



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

**RAFAEL DE ARAÚJO COSTA**

**PROBLEMA DE MINIMIZAÇÃO DE CUSTO DE PROPAGAÇÃO DE INFLUÊNCIA  
EM REDES SOCIAIS: UMA ABORDAGEM HEURÍSTICA**

**RUSSAS**

**2019**

RAFAEL DE ARAÚJO COSTA

PROBLEMA DE MINIMIZAÇÃO DE CUSTO DE PROPAGAÇÃO DE INFLUÊNCIA EM  
REDES SOCIAIS: UMA ABORDAGEM HEURÍSTICA

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

Orientadora: Prof. Ms. Tatiane Fernan-  
des Figueiredo

RUSSAS

2019

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)

---

C875p Costa, Rafael de Araújo.  
Problema de minimização de custo de propagação de influência em redes sociais : uma abordagem heurística / Rafael de Araújo Costa. – 2019.  
37 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia de Software, Russas, 2019.  
Orientação: Prof. Me. Tatiane Fernandes Figueiredo.

1. Meta-Heurísticas. 2. Redes Sociais. 3. Algoritmo Genético. I. Título.

CDD 005.1

---

RAFAEL DE ARAÚJO COSTA

PROBLEMA DE MINIMIZAÇÃO DE CUSTO DE PROPAGAÇÃO DE INFLUÊNCIA EM  
REDES SOCIAIS: UMA ABORDAGEM HEURÍSTICA

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

Aprovada em:

BANCA EXAMINADORA

---

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

---

Prof. Dr. Marcio Costa Santos  
Universidade Federal do Ceará - UFC

---

Prof. Dr. Bonfim Amaro Júnior  
Universidade Federal do Ceará - UFC

## **AGRADECIMENTOS**

A Prof. Dr. Tatiane Fernandes, pela orientação, pelos conselhos e por todas as vivências, conversas e broncas.

Ao meu vizinho, Lucas Bezerra e a todos meus amigos do grupo dos "Cornos e Simpatizantes" por todo o apoio e conversas trocadas.

Aos meus colegas de estágio, Hismael, Cintia e Alex por toda troca de conhecimento e aprendizado.

Por fim, a minha família, em especial a minha vó Dilma e minha tia Berlanea por me darem estadia durante o período de graduação e ao meu pai minha mãe e meu irmão por todo o apoio.

“Conhecimento não é aquilo que você sabe, mas  
o que você faz com aquilo que você sabe.”

(Aldous Huxley)

## RESUMO

Com o crescimento das redes sociais nos últimos anos, problemas relacionados a propagação de informação nestas redes tem ganhado cada vez mais notoriedade na comunidade científica. Muitos desses problemas tratam de encontrar indivíduos que sejam cruciais para o processo de propagação de influência, entre eles encontra-se o Problema de Minimização de Custo de Propagação de Influência em Redes Sociais (PMCPIRS), que consiste em encontrar um conjunto de indivíduos chaves que consigam influenciar um determinado número de usuários com o menor custo possível para contratá-los. Atualmente, as técnicas apresentadas na literatura para resolução deste problema concentram-se apenas em métodos exatos. Em contraponto, esse trabalho apresenta o desenvolvimento de uma meta-heurística como opção para resolução de diferentes instâncias, ou seja, caracterizadas por pequenas, médias ou grandes propagações. Em termos de GAP, experimentos computacionais realizados em um conjunto de 108 instâncias atestam a eficiência do método para obtenção de boas soluções PMCPIRS.

**Palavras-chave:** Meta-Heurísticas. Redes Sociais. Algoritmo Genético.

## ABSTRACT

With the growth of social networks in recent years, problems related to the dissemination of information on these networks have gained more and more notoriety in the scientific community. Many of these problems are about finding individuals who are crucial to the process of spreading influence, including the Social Network Influence Propagation Cost Minimization Problem (PMCPIRS), which is about finding a set of key individuals who can influence a certain number of users at the lowest possible cost to hire them. Currently, the techniques presented in the literature for solving this problem focus only on exact methods. In contrast, this paper presents the development of a metaheuristic as an option for resolution of different instances, that is, characterized by small, medium or large propagations. In terms of GAP, computational experiments performed on a set of 108 instances attest to the efficiency of the method to obtain good PMCPIRS solutions.

**Keywords:** Heuristics. Social Networks. Genetic Algorithm

## LISTA DE FIGURAS

Figura 1 – Representação parcial de uma solução. Em verde, destaca-se as arestas escolhidas para propagar influência. . . . .	26
Figura 2 – GAPs do Algoritmo Genético proposto. . . . .	34
Figura 3 – Comparação das instâncias com $ V  = 50$ e $\alpha = 1.0$ . . . . .	35

## LISTA DE TABELAS

Tabela 1 – Análise comparativa entre as diferentes instâncias da heurísticas proposta. . . . .	33
Tabela 2 – Comparação das instâncias com $ V  = 50$ e $\alpha = 1.0$ . . . . .	34

## LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo Genético Clássico . . . . .	19
Algoritmo 2 – Algoritmo Genético . . . . .	27
Algoritmo 3 – Nova Solução . . . . .	28
Algoritmo 4 – Crossover . . . . .	30
Algoritmo 5 – Mutação . . . . .	31

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>1.1</b>	<b>Objetivos</b>	<b>14</b>
<i>1.1.1</i>	<i>Objetivo geral</i>	<i>14</i>
<i>1.1.2</i>	<i>Objetivos específicos</i>	<i>14</i>
<b>1.2</b>	<b>Procedimentos Metodológicos</b>	<b>14</b>
<i>1.2.1</i>	<i>Estudo da Área</i>	<i>14</i>
<i>1.2.2</i>	<i>Definição e análise do problema</i>	<i>14</i>
<i>1.2.3</i>	<i>Criação das instâncias para os testes computacionais</i>	<i>15</i>
<i>1.2.4</i>	<i>Desenvolvimento de um procedimento computacional</i>	<i>15</i>
<i>1.2.5</i>	<i>Teste e aprimoramento do modelo</i>	<i>15</i>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
<b>2.1</b>	<b>Teoria dos Grafos</b>	<b>16</b>
<b>2.2</b>	<b>Pesquisa Operacional</b>	<b>16</b>
<b>2.3</b>	<b>Modelos Limiares de Comportamento Coletivo</b>	<b>17</b>
<i>2.3.1</i>	<i>Modelo de Limiar Linear</i>	<i>17</i>
<b>2.4</b>	<b>Heurística</b>	<b>18</b>
<i>2.4.1</i>	<i>Meta-heurísticas</i>	<i>18</i>
<i>2.4.1.1</i>	<i>Algoritmo Genético</i>	<i>19</i>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>21</b>
<b>3.1</b>	<i>Modelo de limiar linear dinâmico para Amigos e Inimigos</i>	<i>21</i>
<b>3.2</b>	<i>Seleção Combinatória de Conjuntos Alvos</i>	<i>21</i>
<b>3.3</b>	<i>Maximizando a propagação da influência através de um social Rede</i>	<i>22</i>
<b>3.4</b>	<i>Maximização de Influência em Grande Escala através de Cobertura Máxima Local</i>	<i>22</i>
<b>4</b>	<b>O PROBLEMA DE MINIMIZAÇÃO DE CUSTO DE PROPAGAÇÃO DE INFLUÊNCIA EM REDES SOCIAIS</b>	<b>24</b>
<b>4.1</b>	<b>Definição do Problema</b>	<b>24</b>
<i>4.1.1</i>	<i>Notações e Premissas</i>	<i>25</i>
<b>4.2</b>	<b>Representação da Solução</b>	<b>25</b>
<b>4.3</b>	<b>Heurística fundamentada em Algoritmo Genético</b>	<b>26</b>

4.3.1	<i>Geração da População Inicial</i>	27
4.3.2	<i>Aptidão de uma Solução</i>	28
4.3.3	<i>Critérios de Parada</i>	29
4.3.4	<i>Crossover</i>	29
4.3.5	<i>Mutação</i>	31
4.3.6	<i>Seleção da Nova População</i>	31
5	<b>RESULTADOS</b>	32
5.1	<b>Configuração do Ambiente Computacional</b>	32
5.2	<b>Instâncias</b>	32
5.3	<b>Estudo Comparativo entre os Algoritmos</b>	33
6	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	36
	<b>REFERÊNCIAS</b>	37

## 1 INTRODUÇÃO

Com o crescimento das redes sociais nos últimos anos, influenciadores digitais tem se mostrado cada vez mais importantes em campanhas de marketing digital. De acordo com o Iinterativa (2018), 75% das marcas já utilizam influenciadores digitais em suas estratégias de marketing. Esse crescimento vem sendo notado a partir de 2017, onde houve um crescimento de 198% de posts patrocinados no Instagram, à medida que 75% dos usuários do Instagram já tomaram alguma ação após a exibição de um post patrocinado. Desta forma, podemos concluir que problemas relacionados a propagação de informação em redes sociais tem ganhado cada vez mais espaço no âmbito acadêmico.

Muitos desses problemas, como o descrito em Kempe *et al.* (2003), tratam de encontrar indivíduos-chaves que sejam cruciais para o processo de propagação de influência. Como exemplo, considere uma empresa de marketing digital que se utiliza de uma rede social para lançar um produto, logo, nota-se necessário a contratação de alguns indivíduos para convencer um subconjunto de usuários a adotar o produto. Sendo assim, alguns questionamentos devem ser feitos: quais indivíduos deve-se contratar de modo que aumente a quantidade de usuários convencidos? Como desencadear um processo de propagação de influência e quais indivíduos iremos atingir?

Em Fischetti *et al.* (2018) é definido o Problema de Minimização de Custo de Propagação de Influência em Redes Sociais ( $PMCP_{RS}$ ), que consiste em encontrar um conjunto de indivíduos-chaves que consigam influenciar um determinado número de usuários com o menor custo possível para contratá-los. Neste trabalho, os autores apresentam uma abordagem exata para sua resolução, para tal, utiliza-se como regra de propagação o Modelo de Limiar Linear, proposto por Granovetter (1978), no qual os indivíduos em uma rede social se tornam ativos (por exemplo, adotam uma informação ou um produto) se a soma das influências recebidas de seus vizinhos ativos atingir um limiar predefinido.

É importante mencionar, que devido a contemporaneidade do  $PMCP_{RS}$ , os trabalhos apresentados na literatura concentram-se em apresentações de soluções utilizando métodos exatos. Este trabalho tem como foco o desenvolvimento de um algoritmo heurístico, fundamentado na meta-heurística denominada Algoritmo Genético, que seja eficiente para o  $PMCP_{RS}$ . O presente trabalho é dividido em seis capítulos: Introdução (Seção 1); Fundamentação Teórica (Seção 2); Trabalhos Relacionados (Seção 3); Definição do Problema (Seção 4); Resultados (Seção 5) e Conclusões e Trabalhos Futuros (Seção 6).

## **1.1 Objetivos**

### ***1.1.1 Objetivo geral***

Obter soluções para o Problema de Minimização de Custo de Propagação de Influência em Redes Sociais através de um algoritmo heurístico fundamentado na meta-heurística denominada Algoritmo Genético.

### ***1.1.2 Objetivos específicos***

- Criar e implementar um algoritmo heurístico, fundamentado na meta-heurística denominada Algoritmo Genético, para o Problema de Minimização da Propagação de Influência em Redes Sociais;
- Desenvolver um gerador de instâncias para o problema, baseado em problemas semelhantes existentes na literatura;
- Fornecer uma análise estatística do desempenho do algoritmo heurístico apresentado.

## **1.2 Procedimentos Metodológicos**

Dada a natureza aplicada da pesquisa realizada, o presente trabalho foi desenvolvido com base nas fases padronizadas da Pesquisa Operacional abordadas por Hillier e Lieberman (2013), apresentadas a seguir:

### ***1.2.1 Estudo da Área***

Primeiramente, foi realizado um estudo na área de Teoria dos Grafos e Pesquisa Operacional com foco em técnicas de programação meta-heurísticas, além da pesquisa e do entendimento de problemas semelhantes descritos na literatura.

### ***1.2.2 Definição e análise do problema***

Dada a definição formal do problema, foi realizada uma análise a cerca das possíveis entradas e restrições de viabilidade do mesmo. Após a etapa de análise, desenvolveu-se um pseudocódigo para representar as etapas do algoritmo heurístico proposto para resolução do problema.

### **1.2.3 Criação das instâncias para os testes computacionais**

Foi desenvolvido um gerador de instâncias em Python tomando como base as instâncias utilizadas em Fischetti *et al.* (2018). Foram gerados três mundos de instancias (WATTS; STROGATZ, 1998), com cada um possuindo 36 grafos direcionados, variando em numero de vértices, grau, probabilidade de religação de arestas e percentual de ativação. Ao todo foram gerados 108 instâncias.

### **1.2.4 Desenvolvimento de um procedimento computacional**

Nessa etapa, foi implementado o algoritmo heurístico proposto na etapa 1.2.2, assim como o método exato apresentado por Fischetti *et al.* (2018). A linguagem de programação utilizada para a implmentação do algoritmo heurístico foi a linguagem *Python-3.5.6* , enquanto para o método exato, utilizou-se a linguagem *Java-8* com auxílio da pacote de otimização *CPLEX-12.8*.

### **1.2.5 Teste e aprimoramento do modelo**

Teste computacionais foram realizados para validação do algoritmo. Através da análises estatísticas dos resultados realizou-se alterações e melhoramentos no algoritmo proposto. Para os critérios de comparação e análise dos resultados finais obtidos, utilizou-se a medida conhecida como *GAP*, que é a distância da solução final obtida pelo algoritmo heurístico em comparação com a solução encontrada através do método exato proposto em Fischetti *et al.* (2018), não é necessariamente sendo a solução ótima, devido as restrições de configurações impostas ao algoritmo.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Teoria dos Grafos

Um *grafo orientado*, ou *digrafo*  $G = (V(G), E(G))$  consiste de um conjunto finito não vazio  $V(G)$  de elementos, chamados *vértices* ou *nós*, e um conjunto de pares ordenados de elementos distintos de  $V(G)$ , chamado *arcos* ou *arestas orientadas*, isto é,  $E(G) \subseteq \{(u, v) | u, v \in V(G), u \neq v\}$ . Se  $a = (u, v)$  é um arco, dizemos que  $a$  incide em  $u$  e em  $v$ , onde  $u$  e  $v$  são seus extremos. Além disso, também nos é referido a  $u$  como *inicio* ou *origem* e a  $v$  como *término* ou *destino*.

O *grau de entrada* ou *grau de recepção* de um vértice  $v \in V(G)$  corresponde ao número de arcos que incidem em  $v$  partindo de qualquer outro vértice  $u \in V(G) \setminus v$ . Em contrapartida, o *grau de saída* ou *grau de emissão* de um vértice  $v \in V(G)$  é definido pelo número de arcos que partem de  $v$ .

Um *passeio* em um grafo  $G = (V(G), E(G))$  é definido por uma sequência de vértices  $v \in V(G)$  dotada das seguintes propriedades: se  $u$  e  $v$  são vértices consecutivos na sequência, então,  $uv$  é um arco do grafo  $G$ , onde o inverso de um passeio não é, necessariamente, um passeio. Um *arco do passeio* é qualquer arco  $uv \in E(G)$  tal que  $v$  é o sucessor de  $u$  no passeio. Um passeio é *fechado* se tem pelo menos dois arcos e seu primeiro vértice coincide com o último.

Um *caminho* em um grafo  $G = (V(G), E(G))$  é definido por um passeio com uma sequência de vértices  $(v_0, v_1, \dots, v_k)$  de  $V(G)$  distintos entre si, ou seja, um passeio no qual todos os arcos são diferentes entre si. Um *ciclo* em um grafo  $G$  é definido por um caminho fechado, ou seja, um caminho que possui dois ou mais arcos, onde o primeiro e o último vértice são iguais.

### 2.2 Pesquisa Operacional

A Pesquisa Operacional (PO), é um ramo interdisciplinar da matemática aplicada que faz uso de modelos matemáticos, estatísticos e de algoritmos na ajuda à tomada de decisão. É aplicada a problemas envolvendo como conduzir e coordenar as operações (atividades) em uma organização (HILLIER; LIEBERMAN, 2013), sendo utilizada, sobretudo, para analisar sistemas complexos do mundo real, tipicamente com o objetivo de melhorar ou otimizar a performance. Para isso, a PO, pode-se utilizar de funções que maximizem ou minimizem uma função objetivo, afim de encontrar a melhor solução possível. Uma solução é dita ótima, caso seja uma solução

viável e de custo ótimo, ou seja, além de não violar nenhuma restrição do problema, uma solução ótima apresenta o melhor custo dentre todas as soluções. Caso uma das restrições seja violada, a solução é dita inviável.

Devido ao fato de que alguns problemas do mundo real possuem um vasto conjunto de soluções, pode não ser possível encontrar uma solução ótima em tempo computacional razoável. Sendo assim, os métodos heurísticos, fazem-se necessários nesses casos para encontrar uma solução viável razoavelmente boa em tempo hábil. O foco de estudo deste trabalho consiste na aplicação de uma meta-heurística para resolução do Problema de Minimização de Custo de Propagação de Influência em Redes Sociais.

### 2.3 Modelos Limiares de Comportamento Coletivo

Granovetter (1978) foi um dos primeiros a propor um modelo que sintetizasse o processo de disseminação de uma ideia ou inovação através de uma rede social, onde um ator qualquer possui duas alternativas, nas quais seus custos e/ou benefícios dependem das alternativas escolhidas por outros atores.

Um dos conceitos chave desse tipo de modelo é o de limiar de decisão.

Entrar ou não em greve, adotar ou não uma ideia, eleger A ou B, migrar ou permanecer em seu país, etc. Aqui são decisões binárias que constituem um dos requisitos do modelo, aquelas em que o ator possui dois comportamentos distintos e mutuamente exclusivos. Nesse caso e em muitos outros, precisar o limiar de decisão dos indivíduos seria responder à seguinte pergunta: qual o número ou proporção de outros indivíduos que devem tomar uma certa decisão antes que um determinado ator decida-se a tomá-la também? O limiar é esse ponto onde a percepção de benefício para um indivíduo tomar sua decisão excede a percepção dos custos (LEÃO, 2004).

Sendo assim, em meio aos modelos de comportamento coletivo apresentados por Granovetter (1978), o que descreveremos nesse trabalho é o Modelo de Limiar Linear.

#### 2.3.1 Modelo de Limiar Linear

Tendo como base um grafo direcionado  $G = (V, E)$ , onde cada vértice  $i \in V$  representa um ator e cada aresta  $i, j \in E$  representa a ligação entre  $i$  e  $j$ . Nesse modelo, cada nó  $i$ , é influenciado por seu vizinho  $j$  de acordo com o peso de sua ligação  $w_{i,p}$ . A dinâmica desse processo acontece da seguinte forma. Cada nó  $i$  possui um limiar  $\theta_i$  uniformemente randômico no intervalo de  $[0, 1]$ . Além disso, dizemos que um nó  $i$  é ativado, quando:

$$\sum_{i,j \in E} w_{i,j} \geq \theta_i$$

Sendo assim, o processo de difusão se desdobra deterministicamente em etapas discretas, onde na etapa  $t$ , todos os nós que estavam ativos na etapa  $t - 1$  permanecem ativos e ativamos qualquer nó  $v$  que tenha seu limiar atingido.

## 2.4 Heurística

Heurísticas podem ser definidas como estratégias que ignoram parte da informação de um problema com o objetivo de tornar mais fácil e rápida a escolha de uma solução. Provavelmente, uma heurística encontrará uma solução viável, mas não necessariamente uma solução ótima, para o problema específico que se deseja tratar (HILLIER; LIEBERMAN, 2013). No entanto, apesar de não possuir a garantia de encontrar uma solução ótima, um algoritmo heurístico bem elaborado, geralmente, é capaz de fornecer uma solução próxima do ótimo, ou concluir que tais soluções não existem.

Dentre diversas aplicações para algoritmos heurísticos, destaca-se o seu uso para resolução de problemas de otimização, que em sua grande maioria, são caracterizados por grandes espaços de soluções, o que dificulta a obtenção de uma solução ótima em tempo computacional razoável. Por esse motivo, as heurísticas abrem mão da otimalidade e buscam as melhores soluções possíveis de um problema em tempo viável.

Dentre as heurísticas conhecidas e catalogadas na literatura, pode-se classificá-las em: Heurísticas Construtivas, Heurísticas de Busca em Vizinhança, Heurísticas Sistemáticas, Heurísticas Híbridas e Meta-Heurísticas, onde esta última é o foco de estudo deste trabalho.

### 2.4.1 Meta-heurísticas

Uma meta-heurística é um método heurístico utilizado para resolver de forma genérica problemas de otimização. De forma geral, utilizam combinação de escolhas aleatórias e conhecimento dos resultados obtidos anteriormente para se guiarem na busca de soluções cada vez melhores. O objetivo deste trabalho reside na aplicação da meta-heurística Algoritmo Genético para resolução do Problema de Minimização de Custo de Propagação de Influência em Redes Sociais.

### 2.4.1.1 Algoritmo Genético

Desenvolvido por Holland *et al.* (1992) o Algoritmo Genético é um método de otimização inspirado pela teoria evolucionária de Charles Darwin, onde as populações evoluem de acordo com os princípios de seleção natural e os indivíduos mais bem adaptados ao ambiente têm maiores chances de sobreviver e de se reproduzir. Assim, pelo decorrer das gerações, a população de uma espécie evolui para melhor adaptar ao ambiente onde reside. Dessa forma, o ponto chave dos algoritmos genéticos é simular, de forma efetiva, termos vindo da genética, sendo eles:

- **População:** conjunto de indivíduos;
- **Cromossomo** (ou indivíduo): coleção de genes. Cada cromossomo é uma estrutura de dados que codifica a solução de um problema;
- **Gene:** menor unidade de um cromossomo. Codifica um simples parâmetro do problema;
- **Alelos:** valores que o gene pode assumir;
- **Cruzamento** (ou *crossover*): processo de combinar os genes dos cromossomos pais a fim de gerar novos indivíduos, garantindo assim a diversidade e a evolução populacional;
- **Mutação:** processo de alteração aleatória de genes, seja no seu conteúdo ou na sua localização. Como na biologia, esta anomalia possui baixa probabilidade de ocorrência;
- **Função de aptidão** (ou *fitness*): função que avalia quão bom é um indivíduo. Em geral, a aptidão é o valor da solução.

---

#### Algoritmo 1: Algoritmo Genético Clássico

---

**Input:** Instância do Problema.

**Output:** Melhor solução encontrada.

```

1 begin
2   Gerar população inicial;
3   Avaliar a aptidão da população;
4   while Critérios de parada não forem atendidos do
5     Selecionar indivíduos para reprodução;
6     Reprodução, de acordo com a taxa de reprodução;
7     Mutação, de acordo com a taxa de mutação;
8     Selecionar nova população;
9   return melhor solução encontrada
```

---

De forma mais detalhada, primeiramente é gerada uma população inicial (linha 2) e

avaliada a aptidão de cada indivíduo (linha 3). Enquanto o critério de parada não for atendido (linha 4), a população irá se reproduzir, podendo ou não, ocorrer mutações durante o processo de reprodução. Os indivíduos que serão selecionados para reprodução, assim como a decisão da aplicação das regras de cruzamento (*crossover*) e mutação são definidas na linha 5 e 6 e 7. Por fim, os melhores indivíduos serão selecionados na linha 8. Ao final do algoritmo será retornado o melhor de todos os indivíduos da última geração. Os critérios de parada podem ser os mais variados possíveis. Entre os mais comuns temos: tempo de processamento e quantidade de gerações e convergência, ou seja, não possuir melhora considerável na solução durante um determinado número de gerações.

É importante ressaltar, que o entendimento do problema que se deseja resolver e suas prioridades é essencial para prover bons resultados, tendo em vista que os parâmetros do algoritmo devem ser ajustado de acordo com o problema, assim como a escolha dos critérios de parada que devem ser utilizados.

### 3 TRABALHOS RELACIONADOS

Este capítulo descreve trabalhos da literatura mais relevantes para a contextualização do problema proposto nesta monografia. Na Seção 3.1 é apresentada um modelo atualizado de propagação de informação em redes sociais. Na Seção 3.2 é mostrado um modelo combinatório para o problema de seleção de alvos, por fim na Seção 3.3 e 3.4 é apresentado o Problema de Maximização de Influência apresentando diferentes formas para sua resolução.

#### 3.1 Modelo de limiar linear dinâmico para Amigos e Inimigos

Tal como o Modelo de Limiar Linear o *Modelo de limiar linear dinâmico para Amigos e Inimigos* ( $ML^2DAI$ ), formulado por Caliò e Tagarelli (2018), é um modelo de difusão que visa representar, da forma mais natural possível, o fenômeno de propagação de influência em cenários competitivos e não competitivos.

Logo, mesmo tomando como base o *Linear Threshold Model*, inicialmente proposto por Granovetter (1978), o  $ML^2DAI$  se difere ao tratar dos seguintes aspectos: o nível de confiança entre as relações, podendo ser tanto negativo quanto positivo; alguma possível mudança na opinião de um usuário já ativo; a hesitação em adotar alguma informação; o intervalo de tempo entre o instante da ativação e o início da propagação; o tempo do processo de propagação; múltiplas cascatas de informação que podem ocorrer de forma competitiva ou não competitiva.

Por fim, o modelo ainda traz algumas pequenas variações para cenários não competitivos ( $nC - ML^2DAI$ ), competitivos semi-progressivos ( $spC - ML^2DAI$ ) e competitivos não progressivos ( $npC - ML^2DAI$ ). Além disso, os autores ainda definem quatro estratégias para a seleção de indivíduos iniciais que são devidamente testadas e validadas.

#### 3.2 Seleção Combinatória de Conjuntos Alvos

Tendo como base o *Linear Threshold Model*, dado dois inteiros  $k$  e  $l$ , um grafo direcionado  $G = (V, E)$  e que  $t : V \rightarrow \mathbb{N}$  seja uma função de limiar associada aos vértices de  $G$ , o *Seleção Combinatória de Conjuntos Alvos* ( $SCCA$ ) consiste em encontrar um conjunto  $S \subseteq V$ , tal que  $|S| \leq k$ , e que ao fim do processo de propagação o número de vértices ativos seja igual ou superior a  $l$ .

Sendo assim, em Ackerman *et al.* (2010), é feito um modelo combinatório para o TSS com as mesmas entradas, porém, sendo necessário encontrar  $S \subseteq V$ , tal que  $|S| \leq k$ , havendo

um conjunto  $A \subseteq V$ , tal que  $S \subseteq A$ ,  $|A| \geq l$ , além da possibilidade de remover arestas de tal forma que  $G[A]$  seja acíclico e  $\deg_{in}(v) \geq t(v)$  para cada vértice  $v \in A \setminus S$ . Onde  $\deg_{in}(v)$  denota o grau de um vértice  $v$  em  $G$ , sendo  $S$  uma solução do *SCCA*, se e somente se, for uma solução do *Seleção Combinatória de Conjuntos Alvos*.

Note que, uma simples redução do *SCCA* para um problema de cobertura de vértices mostra que é NP-Difícil decidir se existe, ou não, tal conjunto  $S$ . Logo, definindo  $l = n$  e  $t(v) = \deg_{in}(v)$ , percebe-se que  $G$  tem uma cobertura de vértice de tamanho  $k$ , se e somente se, tiver um conjunto de alvos de tamanho  $k$ . Além disso, o fato de muitos casos buscar encontrar um conjunto pequeno de alvos que ative um grande número de vértices, motivou a criação de algumas variações do *SCCA*, sendo elas o *Conjunto Máximo Ativo* e o *Menor conjunto selecionado*. Ademais, o autor ainda deriva alguns limites combinatórios para o tamanho mínimo do conjunto de alvos  $k$  em termos de vértices, graus e limiares.

### 3.3 Maximizando a propagação da influência através de um social Rede

Inicialmente formulado por Kempe *et al.* (2003), o problema de maximização de influência através de uma rede social consiste em achar um conjunto de usuários  $k$  que maximize a quantidade de vértices ativos, seguindo um processo estocástico de propagação chamado de *Independent Cascade Model*, onde um usuário influenciará outro usuário adjacente com a probabilidade proporcional a relação entre eles.

Além disso, Kempe *et al.* (2003) também provou que é NP-Difícil determinar o ótimo para o problema de maximização de influência. O autor também propôs um algoritmo guloso com um fator de aproximação de  $(1 - 1/e - \epsilon)$  tanto para o *Independent Cascade Model*, como para o *Linear Threshold Model*. Onde  $e$  é a base do logaritmo natural e  $\epsilon$  é qualquer número positivo. Vale salientar que a estratégia gulosa proposta garante um desempenho de aproximadamente 63%, que é provada utilizando de técnicas da teoria de funções sub-modulares.

### 3.4 Maximização de Influência em Grande Escala através de Cobertura Máxima Local

O Problema de Locação Máxima de Cobertura (*Maximal Covering Location Problem*), definido por Church e ReVelle (1974), procura encontrar o maior número de consumidores que podem ser servidos dentro de uma área de cobertura ou por um dado tempo da utilização de um número limitado de facilidade. Sendo assim, em Güney *et al.* (2018), os autores mostram que

o Problema de Maximização de Influência (*IMP*) é um caso especial do Problema de Localização Máxima de Cobertura com incertezas nas relações entre consumidores e facilidades.

Logo, o ponto principal do modelo está em perceber que o conjunto de facilidades e de demanda corresponde diretamente ao conjunto de vértices  $V$  e que toda demanda é igual a um. Além disso, os autores ainda utilizam da técnica de *Decomposição de Benders* para a utilização do modelo em larga escala, adquirindo resultados computacionais que superam o desempenho do estado da arte em muitos casos por ordem de magnitude.

## 4 O PROBLEMA DE MINIMIZAÇÃO DE CUSTO DE PROPAGAÇÃO DE INFLUÊNCIA EM REDES SOCIAIS

Neste capítulo, é apresentado o Problema de Minimização de Custo de Propagação de Influência em Redes Sociais ( $PMCPI_{RS}$ ), foco de estudo deste trabalho. A definição formal do problema é apresentada na Seção 4.1. Na Seção 4.1.1 é apresentada algumas premissão e considerações que auxiliam no entendimento a cerca da definição viabilidade de uma solução para este problema, enquanto na Seção 4.2 é ilustrada como uma solução é representada neste trabalho. Por fim, na Seção 4.3 é apresentado o algoritmo proposto para resolução do  $PMCPI_{RS}$ .

### 4.1 Definição do Problema

A definição que usaremos para o  $PMCPI_{RS}$  é a mesma utilizada por Fischetti *et al.* (2018). Dado um grafo direcionado  $G = (V, A)$ , no qual  $V$  representa os usuários de uma rede social e  $A$  as conexões entre eles. Denominamos como *força de influência* de um vértice  $i \in V$  sobre um vértice  $j \in V$ , o valor  $d_{i,j} > 0$  associado a todo arco  $(i, j) \in A$ ; e definimos como *obstáculo*, o valor  $h_i > 0$  para todo  $i \in V$ , que representa o menor valor de influência que o vértice deve receber para ser considerado ativo. É importante mencionar que um vértice ativo representa um usuário que irá dar continuidade ao processo de propagação. Além disso, os possíveis incentivos são definidos pelo conjunto  $P_i$  juntamente com um custo  $w_{i,p} \geq 0$  para oferecer um incentivo de  $p \in P_i$  para o vértice  $i \in V$ .

Uma possível solução para o  $PMCPI_{RS}$  é dada por  $S = (V^s, N^s, p^s)$ , sendo  $V^s \subset V$  o conjunto de vértices ativos, de tal maneira que  $|V^s| \geq \lceil \alpha |V| \rceil$ . O conjunto de vizinhos influentes de  $i$  definido por  $N_i^s$  de tal maneira que  $N_i^s \subseteq N_i = \{j \in V | (j, i) \in A\}$ . Por fim, para cada vértice  $i \in V^s$ ,  $p_i^s \in P_i$  é um incentivo, de tal forma que  $i$  pode ser ativado pelos incentivos vindos de  $N_i^s + p_i^s$  (conforme a definição 1). A função de ativação utilizada foi baseada no Modelo de Limiar Linear proposto por Granovetter (1978), se diferenciando ao considerar o incentivo vindo de  $p_i^s$  como um fator de ativação. Portanto, para ativar um vértice  $i$  é necessário que  $p_i^s + \sum_{j \in N_i} w_{i,j} \geq h_i$ . Sendo assim, o objetivo do  $PMCPI_{RS}$  é encontrar os melhores incentivos  $p \in P_i$ , para todo  $i \in V$  que minimizem o total de custos  $\sum_{i \in V} w_{i,p_i^s}$  por incentivo pago.

### 4.1.1 Notações e Premissas

Para auxiliar no entendimento do problema e das definições utilizadas é importante ter o conhecimento de algumas notações. Além da definição de vizinhança,  $N_i = \{j \in V \mid (j, i) \in A\}$  para denotar o conjunto dos vértices vizinhos que podem influenciar o vértice  $i$  diretamente, usaremos  $\eta_i = 2^{N_i}$  para determinar a força de  $N_i$ . Além disso, é importante salientar que cada  $P_i$  contém um elemento especial 0, no qual nenhum incentivo é fornecido e nenhum custo é cobrado. Sendo assim, qualquer incentivo  $p \in P_i / \{0\}$  satisfaz  $p \geq 0$  e  $w_{ip} \geq 0$ . Ademais, existe pelo menos um elemento  $p \in P_i$  que pode ativar o indivíduo  $i \in V$  por completo, ou seja sem nenhum incentivo vindo de  $N_i$ , de tal forma que  $f_i(\emptyset, p) \geq h_i$ . Em seguida é apresentada algumas definições para auxiliar a modelagem do problema.

**Definição 1 (Função de Ativação)** *Uma função monótona  $f_i : \eta_i \times P_i \rightarrow \mathbb{R}_+$  definida para  $i \in V$  é uma função de ativação se satisfazer  $f_i(\emptyset, 0) = 0$ . Logo, a função  $f_i$  é monótona, se e somente se,  $U \subset U' \in \eta_i$  implica que  $f_i(U, p) \leq f_i(U', p)$ ,  $\forall p \in P_i$  e  $p \leq p' \in P_i$  implica que  $f_i(U, p) \leq f_i(U, p')$ ,  $\forall U \in \eta_i$ .*

**Definição 2 (Função de Ativação de Influência-Monótona)** *Uma função de ativação  $f_i : \eta_i \times P_i \rightarrow \mathbb{R}_+$  do nó  $i \in V$  é dita de influência monótona se e somente se  $f_i(U \cup \{j\}, p) \geq f_i(U \cup \{k\}, p)$  vale para cada incentivo  $p \in P_i$ , cada conjunto  $U \in \eta_i$ , e todo par de nós  $j \neq k \in N_i \setminus U$  tal que  $d_{ji} \geq d_{ki}$ .*

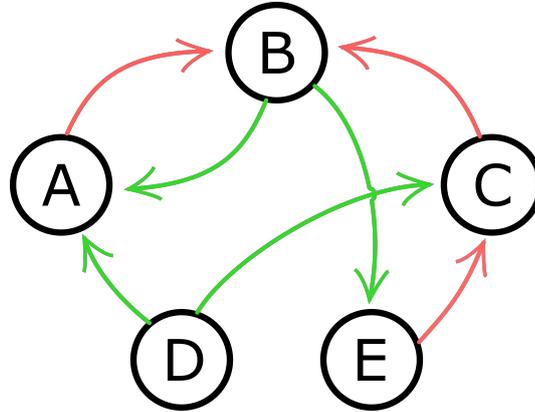
**Definição 3 (Grafo de Propagação Induzida)** *Sendo  $S = (V^s, N^s, p^s)$  uma possível solução para o  $PMCP_{IRS}$ . Um subgrafo  $G' = (V^s, A')$  de  $G$  é chamado de Grafo de Propagação Induzida de  $S$  se o somente o  $G'$  satisfazer as seguintes condições: (i)  $A' = \{(j, i) \in A \mid i, j \subseteq V^s, j \in N_i^s\}$  incluindo todo arco na qual a influência é exercida, e (ii)  $G'$  é acíclico, ou seja, não contém um ciclo direcionado.*

## 4.2 Representação da Solução

Neste trabalho, uma solução para o  $PMCP_{IRS}$ , dada pelo algoritmo apresentado em seguida, é retratado por uma matriz de incidência  $Z_{|V| \times |A|}^s$ , representando as arestas selecionadas para propagar a influência e um vetor  $V^s$  com a sequência dos vértices que serão ativados de forma a evitar ciclos. Por fim, o valor objetivo da solução, representado por  $\varphi^s$ , é dado

pelo incentivo  $p \in P_i$  de menor custo, tal que  $f_i(\eta_i^s, p) \geq h_i$  para todo  $i \in V^s$ , considerando  $N_i^s = \{j \in V^s | (j, i) \in A, Z_{ji}^s = 1\}$ .

**Figura 1** – Representação parcial de uma solução. Em verde, destaca-se as arestas escolhidas para propagar influência.



Fonte: Elaborado pelo autor (2019).

Exemplificando, a Figura 1 representa parcialmente uma possível solução  $S = (V^s, Z^s, \varphi^s)$  para o dígrafo  $G = (V, A)$ , onde  $V^s$  é dado pelo vetor  $[D, B, C, E, A]$ . Sendo assim, considerando  $i$  e  $j$  posições de  $V^s$ , nenhum incentivo pode ser propagado do vértice  $V_i^s$  para o vértice  $V_j^s$ , para todo  $i < j$ . É importante salientar, que ao representar a solução dessa maneira, o  $PMCP_{IRS}$  se torna um problema de ordenação de vetores, o que simplifica a resolução do mesmo.

### 4.3 Heurística fundamentada em Algoritmo Genético

Nesta seção, é descrito um algoritmo heurístico que fornece uma boa solução para o  $PMCP_{IRS}$ . Este algoritmo baseia-se na meta-heurística denominada Algoritmo Genético, onde, a partir de um conjunto inicial de soluções viáveis (população), a cada iteração obtém-se novas soluções que podem ser selecionadas para a próxima população, respeitando sempre as características de viabilidade das soluções futuras.

Abaixo é apresentado as entradas para o Algoritmo Genético, juntamente com os melhores valores encontrados para a calibração do algoritmo:

- $\beta$ : O tamanho das populações geradas, com  $\beta = 300$ ;
- $G$ : Um grafo direcionado representado por uma matriz de adjacência;
- $\alpha$ : O percentual de vértices que devem estar ativos ao fim do processo;
- $n$ : O número de vértices de  $G$ ;

- $\gamma$ : A quantidade de épocas do Algoritmo Genético, com  $\gamma = 200$ ;
- $P$ : Uma matriz com todos os possíveis contratos de incentivos para todos os vértices  $i \in G$ ;
- $\mu$ : A taxa de possíveis mutações, com  $\mu = 0.02$ ;
- $\theta$ : Taxa de tolerância de repetição dos melhores indivíduos entre épocas, com  $\theta = 50$ .

---

**Algoritmo 2:** Algoritmo Genético
 

---

**Input:**  $(\beta, G, \alpha, n, \gamma, P, \mu, \theta)$

**Output:** Melhor solução encontrada para o  $PMCP_{RS}$

```

1 populacao  $\leftarrow$  gerar_populacao( $\beta$ );
2 while criterio de parada nao for atendido do
3   | populacao  $\leftarrow$  populacao  $\cup$  reproducao(crossover, populacao);
4   | populacao  $\leftarrow$  selecionar_nova_populacao(populacao,  $\beta$ );
5 return Melhor solução encontrada

```

---

#### 4.3.1 Geração da População Inicial

Primeiramente, o método  $gerar\_populacao(\beta)$  gera um conjunto de soluções iniciais de tamanho  $\beta$ , denominado população inicial, fazendo a chamada do método  $nova\_solucao()$   $\beta$  vezes, que pode ser visto no Algoritmo 3. Este por sua vez irá retornar uma possível solução

aleatória  $S = (Z^s, V^s, \varphi^s)$ .

---

**Algoritmo 3:** Nova Solução

---

**Input:**  $(G, \alpha, n)$

**Output:**  $S(Z, V, \varphi)$

```

1  $Z \leftarrow$  matriz de  $n \times n$  posições;
2  $V \leftarrow$  vetor de  $n$  posições;
3  $T \leftarrow$  vetor com  $[0, 1, 2 \dots n]$ ;
4  $\varphi \leftarrow 0$ ;
5 while tamanho de  $V < \lceil \alpha n \rceil$  do
6      $\text{novo\_individuo} \leftarrow$  elemento aleatorio de  $T$ ;
7      $T.remove(\text{novo\_individuo})$ ;
8      $V.adiciona(\text{novo\_individuo})$ ;
9     for  $i : V[0 : -1]$  do
10         if  $G[\text{novo\_elemento}][i] > 0$  then
11              $Z[\text{novo\_elemento}][i] \leftarrow 1$ ;
12  $\varphi \leftarrow calcular\_aptidao(V, Z)$ ;
13 return  $S(Z, V, \varphi)$ ;
```

---

De forma mais detalhada, da linha 2 a linha 5, temos a instanciação das variáveis utilizadas. Note que o vetor  $T$  possui os vértices candidatos a entrarem na solução. O laço de repetição, da linha 6 a linha 15, acontecerá enquanto não forem selecionados os  $\lceil \alpha n \rceil$  vértices pertencentes a solução. Na linha 7 e 8, um novo vértice é selecionado de forma aleatória e removido de  $T$ . Na linha 9, o vértice removido de  $T$  é adicionado ao vetor de soluções  $V$ . O bloco da linha 10 a linha 14 é responsável por selecionar as arestas capazes de passar influência, sendo atualizada a matriz  $Z$  na linha 12. Por fim, a aptidão da nova solução é calculada na linha 16. Na linha 17, é retornada a nova solução completa.

### 4.3.2 Aptidão de uma Solução

A aptidão (eficiência) de uma solução é representado por  $\varphi$  e é calculada através da  $funcao\_objetivo(V^s, Z^s)$  que representa o valor total de contratação dos incentivos necessários para ativar  $|V^s| \geq \lceil \alpha |V| \rceil$ . Logo, a aptidão de uma solução é dada pelo valor de contratação dos menores incentivos possível necessários para ativar todos os vértices  $i \in V^s$ , podendo ser representada por:

$$\varphi(S) = \sum_{i \in V^s} \min(w_{ip} | f_i(\eta_i^s, p) \geq h_i, p \in P_i)$$

### 4.3.3 Critérios de Parada

Foram utilizados dois critérios de parada neste algoritmo: número máximo de gerações ( $\gamma$ ) e repetição de população. Para o segundo caso, considerou como repetição de população a circunstância quando os melhores indivíduos da população se mantiveram estáveis por um número  $\theta$  de gerações.

### 4.3.4 Crossover

Dado duas soluções  $S_i = (V^i, Z^i, \varphi^i)$  e  $S_k = (V^k, Z^k, \varphi^k)$ , o *crossover* (Algoritmo 4) é feito realizando dois cortes em locais aleatórios no gene relacionado a  $V^i$  e a  $V^k$ . Desta forma dividindo-os em três partes que denominaremos de:  $V_1^i, V_2^i, V_3^i, V_1^k, V_2^k$  e  $V_3^k$ . Após as partições realizadas, faz-se-á a junção de  $V_1^i$ , com  $V_2^k$  e  $V_3^i$  gerando uma possível nova combinação de vértices  $V_1^n$ . Da maneira similar, junta-se  $V_1^k$  com  $V_2^i$  e  $V_3^k$  para gerar  $V_2^n$ . Assim, duas novas soluções  $S_1^n = (V_1^n, Z_1^n, \varphi_1^n)$  e  $S_2^n = (V_2^n, Z_2^n, \varphi_2^n)$  são criadas, realizando também as ativações das arestas necessárias em  $Z_1^n$  e em  $Z_2^n$ . Por fim, é efetuado o cálculo de  $\varphi_1^n$  e  $\varphi_2^n$ . Caso haja repetição de alelos no gene  $V^n$ , em qualquer nova solução gerada, o indivíduo é eliminado

automaticamente.

---

**Algoritmo 4:** Crossover

---

**Input:**  $(S_i = (V^i, Z^i, \varphi^i), S_k = (V^k, Z^k, \varphi^k))$

**Output:**  $S_1^n(Z_1^n, V_1^n, \varphi_1^n), S_2^n(Z_2^n, V_2^n, \varphi_2^n)$

```

1  $Z_1^n \leftarrow$  matriz de  $n \times n$  posicoes;
2  $Z_2^n \leftarrow$  matriz de  $n \times n$  posicoes;
3  $V_1^n \leftarrow$  vetor de  $n$  posicoes;
4  $V_2^n \leftarrow$  vetor de  $n$  posicoes;
5  $\varphi_1^n \leftarrow 0$ ;
6  $\varphi_2^n \leftarrow 0$ ;
7  $S_1^n \leftarrow$  Falso;
8  $S_2^n \leftarrow$  Falso;
9  $corte\_1 \leftarrow$  Valor randomico entre 1 e  $n/2$ ;
10  $corte\_2 \leftarrow$  Valor randomico entre  $n/2$  e  $n - 1$ ;
11 if nao existe qualquer intersecao entre  $V^i[0 : corte\_1]$ ,  $V^k[corte\_1 : corte\_2]$  e
     $V^i[corte\_2 : ]$  then
12    $V_1^n \leftarrow V^i[0 : corte\_1] \cup V^k[corte\_1 : corte\_2] \cup V^i[corte\_2 : ]$ ;
13    $V_1^n \leftarrow$  realizar_mutacao( $V_1^n$ );
14    $Z_1^n \leftarrow$  preencher_matriz( $V_1^n$ )
15    $\varphi_1^n \leftarrow$  calcular_aptidao( $V_1^n, Z_1^n$ );
16 if nao existe qualquer intersecao entre  $V^k[0 : corte\_1]$ ,  $V^i[corte\_1 : corte\_2]$  e
     $V^k[corte\_2 : ]$  then
17    $V_2^n \leftarrow V^k[0 : corte\_1] \cup V^i[corte\_1 : corte\_2] \cup V^k[corte\_2 : ]$ ;
18    $V_2^n \leftarrow$  realizar_mutacao( $V_2^n$ );
19    $Z_2^n \leftarrow$  preencher_matriz( $V_2^n$ )
20    $\varphi_2^n \leftarrow$  calcular_aptidao( $V_2^n, Z_2^n$ );
21 return  $S_1^n(Z_1^n, V_1^n, \varphi_1^n), S_2^n(Z_2^n, V_2^n, \varphi_2^n)$ ;

```

---

De forma mais detalhada, da linha 2 a linha 11, é instanciadas as variáveis utilizadas, entre elas, na linha 10 e 11, é gerada as posições onde os vetores  $V^i$  e  $V^k$  serão cortados. A condição estabelecida na linha 12 garante a não repetição de vértices no gene relacionado a  $V^n$ . Na linha 13,  $V^n$  é preenchido e passa por uma possível mutação na linha 14. Na linha 15, a matriz  $Z^n$  é preenchida de forma que o vértice  $V^n[i]$  não possua arestas propagando influência

para qualquer  $V^n[j]$  para todo  $j < i$ . Na linha 16 é calculada a aptidão de  $S_1^n$ . De mesma forma, da linha 18 a linha 23, o processo se repete. Por fim, é retornado as novas soluções, caso existam.

#### 4.3.5 Mutação

Dado uma solução  $S^n = (V^n, Z^n, \varphi^n)$ , a mutação é realizada considerando o gene  $V^n$ , da seguinte forma: seleciona-se um número aleatório entre 0 e  $n\mu$ , representando a quantidade de alelos que podem sofrer mutação. Dado a quantidade de vértices mutáveis, escolhidos de forma randômica, substitui-se cada vértice por vértices que não estão presente em  $V^n$ . Dessa forma, a mutação não é realizada quando  $\alpha = 1.0$ . Sendo assim, temos:

---

##### Algoritmo 5: Mutação

---

**Input:**  $(V^n, \mu, n)$

**Output:**  $V^n$

```

1 begin
2   contador  $\leftarrow$  0;
3   while contador < valor aleatorio entre 0 e nμ do
4     posicao  $\leftarrow$  valor_randomico(0, tamanho( $V^n$ ));
5      $f_1 \leftarrow V^n[\textit{posicao}]$ ;
6      $f_2 \leftarrow \textit{valor\_randomico}$ (0,  $n - 1$ );
7     while  $V^n \textit{conter } f_2 \textit{ ou } f_1 \textit{ igual a } f_2$  do
8        $f_2 \leftarrow \textit{valor\_randomico}$ (0,  $n - 1$ );
9      $V^n[\textit{posicao}] \leftarrow f_2$ ;
10    contador  $\leftarrow$  contador + 1;
11    return  $V^n$ ;
12
```

---

#### 4.3.6 Seleção da Nova População

Ao final de cada geração (iteração) do algoritmo, uma nova população deve ser selecionada. Todas as soluções geradas pelo *crossover* são adicionadas a população. Por fim, a população é ordenada pelos valores de aptidão e são selecionadas as soluções para a próxima época, onde 80% delas fazem parte dos melhores indivíduos e 20% dos piores. Dessa forma, é possível garantir uma maior variabilidade genética.

## 5 RESULTADOS

Neste capítulo, são comparadas duas implementações para o  $PMCPIRS$ : o algoritmo exato proposto por Fischetti *et al.* (2018) e o algoritmo heurístico proposto neste trabalho. Na Seção 5.1 é apresentado o ambiente computacional utilizado para realização dos testes. Na Seção 5.2 são apresentadas as principais características das instâncias utilizadas na realização dos testes computacionais. Por fim, é realizado, na Seção 5.3, um estudo comparativo entre as implementações do modelo exata existente na literatura e o algoritmo proposto. Vale salientar, que os parâmetros utilizados na heurística proposta foram ajustados a partir de testes empíricos.

### 5.1 Configuração do Ambiente Computacional

A implementação da heurística apresentada neste trabalho foi realizada na linguagem de programação Python, versão 3.5.6. O algoritmo exato implementado segue as mesmas restrições de Fischetti *et al.* (2018), a linguagem de programação utilizada foi Java com uso do pacote de otimização CPLEX - versão 12.8, sendo limitado pelo tempo de execução de 2 horas e o uso de memória de 8GB. Os experimentos realizados foram executados utilizando uma máquina com processador Intel Core i5-7200U,  $4 \times 2.50$  GHz, 8 GB de memória RAM e sistema operacional Windows 10.

### 5.2 Instâncias

As duas implementações para o  $PMCPIRS$ , tanto a exata quanto a heurística, foram avaliadas através de testes computacionais utilizando 108 instâncias empregadas em Fischetti *et al.* (2018) e geradas da seguinte forma: primeiro são gerados três mundos de grafos direcionados (WATTS; STROGATZ, 1998). Dentro de cada mundo é gerada uma instância para cada conjunto de tamanho de nós, onde  $|V| \in \{50, 75, 100\}$  com grau médio dos nós  $k \in \{4, 8\}$ . A probabilidade de religação  $\beta \in \{0.1, 0.3\}$ . Os valores de influência  $d_{ij}$  do arco  $(i, j) \in A$  são distribuídos de forma pseudo-randômica uniforme em  $\{1, \dots, 10\}$ . Já a quantidade total de influência que pode ser exercida sobre o nó  $i$  vindo de seus vizinhos é representado por  $D_i = \sum_{j \in N_i} d_{ji}$ . Então,  $h_i = \max\{1, \min\{\zeta_i, D_i\}\}$  é arredondado para o próximo número inteiro, onde  $\zeta_i \sim N(0.7D_i, D_i/|N_i|)$  é uma variável aleatória normalmente distribuída. A fração mínima de nós a serem ativados é definida como  $\alpha \in \{0.1, 0.5, 1.0\}$ . Além disso, o conjunto discreto de incentivos baseia-se no maior valor de obstáculo de todos os nós  $\hat{h} = \max_{i \in V} h_i$  e é definido como

$P_i = \{0, 0.25\hat{h}, 0.5\hat{h}, 0.75\hat{h}, \hat{h}\}$  para todo  $i \in V$ . Por fim, o custo de utilizar um incentivo não é necessariamente proporcional ao seu valor, sendo assim, foi utilizado uma escala econômica definida como  $w_{ip} := p^{0.9}$ .

### 5.3 Estudo Comparativo entre os Algoritmos

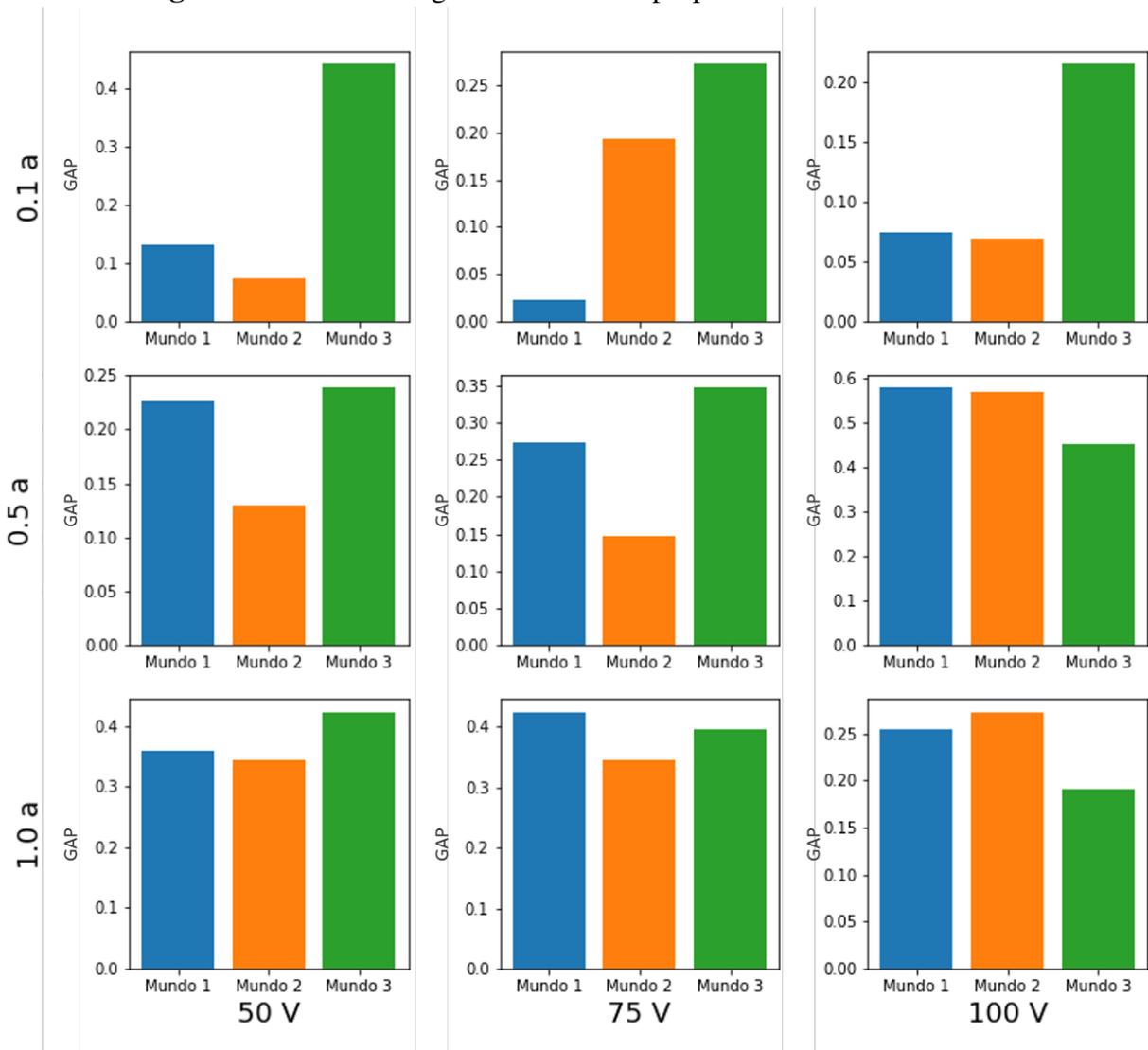
A Tabela 1 apresenta os resultados médios tanto para o GAP, quanto o tempo médio de resolução, em segundos, para cada mundo de instância gerado. Como é possível observar quanto menor o número de vértices que se deve ativar ao fim do processo, menor são os GAPs obtidos pelo algoritmo. Em contra partida, quando  $\alpha = 1$ , os tempos de execução do algoritmo caem drasticamente, isso se deve ao fato de que nas instâncias com essa configuração o  $PMCP_{RS}$  se torna um problema cujo objetivo é trocar o posicionamento das casas de um vetor.

Os gráficos apresentados na Figura 2 mostram o quanto os GAPs podem variar de um mundo de instância para outro. Possuindo uma maior constância em instâncias difíceis, ou seja, onde o  $\alpha$  é igual a 1.0, sendo necessário ativar todos os vértices. Embora o algoritmo proposto ainda tenha grandes dificuldades de encontrar boas soluções para as instâncias difíceis, seu tempo de processamento sempre é abaixo, 43.8727 segundos. Em muitos destes casos, o algoritmo exato utiliza todo o tempo disponível (7200 segundos - 2 horas) sem encontrar a solução ótima, (*timeout*).

Tabela 1 – Análise comparativa entre as diferentes instâncias da heurísticas proposta.

V	$\alpha$	Mundo 1		Mundo 2		Mundo 3	
		GAP	Tempo	GAP	Tempo	GAP	Tempo
50	0.1	0.13	7.6183	0.07	6.7181	0.43	14.0141
	0.5	0.22	29.4865	0.12	33.7587	0.23	37.1645
	1.0	0.35	1.7869	0.34	1.4274	0.42	1.7392
75	0.1	0.02	26.2532	0.19	28.8951	0.27	24.2473
	0.5	0.27	11.5847	0.14	43.8727	0.34	24.1319
	1.0	0.42	3.2602	0.34	2.6085	0.39	3.9039
100	0.1	0.07	7.8461	0.06	26.1471	0.21	25.7803
	0.5	0.57	41.0137	0.56	22.2165	0.45	22.4121
	1.0	0.25	4.6845	0.27	4.5347	0.19	5.3953

**Figura 2** – GAPs do Algoritmo Genético proposto.



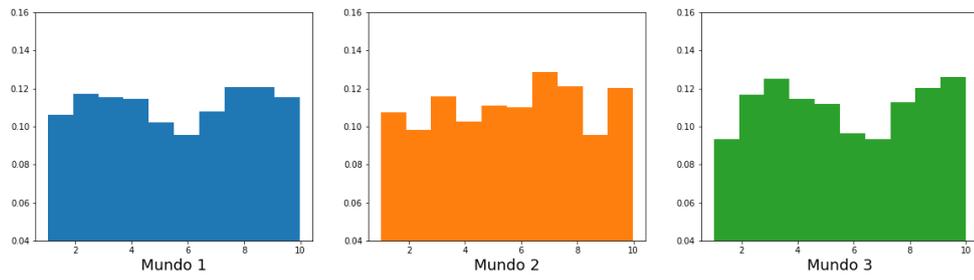
Fonte: Elaborado pelo autor (2019).

Visando entender a grande variação de GAP do mundo 3 para o mundo 1 e 2, a Figura 3 mostra de forma gráfica a comparação dos pesos das arestas presentes nas instâncias de 50 vértices e  $\alpha = 1.0$  de cada mundo. De forma quantitativa, a Tabela 2 traz a Variância (medida de dispersão, que indica quanto os dados estão afastados da média) dos dados da Figura 3. Dessa forma, podemos observar que um dos fatores que justifique o comportamento adverso do GAP das instâncias geradas no Mundo 3 seja a maior variância das suas arestas.

Tabela 2 – Comparação das instâncias com  $|V| = 50$  e  $\alpha = 1.0$ .

	<b>Mundo 1</b>	<b>Mundo 2</b>	<b>Mundo 3</b>
<b>Variância</b>	6.80	6.64	<b>6.91</b>

**Figura 3** – Comparação das instâncias com  $|V| = 50$  e  $\alpha = 1.0$ .



Fonte: Elaborado pelo autor (2019).

## 6 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho apresentou uma solução através de um algoritmo heurístico para o Problema de Minimização de Custo de Propagação de Influência em Redes Sociais (*PMCP<sub>IRS</sub>*). Técnica ainda não apresentada na literatura. Este é um problema do mundo real, aplicável não apenas a empresas de *marketing*, como também a qualquer um que deseje encontrar os principais influenciadores digitais para propagar uma ideia em uma rede social, visando gastar o menor valor possível em um tempo razoável.

Foi apresentado um algoritmo baseado na meta-heurística denominado algoritmo genético e posteriormente realizados testes computacionais para as instâncias. Os resultados obtidos foram comparados com as soluções alcançadas com o método exato descrito em Fischetti *et al.* (2018). Em termos de GAP, o algoritmo proposto nesse trabalho apresentou bons resultados para instâncias com  $\alpha \in \{0.1, 0.5\}$ , ou seja, o resultado obtido pelo algoritmo heurístico se aproximou bastante do resultado obtido pelo algoritmo exato para determinadas instâncias. Entretanto, para instâncias difíceis,  $\alpha = 1.0$ , o algoritmo apresentado não demonstrou bons resultados. Além disso, apesar de não encontrar o ótimo em muitos casos, o algoritmo proposto demonstrou ser eficiente em questão de tempo de execução. Como trabalhos futuros, espera-se alimentar algoritmo proposto com dados reais coletados através de *web-crawlers*. Além disso, outras heurísticas e meta-heurísticas podem ser investigadas para este novo problema, do mesmo modo que outros parâmetros para o algoritmo podem ser testados, afim de solucionar as falhas apresentadas pelo algoritmo proposto.

## REFERÊNCIAS

- ACKERMAN, E.; BEN-ZWI, O.; WOLFOVITZ, G. Combinatorial model and bounds for target set selection. **Theoretical Computer Science**, v. 411, n. 44, p. 4017 – 4022, 2010. ISSN 0304-3975. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0304397510004561>>.
- Caliò, A.; Tagarelli, A. Trust-based dynamic linear threshold models for non-competitive and competitive influence propagation. In: **2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)**. [S.l.: s.n.], 2018. p. 156–162. ISSN 2324-9013.
- CHURCH, R.; REVELLE, C. The maximal covering location problem. **Papers of the Regional Science Association**, v. 32, n. 1, p. 101–118, Dec 1974. ISSN 1435-5957. Disponível em: <<https://doi.org/10.1007/BF01942293>>.
- FISCHETTI, M.; KAHR, M.; LEITNER, M.; MONACI, M.; RUTHMAIR, M. Least cost influence propagation in (social) networks. **Mathematical Programming**, Springer, v. 170, n. 1, p. 293–325, 2018.
- GRANOVETTER, M. Threshold models of collective behavior. **American Journal of Sociology**, v. 83, n. 6, p. 1420–1443, 1978. Disponível em: <<https://doi.org/10.1086/226707>>.
- GÜNEY, E.; LEITNER, M.; RUTHMAIR, M.; SINNL, M. Large-scale influence maximization via maximal covering location. 12 2018.
- HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. [S.l.]: McGraw Hill Brasil, 2013.
- HOLLAND, J. H. *et al.* **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: MIT press, 1992.
- IINTERATIVA. **Infográfico – O que mudou no mercado dos influenciadores?** 2018. [Http://www.iinterativa.com.br/infografico-mudou-mercado-dos-influenciadores/](http://www.iinterativa.com.br/infografico-mudou-mercado-dos-influenciadores/).
- KEMPE, D.; KLEINBERG, J.; TARDOS, É. Maximizing the spread of influence through a social network. In: ACM. **Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining**. [S.l.], 2003. p. 137–146.
- LEÃO, L. **Derivas: cartografias do ciberespaço**. [S.l.]: AnnaBlume, 2004.
- WATTS, D. J.; STROGATZ, S. H. Collective dynamics of ‘small-world’ networks. **nature**, Nature Publishing Group, v. 393, n. 6684, p. 440, 1998.