

Explorando Horizontes Determinísticos: A Síntese de Scatter Search e Busca Tabu para Reduzir Tempos no Problema de Tempo de Transmissão Mínimo

Pedro Baldotto¹, Rafael Crevelari¹

¹Departamento de Informática – Universidade Federal de Viçosa (UFV)
Viçosa – MG – Brasil

pedro.baldotto@ufv.br, rafael.crevelari@ufv.br

Abstract. *This paper addresses the Minimum Transmission Time problem, which involves disseminating data among devices, where each device can transmit information only to its adjacent devices in a sequential manner. The goal is to send a message to all devices from a set of source devices. The problem is applicable in distributed systems, such as traffic light synchronization and clock synchronization. Being an NP-complete problem, an efficient exact solution is not expected. Therefore, the study of heuristics and metaheuristics is crucial. In this work, we present two deterministic algorithms: Scatter Search and Tabu Search to solve the Minimum Transmission Time problem. The proposed algorithm was compared with existing stochastic and approximate techniques for the problem, and the experimental results demonstrated the superiority of our algorithm over these techniques.*

Keywords: *Dissemination in Graphs, Minimum Broadcast Time, MBT, Population Algorithms, Scatter Search e Busca Tabu*

Resumo. *Este documento aborda o problema do Tempo de Transmissão Mínimo, que consiste em disseminar dados entre dispositivos, onde cada dispositivo pode transmitir informações apenas para seus dispositivos adjacentes, de forma sequencial. O objetivo é enviar uma mensagem a todos os dispositivos a partir de um conjunto de dispositivos fonte. O problema é aplicável em sistemas distribuídos, como sincronização de semáforos e relógios. Por ser um problema NP-completo, não se espera uma solução exata eficiente. Portanto, o estudo de heurísticas e metaheurísticas é fundamental. Neste trabalho, apresentamos dois algoritmos determinísticos: Scatter Search e Busca Tabu para resolver o Tempo de Transmissão Mínimo. O algoritmo proposto foi comparado com técnicas estocásticas e aproximadas existentes para o problema, e os resultados experimentais demonstraram a superioridade do nosso algoritmo em relação a essas técnicas.* **Palavras-chave:** *Disseminação em Grafos, Tempo de Transmissão Mínimo, Algoritmo Populacional, Metaheurísticas, Scatter Search e Busca Tabu.*

1. Introdução

A disseminação e transferência de dados em redes de computadores são tópicos amplamente explorados, com inúmeras pesquisas buscando melhorar esses aspectos. No entanto, ainda existem lacunas a serem estudadas, e um exemplo disso é o problema do

Tempo de Transmissão Mínimo (MBT - Minimum Broadcast Time) [Farley et al. 1979]. O problema pode ser descrito da seguinte maneira: suponha que um ou vários dispositivos possuam uma mensagem a ser transmitida para os demais dispositivos. Todos os dispositivos pertencem à mesma rede e devem receber a mensagem no menor tempo possível, considerando uma escala discreta. No entanto, existem algumas limitações. Cada dispositivo só pode realizar uma transferência por vez para seus vizinhos imediatos. Quando um dispositivo recebe a mensagem, ele também é responsável por enviá-la para seus vizinhos imediatos correspondentes.

A Figura 1 ilustra um simples exemplo de rede com o V1 detendo a informação inicial para entendimento do MBT. Essa configuração inicial possui duas possíveis soluções, uma viável e outra ótima, que são ilustradas pelas Figuras 2 e 3, respectivamente.

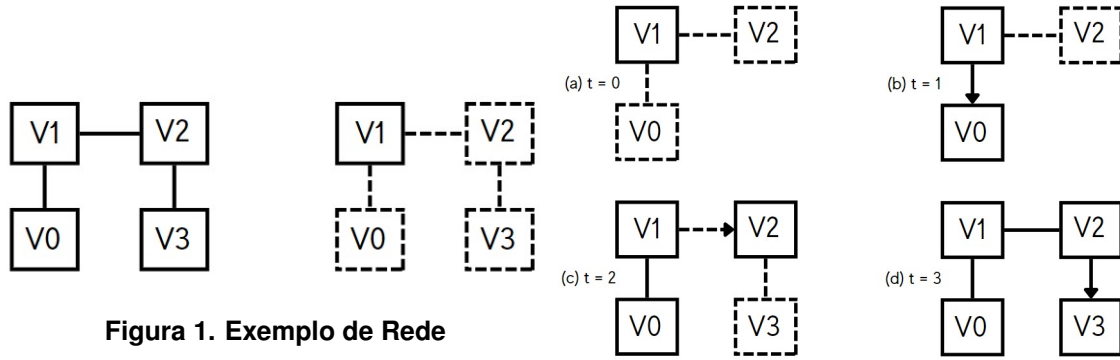


Figura 1. Exemplo de Rede

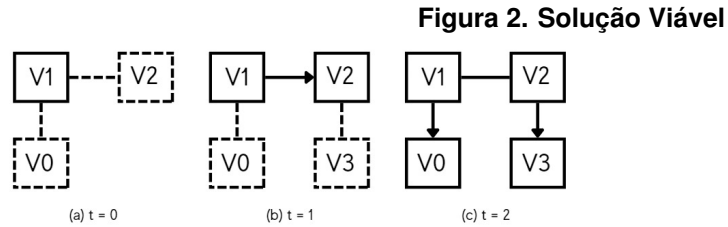


Figura 2. Solução Viável

Figura 3. Solução Ótima

Neste documento, exploramos duas meta-heurísticas determinísticas para abordar eficientemente o problema do Tempo de Transmissão Mínimo.

2. Trabalhos Relacionados

Para embasar nossos estudos e comparação de resultados, analisamos o artigo de [Silva et al. 2020], que propõe uma solução totalmente estocástica para o problema do Tempo de Transmissão Mínimo. Buscaremos melhorar os resultados obtidos nesse trabalho. Além disso, também nos baseamos no artigo proposto por [de Sousa et al. 2018], que apresenta dois métodos determinísticos capazes de encontrar soluções ótimas para alguns dos experimentos propostos.

3. Representação da Solução

Em nossa abordagem, utilizamos uma representação de solução baseada em um vetor de prioridade, onde o valor de cada posição representa um vértice. A ordem dos elementos

no vetor reflete a prioridade dos dispositivos, sendo que estar mais à esquerda indica uma maior prioridade tanto para enviar quanto para receber informações.

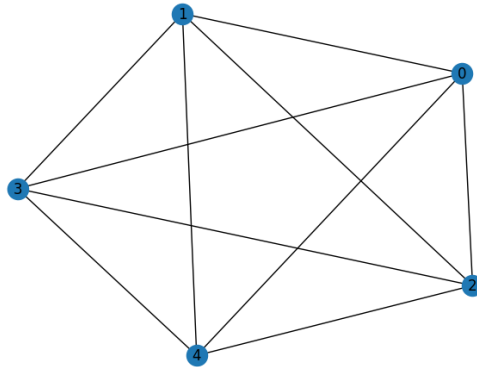


Figura 4. Grafo de Harari (H4,5)

Com base na análise da estrutura do grafo apresentado na Figura 4, é possível identificar e extrair várias soluções potenciais. Ao examinar os diferentes caminhos ou conjuntos de arestas do grafo, podemos identificar diferentes trajetórias que levam a um resultado desejado. Cada trajetória ou conjunto de arestas representa uma solução viável para o problema em questão. A seguir, apresentamos algumas das possíveis soluções para o grafo da Figura 4.

- **Solução 1:** $[0,4,2,1,3]$
- **Solução 2:** $[4,1,3,2,0]$
- **Solução 3:** $[1,4,2,0,3]$

3.1. Avaliação de Solução

Uma vez que tenhamos uma solução em mãos, é necessário realizar a sua avaliação. Utilizaremos o exemplo anterior para ilustrar o processo de cálculo da avaliação da solução.

Passo a passo na execução no Tempo 1 (Figura 5):

Solução: $[0,4,2,1,3]$

Lista Visitados: $[V,F,F,F,V]$

Devemos verificar o vizinho de 0 com maior prioridade, que ainda não detêm a informação. Nesse caso 4, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.

Atualização Lista Visitados: $[V,F,F,F,V]$

Como não existe outro item que detêm a informação (passada no início do tempo), não há mais iterações do tempo 1.

Passo a passo na execução no Tempo 2 (Figura 6):

Solução: $[0,4,2,1,3]$

Lista de Visitados: $[V,F,F,F,V]$

Devemos verificar o vizinho de 0 com maior prioridade, que ainda não detêm a informação. Nesse caso 2, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.

Atualização Lista Visitados: $[V,F,V,F,V]$

Além disso, devemos verificar o vizinho de 4 com maior prioridade, que ainda não detêm a informação. Nesse caso 1, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.

Atualização Lista Visitados: [V,V,V,F,V]

Como não existe outro item que detêm a informação (passada no início do tempo), acabaram as iterações do tempo 2.

Passo a passo na execução no Tempo 3 (Figura 7):

Solução: [0,4,2,1,3]

Lista de Visitados: [V,V,V,F,V]

Devemos verificar o vizinho de 0 com maior prioridade, que ainda não detêm a informação. Nesse caso 3, desse modo, enviamos a informação para ele, e atualizamos a lista de visitados.

Atualização Lista de Visitados: [V,V,V,V,V]

Como a lista de visitados foi completa, retornamos o tempo atual, ou seja, 3.

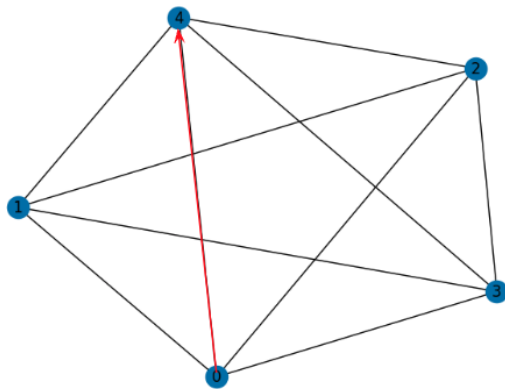


Figura 5. Vértice com Informação no Tempo 1

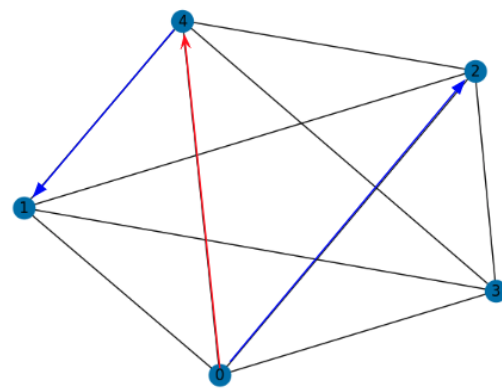


Figura 6. Vértice com Informação no Tempo 2

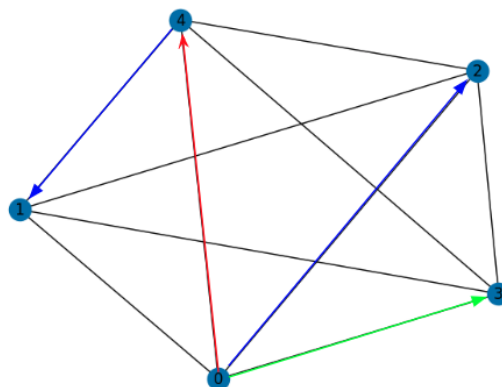


Figura 7. Vértice com Informação no Tempo 3

4. Scatter Search para o MBT

Optamos por utilizar o algoritmo Scatter Search para abordar o problema do Tempo de Transmissão Mínimo (MBT) devido à sua natureza determinística. A escolha de um algoritmo determinístico como o Scatter Search permite uma abordagem mais controlada e

previsível na busca por soluções otimizadas para o MBT. Através desse algoritmo, buscamos explorar de forma sistemática o espaço de busca, combinando diferentes soluções parciais para encontrar soluções que minimizem o tempo de transmissão. Essa abordagem determinística nos permite obter resultados mais consistentes e repetíveis, contribuindo para uma melhor compreensão e análise do problema do Tempo de Transmissão Mínimo.

Algorithm 1 Algoritmo Scatter Search

```

populacao ← GerarPopulacao()
conjuntoReferencia ← CriarConjuntoDeReferencia()
iteracoesSemMelhora ← 0
while iteracoesSemMelhora < 5 do
    pais ← gerarPais()
    filhos ← gerarFilhos()
    melhorIndividuoAntesAtualizar ← conjuntoReferencia[0]
    atualizarConjuntoReferencia()
    melhorIndividuoAposAtualizar ← conjuntoReferencia[0]
    if melhorIndividuoAntesAtualizar ≠ melhorIndividuoAposAtualizar then
        | iteracoesSemMelhora ← 0
    end
    else
        | iteracoesSemMelhora ← iteracoesSemMelhora + 1
    end
end
return conjuntoReferencia[0].solucao

```

4.1. Representação do Indivíduo

Assim como em outros algoritmos populacionais, é importante compreender a representação dos indivíduos em nosso contexto. Cada indivíduo é codificado por meio de um vetor de números, conhecido como cromossomo. Esse cromossomo é responsável por codificar uma possível solução para o problema do Tempo de Transmissão Mínimo (MBT). Por exemplo, considere o seguinte cromossomo:

- **Cromossomo:** $[3, 2, 0, 4, 1]$

Nesse caso, cada elemento do cromossomo representa a prioridade associada a um dispositivo específico. A ordem dos valores no cromossomo indica a ordem de prioridade para a transmissão das informações. Ao aplicar o cromossomo em um processo de decodificação, obtemos a solução gerada:

- **Solução Gerada:** $[2, 4, 1, 0, 3]$

Assim, o valor presente em $\text{Cromossomo}[i]$ (valor real) indica qual é a prioridade associada ao dispositivo V_i na solução gerada. Essa representação por meio de cromossomos nos permite trabalhar com diferentes configurações e buscar soluções ótimas para o problema do Tempo de Transmissão Mínimo.

4.2. Geração da População

Em seguida, a população inicial é gerada de forma determinística, combinando diferentes tipos de indivíduos. Os cromossomos são criados a partir de permutações dos grupos da solução trivial ($[0, 1, 2, 3, \dots, n-1]$) e das permutações do inverso da solução trivial.

Além disso, uma abordagem gulosa é aplicada utilizando o grafo do experimento. Os vértices de maior grau são selecionados e colocados em uma lista de prioridade em ordem decrescente. Essa solução gulosa é então transformada em um cromossomo de um indivíduo, que servirá como base para a criação dos indivíduos restantes por meio da permutação desses cromossomos.

Com base nesses diferentes tipos de indivíduos, nossa população inicial é dividida da seguinte maneira:

- **1/3 da população:** *é gerada a partir das permutações do cromossomo trivial.*
- **1/3 da população:** *é gerada a partir das permutações do cromossomo trivial inverso.*
- **1/3 da população:** *é gerada a partir das permutações do cromossomo guloso.*

Essa estratégia garante que a população inicial seja diversificada, abrangendo uma ampla gama de possíveis soluções. Ao combinar a criação de cromossomos a partir das permutações do trivial, a abordagem gulosa e a permutação dos cromossomos, buscamos explorar diferentes aspectos do problema e aumentar as chances de encontrar soluções ótimas.

Dessa forma, essa população inicial variada e bem explorada nos permite realizar uma busca mais abrangente e eficiente por soluções ótimas.

4.3. Criação do Conjunto de Referência

Após definir um tamanho para o conjunto de referência, é necessário criar esse conjunto a partir da população. A criação do conjunto de referência segue a seguinte abordagem:

- **1/2 do conjunto de referência:** *Consiste nos indivíduos da população que obtiveram os menores tempos de transmissão para as instâncias do problema. Esses indivíduos são selecionados com base no critério de desempenho, ou seja, aqueles que alcançaram os melhores resultados em termos de tempo de transmissão.*
- **1/2 do conjunto de referência:** *Inclui os indivíduos que não estão presentes no conjunto de referência e possuem o maior número de valores diferentes em suas soluções em comparação com todos os indivíduos que já estão no conjunto de referência.*

Essa abordagem visa garantir que o conjunto de referência seja composto tanto pelos melhores indivíduos em termos de desempenho quanto por indivíduos que apresentam soluções distintas das já existentes. Dessa forma, o conjunto de referência abrange uma diversidade de soluções promissoras, proporcionando um leque de opções a serem exploradas durante o processo de otimização. O conjunto de referência está sempre ordenado em forma crescente de soluções dos indivíduos, ou seja, o primeiro indivíduo é sempre o melhor.

Essa estratégia de criação do conjunto de referência contribui para uma busca mais abrangente no espaço de soluções e ajuda a evitar a convergência prematura para soluções subótimas.

4.4. Criação dos Pais

Após a criação do conjunto de referência, é necessário criar o conjunto de pais, que consiste em todas as combinações possíveis de indivíduos presentes no conjunto de referência, tomados dois a dois. Essa etapa visa gerar uma variedade de combinações para

explorar diferentes interações entre os indivíduos. Por exemplo, considere o seguinte exemplo:

- **Indivíduos:** [1,2,3]
- **Conjunto de Pais::** [1-2,1-3, 2-1, 2-3, 3-1, 3-2]

Essa abordagem permite explorar diferentes interações entre os indivíduos, facilitando a busca por soluções promissoras para o problema.

4.5. Criação dos Filhos

Com o conjunto de pais formado, eles são submetidos a quatro tipos de reprodução para gerar novos filhos. Todas as reproduções ocorrem em torno dos cromossomos dos indivíduos. Os quatro métodos de reprodução são os seguintes:

- **Primeira Reprodução: Média** - *Os cromossomos dos pais são combinados através da média aritmética. O cromossomo do filho é obtido calculando a média entre os cromossomos dos pais.*
- **Segunda Reprodução: Média Geométrica** - *Os cromossomos dos pais são combinados através da média geométrica. O cromossomo do filho é obtido calculando a raiz quadrada do produto dos cromossomos dos pais.*
- **Terceira Reprodução: Intercalada** - *Os cromossomos dos pais são intercalados de forma alternada. O primeiro gene é herdado do pai, o segundo gene da mãe, o terceiro gene do pai e assim por diante.*
- **Quarta Reprodução: Distância Euclidiana** - *Os cromossomos dos pais são combinados utilizando a distância euclidiana. O cromossomo do filho é obtido calculando o quadrado da diferença entre os cromossomos dos pais.*
- **Quinta Reprodução: Busca Local** - *O primeiro indivíduo do par de pais é submetido a uma busca local usando a heurística de "first improvement" e movimentos de swap em sua solução gerada. Em seguida, esse novo cromossomo é gerado. Se essa nova solução for melhor que a solução do outro indivíduo do par de pais, o filho é criado com esse novo cromossomo. Caso contrário, o cromossomo do indivíduo que não sofreu a busca local é propagado para o filho.*

Esses métodos de reprodução são empregados para criar variedade genética nos filhos, combinando os cromossomos dos pais de diferentes maneiras. Cada método possui suas próprias características e pode levar a diferentes resultados na prole, enriquecendo a diversidade e explorando diferentes regiões do espaço de busca.

4.6. Atualização do Conjunto de Referência

Após a geração dos filhos, é importante realizar uma troca entre os melhores filhos e os piores indivíduos do conjunto de referência. Nesse processo, caso um filho seja considerado superior a qualquer indivíduo do conjunto de referência, ele substituirá o membro correspondente no conjunto.

Essa estratégia de substituição tem como objetivo promover a evolução contínua da população, privilegiando os filhos mais promissores em relação aos indivíduos menos adaptados presentes no conjunto de referência. Dessa forma, o conjunto de referência é constantemente atualizado com os indivíduos mais aptos, permitindo que a próxima geração seja formada por descendentes que possuam características genéticas superiores.

4.7. Iteração sem Melhora

Após uma iteração sem melhora, ou seja, quando o indivíduo com o menor tempo encontrado não apresenta melhorias em relação às iterações anteriores, é necessário alterar o método de reprodução utilizado. Nesse caso, adota-se uma abordagem sequencial de tentativas, em que são testados diferentes métodos de reprodução até encontrar aquele que possa gerar melhorias no desempenho.

Se, em qualquer uma das tentativas, houver melhora no desempenho, os métodos de reprodução são reiniciados, e o processo continua iterando. Porém, caso nenhuma melhora seja obtida após todas as tentativas, o algoritmo do Scatter Search é finalizado, retornando a solução do melhor indivíduo.

5. Busca Tabu para o MBT

A solução inicial é o vetor de prioridade guloso(conforme descrito anteriormente). Os movimentos de melhoria são baseados na troca de elementos (swap), testando todas as combinações possíveis. Os movimentos válidos são aqueles que não estão na lista tabu ou que atendem ao critério de aspiração (melhor solução). Após encontrar todos os movimentos válidos, escolhemos o melhor e atualizamos a lista tabu. Esse processo de busca é repetido até que um tempo definido seja alcançado.

Algorithm 2 Algoritmo de Busca Tabu

Entrada: solucaoInicial, custoSolucaoInicial, limiteDeTempo

Saída : melhorSolucao

solucaoCorrente \leftarrow *solucaoInicial*

custoSolucaoCorrente \leftarrow *custoSolucaoInicial*

melhorSolucao \leftarrow *solucaoInicial*

custoMelhorSolucao \leftarrow *custoSolucaoInicial*

listaTabu $\leftarrow \emptyset$

while *limiteDeTempo* **do**

movimento \leftarrow *melhorMovimentoVizinho*(*solucaoCorrente*)

if $|listaTabu| = tamanhoLimite$ **then**

listaTabu \leftarrow *listaTabu* $- \{movimentoMaisAntigo\}$

listaTabu \leftarrow *listaTabu* $+ \{movimento\}$

end

else

listaTabu \leftarrow *listaTabu* $+ \{movimento\}$

end

solucaoCorrente \leftarrow *gerarSolucaoCorrente*(*movimento*)

custoSolucaoCorrente \leftarrow *f*(*solucaoCorrente*)

if *custoSolucaoCorrente* $<$ *custoMelhorSolucao* **then**

melhorSolucao \leftarrow *solucaoCorrente*

custoMelhorSolucao \leftarrow *custoSolucaoCorrente*

end

end

return *melhorSolucao*

6. Resultados Experimentais

Todas as instâncias utilizadas em nossos experimentos são grafos de Harari, também mencionados nos artigos propostos por [Silva et al. 2020] e [de Sousa et al. 2018]. Nosso objetivo principal é obter resultados superiores, reduzindo o tempo de execução em comparação com suas implementações.

Para obter os melhores resultados em nossos experimentos, optamos por utilizar os seguintes parâmetros em nossos algoritmos:

- **Número de Indivíduos:** 800
- **Tamanho do Conjunto de Referência:** 15
- **Tamanho Lista Tabu:** *Número de Vértices do Grafo * 0,30*
- **Tempo Máximo Tabu:** 120 segundos

Após aplicarmos os algoritmos desenvolvidos nas instâncias, obtivemos os seguintes resultados e o respectivo tempo em segundos:

Instância	TreeBlock	BRKGA (tempo)	Scatter Se- arch (tempo)	Busca Tabu (tempo)
H2,17	9 (-)	9 (0,20)	9 (0,02)	9 (2)
H2,30	15 (-)	15 (0,40)	15 (0,08)	15 (2)
H2,50	25 (-)	25 (0,11)	25 (0,37)	25 (2)
H2,100	50 (-)	50 (0,38)	50 (3,84)	50 (3,56)
H3,17	5 (-)	5 (0,20)	5 (0,01)	5 (2)
H3,30	9 (-)	9 (0,40)	9 (0,06)	9 (2)
H3,50	14 (-)	14 (0,90)	14 (0,28)	14 (2)
H5,17	5 (-)	5 (0,20)	5 (0,01)	5 (2)
H8,30	6 (-)	5 (1,15)	5 (0,08)	5 (2)
H11,50	7 (-)	6 (0,10)	6 (0,15)	6 (2)
H6,17	5 (-)	5 (0,20)	5 (0,01)	5 (2)
H9,30	6 (-)	5 (0,60)	5 (0,04)	5 (2)
H20,50	6 (-)	5 (0,14)	5 (0,18)	5 (2)
H7,17	5 (-)	5 (0,30)	5 (0,01)	5 (2)
H10,30	6 (-)	5 (0,60)	5 (0,05)	5 (2)
H21,50	7 (-)	6 (0,14)	6 (0,15)	6 (2)

Tabela 1. Tabela de Resultados dos Experimentos

Esses resultados mostram o desempenho e a eficácia dos algoritmos propostos na busca por soluções com tempo de transmissão mínimo. Eles representam uma melhoria significativa em relação aos métodos previamente utilizados, proporcionando soluções ótimas em todos os casos, além de mais rápidas e eficientes na maioria dos casos. Esses resultados validam a abordagem adotada e demonstram o potencial dos algoritmos desenvolvidos para lidar com o problema do tempo de transmissão mínimo em grafos de Harari.

7. Implementação

O conteúdo utilizado no desenvolvimento deste projeto está disponível no repositório do Github: <https://github.com/PedroFiorio/Trabalho-Final-INF-284>.

O repositório contém a implementação dos algoritmos utilizados neste documento. Além disso, no repositório existe informações detalhadas sobre como utilizar a implementação corretamente.

Certifique-se de visitar o repositório para obter acesso ao código-fonte, recursos e instruções de uso adequado.

8. Conclusão e Trabalhos Futuros

Em resumo, o trabalho desenvolvido foi altamente produtivo, resultando em avanços significativos no campo da busca por soluções para o problema do Tempo Mínimo de Transmissão (Minimum Broadcast Time). Nossa abordagem se destaca ao lidar com contextos em que a aleatoriedade na construção das soluções não é viável. Nos testes realizados, conseguimos obter resultados ótimos para as instâncias utilizadas por [de Sousa et al. 2018].

Como trabalhos futuros, pretendemos expandir nossa pesquisa encontrando novas instâncias desafiadoras para o problema, a fim de realizar testes mais robustos em nossos algoritmos. Além disso, buscamos enriquecer ainda mais nossos algoritmos com novas ideias e aprimoramentos, visando a melhoria contínua e a obtenção de soluções ainda melhores para o problema do Tempo Mínimo de Transmissão.

9. Atividades da Dupla

Durante o projeto, as atividades foram conduzidas em dupla utilizando a técnica de pair-programming. Essa abordagem envolve o acompanhamento simultâneo do desenvolvimento do código, com um membro do grupo observando enquanto o outro codifica, e alternando esses papéis periodicamente. Essa prática garantiu que ambos os membros da equipe adquirissem um conhecimento TOTAL de toda a implementação do projeto.

Referências

- de Sousa, A., Gallo, G., Gutierrez, S., Robledo, F., Rodríguez-Bocca, P., and Romero, P. (2018). Heuristics for the minimum broadcast time. *Instituto de Telecomunicações*.
- Farley, A., Hedetniemi, S., Mitchell, S., and Proskurowski, A. (1979). Minimum broadcast graphs. *Discrete Mathematics*, 25(2):189–193.
- Silva, A. L. M., Pinheiro, R. G. S., Nogueira, B. C. e. S., and Peixoto, R. J. S. (2020). Algoritmo genético de chaves aleatórias viciadas para o problema do tempo de transmissão mínimo. *Instituto de Computação – Universidade Federal de Alagoas*.