo de caso.	43
Tabela 4 – Execução das instâncias mais simples do grupo GKD.	49
Tabela 5 – Execução das instâncias mais robustas do grupo GKD.	50
Tabela 6 – Execução das instâncias dos grupos MDG e SOM.	51

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo Guloso genérico	23
Algoritmo 2 – Heurística Roleta Russa	33
Algoritmo 3 – Heurística Senhor Barriga	35
Algoritmo 4 – Algoritmo de Busca Local	37

LISTA DE ABREVIATURAS E SIGLAS

ABO	Aprendizagem Baseada em Oposição
AG	Algoritmo Guloso
AM	Algoritmo Memético
BL	Busca Local
BT	Busca Tabu
EUA	Estados Unidos da América
H-RR	Heurística Roleta Russa
H-SB	Heurística Senhor Barriga
HS	Heurística de SOARES
ONP	Otimização por Nuvem de Partículas
PDM	Problema da Diversidade Máxima
PL	Programação Linear
PLI	Programação Linear Inteira
PO	Pesquisa Operacional
TAE	Técnico-Administrativo em Educação

LISTA DE SÍMBOLOS

\subseteq	É subconjunto próprio de
\in	Pertence
\notin	Não pertence
\neq	Diferente de
<i>s.a</i>	Sujeito à
<i>max</i>	Maximizar
<i>min</i>	Minimizar
\geq	Maior igual
\forall	Para todo
$<$	Menor que
\cup	União

SUMÁRIO

1	INTRODUÇÃO	16
1.1	Organização do Trabalho	17
2	OBJETIVOS	18
2.1	Objetivo Geral	18
2.2	Objetivos Específicos	18
3	FUNDAMENTAÇÃO TEÓRICA	19
3.1	Conceitos Importantes sobre Teoria dos Grafos	19
3.2	Conceitos Importantes sobre Pesquisa Operacional	20
3.2.1	<i>Programação Linear</i>	20
3.2.2	<i>Programação Linear Inteira</i>	21
3.2.3	<i>Métodos Heurísticos</i>	22
3.2.4	<i>Procedimento de Busca Local</i>	22
3.2.5	<i>Algoritmos Gulosos</i>	23
3.3	Problema da Diversidade Máxima	24
3.3.1	<i>Aplicações</i>	24
4	TRABALHOS RELACIONADOS	26
5	METODOLOGIA	28
5.1	Definição do Contexto de Aplicação	28
5.2	Elaboração de Questionários e Coleta dos Dados	29
5.3	Construção dos Grafos	29
5.4	Construção das Heurísticas e Adequação do Método exato de Soares <i>et al.</i> (2018)	32
5.5	Execução e Estudos Comparativos	32
6	ALGORITMOS PROPOSTOS	33
6.1	Heurística Roleta Russa	33
6.2	Heurística Senhor Barriga	35
6.3	Método de Busca Local	37
7	EXPERIMENTOS COMPUTACIONAIS	38
7.1	Configuração do Ambiente Computacional	38
7.2	Características das Instâncias	38

7.2.1	<i>Instâncias da literatura</i>	38
7.2.2	<i>Instâncias dos estudos de caso</i>	39
7.3	Maximização da Formação dos Monitores da Sala de Aula	40
7.4	Diversificação do Comitê dos Professores/Servidores	42
7.5	Execução das Instâncias da Literatura	44
7.5.1	<i>Instâncias Mais Simples do Grupo GKD</i>	44
7.5.2	<i>Instâncias Intermediárias do Grupo GKD</i>	44
7.5.3	<i>Instâncias do Grupo MDG e SOM</i>	45
8	CONCLUSÕES E TRABALHOS FUTUROS	52
8.1	Trabalhos Futuros	53
	REFERÊNCIAS	54

1 INTRODUÇÃO

O Problema da Diversidade Máxima (PDM) pertence à família dos problemas de dispersão e diversidade que, devido ao seu amplo domínio de aplicações, dentre elas: genética (AGRAFIOTIS, 1997); locação (ERKUT; NEUMAN, 1991); preservação ecológica (GLOVER *et al.*, 1995); e gerenciamento de recursos humanos (DHIR *et al.*, 1993) etc, vem estimulando o interesse de inúmeros pesquisadores nas áreas de otimização combinatória (KUO *et al.*, 1993; GLOVER *et al.*, 1995; AGRAFIOTIS, 1997; ZHOU *et al.*, 2017) com o objetivo de desenvolver algoritmos, sejam eles exatos ou heurísticos, para solucionar o PDM em tempo viável e com obtenção de resultados significativos.

Existem limitações sobre o uso da abordagem exata para a resolução do PDM. Por ser um problema NP-Difícil (KUO *et al.*, 1993), resolvê-lo de forma ótima é considerado um desafio computacional, até mesmo para instâncias de tamanho moderado, com cerca de 125 vértices (SOARES *et al.*, 2018). Para tratar instâncias mais robustas, é necessário recorrer ao uso de algoritmos que forneçam resultados próximos da solução ótima, tais como heurísticas ou meta-heurísticas (BONOTTO; CABRAL, 2016; ZHOU *et al.*, 2017).

Neste trabalho, foram desenvolvidas duas heurísticas que fazem uso do procedimento de Busca Local (BL) em suas estruturas para resolver o PDM, sendo a primeira baseada em probabilidades como critério de solução e a segunda fundamentada sobre o princípio guloso. As Heurísticas foram comparadas através de instâncias *benchmark* disponíveis na literatura e também por meio de instâncias obtidas por intermédio de dois estudos de caso, que foram introduzidos na Universidade Federal do Ceará — Campus Russas. O primeiro estudo tem o intuito de maximizar a formação dos monitores da turma de Processos de Software. Já o segundo, visa a diversificação da formação de um grupo de representantes da universidade, formado por docentes e técnicos administrativos do campus.

O método exato de Soares *et al.* (2018) foi utilizado para obter a solução ótima dos estudos de caso. Além disso, uma análise comparativa, até então pouco utilizada na literatura, foi desenvolvida. O estudo consiste na comparação entre os resultados gerados pelas heurísticas desenvolvidas na pesquisa e o valor ótimo obtido pelo método de Soares *et al.* (2018), a fim de verificar a viabilidade no uso de métodos heurísticos para a resolução do PDM.

A metodologia adotada para o desenvolvimento da pesquisa consiste em 5 fases sequenciais, que foram definidas da seguinte forma: A primeira etapa consiste na escolha do contexto de aplicação do PDM. A segunda foca na elaboração dos questionários e coleta dos dados necessários para a formulação das instâncias dos estudos de caso. Na terceira fase, os grafos foram formulados de acordo com os dados coletados na etapa anterior. O desenvolvimento dos algoritmos heurísticos e a adequação do método exato de Soares *et al.* (2018) foram realizados na quarta fase. Por fim, a quinta etapa consiste na execução das instâncias dos estudos de caso e das instâncias *benchmark* da literatura juntamente com a realização dos devidos estudos comparativos.

1.1 Organização do Trabalho

Este trabalho se divide em 8 capítulos. O Capítulo 1 introduz a problemática de forma breve, juntamente com a importância do problema abordado na pesquisa. Os objetivos esperados no desdobramento do trabalho são retratados no Capítulo 2. O Capítulo 3 apresenta a definição dos conceitos essenciais para a compreensão e resolução do problema, juntamente com a definição matemática do PDM. O Capítulo 4 descreve o estado da arte presente na literatura. O Capítulo 5 apresenta a metodologia utilizada para o desenvolvimento da pesquisa. O capítulo 6 apresenta e detalha os algoritmos que foram desenvolvidos neste trabalho. O Capítulo 7 descreve os resultados que foram obtidos na pesquisa. Por fim, as conclusões da pesquisa e os trabalhos futuros são descritos no Capítulo 8.

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver dois algoritmos heurísticos para resolução e aplicação de dois estudos de caso do Problema da Diversidade Máxima na Universidade Federal do Ceará — Campus Russas. O primeiro propôs a maximização da formação dos monitores da turma de processo de software, do curso de Engenharia de Software, e o segundo a diversificação do comitê dos professores e técnicos da universidade.

2.2 Objetivos Específicos

- Gerar as instâncias dos estudos de caso;
- Obter o valor ótimo das instâncias geradas através do método exato de Soares *et al.* (2018);
- Analisar as soluções obtidas pelas heurísticas produzidas com as instâncias dos estudos de caso;
- Comparar as soluções geradas pelas heurísticas desenvolvidas com a heurística elaborada por Soares *et al.* (2018) em relação as instâncias *benchmark* disponíveis na literatura;
- Fomentar a literatura com uma nova abordagem comparativa acerca do PDM.

3 FUNDAMENTAÇÃO TEÓRICA

Para a compreensão do problema abordado na pesquisa e sobre como será modelada a solução para tal, é necessária a assimilação de alguns conceitos básicos nas áreas de Teoria dos Grafos e Pesquisa Operacional, que serão apresentados nas Seções 3.1 e 3.2 respectivamente. A Seção 3.3 aborda a definição matemática do PDM juntamente com uma parte das suas áreas de aplicação.

3.1 Conceitos Importantes sobre Teoria dos Grafos

Segundo Prestes (2016), um grafo $G = (V(G), E(G))$ é composto de um conjunto finito não vazio $V(G)$ de elementos, chamados vértices, e um conjunto de pares não-ordenados de elementos distintos de $V(G)$, chamado arestas, isto é, $E(G) \subseteq \{\{u, v\} | u, v \in V(G), u \neq v\}$. Um grafo é chamado de completo se e somente se, todos os seus vértices possuírem adjacência entre si, ou seja, a partir de um vértice de $V(G)$, este possui uma aresta entre ele e todos os outros vértices de $V(G)$ (CORMEN *et al.*, 2012).

De acordo com Szwarcfiter (1988), seja um grafo $G = (V(G), E(G))$, dizemos que o grafo $G' = (V'(G'), E'(G'))$ é subgrafo de G se e somente se:

1. $V'(G') \subseteq V(G)$;
2. $E'(G') \subseteq E(G)$.

Um grafo $G = (V(G), E(G))$ é caracterizado como direcionado se, para cada par ordenado de vértices $\{x, y\} \in E(G)$, determina um arco com origem em x e destino em y , ou seja $\{x, y\} \neq \{y, x\}$. Quando não existe direção no par ordenado $\{x, y\}$, o grafo é denominado como não direcionado, isto é, $\{x, y\} = \{y, x\}$ (PRESTES, 2016).

Para Prestes (2016), uma clique em um grafo G , é um subgrafo completo de G , o que implica que todos os vértices na clique possuem adjacência entre si. Uma clique é maximal quando não for um subgrafo próprio de outro subgrafo completo. Na Figura 1 serão apresentados os principais grafos que serão utilizados no decorrer deste trabalho, onde G representa um grafo completo e não orientado, G' equivale a um subgrafo não orientado de G , e G'' corresponde a uma clique não orientada de G .

Figura 1 – Representação de grafo não orientado, subgrafo e clique.



Fonte: Elaborado pelo autor (2019).

3.2 Conceitos Importantes sobre Pesquisa Operacional

Segundo Hillier e Lieberman (2013) a Pesquisa Operacional (PO) é um ramo interdisciplinar da matemática aplicada que recorre ao uso de modelos matemáticos, estatísticos e algoritmos para maximizar a tomada de decisões. Seu uso é de fundamental importância para a análise de problemas complexos do mundo real, com o objetivo de melhorar e/ou otimizar o desempenho. Os problemas determinísticos de PO podem ser apontados em categorias genéricas: programação linear; simulação; teoria dos jogos; programação dinâmica; programação inteira e programação não-linear. Existem ocorrências em que os modelos de PO são tão substanciais, que se tornam impraticáveis na busca por uma solução ótima. Nesse caso, normalmente utiliza-se métodos heurísticos para alcançar ao menos uma boa solução viável.

3.2.1 Programação Linear

Para Ferreira *et al.* (2001) a Programação Linear (PL) consiste em maximizar ou minimizar uma função linear estabelecida sobre um complexo de variáveis, que devem satisfazer um sistema de equações e inequações lineares. A Programação Linear denota algoritmos profundamente eficientes que podem ser resolvidos utilizando o computador, a PL é um método que desfruta de instrumentos matemáticos que permitem a otimização de operações, e é abordada na maioria dos casos na resolução de problemas que possuam seus modelos representados através de expressões regulares (GOLDBARG; LUNA, 2000).

Segundo Garcia *et al.* (1997) matematicamente, pode-se formular o seguinte modelo de um problema de otimização linear seguindo a seguinte estrutura:

$$\begin{aligned}
 \text{max ou min: } Z &= c_1x_1 + c_2x_2 + \dots + c_nx_n \\
 \text{s.a} \quad &a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n < b_1 \\
 &a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n < b_2 \\
 &\vdots \\
 &a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n < b_m \\
 &x_i \geq 0 \quad \forall i = 1, 2, \dots, n \\
 &b_j \geq 0 \quad \forall j = 1, 2, \dots, m
 \end{aligned}$$

onde:

Z é a função a ser maximizada ou minimizada, respeitando o conjunto de elementos do problema;

x_i são as variáveis que retratam as quantidades ou recursos que serão determinados para otimizar o resultado global;

c_i são os coeficientes de ganho ou custo que cada variável é capaz de gerar;

b_j são a quantidade disponível de cada recurso;

a_{ij} são a quantidade de recurso que cada variável decisória consome.

3.2.2 Programação Linear Inteira

Os problemas de Programação Linear Inteira (PLI) são reconhecidos pelo fato de todas as suas variáveis possuírem valores inteiros. Caso apenas uma parte das variáveis sejam inteiras, o método passa a ser caracterizado como Programa Linear Inteiro Misto. Outra restrição peculiar, é quando as variáveis são inteiras e possuem seus valores restritos a 0 e 1, o que caracteriza um programa Linear Inteiro Binário (CORMEN *et al.*, 2012).

3.2.3 Métodos Heurísticos

Segundo Hillier e Lieberman (2013) um método heurístico é um procedimento que visa encontrar uma boa solução viável, o que não necessariamente será uma solução ótima para o problema que ela está aplicada. Não é possível garantir a qualidade do resultado obtido, entretanto, um método heurístico normalmente é capaz de disponibilizar uma solução ao menos próxima da ótima. O procedimento deve ser suficientemente eficaz de lidar com problemas muito grandes. O método geralmente é um algoritmo iterativo completo onde cada iteração envolve a condução de uma pesquisa por uma nova solução que, por ventura, venha a ser superior que a melhor solução encontrada anteriormente. Quando o algoritmo termina, o resultado retornado por ele é o melhor que foi encontrado durante o período de iterações.

3.2.4 Procedimento de Busca Local

Segundo Johnson (1990), os Algoritmos de Busca Local (BL), também conhecidos como Busca de Vizinhança, definem, para cada solução, uma vizinhança composta por um conjunto de soluções com características semelhantes. Dada uma solução corrente, uma das formas de implementar um algoritmo de BL consiste em percorrer a vizinhança dessa solução em busca de outra com valor menor (quando se lida com problemas de minimização), ou maior (quando se trata de problemas de maximização). Se tal solução vizinha for encontrada, torna-se a nova solução corrente e o algoritmo prossegue com sua execução. Caso contrário, a solução corrente é um ótimo local em relação a vizinhança empregada. Dependendo do explorado, métodos de BL tendem a ficar presos em ótimos locais, não gerando, conseqüentemente, uma solução ótima global para um determinado problema.

3.2.5 Algoritmos Gulosos

Um Algoritmo Guloso (AG) toma decisões com base apenas nas informações disponíveis, sem se preocupar com os efeitos futuros, isto é, ele nunca reconsidera as decisões tomadas, independente das consequências. Não há, portanto, a necessidade de avaliar as alternativas e nem empregar procedimentos elaborados permitindo que decisões anteriores sejam desfeitas. Devido a essas características, esse tipo de algoritmo é considerado fácil de ser formulado e implementado, sendo eficiente quando funciona corretamente (BRASSARD; BRATLEY., 1995). A seguir será apresentado o pseudocódigo do modelo genérico de um AG:

Algoritmo 1: Algoritmo Guloso genérico

Input: Conjunto de candidatos: C

Output: Melhor solução obtida

```

1 begin
2   result  $\leftarrow$  0
3   while  $C \neq 0$  and not solução(result) do
4     x  $\leftarrow$  seleciona C
5     C  $\leftarrow$  C  $\setminus$  {x}
6     if É viável result  $\cup$  {x} then
7       result  $\leftarrow$  result  $\cup$  {x}
8     end
9   end
10  if solução(result) then
11    Retorne result
12  else
13    Retorne "Não existe uma solução"
14  end
15 end

```

3.3 Problema da Diversidade Máxima

De acordo com Zhou *et al.* (2017) o Problema da Diversidade Máxima faz parte de uma família de problemas de dispersão e diversidade, cujo objetivo é determinar, a partir de um conjunto V com n elementos, um subconjunto S de V com m ($m < n$) elementos, de tal forma que a diversidade entre os pares ordenados $\{x, y\}$ de S seja a maior possível. O PDM pode ser conceituado sobre um grafo não direcionado $G = (V, E)$, com arestas ponderadas, onde $V = \{1, \dots, n\}$ e $c_{ij} \in \mathbb{R}_+$ denota o peso da aresta $\{i, j\}$. Segundo Kuo *et al.* (1993), o PDM pode ser formulado como o seguinte problema quadrático binário:

$$\begin{aligned} \max \quad & \sum_{i=1}^{n-1} \sum_{j=i+1}^n c_{ij} x_i x_j \\ \text{s.a.} \quad & \sum_{i=1}^n x_i = m, \\ & x_i \in \{0, 1\} \quad 1 \leq i \leq n, \end{aligned}$$

onde a variável binária $x_i = 1$ se o vértice $i \in S$, e $x_i = 0$ se $i \notin S$. A equação de cardinalidade garante que uma solução viável x contém exatamente m vértices.

3.3.1 Aplicações

Na indústria farmacêutica, um problema que ocorre frequentemente, é a seleção de um subconjunto de componentes químicos para síntese e avaliação biológica. O principal objetivo é maximizar a diversidade molecular seguindo um conjunto de critérios preestabelecidos. Agrafiotis (1997) desenvolveu uma família de algoritmos que codifica qualquer critério de seleção almejado, sendo capaz de obter as mais diversas categorias de componentes químicos.

Na área de preservação ecológica, esforços vem sendo realizados para promover a preservação das espécies que correm risco de extinção, no entanto, os recursos disponíveis são limitados. A partir desse ponto, surge a importância de gerenciar os recursos da melhor maneira possível, a fim de atender as necessidades de todas as espécies. Glover *et al.* (1995) desenvolveram um modelo de programação quadrática 0 – 1 com o intuito de maximizar a biodiversidade sujeita a restrição de recursos.

Na área de gerenciamento de recursos humanos, um dos principais desafios a ser resolvido, é o gerenciamento de portfólios. Dhir *et al.* (1993) abordaram um exemplo sobre a análise de seguros, que tem como objetivo buscar a minimização dos riscos de uma organização de acordo com a variedade, quantidade e qualidade dos investimentos realizados. O PDM tem alto índice de desempenho quando aplicado a esse tipo de problema.

Na área de locação, Erkut e Neuman (1991) estudaram sobre a dificuldade de designação dos cargos para os profissionais de uma empresa, de modo que cada empregado seja direcionado para a sua área de maior identidade, com o intuito de maximizar a agilidade, produtividade e renda da empresa. Uma das formas de resolver esse problema de maneira satisfatória, seria através da aplicação do PDM.

4 TRABALHOS RELACIONADOS

Neste Capítulo, será apresentado o estado da arte presente na literatura, que engloba trabalhos que estudaram a gama de aplicações do Problema da Diversidade Máxima, e pesquisas que desenvolveram algoritmos para resolução do PDM, sejam elas através de métodos heurísticos ou algoritmos exatos. Também será disponibilizada uma tabela para nível de comparação entre os trabalhos abordados neste Capítulo e este trabalho, afim de especificar o diferencial desta pesquisa em relação ao estado da arte.

McConnell (2013) realizou estudos sobre o incessante crescimento da imigração nos Estados Unidos da América (EUA), e a forma de como o País deseja selecionar os mais “aptos” a passar pela fronteira. Na pesquisa são listadas inúmeras características que são levadas em consideração para a tomada de decisão, tais como: o drama emocional, os laços familiares, a naturalidade e a mão-de-obra qualificada. O objetivo é aperfeiçoar o programa de imigração dos EUA, a fim de atender e satisfazer a todos os critérios listados anteriormente. Uma forma de solucionar esse problema, é através da aplicação do Problema da Diversidade Máxima, mapeando os candidatos a vaga no programa, e a partir daí, selecionar os que possuem a maior diversidade de acordo com os parâmetros do algoritmo.

Bonotto e Cabral (2016) propuseram um algoritmo híbrido que faz uso da meta-heurística Otimização por Nuvem de Partículas (ONP), que simula o comportamento de um bando de pássaros em voo com seu movimento localmente aleatório, mas globalmente determinado, afim de encontrar ótimos locais, e da meta-heurística Busca Tabu (BT), que é um procedimento adaptativo que faz uso de uma estrutura de memória para acumular as soluções geradas, para direcionar um algoritmo de busca local. O método proposto pelos autores, obteve resultados satisfatórios, já que para todas as instâncias testadas, foram geradas as melhores soluções encontradas na literatura.

Zhou *et al.* (2017) produziram um Algoritmo Memético (AM), que é uma abordagem de pesquisa híbrida que combina pesquisa e otimização local, capaz de resolver problemas NP-difícil, a fim de solucionar o Problema da Diversidade Máxima. O AM faz uso da Aprendizagem Baseada em Oposição (ABO), que engloba uma variedade de novos modelos teóricos e métodos técnicos para lidar com problemas de otimização, a fim de reforçar a diversidade populacional e aperfeiçoar a pesquisa evolutiva. Como resultados computacionais, os autores obtiveram resultados aprimorados para 22 dos 80 casos mais populares do *benchmark* PDM.

Soares *et al.* (2018) desenvolveram um algoritmo exato, que usa uma heurística gulosa com busca local para gerar a solução inicial, para a resolução do Problema da Diversidade Máxima. Para a construção do método, os autores revisaram a formulação quadrática do PDM para aplicar a técnica da *t-linearização* e revigorar as restrições resultantes. Após isso, novas restrições válidas para o problema foram propostas, para que a partir desse ponto, o algoritmo pudesse ser desenvolvido embasado-se no método *branch-and-bound*. Os resultados computacionais obtidos neste trabalho foram considerados satisfatórios, em consequência de o método ser capaz de resolver o problema com instâncias de até 125 elementos, e ser mais eficiente que o melhor algoritmo exato disponível na literatura. Mediante aos bons resultados analisados, o método exato de Soares *et al.* (2018) será usado nesta pesquisa.

A seguir, será apresentada a Tabela 1, que expõe o que cada autor desenvolveu em seu trabalho. Vale ressaltar que o X vermelho simboliza o diferencial desta pesquisa em relação ao estado da arte.

Tabela 1 – Comparação desta pesquisa em relação ao estado da arte.

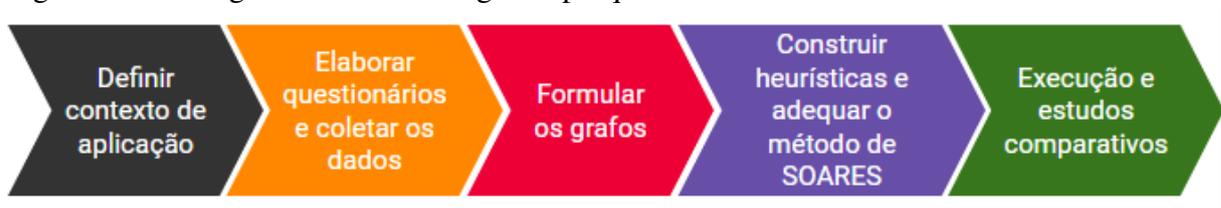
Lista de atividades	Trabalhos comparados				
	MCconnell (2013)	Cabral (2016)	Zhou (2017)	Soares (2018)	TCC Humberto
Desenvolver método heurístico		X	X	X	X
Desenvolver método exato				X	
Aplicação do PDM em ambiente real	X		X		X
Estudo comparativo entre heurísticas		X	X		X
Estudo comparativo entre métodos exatos				X	
Estudo comparativo entre heurística e método exato para resolver o PDM					X

Fonte: Elaborada pelo autor (2020)

5 METODOLOGIA

Este Capítulo descreve as cinco etapas que foram utilizadas para a realização deste trabalho: A escolha do contexto de aplicação do PDM; a elaboração dos questionários para coletar os dados; a geração das instâncias dos estudos de caso com base nos dados coletados; a construção de duas heurísticas e a adequação do método proposto por Soares *et al.* (2018) para resolver o PDM; por fim, a execução dos algoritmos e a realização dos estudos comparativos. A Figura 2 que será exibida a seguir, retrata a ordem das etapas realizadas neste trabalho.

Figura 2 – Fluxograma da metodologia de pesquisa.



Fonte: Elaborado pelo autor (2020).

5.1 Definição do Contexto de Aplicação

Devido ao amplo domínio de aplicações do PDM, os estudos de caso foram formulados para solucionar dois problemas presentes na própria universidade. O primeiro tem como objetivo maximizar a formação dos monitores da turma de processos de software, com o intuito de melhorar o nível de aprendizagem dos alunos, de modo que os selecionados, sejam os mais qualificados para trabalhar em conjunto e repassar o conteúdo aos demais estudantes, aumentando assim o rendimento da sala de aula. Já o segundo estudo de caso visa diversificar a formação do conselho dos professores e técnicos administrativos da faculdade, fazendo com que os selecionados sejam os mais discrepantes, para que possam buscar recursos para todas as áreas de pesquisa do campus, melhorando a infraestrutura e a reputação da universidade.

5.2 Elaboração de Questionários e Coleta dos Dados

Para a formulação dos grafos gerados a partir dos estudos de caso propostos, dois questionários foram elaborados e aplicados na universidade, com o intuito de coletar as características que serviram de base para a ponderação das arestas das instâncias que foram executadas na pesquisa. Para o problema da maximização da formação dos monitores da turma de Processos de Software, os dados que foram levados em consideração para a formulação das instâncias foram:

- O conhecimento prévio da disciplina (de 1 a 10);
- A idade do aluno;
- A cidade que o aluno reside atualmente;
- O semestre do aluno;
- O IRA individual do discente.

Já para o problema da diversificação da formação do comitê dos professores e servidores Técnico-Administrativo em Educação (TAE) do campus, as informações que foram levados em consideração para a formulação das instâncias foram:

- O cargo na universidade (Docente ou TAE);
- A idade do respondente;
- A cidade que o servidor reside atualmente;
- A formação acadêmica;
- A titulação (Especialista, Técnico, Graduado, Mestre ou Doutor);
- O tempo de experiência profissional (medido em anos).

5.3 Construção dos Grafos

Após a coleta dos dados, foi realizada uma adaptação na formulação matemática criada por Bonotto e Cabral (2016), para determinar o valor de todas as arestas de um grafo, de acordo com características que foram estabelecidas previamente. Vale ressaltar que os grafos que foram usados na pesquisa, são não orientados e completos.

Para o problema da maximização da formação dos monitores da sala de aula, a formulação de Bonotto e Cabral (2016) foi adaptada da seguinte forma:

$$d_{xy} = \sqrt{\sum_{k=1}^n |d_{xk} + d_{yk}|^2},$$

onde

1. d_{xy} é peso da aresta $\{xy\}$;
2. n é a quantidade de características levadas em consideração;
3. $|d_{xk} + d_{yk}|$ é o módulo da soma dos atributos definidos anteriormente, entre x e y .

Desse modo, quanto mais características em comum x e y possuírem, maior será o valor da aresta que os interliga. A seguir será apresentado um exemplo fictício sobre como funciona o cálculo do peso da aresta entre dois vértices para este estudo de caso. Sejam os vértices:

- **Vértice x:** Um aluno de 20 anos de idade, com IRA 7.0, residente na cidade de Russas, cursando o 2º semestre e com nota 9 no conhecimento prévio da disciplina;
- **Vértice y:** Um aluno de 18 anos de idade, com IRA 6.5, residente na cidade de Morada Nova, cursando o 6º semestre e com nota 6 no conhecimento prévio da disciplina.

$$d_{xy} = \sqrt{|9 + 6|^2 + |9|^2 + |60|^2 + |8|^2 + |7.0 + 6.5|^2},$$

onde

1. d_{xy} é peso da aresta $\{xy\}$;
2. $|9 + 6|^2$ é o quadrado da soma do conhecimento prévio da disciplina de x e y ;
3. $|9|^2$ é o quadrado do intervalo entre as idades de x e y . Quanto maior for a diferença de idade, menor será o valor;
4. $|60|^2$ é o quadrado da distância entre as cidades que x e y residem;
5. $|8|^2$ é o quadrado do intervalo dos semestres que x e y cursam. Quanto maior for a diferença de semestres entre eles, menor será o valor;
6. $|7.0 + 6.5|^2$ é o quadrado da soma do valor do IRA de x e y .

Já para o problema da formação do comitê dos representantes dos professores / técnicos administrativos da universidade, a formulação de Bonotto e Cabral (2016) foi adaptada de outra maneira:

$$d_{xy} = \sqrt{\sum_{k=1}^n |d_{xk} - d_{yk}|^2},$$

de modo que

1. d_{xy} é peso da aresta $\{xy\}$;
2. n é a quantidade de características levadas em consideração;
3. $|d_{xk} - d_{yk}|$ é o módulo da diferença dos atributos definidos anteriormente, entre x e y .

Dessa maneira, quanto menos atributos em comum x e y possuíam, maior será o peso da aresta que os associa. Um exemplo fictício demonstrando o funcionamento do cálculo do valor da aresta entre dois vértices deste estudo de caso será exibido a seguir. Sejam os vértices:

- **Vértice x:** Um servidor TAE de 35 anos de idade, residente na cidade de Limoeiro do Norte, técnico em biblioteconomia, com 5 anos de experiência profissional;
- **Vértice y:** Um professor de 42 anos de idade, residente na cidade de Russas, Doutor em Ciência da Computação, com 15 anos de experiência profissional.

$$d_{xy} = \sqrt{|10|^2 + |35 - 42|^2 + |30|^2 + |10|^2 + |6|^2 + |5 - 15|^2},$$

onde

1. d_{xy} é peso da aresta $\{xy\}$;
2. $|10|^2$ é o quadrado do valor quando x e y não pertencem ao mesmo grupo de servidor. Se forem do mesmo tipo, o valor será $|7|^2$;
3. $|35 - 42|^2$ é o quadrado da diferença das idades de x e y ;
4. $|30|^2$ é o quadrado da distância entre as cidades que x e y residem;
5. $|10|^2$ é o quadrado da diferença das formações acadêmicas de x e y . Se pertencerem a mesma área de conhecimento, maior será o valor, caso contrário, menor será o valor;
6. $|6|^2$ é o quadrado da diferença das titulações de x e y . Quanto mais próxima for a titulação entre ambos, menor será o valor;
7. $|5 - 15|^2$ é o quadrado da diferença de experiência profissional entre x e y .

5.4 Construção das Heurísticas e Adequação do Método exato de Soares *et al.* (2018)

Após a formulação dos grafos, foram implementados através do Framework *Jupyter Notebook*, com utilização da linguagem de programação Python, dois métodos heurísticos, sendo o primeiro chamado de Heurística Roleta Russa, que é baseada em probabilidades como critério de seleção. Já o segundo, denominado Heurística Senhor Barriga, é embasada sobre o princípio guloso. Vale ressaltar que ambas as heurísticas realizam um procedimento de busca local antes de retornar o subconjunto solução obtido. Também foi realizada uma adequação do método exato de Soares *et al.* (2018) para obter a solução ótima dos estudos de caso tratados na pesquisa.

5.5 Execução e Estudos Comparativos

Com os métodos implementados e adequados ao problema, os grafos formulados foram executados para que os algoritmos obtenham a solução dos estudos de caso propostos na pesquisa. Vale ressaltar que a saída dos métodos é uma clique de G , onde G equivale a cada instância que foi testada nos algoritmos. Após essa etapa, foi realizada uma análise comparativa entre as heurísticas e o método exato de Soares *et al.* (2018), para verificar se as soluções geradas pelas heurísticas foram relevantes em relação as obtidas pelo método exato de Soares *et al.* (2018). Além disso, também foram executadas nos algoritmos desenvolvidos e comparadas com as soluções obtidas pela heurística que compõe a estrutura do método exato de Soares *et al.* (2018), um conjunto de instâncias *benchmark* presentes na literatura, afim de avaliar o nível de precisão das soluções geradas pelas heurísticas desenvolvidas.

6 ALGORITMOS PROPOSTOS

6.1 Heurística Roleta Russa

Nesta Seção será apresentada a descrição da Heurística Roleta Russa (H-RR). A seleção do subconjunto de tamanho m consiste na escolha dos m elementos que possuem a maior probabilidade ao final da execução do algoritmo. Essa probabilidade é alterada continuamente no decorrer de cada iteração, podendo ser aumentada ou diminuída, dependendo do valor da soma do subconjunto encontrado na respectiva iteração do método. A etapa final da execução consiste em submeter a solução com maior probabilidade a um procedimento de BL, descrito na Seção 6.3, visando refinar o resultado do algoritmo. Ao final da execução, o algoritmo retorna a maior solução encontrada. A seguir será exibido o pseudocódigo da heurística juntamente com a descrição de algumas funções que são usadas na H-RR:

Algoritmo 2: Heurística Roleta Russa

Input: Grafo: G , Tamanho do subconjunto da solução: m

Output: Maior solução obtida

```

1 begin
2    $qtdSoluçãoRepetida \leftarrow 0$ 
3    $maiorSolução \leftarrow 0$ 
4    $subconjuntoSolução \leftarrow \text{vazio}$ 
5    $vetProbabilidades \leftarrow \text{iniciaProbabilidades}(G)$ 
6   while  $qtdSoluçãoRepetida < 10$  do
7      $vetorSolução \leftarrow \text{escolheSubconjunto}(G, m, vetProbabilidades)$ 
8      $somaSolução \leftarrow \text{somaArestas}(G, subconjuntoSolução)$ 
9     if  $somaSolução > maiorSolução$  then
10       $maiorSolução \leftarrow somaSolução$ 
11       $subconjuntoSolução \leftarrow vetorSolução$ 
12       $vetProbabilidades \leftarrow \text{aumentaProbabilidades}(G, subconjuntoSolução)$ 
13    else if  $somaSolução = maiorSolução$  then
14       $vetProbabilidades \leftarrow \text{aumentaProbabilidades}(G, subconjuntoSolução)$ 
15       $qtdSoluçãoRepetida \leftarrow qtdSoluçãoRepetida + 1$ 
16    else
17       $vetProbabilidades \leftarrow \text{diminuiProbabilidades}(G, subconjuntoSolução)$ 
18       $qtdSoluçãoRepetida \leftarrow 0$ 
19    end
20  end
21   $resultado \leftarrow \text{buscaLocal}(G, maiorSolução, vetorSolução)$ 
22  Retorne resultado
23 end

```

1. **iniciaProbabilidades(G)**: Função usada na linha 5 da H-RR, responsável por iniciar as probabilidades do grafo que é passado por parâmetro. Essa probabilidade é calculada da seguinte forma: $1/\text{qtdVértices}$, onde qtdVértices equivale ao total de vértices do grafo G. Essa condição garante que todos os vértices do grafo tenham a mesma chance de serem selecionados no início da execução do algoritmo;
2. **escolheSubconjunto(G, m, vetProbabilidades)**: Função usada na linha 7 da H-RR, encarregada de selecionar um subconjunto solução (cujo tamanho é informado por parâmetro) mediante a uma determinada população de vértices. O processo de seleção leva em consideração as probabilidades de cada elemento do grafo;
3. **somaArestas(G, subconjuntoSolução)**: Responsável por somar a ponderação das arestas de um subconjunto que é passado como parâmetro. É através deste método que a diversidade máxima é calculada. Foi utilizada na linha 8 da H-RR;
4. **aumentaProbabilidades(G, subconjuntoSolução)**: Função utilizada nas linhas 12 e 14 da H-RR, encarregada de aumentar os valores de probabilidade dos elementos do subconjunto solução, e diminuir a probabilidade dos elementos que não fazem parte do referido conjunto. O valor tanto do acréscimo quanto do decréscimo da probabilidade é calculado de acordo com a quantidade de vértices do grafo;
5. **diminuiProbabilidades(G, subconjuntoSolução)**: Função utilizada na linha 17 da H-RR, responsável por diminuir os valores de probabilidade dos elementos do subconjunto solução cuja diversidade não seja a maior encontrada no decorrer da execução do algoritmo, e aumentar a probabilidade dos demais elementos. O valor tanto do decréscimo quanto do acréscimo da probabilidade é calculado levando em consideração a quantidade de vértices do grafo.

6.2 Heurística Senhor Barriga

Esta Seção disserta sobre a Heurística Senhor Barriga (H-SB), que tem a sua estrutura baseada no princípio guloso, visando encontrar uma solução localmente ótima para cada iteração. Em cada iteração, a formação do subconjunto solução é realizada em duas etapas complementares, a primeira parte é selecionada de forma aleatória, já a segunda parte é selecionada de forma gulosa, que consiste em buscar os elementos que ocasionem na maior diversidade possível ao se juntarem com o conjunto selecionado de forma aleatória. Ao final da execução do algoritmo, a melhor solução obtida é submetida a um procedimento de BL, descrito na Seção 6.3, com o intuito de aprimorar ainda mais o resultado do algoritmo. A seguir será exibido o pseudocódigo da heurística juntamente com a descrição de algumas funções que são usadas na H-SB:

Algoritmo 3: Heurística Senhor Barriga

Input: Grafo: G , Tamanho do subconjunto da solução: m

Output: Maior solução obtida

```

1 begin
2   maiorSolução  $\leftarrow$  0
3   subconjuntoSolução  $\leftarrow$  vazio
4   incremento  $\leftarrow$  0
5   tamSubconjuntoAleatorio  $\leftarrow$   $m / 2$ 
6   coeficienteDeExecução  $\leftarrow$  (qtdArestas( $G$ ) * 0.8)
7   while incremento < coeficienteDeExecução do
8     vetorAleatorio  $\leftarrow$  escolheSubconjuntoAleatorio( $G$ , tamSubconjuntoAleatorio)
9     vetorSolução  $\leftarrow$  selecionaConjuntoGuloso( $G$ , vetorAleatorio, ( $m / 2$ ))
10    somaSolução  $\leftarrow$  somaArestas( $G$ , vetorSolução)
11    if somaSolução > maiorSolução then
12      maiorSolução  $\leftarrow$  somaSolução
13      subconjuntoSolução  $\leftarrow$  vetorSolução
14    end
15    incremento  $\leftarrow$  incremento + 1
16  end
17  resultado  $\leftarrow$  buscaLocal( $G$ , maiorSolução, subconjuntoSolução)
18  Retorne resultado
19 end

```

1. **escolheSubconjuntoAleatorio(G, tamSubconjuntoAleatorio):** Função utilizada na linha 8 da H-SB, responsável por selecionar uma parte do subconjunto solução (equivalente a metade do tamanho do subconjunto, que é passado como parâmetro para o algoritmo) de forma totalmente aleatória;
2. **selecionaConjuntoGuloso(G, vetorAleatorio, (m / 2)) :** Função usada na linha 9 da H-SB, encarregada de selecionar de forma gulosa a outra metade do subconjunto solução. O processo de seleção dos elementos consiste na escolha dos vértices que ao se juntarem ao subconjunto que foi selecionado de forma aleatória, formem um conjunto cujo os elementos formem a maior diversidade possível;
3. **somaArestas(G, subconjuntoSolução):** Responsável por somar a ponderação das arestas de um subconjunto que é passado como parâmetro. É através deste método que a diversidade máxima é calculada. Foi utilizada na linha 10 da H-SB.

6.3 Método de Busca Local

Nesta Seção será exibido o pseudocódigo do algoritmo de Busca Local que foi desenvolvido e utilizado pelas heurísticas Roleta Russa e Senhor Barriga, visando refinar ainda mais o subconjunto solução obtido pelos algoritmos propostos. Basicamente, o método consiste em trocar os vértices do subconjunto solução de entrada por outros vértices que inicialmente não fazem parte da solução, visando um possível aperfeiçoamento das soluções geradas pelos algoritmos. A seguir será exibido o pseudocódigo do método de BL:

Algoritmo 4: Algoritmo de Busca Local

Input: Grafo: G , Subconjunto solução: s , maior Solução: $maxResult$

Output: subconjunto solução refinado

```

1 begin
2   resultado  $\leftarrow$  vazio
3   while houver um vértice de  $G$  que ainda não iterou sobre  $s$  do
4     | Seleccione um vértice de  $G$  que não esteja em  $s$ ;
5     | Troque um vértice de  $s$  por vez, pelo vértice selecionado anteriormente;
6   end
7   Some as soluções de todos os novos conjuntos obtidos;
8   Atribua a variável resultado a melhor solução encontrada.
9   if valorSolução(resultado) > maxResult then
10    | Retorne resultado
11  else
12    | Retorne "Não encontrei nenhuma solução melhor."
13  end
14 end

```

7 EXPERIMENTOS COMPUTACIONAIS

Este Capítulo apresenta os resultados obtidos neste trabalho. A Seção 7.1 descreve o ambiente de desenvolvimento sobre qual os experimentos computacionais foram realizados. A Seção 7.2 apresenta as características e peculiaridades de todos os grupos de instâncias que foram utilizadas na pesquisa. As Seções 7.3 e 7.4 denotam acerca dos resultados obtidos na resolução dos estudos de caso que foram propostos nesta pesquisa, juntamente com uma análise comparativa entre as soluções obtidos pelos algoritmos desenvolvidos neste trabalho e os resultados geradas pelo método exato de Soares *et al.* (2018). Por fim, a Seção 7.5 exibe as resoluções obtidas pelas heurísticas H-RR e H-SB em comparação as alcançadas pela Heurística de SOARES (HS), em relação as instâncias *benchmark* da literatura.

7.1 Configuração do Ambiente Computacional

Nesta Seção são descritas as especificações do ambiente computacional em que foram realizados os experimentos computacionais. A implementação das heurísticas H-RR, H-SB e do algoritmo de BL foi realizada através do Framework de desenvolvimento de código livre *Jupyter Notebook*, por intermédio da linguagem de programação Python. Os experimentos computacionais foram realizados em um equipamento com processador Intel Core I5-4570 (3.20GHz) 4 Threads com 16 GB de RAM e sistema operacional Ubuntu 16:04 LTS. Vale salientar que para a realização dos experimentos, tanto para as instâncias dos estudos de caso, quanto para as instâncias da literatura, foi considerado um tempo limite de execução de 1000 segundos para todos os algoritmos analisados.

7.2 Características das Instâncias

7.2.1 Instâncias da literatura

Os experimentos computacionais para sustentação / aprimoramento das soluções geradas pelos algoritmos desenvolvidos nesta pesquisa foram realizados sobre uma amostragem de 130 instâncias *benchmark* presentes na literatura, que são divididas em 3 subconjuntos com características dessemelhantes. Esse grupo de instâncias pode ser encontrado em <http://www.grafo.etsii.urjc.es/opticom/mdp/>. A seguir, são apresentadas as principais características de cada grupo das instâncias:

- **GKD:** Grupo de instâncias com complexidade moderada / robusta, desenvolvidas por Duarte e Martí (2007), cujo os dados consistem em 145 matrizes para as quais os valores foram calculados como as distâncias euclidianas de pontos gerados aleatoriamente com coordenadas no intervalo de 0 a 10. A quantidade de vértices das instâncias (n) varia entre 10, 15, 30, 50, 100, 125 e 150. Os testes foram efetuados com m alternando entre 2, 3, 5, 6, 8, 9, 12, 15, 37 e 45;
- **MDG:** Conjunto de instâncias mais robustas e complexas, desenvolvidas por Palubeckis (2007), cujo os dados consistem em 100 matrizes com números reais selecionados aleatoriamente entre 0 e 10 por intermédio de uma distribuição uniforme. A quantidade de vértices das instâncias (n) altera entre 500 e 2000. Os testes foram executados com o m alternando entre 50 e 200;
- **SOM:** Coleção de instâncias de complexidade intermediária, produzidas por Martí *et al.* (2010) com um gerador desenvolvido por Silva *et al.* (2004), cujo os dados consistem em 70 matrizes com números aleatórios entre 0 e 9 gerados a partir de uma distribuição uniforme inteira. A quantidade de vértices das instâncias (n) varia entre 100, 200, 300, 400 e 500. Os testes foram realizados com o m variando entre 10, 20, 30, 40, 50, 60, 80, 90, 100, 120, 150, 160 e 200.

7.2.2 Instâncias dos estudos de caso

As instâncias geradas através dos estudos de caso que foram introduzidos na universidade podem ser encontradas em <<https://github.com/lberTT/Instancias-tcc-humberto>>. A seguir, são apresentadas as principais propriedades e peculiaridades das duas instâncias que foram geradas neste trabalho:

- **Maximização da formação dos monitores da turma de processos de software:** Instância que possui o $n = 50$. Os testes foram realizados com o m variando entre 2, 5, 10, 15, 20 e 25. A ponderação de suas arestas vai de 48.28 a 75.64;
- **Diversificação na formação do comitê dos professores / servidores da universidade:** Instância cujo valor $n = 65$. Para a realização dos testes, o valor de m varia entre 2, 5, 12, 19, 26 e 32. A ponderação de suas arestas vai de 10.77 a 78.35.

7.3 Maximização da Formação dos Monitores da Sala de Aula

Os resultados obtidos pelas heurísticas H-RR e H-SB em comparação com o Método Exato de Soares *et al.* (2018) para a resolução do estudo de caso da maximização da formação dos monitores da turma de Processos de Software serão exibidos nesta Seção. A Tabela 2 mostra os resultados obtidos pelos algoritmos testados (H-RR, H-SB, o método exato de Soares *et al.* (2018)), sendo composta pelas seguintes colunas: Instância (EC-1); tamanho de vértices na instância (n); tamanho do subconjunto solução (m); valor obtido; e tempo de execução medido em segundos. Os valores destacados em negrito representam os melhores resultados de acordo com a descrição dos experimentos. A Figura 3 retrata um gráfico que apresenta a variação do tempo de execução dos algoritmos para cada valor de subconjunto solução (m) testado.

Os resultados apresentados na Tabela 2 certificam que as heurísticas desenvolvidas nesta pesquisa alcançaram resultados expressivos, na medida em que foram capazes de obter o resultado exato do respectivo estudo de caso em todos os testes realizados, com exceção da Heurística H-SB, que não obteve o valor ótimo no teste com o valor do subconjunto solução $m = 25$. Após a realização dos testes, foi possível identificar que dois elementos estavam presentes em todos os subconjuntos soluções obtidos, sendo eles:

- **Vértice 23:** Um aluno de 20 anos com IRA 8.0, residente na cidade de Russas, cursando o 4º semestre e com nota 9 no conhecimento prévio da disciplina;
- **Vértice 27:** Um aluno de 20 anos com IRA 8.3, residente na cidade de Russas, cursando o 3º semestre e com nota 8 no conhecimento prévio da disciplina.

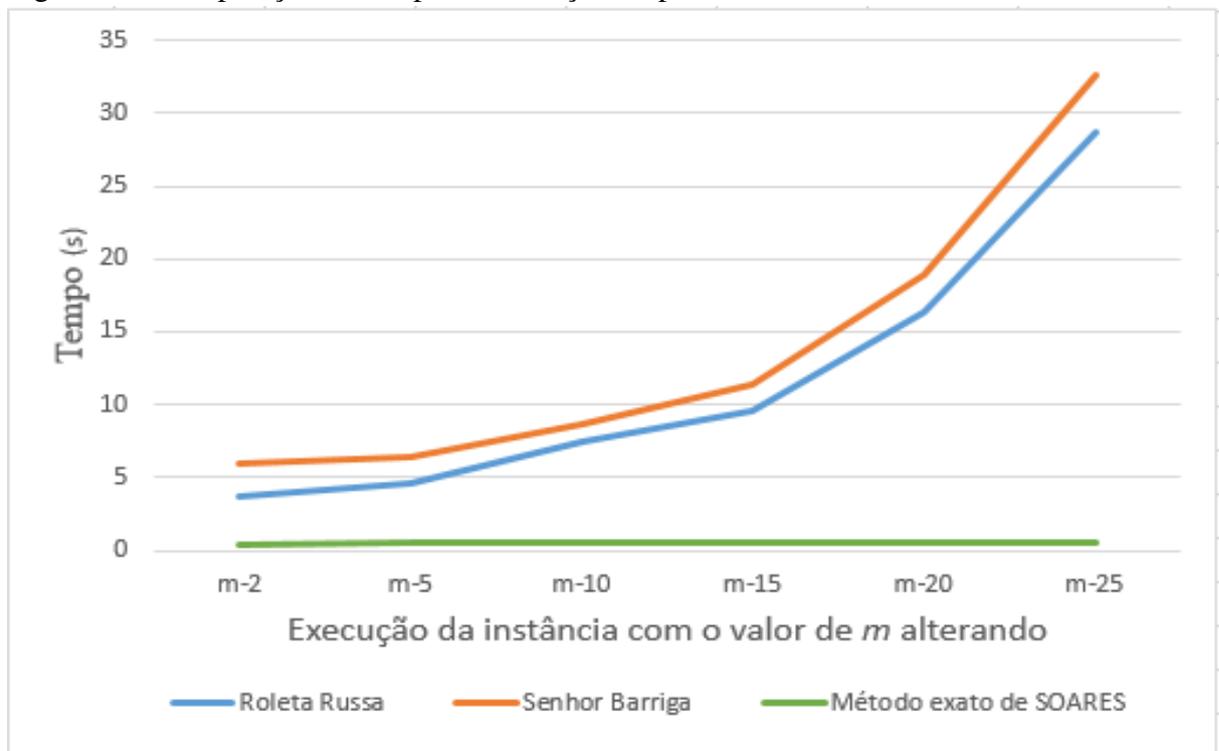
Mediante a análise do gráfico exibido na Figura 3, é plausível constatar que o tempo de execução dos algoritmos desenvolvidos neste trabalho tende a aumentar na medida em que o valor do subconjunto solução (m) aumenta. Em contrapartida, o Método Exato de Soares *et al.* (2018) mantém um desempenho constante para todos os valores de m testados. Através desta Figura, também é possível corroborar que a H-RR obteve em todos os testes realizados, um subconjunto solução em menos tempo que a H-SB.

Tabela 2 – Execução da instância do primeiro estudo de caso.

Instância	n	m	H-RR		H-SB		Método exato de SOARES	
			Valor	Tempo	Valor	Tempo	Valor	Tempo
EC-1	50	2	75.64	3.76	75.64	5.88	75.64	0.32
EC-1	50	5	741.30	4.61	741.30	6.43	741.30	0.51
EC-1	50	10	3243.66	7.45	3243.66	8.68	3243.66	0.54
EC-1	50	15	7377.61	9.55	7377.61	11.44	7377.61	0.52
EC-1	50	20	13105.69	16.37	13105.69	18.96	13105.69	0.52
EC-1	50	25	20411.18	28.72	20365.38	32.54	20411.18	0.52

Fonte: Elaborada pelo autor (2020)

Figura 3 – Comparação do tempo de execução do primeiro Estudo de Caso.



Fonte: Elaborado pelo autor (2020).

7.4 Diversificação do Comitê dos Professores/Servidores

Nesta Seção serão exibidos os resultados obtidos pelas heurísticas H-RR e H-SB em comparação com o Método Exato de Soares *et al.* (2018) para a resolução do estudo de caso da diversificação do comitê dos docentes e técnicos administrativos do campus. A Tabela 3 é composta pelas colunas de identificação da instância (EC-2), quantidade de vértices da instância (n), tamanho do subconjunto solução (m), valor obtido e tempo de execução medido em segundos, alcançados pelos algoritmos testados (H-RR, H-SB e o Método exato de Soares *et al.* (2018)). Os valores destacados em negrito correspondem aos melhores resultados de acordo com a descrição dos experimentos. A Figura 4 apresenta um gráfico que contempla a variação do tempo de execução dos algoritmos para cada subconjunto solução(m) testado.

Os resultados apresentados na Tabela 3 evidenciam que as Heurísticas H-RR e H-SB obtiveram resultados satisfatórios, já que foram competentes em obter o valor ótimo do devido estudo de caso em todos os testes executados, com ressalva a H-RR, que não obteve o resultado exato no teste com $m = 32$, e a H-SB, que só obteve o valor ótimo nos teste com $m = 2, 5, 12$ e 19 . Após a realização do testes foi possível identificar a existência de 2 vértices que fazem parte de todas as soluções geradas, sendo eles:

- **Vértice 33:** Um servidor TAE de 22 anos de idade, residente na cidade de Limoeiro do Norte, graduado em Engenharia Civil, com 1 ano e meio de experiência profissional;
- **Vértice 57:** Um professor de 50 anos de idade, residente na cidade de Russas, Doutor em Engenharia Mecânica, com 25 anos de experiência profissional.

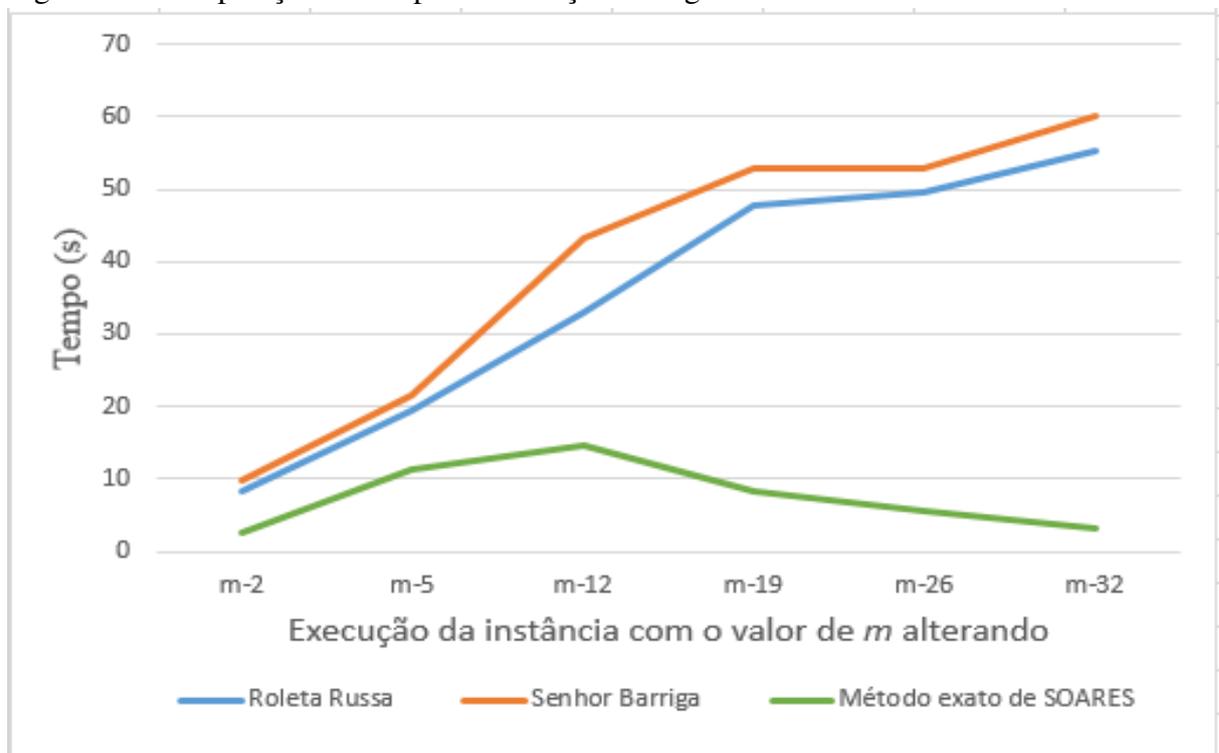
Minuciando a Figura 4, é possível inferir que a H-RR encontrou, em todos os testes realizados, um subconjunto solução em menos tempo que a H-SB. Também é possível deduzir através desta, que os algoritmos H-RR e H-SB levam mais tempo para encontrar o resultado de uma instância na medida em que o tamanho do subconjunto solução (m) aumenta. Por outro lado, o Método exato de Soares *et al.* (2018) demandou mais tempo para obter um subconjunto solução para valores intermediários de m (5, 12 e 19).

Tabela 3 – Execução da instância do segundo estudo de caso.

Instância	n	m	H-RR		H-SB		Método exato de SOARES	
			Valor	Tempo	Valor	Tempo	Valor	Tempo
EC-2	65	2	78.35	8.33	78.35	9.85	78.35	2.58
EC-2	65	5	512.76	19.32	512.76	21.43	512.76	11.42
EC-2	65	12	2862.61	32,98	2862.61	43.35	2862.61	14.53
EC-2	65	19	6851.62	47.65	6851.62	52.81	6851.62	8.33
EC-2	65	26	12324.24	49.47	12308.84	52.92	12324.24	5.54
EC-2	65	32	18068.74	55.34	18018.92	60.21	18103.19	3.28

Fonte: Elaborada pelo autor (2020)

Figura 4 – Comparação do tempo de execução do segundo Estudo de Caso.



Fonte: Elaborado pelo autor (2020).

7.5 Execução das Instâncias da Literatura

Esta Seção apresenta os resultados obtidos pelos algoritmos H-RR e H-SB em comparação com a Heurística que compõe a estrutura do método exato de Soares *et al.* (2018) com relação as instâncias *benchmark* disponíveis na literatura. As Tabelas 4, 5 e 6 são compostas pelas colunas de identificação da instância (GKD, MDG ou SOM), quantidade de vértices da instância (n), tamanho do subconjunto solução (m), valor obtido e tempo de execução medido em segundos, atingidos pelos algoritmos testados (H-RR, H-SB e a HS). Os valores destacados em negrito correspondem aos melhores resultados conforme a descrição dos experimentos. As Figuras 5, 6, 7, 8 e 9 apresentam gráficos que contemplam as variações de tempo de execução dos algoritmos que foram comparados para cada grupo de instâncias testadas.

7.5.1 Instâncias Mais Simples do Grupo GKD

Os resultados apresentados na Tabela 4 evidenciam que as heurísticas H-RR e H-SB alcançaram resultados no mínimo, iguais aos que a HS obteve com as instâncias mais simples do grupo GKD (com n alternando entre 10, 15, e 30), sendo superiores em 20% dos experimentos realizados, que correspondem as linhas GKD-2, GKD-5, GKD-7, GKD-8, GKD-24, GKD-25, GKD-41, GKD-43 e GKD-44. Em relação ao tempo de execução dos algoritmos, é possível notar que os algoritmos H-RR e H-SB foram capazes de obter um subconjunto solução em um intervalo de tempo igual ao obtido pela HS em todas as instâncias testadas, com exceção dos experimentos referentes as linhas GKD-41, GKD-42, GKD-43, GKD-44 e GKD-45.

7.5.2 Instâncias Intermediárias do Grupo GKD

Os resultados retratados na Tabela 5 apontam que a H-RR obteve, em 77.5% dos experimentos realizados, resultados tão bons ou melhores quanto os atingidos pela HS com as instâncias de tamanho intermediário (com n alternando entre 50, 100, 125 e 150) do grupo GKD, sendo inferior apenas nas instâncias GKD-62, GKD-64, GKD-67, GKD-71, GKD-73, GKD-77, GKD-83, GKD-84 e GKD-85. Já a H-SB alcançou em 80% dos testes realizados, resultados similares aos obtidos pela HS, sendo pior apenas somente nas instâncias GKD-58, GKD-63, GKD-67, GKD-71, GKD-73, GKD-79, GKD-83 e GKD-85. Também podemos notar que a H-SB obteve resultados melhores ou análogos aos obtidos pela H-RR em todos os casos, com exceção ao grupo de experimentos com $n = 100$ e $m = 10$.

No que diz respeito ao tempo de execução, mediante a análise das Figuras 5 e 6, percebe-se que a HS obteve, em todas as 45 instâncias testadas, um subconjunto solução em menos tempo que a H-RR, que por sua vez foi mais rápida que a H-SB em todos os experimentos. Também é possível atentar que para as instâncias intermediárias do grupo GKD (com n alternando entre 50, 100, 125 e 150), quanto maior for o tamanho do conjunto de entrada (n) e o tamanho do subconjunto solução (m), maior é o tempo que as heurísticas H-RR e H-SB levam para obter uma solução válida.

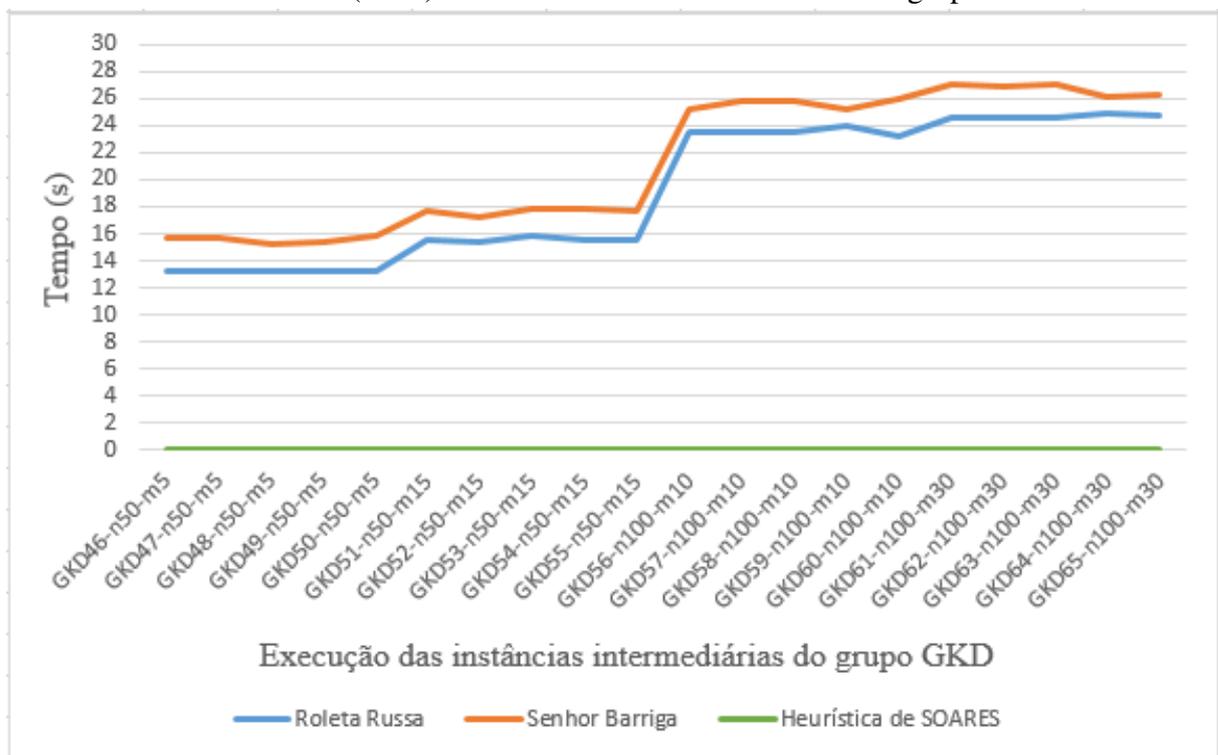
7.5.3 *Instâncias do Grupo MDG e SOM*

Mediante a análise da Tabela 6, verifica-se que a H-RR obteve, em 68% dos experimentos, resultados equivalentes ou superiores aos alcançados pela HS, com as instâncias mais robustas (com n alternando entre 500 e 2000) do grupo MDG (equivalente as linhas MDG-1, ..., MDG-25), obtendo resultados inferiores apenas nos testes documentados nas linhas MDG-5, MDG-11, MDG-13, MDG-14, MDG-18, MDG-20, MDG-21 e MDG-25. Já a H-SB conseguiu, em 52% dos testes, soluções análogas ou melhores que as obtidas pela HS, obtendo resultados inferiores somente nos experimentos atestados nas linhas MDG-2, MDG-5, MDG-7, MDG-9, MDG-13, MDG-14, MDG-17, MDG-18, MDG-20, MDG-21, MDG-23 e MDG-25.

Examinando os testes com as instâncias de complexidade intermediária / robusta (com n variando entre 100, 200, 300, 400 e 500) do grupo SOM (linhas SOM-1, ..., SOM-20 da Tabela 6), nota-se que a H-RR alcançou, em 80% dos testes, resultados melhores ou similares aos obtidos pela HS, sendo inferior apenas na execução das instâncias documentadas nas linhas SOM-15, SOM-16, SOM-19 e SOM-20. Já a H-SB obteve, em 70% dos testes, soluções superiores ou equivalentes as obtidas pela HS, sendo inferior apenas nos experimentos atestados nas linhas SOM-8, SOM-12, SOM-15, SOM-16, SOM-19 e SOM-20. Também é possível notar que a H-RR obteve resultados melhores ou iguais aos obtidos pela H-SB, tanto com os testes das instâncias do grupo MDG, quanto nos experimentos das instâncias do grupo SOM, com exceção do experimento apresentado na linha MDG-11, onde a H-RR obteve um solução melhor.

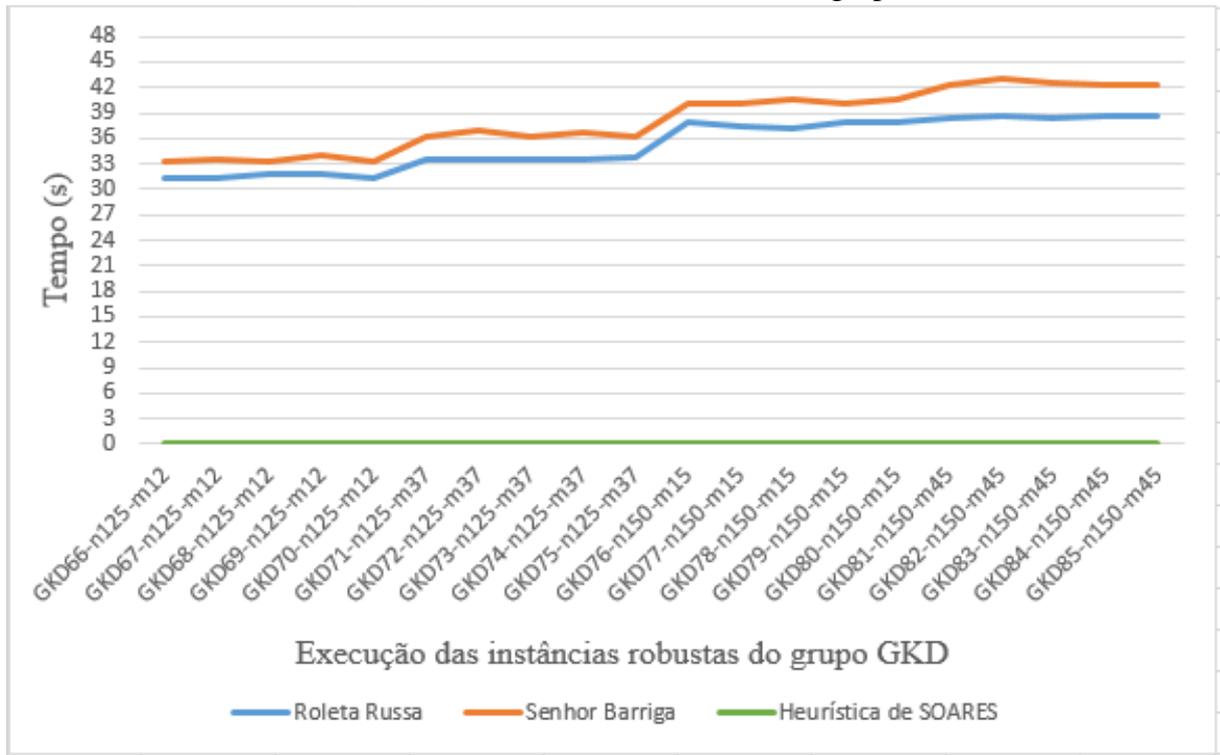
Com relação ao tempo de execução, nas instâncias do grupo MDG, mediante a análise das Figuras 7 e 8, percebe-se que a HS obteve uma solução para todos os experimentos realizados, em menos tempo que a H-RR, que por sua vez foi mais rápida que a H-SB em todos os testes. No tocante as instâncias do grupo SOM, examinando a Figura 9, nota-se que a HS também foi mais rápida que a H-RR em todos os testes, que por sua vez obteve um resultado para todos os experimentos em menos tempo que a H-SB. Também é possível verificar que tanto para as instâncias do grupo MDG, quanto para instâncias do grupo SOM, quanto maior for o tamanho do conjunto de entrada (n) e o tamanho do subconjunto solução (m), maior é o tempo que as heurísticas H-RR e H-SB levam para obter uma solução válida.

Figura 5 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares *et al.* (2018) com as instâncias intermediárias do grupo GKD.



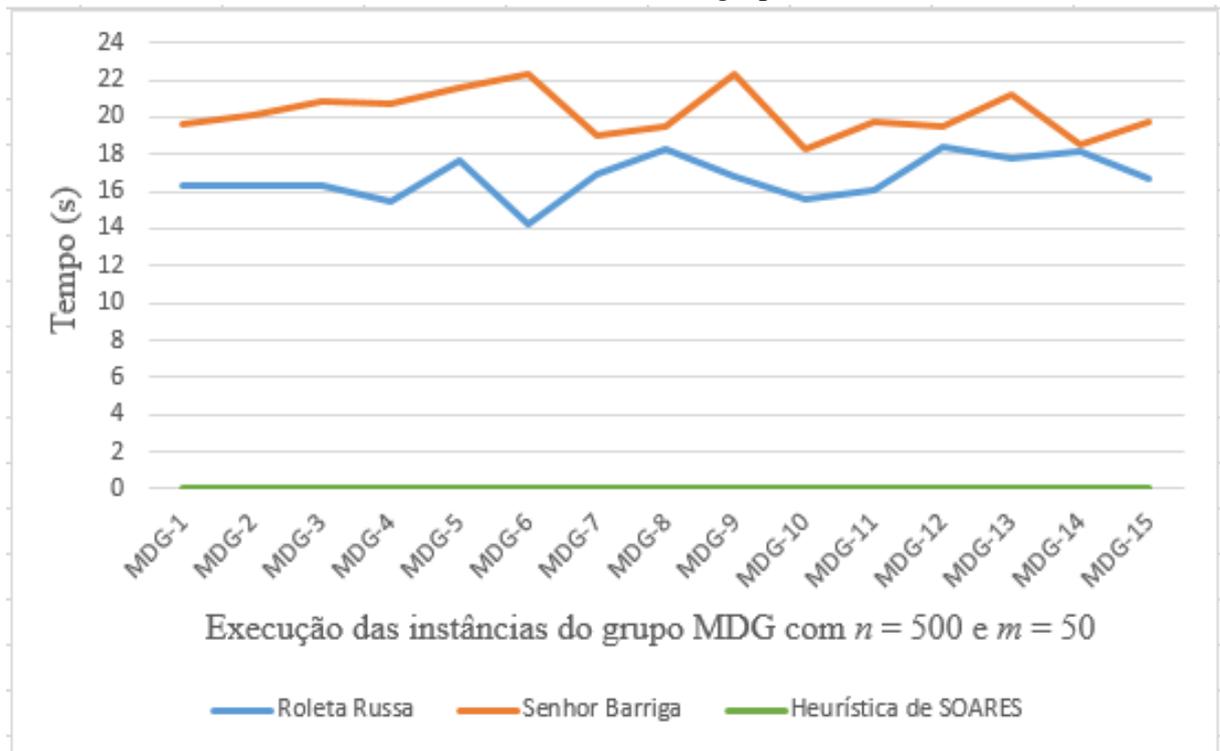
Fonte: Elaborado pelo autor (2020).

Figura 6 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares *et al.* (2018) com as instâncias robustas do grupo GKD.



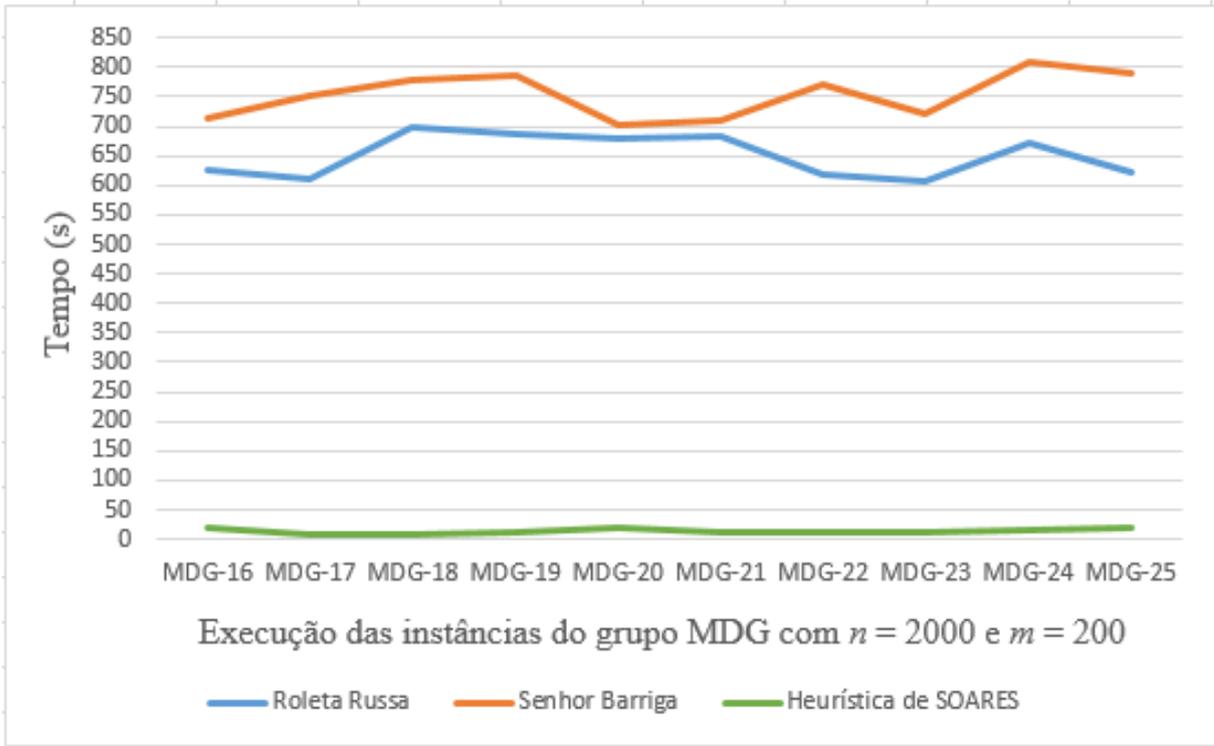
Fonte: Elaborado pelo autor (2020).

Figura 7 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares *et al.* (2018) com as instâncias do grupo MDG com $n = 500$ e $m = 50$.



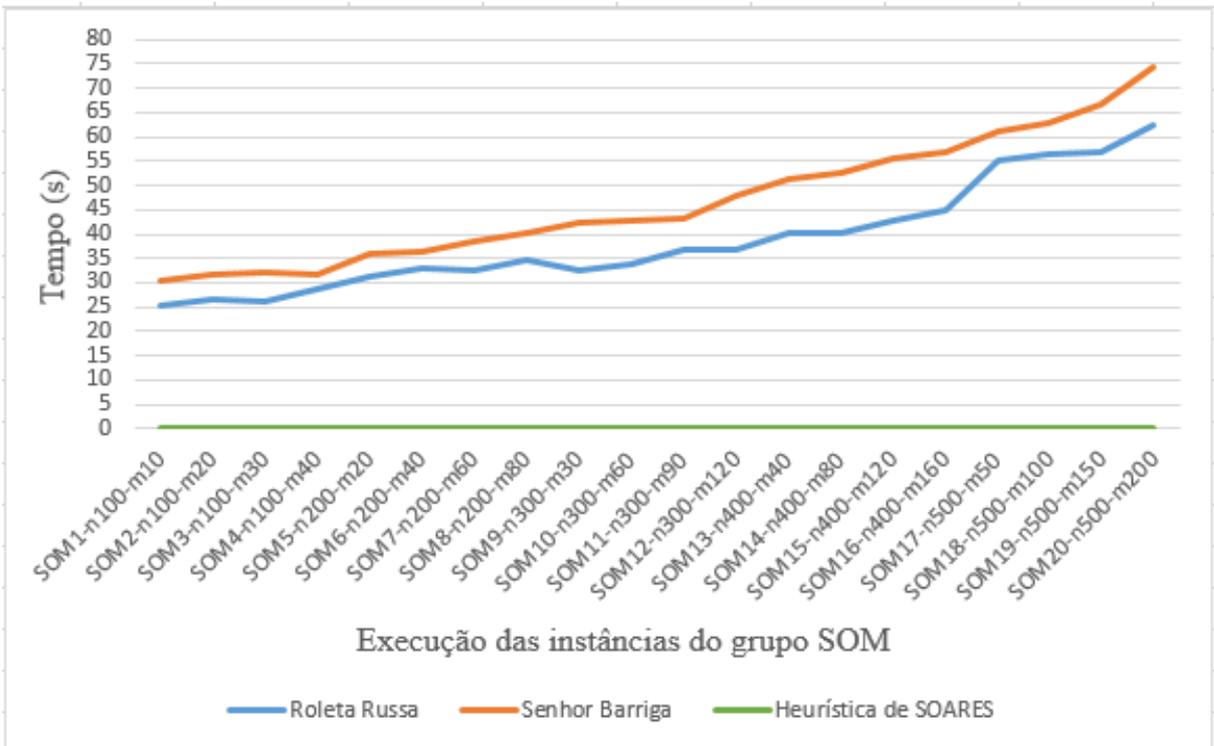
Fonte: Elaborado pelo autor (2020).

Figura 8 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares *et al.* (2018) com as instâncias do grupo MDG com $n = 2000$ e $m = 200$.



Fonte: Elaborado pelo autor (2020).

Figura 9 – Comparação do tempo de execução entre as heurísticas desenvolvidas e a heurística de Soares *et al.* (2018) com as instâncias do grupo SOM.



Fonte: Elaborado pelo autor (2020).

Tabela 4 – Execução das instâncias mais simples do grupo GKD.

Instância	n	m	H-RR		H-SB		HS	
			Valor	Tempo	Valor	Tempo	Valor	Tempo
GKD-1	10	2	243.97	0.00	243.97	0.00	243.97	0.00
GKD-2	10	2	210.65	0.00	210.65	0.00	191.44	0.00
GKD-3	10	2	146.61	0.00	146.61	0.00	146.61	0.00
GKD-4	10	2	186.18	0.00	186.18	0.00	186.18	0.00
GKD-5	10	2	209.61	0.00	209.61	0.00	207.35	0.00
GKD-6	10	3	254.13	0.00	254.13	0.00	254.13	0.00
GKD-7	10	3	470.04	0.00	470.04	0.00	450.57	0.00
GKD-8	10	3	363.24	0.00	363.24	0.00	357.38	0.00
GKD-9	10	3	541.08	0.00	541.08	0.00	541.08	0.00
GKD-10	10	3	435.71	0.00	435.71	0.00	435.71	0.00
GKD-11	10	6	2106.73	0.00	2106.73	0.00	2106.73	0.00
GKD-12	10	6	2349.91	0.00	2349.91	0.00	2349.91	0.00
GKD-13	10	6	1850.86	0.00	1850.86	0.00	1850.86	0.00
GKD-14	10	6	2496.47	0.00	2496.47	0.00	2496.47	0.00
GKD-15	10	6	1387.79	0.00	1387.79	0.00	1387.79	0.00
GKD-16	10	8	5279.08	0.00	5279.08	0.00	5279.08	0.00
GKD-17	10	8	3441.99	0.00	3441.99	0.00	3441.99	0.00
GKD-18	10	8	2720.61	0.00	2720.61	0.00	2720.61	0.00
GKD-19	10	8	3506.08	0.00	3506.08	0.00	3506.08	0.00
GKD-20	10	8	3612.46	0.00	3612.46	0.00	3612.46	0.00
GKD-21	15	3	332.76	0.00	332.76	0.00	332.76	0.00
GKD-22	15	3	670.80	0.00	670.80	0.00	670.80	0.00
GKD-23	15	3	631.81	0.00	631.81	0.00	631.81	0.00
GKD-24	15	3	521.45	0.00	521.45	0.00	490.33	0.00
GKD-25	15	3	575.13	0.00	575.13	0.00	571.47	0.00
GKD-26	15	6	2805.25	0.00	2805.25	0.00	2805.25	0.00
GKD-27	15	6	2793.15	0.00	2793.15	0.00	2793.15	0.00
GKD-28	15	6	2116.50	0.00	2116.50	0.00	2116.50	0.00
GKD-29	15	6	2987.21	0.00	2987.21	0.00	2987.21	0.00
GKD-30	15	6	2421.25	0.00	2421.25	0.00	2421.25	0.00
GKD-31	15	9	7271.14	0.00	7271.14	0.00	7271.14	0.00
GKD-32	15	9	6366.32	0.00	6366.32	0.00	6366.32	0.00
GKD-33	15	9	4319.72	0.00	4319.72	0.00	4319.72	0.00
GKD-34	15	9	5574.20	0.00	5574.20	0.00	5574.20	0.00
GKD-35	15	9	4191.09	0.00	4191.09	0.00	4191.09	0.00
GKD-36	15	12	10305.45	0.00	10305.45	0.00	10305.45	0.00
GKD-37	15	12	13126.98	0.00	13126.98	0.00	13126.98	0.00
GKD-38	15	12	11100.18	0.00	11100.18	0.00	11100.18	0.00
GKD-39	15	12	7902.54	0.00	7902.54	0.00	7902.54	0.00
GKD-40	15	12	10145.57	0.00	10145.57	0.00	10145.57	0.00
GKD-41	30	6	3113.91	1.55	3113.91	2.1	3097.71	0.00
GKD-42	30	6	3131.73	1.73	3131.73	2.1	3131.73	0.00
GKD-43	30	6	2709.70	1.55	2709.70	2.1	2699.49	0.00
GKD-44	30	6	1912.59	1.55	1912.59	2.1	1905.50	0.00
GKD-45	30	6	1903.68	1.55	1903.68	2.1	1903.68	0.00

Fonte: Elaborada pelo autor (2020)

Tabela 5 – Execução das instâncias mais robustas do grupo GKD.

Instância	<i>n</i>	<i>m</i>	H-RR		H-SB		HS	
			Valor	Tempo	Valor	Tempo	Valor	Tempo
GKD-46	50	5	1790.95	13.31	1790.95	15.75	1790.95	0.00
GKD-47	50	5	2182.36	13.31	2182.36	15.75	2182.36	0.00
GKD-48	50	5	1365.70	13.31	1365.70	15.31	1365.70	0.00
GKD-49	50	5	1291.36	13.31	1291.36	15.43	1291.36	0.00
GKD-50	50	5	1890.25	13.31	1890.25	15.75	1890.25	0.00
GKD-51	50	15	10842.42	15.48	10842.42	17.72	10842.42	0.00
GKD-52	50	15	7659.77	15.33	7659.77	17.26	7659.77	0.00
GKD-53	50	15	13862.29	15.86	13862.29	17.78	13862.29	0.00
GKD-54	50	15	14406.56	15.48	14406.56	17.89	14406.56	0.00
GKD-55	50	15	14350.75	15.48	14350.75	17.72	14350.75	0.00
GKD-56	100	10	5402.30	23.55	5402.30	25.22	5402.30	0.00
GKD-57	100	10	8068.11	23.55	8068.11	25.78	8068.11	0.00
GKD-58	100	10	5543.69	23.52	5529.94	25.82	5543.69	0.00
GKD-59	100	10	9473.17	23.98	9473.17	25.29	9473.17	0.00
GKD-60	100	10	7708.01	23.23	7708.01	25.22	7708.01	0.00
GKD-61	100	30	47683.49	24.55	47683.49	26.97	47683.49	0.00
GKD-62	100	30	42668.83	24.52	42690.59	26.92	42690.59	0.00
GKD-63	100	30	92960.15	24.59	92936.52	26.98	92960.15	0.00
GKD-64	100	30	81903.73	24.88	81918.67	26.31	81918.67	0.00
GKD-65	100	30	53634.54	24.73	53634.54	26.18	53634.54	0.00
GKD-66	125	12	5393.84	31.24	5393.84	33.21	5393.84	0.00
GKD-67	125	12	12614.36	31.29	12614.36	33.49	12648.60	0.00
GKD-68	125	12	8021.78	31.79	8021.78	33.21	8021.78	0.00
GKD-69	125	12	13328.91	31.88	13328.91	33.90	13328.91	0.00
GKD-70	125	12	7123.98	31.24	7123.98	33.37	7123.98	0.00
GKD-71	125	37	111436.82	33.41	111442.52	36.28	111485.58	0.00
GKD-72	125	37	129434.47	33.56	129434.47	36.88	129434.47	0.00
GKD-73	125	37	89448.27	33.41	89431.93	36.33	89474.97	0.00
GKD-74	125	37	97638.68	33.49	97638.68	36.76	97638.68	0.00
GKD-75	125	37	120923.32	33.79	120923.32	36.28	120923.32	0.00
GKD-76	150	15	22674.92	37.92	22674.92	40.21	22674.92	0.00
GKD-77	150	15	14389.51	37.33	14419.88	40.14	14419.88	0.00
GKD-78	150	15	12900.74	37.26	12900.74	40.56	12900.74	0.00
GKD-79	150	15	16344.65	37.87	16315.97	40.21	16344.65	0.00
GKD-80	150	15	16665.04	37.92	16665.04	40.51	16665.04	0.00
GKD-81	150	45	160263.19	38.29	160263.19	42.31	160263.19	0.00
GKD-82	150	45	198323.55	38.70	198323.55	42.92	198323.55	0.00
GKD-83	150	45	135989.62	38.36	135996.27	42.44	136023.33	0.00
GKD-84	150	45	199687.29	38.62	199702.54	42.31	199702.54	0.00
GKD-85	150	45	145638.78	38.69	145657.92	42.31	145685.38	0.00

Fonte: Elaborada pelo autor (2020)

Tabela 6 – Execução das instâncias dos grupos MDG e SOM.

Instância	n	m	H-RR		H-SB		HS	
			Valor	Tempo	Valor	Tempo	Valor	Tempo
MDG-1	500	50	7734.92	16.32	7734.92	19.65	7734.92	0.07
MDG-2	500	50	7591.82	16.32	7532.76	20.14	7591.82	0.04
MDG-3	500	50	7699.34	16.27	7699.34	20.86	7699.34	0.03
MDG-4	500	50	7715.70	15.44	7715.70	20.77	7715.70	0.03
MDG-5	500	50	7626.56	17.62	7597.42	21.56	7640.84	0.04
MDG-6	500	50	7614.65	14.21	7614.65	22.31	7614.65	0.04
MDG-7	500	50	7655.63	16.94	7639.78	18.95	7655.63	0.03
MDG-8	500	50	7617.29	18.31	7617.29	19.47	7617.29	0.04
MDG-9	500	50	7651.15	16.79	7637.88	22.35	7651.15	0.02
MDG-10	500	50	7636.64	15.57	7636.64	18.25	7636.64	0.07
MDG-11	500	50	7614.86	16.13	7619.14	19.69	7619.14	0.03
MDG-12	500	50	7701.20	18.41	7701.20	19.44	7701.20	0.05
MDG-13	500	50	7657.74	17.83	7643.85	21.23	7673.66	0.03
MDG-14	500	50	7619.35	18.11	7598.44	18.55	7630.68	0.03
MDG-15	500	50	7651.44	16.66	7651.44	19.71	7651.44	0.04
MDG-16	2000	200	113420.00	625.31	113420.00	714.76	113420.00	18.80
MDG-17	2000	200	112826.00	611.44	112698.00	751.23	112826.00	7.75
MDG-18	2000	200	113358.00	698.51	113358.00	778.41	113397.00	8.08
MDG-19	2000	200	112861.00	685.77	112861.00	786.88	112861.00	10.75
MDG-20	2000	200	112988.00	679.21	112945.00	701.35	113037.00	18.64
MDG-21	2000	200	113597.00	683.53	113586.00	709.44	113610.00	10.63
MDG-22	2000	200	113539.00	619.42	113539.00	769.24	113539.00	12.41
MDG-23	2000	200	113135.00	605.11	113132.00	722.78	113135.00	13.21
MDG-24	2000	200	113358.00	670.49	113358.00	808.15	113358.00	14.50
MDG-25	2000	200	113325.00	621.37	113294.00	788.51	113339.00	17.86
SOM-1	100	10	333.00	25.31	333.00	30.25	333.00	0.00
SOM-2	100	20	1157.00	26.47	1157.00	31.77	1157.00	0.00
SOM-3	100	30	2457.00	26.21	2457.00	32.12	2457.00	0.00
SOM-4	100	40	4128.00	28.56	4128.00	31.69	4128.00	0.00
SOM-5	200	20	1215.00	31.32	1215.00	35.92	1215.00	0.00
SOM-6	200	40	4435.00	32.77	4435.00	36.33	4435.00	0.00
SOM-7	200	60	9342.00	32.54	9342.00	38.62	9342.00	0.01
SOM-8	200	80	16224.00	34.66	16197.00	40.21	16224.00	0.03
SOM-9	300	30	2654.00	32.34	2654.00	42.55	2654.00	0.00
SOM-10	300	60	9556.00	33.87	9556.00	42.78	9556.00	0.02
SOM-11	300	90	20643.00	36.63	20643.00	43.12	20643.00	0.05
SOM-12	300	120	35842.00	36.98	35818.00	47.87	35842.00	0.06
SOM-13	400	40	4559.00	40.31	4559.00	51.31	4559.00	0.01
SOM-14	400	80	16720.00	40.55	16720.00	52.37	16720.00	0.06
SOM-15	400	120	36046.00	42.58	36028.00	55.67	36076.00	0.09
SOM-16	400	160	62352.00	44.79	62316.00	56.86	62364.00	0.17
SOM-17	500	50	7022.00	55.21	7022.00	61.28	7022.00	0.05
SOM-18	500	100	26031.00	56.34	26031.00	62.93	26031.00	0.07
SOM-19	500	150	56364.00	56.87	56349.00	66.67	56373.00	0.40
SOM-20	500	200	97154.00	62.48	97125.00	74.32	97171.00	0.46

Fonte: Elaborada pelo autor (2020)

8 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, duas heurísticas que fazem uso do procedimento de busca local (descrito na Seção 6.3) em suas estruturas, sendo a primeira baseada em probabilidades como critério de seleção e a segunda fundamentada sobre o conceito de AG, foram apresentadas juntamente com uma análise dos seus respectivos desempenhos em comparação com o método exato de Soares *et al.* (2018) e com a HS. A análise foi realizada levando em consideração o tempo de execução e o valor do subconjunto solução obtido por cada algoritmo. Para tal, foram utilizadas uma amostragem de 130 instâncias *benchmark* disponíveis na literatura, divididas em três conjuntos (GKD, MDG e SOM), e duas instâncias obtidas através de dois estudos de caso que foram introduzidos na Universidade Federal do Ceará - Campus Russas, onde o primeiro estudo visa a maximização da formação dos monitores da turma de Processos de Software e o segundo trata sobre a diversificação da formação do comitê dos professores e técnicos administrativos da campus.

Com isso, é possível concluir mediante a análise da Seção 7.3, que a H-RR foi superior em relação a H-SB na resolução do estudo de caso da maximização da formação dos monitores da turma de processos de software, na medida em que obteve, em todos os experimentos, resultados equivalentes aos gerados pelo método exato de Soares *et al.* (2018), diferentemente da H-SB, que não alcançou a solução ótima em um dos testes realizados. Com relação ao estudo de caso da diversificação da formação do comitê dos professores / técnicos administrativos da universidade, é plausível consumir que a H-RR foi mais eficiente que a H-SB, tendo em vista que só não obteve a solução ótima em um dos testes, diferentemente da H-SB, que não alcançou em dois testes a solução ótima, como observado na Seção 7.4. Através da análise da Seção 7.5, é possível chegar a conclusão de que a H-RR obteve resultados mais relevantes que a H-SB com as instâncias mais simples do grupo GKD (como visto na Subseção 7.5.1) e com as instâncias do grupo MDG e SOM (como apresentado na Subseção 7.5.3), que por sua vez foi mais eficiente no tratamento das instâncias de complexidade intermediária do grupo GKD (como observado na Subseção 7.5.2). Isso evidencia que a H-RR é mais confiável que a H-SB para resolver instâncias de complexidade mais simples (com até 100 vértices) e instâncias de complexidade mais robusta (com no mínimo 300 vértices). Por outro lado, a H-SB obtém resultados superiores a H-RR no tratamento de instâncias de complexidade intermediária (entre 100 e 300 vértices).

8.1 Trabalhos Futuros

Espera-se em trabalhos futuros um possível aperfeiçoamento da estrutura das heurísticas Roleta Russa e Senhor Barriga, seja por intermédio da utilização de Meta-heurísticas, algoritmos evolucionárias, Aprendizagem Baseada em Oposição, Algoritmos Meméticos, Otimização por Nuvem de Partículas, Busca Tabu etc, visando a obtenção de resultados mais precisos e relevantes, fazendo com que os algoritmos H-RR e H-SB se tornem competitivos em relação as melhores heurísticas disponíveis na literatura. Outra possível proposta seria o aprimoramento dos coeficientes de iteração dos algoritmos H-RR e H-SB, para tratar instâncias de complexidade mais robusta de maneira mais eficiente, e conseqüentemente, reduzir consideravelmente o tempo de execução dos algoritmos com tais instâncias.

REFERÊNCIAS

- AGRAFIOTIS, D. K. Stochastic algorithms for maximizing molecular diversity. **Journal of chemical information and computer sciences**, v. 37, n. 5, p. 841–851, 1997.
- BONOTTO, E. L.; CABRAL, L. dos A. F. Otimização por nuvem de partículas e busca tabu para o problema da diversidade máxima. **Anais do XLVIII Simpósio Brasileiro de Pesquisa Operacional**, 2016.
- BRASSARD, G.; BRATLEY, P. **Fundamentals of Algorithmics**. Nova Iorque: Prentice Hall, 1995. ISBN 01-333-5068-1.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos, teoria e prática**. 3. ed. Brasil: Editora Campos, 2012. ISBN 978-85-352-3699-6.
- DHIR, K. S.; KUO, C.-C.; GLOVER, F. Optimizing diversity for engineering management. **Engineering Management Conference**, p. 23–26, 1993.
- DUARTE, A.; MARTÍ, R. Busca tabu e compreensão do problema da diversidade máxima. **European Journal of Operational Research**, p. 71–84, 2007.
- ERKUT, E.; NEUMAN, S. Comparison of four models for dispersing facilities. **INFOR: Information Systems and Operational Research**, v. 29, n. 2, p. 68–86, 1991.
- FERREIRA, C. E.; FERNANDES, C.; MIYAZAWA, F.; SOARES, J. A. R.; JR, C. J. P.; GUIMARÃES, K. S.; CARVALHO, M. H.; CERIOLI, M. R.; FEOFILOFF, P.; DAHAB, R.; WAKABAYASHI, Y. **Uma introdução sucinta a algoritmos de aproximação**. Rio de Janeiro: Colóquio Brasileiro de Matemática-IMPA, 2001.
- GARCIA, S.; GUERRERO, R.; CORRAR, J. L. Teoria das restrições e programação linear. **Congresso internacional de Custos**, 1997.
- GLOVER, F.; CHING-CHUNG, K.; DHIR, K. S. A discrete optimization model for preserving biological diversity. **Applied Mathematical Modelling**, v. 19, n. 11, p. 696–701, 1995.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear**. Rio de Janeiro: Campus: [s.n.], 2000.
- HILLIER, F. S.; LIEBERMAN, G. J. **Introdução a pesquisa operacional**. 8. ed. Brasil: McGraw Hill, 2013. ISBN 85-86804-68-1.
- JOHNSON, D. S. Local optimization and the traveling salesman problem. **Lecture Notes in Computer Science**, v. 442, p. 446–461, 1990.
- KUO, C.-C.; GLOVER, F.; DHIR, K. S. Analyzing and modeling the maximum diversity problem by zero-one programming. **Decision Sciences**, v. 24, n. 6, p. 1171–1185, 1993.
- MARTÍ, R.; GALLEGO, M.; DUARTE, A. Um algoritmo branch and bound para o problema de diversidade máxima. **European Journal of Operational Research**, p. 36–44, 2010.
- MCCONNELL, S. The new battle over immigration. **Fortune**, 2013.
- PALUBECKIS, G. Pesquisa tabu iterada para o problema de diversidade máxima. **Applied Mathematics and Computation**, 2007.

PRESTES, E. **Introdução à Teoria dos Grafos**. Rio Grande Do Sul: Instituto de informática: [s.n.], 2016.

SILVA, G.; OCHI, L.; MARTINS, S. Comparação experimental de procedimentos de busca adaptativa randômica gananciosa para o problema de diversidade máxima. **Lecture Notes in Computer Science**, v. 3059, p. 498–512, 2004.

SOARES, P. L. B.; CAMPELO, M.; REBOUÇAS, D. N. Método exato para o problema da diversidade máxima. **Latin-Iberoamerican Conference on Operations Research (CLAIO)**, 2018.

SZWARCFITER, J. L. **Grafos e algoritmos computacionais**. Rio de Janeiro: Campus: [s.n.], 1988. ISBN 85-7001-341-8.

ZHOU, Y.; HAO, J.-K.; DUVAL, B. Opposition-based memetic search for the maximum diversity problem. **IEEE Transactions on Evolutionary Computation**, v. 21, n. 5, p. 731–745, 2017.