

---

**FORMULÁRIO DE RELATÓRIO TÉCNICO-CIENTÍFICO**

( ) Parcial      (X) Final

**1. IDENTIFICAÇÃO DO PROJETO**

Nome completo do Beneficiário	Álvaro Junio Pereira Franco
Nome completo da Instituição de Vínculo	Universidade Federal de Santa Catarina
Título do Projeto	Soluções de problemas reais envolvendo otimização
Nome completo do Edital	Edital de chamada pública FAPESC No. 26/2020 Programa de Ciência, Tecnologia e Inovação aos Grupos de Pesquisa da Universidade Federal de Santa Catarina - UFSC
Instrumento Jurídico No. TR	2021TR000436
Vigência da TR (datas início e fim)	30/05/2021 e 31/05/2024

**2. RESUMO PUBLICÁVEL (até 500 palavras)**

O principal objetivo deste projeto é a solução de problemas reais que envolvem otimização. A equipe é composta por 5 professores da UFSC, 1 mestre e servidor do CIASC e 3 alunos. O projeto trata quatro problemas: espalhamento de doença, dois de alocação de recursos e coloração em redes de parentesco. Na alocação de recursos, temos uma parceria com uma empresa pública de tecnologia. O último problema possui caráter interdisciplinar, com colaborações de colegas antropólogos. No problema de espalhamento de doença é dado um grafo que representa a topologia de uma cidade acrescida de dados epidemiológicos. Queremos encontrar um grafo conexo que minimize o pico de infectados ao longo do tempo. Estudamos métodos que criam árvores geradoras e analisamos os resultados obtidos concluindo que árvores geradoras podem ser consideradas boas alternativas topológicas que podem diminuir o número de infectados no grafo.

No primeiro problema de alocação de recursos, é dada uma lista de professores com as habilidades individuais e os horários disponíveis que cada professor tem para lecionar uma disciplina. Desenvolvemos um modelo de Programação Inteira e um algoritmo genético que aloca para cada par (professor, disciplina), um horário, criando assim uma agenda para a escola. A alocação tenta diminuir aulas não-consecutivas dos professores. No segundo, é dada uma lista de requisições de um sistema. As requisições podem ser de escrita ou leitura em uma ou mais variáveis. As variáveis do sistema estão distribuídas entre  $k$  partes. Utilizando um histórico de requisições, queremos encontrar a melhor maneira de espalhar as variáveis de tal forma a equilibrar cada parte e diminuir as sincronizações durante as requisições envolvendo mais de uma variável. Estudamos algoritmos e desenvolvemos um modelo que particionam o estado de um sistema e realizamos experimentos com instâncias que simulam ambientes de sistemas reais.

Com relação ao problema de coloração de redes de parentesco, um grafo com cores nos vértices é dado. As cores nos vértices indicam alguma propriedade do indivíduo, por exemplo, o clã. Os antropólogos parceiros querem encontrar estruturas chamadas de anéis que representam relações de afinidade entre os indivíduos. Foram desenvolvidos algoritmos que encontram, para cada vértice  $v$  da rede, o que chamamos de  $v$ -super, isto é, o maior número de cores em um caminho saindo do vértice  $v$ . Este algoritmo ajuda a encontrar os anéis nas redes. Desenvolvemos algoritmos polinomiais quando a rede é acíclica e possui um número fixo de cores. Experimentos foram realizados com redes reais fornecidas pelos parceiros antropólogos.

No período do projeto, foram concluídas 1 dissertação de mestrado, 3 monografias de conclusão de curso, 1 relatório de iniciação científica, 2 artigos em conferência nacional, 1 artigo em conferência internacional e 7 apresentações em conferências, encontros e seminários. A IC foi indicada ao Prêmio Destaque na Iniciação Científica e Tecnológica - CNPq. Adicionalmente, 1 artigo foi aceito para publicação em conferência nacional. Podemos dizer que o projeto contribuiu com a formação de recursos humanos e para o desenvolvimento do grupo de pesquisa em Inteligência Artificial e Algoritmos da UFSC.

---

### 3. ETAPAS E METAS

#### 3.1 Correlacionar as etapas e metas previstas com as executadas (conforme plano de trabalho aprovado).

**Etapa 1 prevista:** Revisão da literatura (problema de alocação de recursos).

**Metas da Etapa previstas:** Ler os artigos relacionados ao problema de alocação de recursos.

**O que foi executado:** Com relação a essa primeira etapa, vamos nos concentrar no problema de alocação de recursos da agenda escolar. Foram revisados os primeiros artigos que publicaram sobre este assunto até alguns artigos mais recentes. A seguir listamos os artigos para o problema que foram revisados.

- Babaei, H., Karimpour, J., Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86:43–59.
- Carter, M. W., Laporte, G. (1997). Recent developments in practical course timetabling. In *international conference on the practice and theory of automated timetabling*, pages 3–19. Springer.
- Csima, J., Gotlieb, C. (1964). Tests on a computer method for constructing school timetables. *Communications of the ACM*, 7(3):160–163.
- Daskalaki, S., Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160(1):106–120.
- Even, S., Itai, A., Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 184–193. IEEE.
- Gotlieb, C. C. (1963). The construction of class-teacher timetables. In *Proceedings IFIP Congress*, pages 73–77, Amsterdam, Ho. Nort Holland Publishing Co.
- Lions, J. (1966). A conter-example for Gotlieb’s method for construction of school timetables. *Communications of the ACM*, 9(9):697.
- Lions, J. (1967). The Ontario school scheduling program. *The Computer Journal*, 10(1):14–21.
- Luger, G. (2004). *Inteligência Artificial: Estruturas e estratégias para a solução de problemas complexos*. Bookman.
- Pillay, N. (2014). A survey of school timetabling research. In *Annals of Operations Research*, pages 261–293.

---

**Etapa 2 prevista:** Implementação das melhores soluções existentes.

**Metas da Etapa previstas:** Implementar soluções existentes para o problema de alocação de recursos.

**O que foi executado:** O nosso problema da agenda escolar possui semelhanças com problemas existentes na literatura. No entanto, o nosso problema possui particularidades. Por este motivo, não conseguimos encontrar soluções existentes que resolvessem o nosso problema. Então, nos concentramos no desenvolvimento de soluções para o nosso problema (próxima etapa).

---

---

**Etapa 3 prevista:** Desenvolvimento de soluções para o problema de alocação de recursos.

**Metas da Etapa previstas:** Desenvolver soluções próprias para o problema de alocação de recursos.

**O que foi executado:** Foram desenvolvidas duas soluções para o problema da agenda escolar. Uma solução veio através de um modelo de programação inteira mais uma ferramenta de otimização (Gurobi). A outra solução foi dada através de um algoritmo genético.

---

**Etapa 4 prevista:** Comparação entre as soluções existentes e as desenvolvidas.

**Metas da Etapa previstas:** Analisar a qualidade das diferentes soluções para o problema de alocação de recursos.

**O que foi executado:** Nesta etapa a qualidade das soluções encontradas pelas soluções desenvolvidas foi analisada. Comparamos as soluções encontradas pela ferramenta Gurobi ao receber o modelo de programação inteira, com as soluções encontradas pelo algoritmo genético.

---

**Etapa 5 prevista:** Revisão da literatura (problema do espalhamento de uma doença).

**Metas da Etapa previstas:** Ler os artigos relacionados ao problema do espalhamento de uma doença.

**O que foi executado:** A seguir listamos os artigos para o problema do espalhamento de uma doença que foram revisados.

- Barbosa, W. F., da Costa Moreira, E. B., de Araújo, J. M., Pazin-Filho, A., Bueno, C. D. F. (2021). Modelo seir para avaliação do comportamento da pandemia de covid-19 em marabá-pa. *Revista de Medicina*, 100(4):322–328.
- Carvalho, C., Costa, J., Sales, C. L., Lopes, R., Maia de Oliveira, A. K., and Nisse, N. (2020). On the characterization of networks with multiple arc-disjoint branching flows. Research report, UFC; INRIA; CNRS; Université Côte d’Azur; I3S; LIRMM; Université de Montpellier.
- Chang, S., Pierson, E., Koh, P. W., Gerardin, J., Redbird, B., and Grusky, D. Leskovec, J. (2021). Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589:82–87.
- Franco, Á. J. P. (2020). Epidemic model with restricted circulation and social distancing on some network topologies. In *International Conference on Cellular Automata for Research and Industry*, pages 261–264. Springer.
- Hethcote, H. W. (1989). Three basic epidemiological models. In *Applied Mathematical Ecology*, pages 119–144. Springer.
- Dunning, I., Huchette, J., and Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320.
- Keeling, M. J., Eames, K. T. (2005). Networks and epidemic models. *Journal of the royal society interface*, 2(4):295–307.
- Linge, S., Langtangen, H. P. (2020). Programming for computations-Python: A gentle introduction to numerical simulations with Python 3.6. Springer Nature.
- Liu, C., Wu, X., Niu, R., Wu, X., and Fan, R. (2020). A new sair model on complex networks for analysing the 2019 novel coronavirus (covid-19). *Nonlinear Dynamics*, 101(3):1777–1787.

- 
- Manrique-Abril, F. G., Agudelo-Calderon, C. A., González-Chordá, V. M., Gutiérrez-Lesmes, O., Téllez-Piñerez, C. F., Herrera-Amaya, G. (2020). Modelo sir de la pandemia de covid-19 en colombia. *Revista de Salud Pública*, 22:123–131.
  - Rio, D. (2010a). População residente, por grupos de idade e sexo, segundo as Áreas de planejamento (ap), regiões administrativas (ra) e bairros. Prefeitura da Cidade do Rio de Janeiro. Tabela, 2974.
  - Rio, D. (2010b). População residente, por idade e por grupos de idade, segundo as Áreas de planejamento (ap), regiões administrativas (ra) e bairros. Prefeitura da Cidade do Rio de Janeiro. Tabela, 2973.
  - Rio, D. (2010c). População residente por sexo e razão de sexos, segundo as Áreas de planejamento (ap), regiões de planejamento (rp), regiões administrativas (ra) e bairros no município do rio de janeiro. Prefeitura da Cidade do Rio de Janeiro. Tabela, 3163.
  - Visbal, J. H. W., Pedraza, M. C. C. (2020). Aproximación matemática del modelo epidemiológico sir para la comprensión de las medidas de contención contra la covid-19. *Rev. esp. salud pública*.
  - White, S. H., Del Rey, A. M., Sánchez, G. R. (2007). Modeling epidemics using cellular automata. *Applied Mathematics and Computation*, 186(1):193–202.

Capítulos dos seguintes livros foram lidos e revisados.

- Burden, R. L., Faires, J. D. (2011). *Numerical analysis*. Cengage learning, 9th edition.
- Dasgupta, S., Papadimitriou, C. H., and Vazirani, U. V. (2008). *Algorithms*. McGraw-Hill Higher Education New York.
- Feofiloff, P., Kohayakawa, Y., and Wakabayashi, Y. (2011). Uma introdução sucinta à teoria dos grafos.

---

**Etapa 6 prevista:** Implementação de algumas soluções existentes.

**Metas da Etapa previstas:** Implementar soluções existentes para o problema do espalhamento de uma doença.

**O que foi executado:** Foram implementadas soluções para o problema do espalhamento de uma doença usando os modelos epidemiológicos SIS e SIR. O método numérico utilizado na implementação de cada um dos modelos epidemiológicos foi o método geral de Euler com variação no tamanho do passo no tempo a cada execução. Também foram implementados algoritmos conhecidos que constroem árvores geradoras.

---

**Etapa 7 prevista:** Desenvolvimento de soluções para o problema do espalhamento de uma doença.

**Metas da Etapa previstas:** Desenvolver soluções próprias para o problema do espalhamento de uma doença.

**O que foi executado:** Foi desenvolvido um algoritmo que encontra novas árvores geradoras. Foi implementado um modelo de programação não-linear para encontrar a rede conexa que minimiza o número de infectados na rede ao longo do tempo. Foram construídas duas instâncias de cidades reais (Florianópolis e Rio de Janeiro), ou seja, dois grafos de infecção que consideram as proximidades dos bairros das cidades, o número populacional de cada bairro, e os números epidemiológicos (susceptíveis, infectados e recuperados) em um determinado tempo para cada bairro. A construção dos grafos de

---

infecção considera dados populacionais e epidemiológicos reais. Para o caso do Rio de Janeiro, foram utilizados somente dados populacionais reais.

---

**Etapa 8 prevista:** Comparação entre as soluções existentes e as desenvolvidas.

**Metas da Etapa previstas:** Analisar a qualidade das diferentes soluções para o problema do espalhamento de uma doença.

**O que foi executado:** Foram construídas várias árvores geradoras do grafo de infecção de entrada. Para cada uma delas, o pico do número de indivíduos infectado ao longo do tempo foi obtido e comparado com o pico do número de indivíduos infectados considerando o grafo de infecção original. Em geral, foi constatado que pico de infectados em uma árvore geradora é menor que o pico do grafo de infecção original.

---

**Etapa 9 prevista:** Escrita de artigos científicos e divulgação dos resultados obtidos.

**Metas da Etapa previstas:** Escrever relatórios científicos e divulgar os resultados obtidos.

**O que foi executado:** Vamos descrever aqui os relatórios científicos e divulgação dos resultados referentes ao problema do espalhamento de doença. Foram apresentados trabalhos no Encontro Latino-Americano de Matemática e Aplicações (ELAMAT-2021), no Encontro de Teoria da Computação (ETC-2022), no Simpósio Brasileiro de Pesquisa Operacional (SBPO-2023), no Encontro Acadêmico de Matemática (EAMAT-2023), no Simpósio de Iniciação Científica da UFSC (SIC-2023), um relatório de iniciação científica, um resumo no ELAMAT-2021, um artigo publicado nos anais do SBPO-2023, um artigo publicado nos anais do ETC-2022.

---

**Etapa 10 prevista:** Revisão da literatura (para os novos problemas: caminhos  $v$ -super-coloridos e balanceamento do conjunto de vértices com corte mínimo).

**Metas da Etapa previstas:** Ler os artigos que relacionam grafos com cores nos vértices e que desenvolvem métodos sobre tais grafos.

**O que foi executado:** Vamos relacionar a seguir os principais artigos e livros que foram consultados para ajudar no desenvolvimento de novas soluções para os problemas  $v$ -super-coloridos e balanceamento do conjunto de vértices com corte mínimo (o segundo problema tratado nesta pesquisa no contexto da alocação de recurso).

- Bondy, J. A., Murty, U. S. R. (1976). Graph theory with applications, volume 290. Macmillan London.
- Chartrand, G., Johns, G. L., McKeon, K. A., Zhang, P. (2008). Rainbow connection in graphs. *Mathematica Bohemica*, 133(1):85–98.
- Cowen, L., Goddard, W., Jesurum, C. E. (1997). Defective coloring revisited. *Journal of Graph Theory*, 24(3):205–219.
- Hamberger, K., Houseman, M., Dailliant, I., White, D. R., Barry, L. (2004). Matrimonial ring structures. *Mathématiques et sciences humaines. Mathematics and social sciences*, 168:83–119.
- Poz, J. d., Silva, M. F. d. (2009). Maqpar: a homemade tool for the study of kinship networks. *Vibrant*, 6(2):29–51.

- 
- Franco, Á. J. P., Vendramin, M. E. (2021). Super-colored paths in digraphs. In Anais do VI Encontro de Teoria da Computação, p. 94–97, Porto Alegre, RS, Brasil. SBC.
  - Hassin, R., Monnot, J., Segev, D. (2007). Approximation algorithms and hardness results for labeled connectivity problems. Journal of Combinatorial Optimization, 14.
  - Monnot, J. (2005). The labeled perfect matching in bipartite graphs. Information Processing Letters, 96(3):81–88.
  - Appel, K. I., Haken, W. (1989). Every planar map is four colorable, volume 98. American Mathematical Soc.
  - Cooper, B. F., Silberstein, A., Tam E., Ramakrishnan, R., Sears R., Benchmarking cloud serving systems with ycsb, Proceedings of the 1st ACM symposium on Cloud computing, pp. 143-154, 2010.
  - Nishimura, J., Ugander, J., Restreaming graph partitioning: simple versatile algorithms for advanced balancing, Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1106-1114, 2013.
  - Tsourakakis, C., Gkantsidis, C., Radunovic, B., Vojnovic, M., Fennel: Streaming graph partitioning for massive scale graphs, Proceedings of the 7th ACM international conference on Web search and data mining, pp. 333-342, 2014.
  - Alchieri, E., Dotti, F., Mendizabal, O. M., Pedone, F. Reconfiguring parallel state machine replication, 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), pp. 104-113, 2017.
  - Ayall, T. A., Liu, H., Zhou, C., Seid, A. M., Gereme, F. B., Abishu, H. N., Yacob, Y. H., Graph computing systems and partitioning techniques: A survey, IEEE Access, vol. 10, pp. 118523-118550, 2022.
  - Knuth, D. E., The Stanford GraphBase: a platform for combinatorial computing, New York:AcM Press, vol. 1, 1993.

---

**Etapa 11 prevista:** Desenvolvimento de soluções para os novos problemas: caminhos  $v$ -super-coloridos e balanceamento do conjunto de vértices com corte mínimo.

**Metas da Etapa previstas:** Desenvolver soluções próprias para os problemas caminhos  $v$ -super-coloridos e balanceamento do conjunto de vértices com corte mínimo.

**O que foi executado:** Para o problema  $v$ -super-coloridos, foram desenvolvidos um algoritmo exato, três algoritmos parametrizados (sendo  $k$  o número de cores do grafo, um algoritmo quando  $k$  é um número inteiro fixo, um quando  $k = 2$  e um quando  $k = 3$ ) e um programa inteiro quando a entrada é um grafo dirigido acíclico. Para o problema do balanceamento do conjunto de vértices com corte mínimo, foi desenvolvido um modelo de programação inteira e um protótipo que realiza o particionamento do estado de um sistema utilizando uma heurística. As heurísticas implementadas foram: round-robin, fenel, refenel e metis.

---

**Etapa 12 prevista:** Escrita de artigos científicos e divulgação dos resultados obtidos.

**Metas da Etapa previstas:** Escrever relatórios científicos e divulgar os resultados obtidos.

---

**O que foi executado:** Nesta etapa vamos destacar os relatórios científicos e as divulgações dos relatórios realizadas no período. Foram eles o relatório parcial entregue para membros da pró-reitoria de pesquisa da UFSC (com apresentação dos primeiros resultados), relatório parcial entregue para membros da FAPESC e para a câmara de pesquisa do Departamento de Informática e Estatística da UFSC, e relatório técnico e apresentação do problema da agenda escolar (apresentação realizada no EAMAT-2023).

---

**Etapa 13 prevista:** Escrita de artigos científicos e divulgação dos resultados obtidos para os novos problemas caminhos  $v$ -super-coloridos e balanceamento do conjunto de vértices com corte mínimo.

**Metas da Etapa previstas:** Escrever relatórios científicos e divulgar os resultados obtidos.

**O que foi executado:** Sobre o problema  $v$ -super-colorido, estamos finalizando um trabalho com os resultados obtidos. O trabalho foi submetido para apreciação do comitê de programa do SBPO-2024, e ainda está sob análise (veja a confirmação da submissão na seção **Anexos** deste relatório). Sobre o problema de balanceamento do conjunto de vértices com corte mínimo, um artigo foi apresentado e publicado nos anais de uma conferência internacional - *IEEE International Symposium on Computer Architecture and High Performance Computing* (SBAC-PAD-2023).

### 3.2 Relatar as alterações ou dificuldades encontradas na execução das etapas e metas.

As principais dificuldades estão relacionadas às novas atividades assumidas pelo coordenador durante o período vigente deste projeto. Este projeto teve início em maio de 2021. E em setembro de 2021, o coordenador deste projeto assumiu o cargo de coordenador do curso de graduação em Sistemas de Informação da UFSC. Até o presente momento, o coordenador deste projeto é também coordenador do curso de graduação citado. As atividades administrativas tomaram muito tempo dedicado à pesquisa e isso dificultou avançar mais com as soluções para os problemas tratados. Outra dificuldade que ocorreu foi que durante mais de 1 ano, a secretaria do curso de Sistemas de Informação estava sem secretário. Para não parar o serviço da secretaria, o presente coordenador deste projeto e coordenador de curso, assumiu também as atividades da secretaria. Tais dificuldades impactaram nos resultados obtidos pelo projeto.

## 4. INDICADORES

*Relacionar e quantificar os resultados alcançados, destacando o resultado principal (conforme previsto no plano de trabalho aprovado), respondendo o formulário, clicando aqui.*

Abaixo listamos para cada resultado em destaque a quantidade prevista no plano de trabalho e a quantidade executada durante o período do projeto.

### Principal

Produção bibliográfica	Quantidade prevista	Quantidade executada
Artigos publicados em periódicos e revistas científicas internacionais	1	0

### Outros resultados

Produção bibliográfica	Quantidade prevista	Quantidade executada
Trabalhos publicados em eventos nacionais	2	2
Trabalhos publicados em eventos internacionais	1	1

Produção técnica	Quantidade prevista	Quantidade executada
Softwares	2	2
Relatório de pesquisa	4	4
Inserção social	Quantidade prevista	Quantidade executada
Melhorias na educação	1	1
Processos e produtos	Quantidade prevista	Quantidade executada
Nova tecnologia	2	2
Formação	Quantidade prevista	Quantidade executada
Trabalhos de conclusão de curso de graduação defendidos	1	3
Trabalhos de iniciação científica defendidos	0	1
Trabalhos de mestrado defendidos	0	1
Grupos de pesquisa	Quantidade prevista	Quantidade executada
Apoio a grupos de pesquisa consolidados	1	1
Divulgação/difusão de produção técnica e científica	Quantidade prevista	Quantidade executada
Trabalhos apresentados em eventos nacionais	1	2
Trabalhos apresentados em eventos internacionais	1	1
Objetivos de Desenvolvimento Sustentável (ODS)	Quantidade prevista	Quantidade executada
3 Saúde e bem-estar	1	1
8 Trabalho decente e crescimento econômico	1	1
17 Parcerias e meios de implementação	1	1
Produção bibliográfica	Quantidade prevista	Quantidade executada
Apresentação de trabalhos	2	7

#### 4.1 Salvar o e-mail enviado contendo as respostas do formulário, no formato PDF. Anexar este documento (PDF) na plataforma, juntamente com este relatório técnico.

O formulário deveria ter sido enviado pelo sistema online. Porém, o sistema não enviou o formulário descrito no item 4.1. Decidimos então pegar a tabela presente no plano de trabalho e acrescentar mais uma coluna na tabela indicando a quantidade de resultados efetivamente executados. A tabela está acima.

## 5. RESULTADOS

*Justificar os resultados previstos que não foram alcançados.*

Inicialmente prevemos uma publicação em revista científica internacional. Durante o período do projeto, não foi possível obter tal publicação. Alguns membros da equipe deste projeto, incluindo o coordenador, trabalham na submissão de um artigo para revista científica. As justificativas para não conseguirmos este resultado são:

- Ainda estamos implementando uma nova heurística para o problema dos grafos de infecção. Depois disso, e depois de avaliar os resultados, acreditamos que poderemos submeter o nosso trabalho para uma revista científica internacional e de qualidade.
- O coordenador do presente projeto se envolveu com atividades administrativas na Universidade Federal de Santa Catarina relacionadas a coordenação de curso de graduação. Há um agravante neste ponto pois o curso de graduação ficou sem secretário por mais de um (1) ano.

Acredita-se que os demais resultados foram alcançados.

*Dissertar sobre os resultados do projeto, incluindo todas as informações necessárias (fotos, tabelas, gráficos, texto)*

Em seguida, para cada problema tratado no projeto, os principais resultados obtidos em cada um deles são descritos.

## **Minimizando Infecção em uma Topologia usando Programação Matemática**

### **Trabalho em conjunto com Wesley C. Ataíde, Pedro B. Castellucci e Rafael de Santiago.**

Depois dos primeiros casos de COVID-19, doença causada pelo coronavírus SARS-CoV-2, serem registrados em dezembro de 2019 na China, a infecção se espalhou rapidamente, representando um grande perigo para sistemas de saúde e para a economia mundial. As medidas adotadas por muitos países em resposta consistiram em restrições de movimento, limitando o funcionamento de estabelecimentos e incentivando seus cidadãos a ficarem em casa, com o objetivo de reduzir o contato entre indivíduos e, desse modo, frear o contágio. Com essas ações, entretanto, vem a questão de quando é seguro reabrir, quais lugares podem ser reabertos e quais atividades podem ser permitidas [Chang et al. 2021]. Para responder essas questões, modelos epidemiológicos se tornaram cruciais. Com um volume grande de dados, eles podem identificar grupos socioeconômicos mais vulneráveis, apontar classes de estabelecimentos mais propensas a espalhar a infecção e encontrar padrões para maximizar a efetividade de medidas de restrição. Medidas que diminuem o contágio são os instrumentos mais importantes na luta contra o COVID-19 numa perspectiva social, pois tanto diminuem o número de infectados como ajudam a garantir o tratamento para pessoas que se infectarem [Chang et al. 2021]. Com isso, previsões se tornaram fundamentais para reduzir os impactos negativos da pandemia tanto numa esfera social quanto numa esfera econômica [Liu et al. 2020], ao passo que guiam as políticas públicas adotadas. Foi apresentado neste trabalho um modelo de programação inteira não-linear de um problema epidemiológico onde o objetivo é minimizar o número de pessoas infectadas ao longo do tempo em um grafo conexo.

A modelagem do problema, baseada em [Franco 2020], consiste na minimização do contágio em uma rede com bairros interconectados por estradas, representada por um grafo em que os bairros são os vértices (com uma população associada) e cada estrada entre um bairro  $i$  e um bairro  $j$  é um par de arcos  $(i, j)$  e  $(j, i)$ . Em relação ao modelo epidemiológico, foi escolhido trabalhar com o modelo SIR, que divide a população em três grupos: suscetíveis, que podem se infectar; infectados, que podem transmitir; e recuperados, que não transmitem nem podem ser infectados. Levando isso em conta, a população de cada vértice  $i$  para cada instante  $t$  consiste no número de pessoas suscetíveis  $S_i^t$ , infectadas  $I_i^t$  e recuperadas  $R_i^t$ . O movimento entre vértices também é modelado, porém é restrito de modo que pessoas do vértice  $i$  apenas podem visitar vértices adjacentes de  $i$ . Ainda definimos para cada arco  $(i, j)$  os valores  $\beta_{i \rightarrow j}$  que representam a porcentagem de pessoas que residem no vértice  $i$  e visitam o vértice  $j$ , ou seja, que “utilizam a estrada  $(i, j)$ ”. Também é associado a cada arco  $(i, j)$  uma variável binária  $x_{i \rightarrow j}$  (que restringimos para ter o mesmo valor de  $x_{j \rightarrow i}$ ), que indica que o tráfego está bloqueado se  $x_{i \rightarrow j} = 0$  ou que está livre se  $x_{i \rightarrow j} = 1$ .

É relevante notar que essa construção faz com que o grafo seja mais facilmente analisado como um grafo dirigido da perspectiva dos valores  $\beta$  e como um grafo não-dirigido da perspectiva das variáveis  $x$ . Desse modo, quando nos referimos a uma árvore ou a componentes conexas nesse grafo, estamos

analisando-o no modo não-dirigido. As ruas bloqueadas e liberadas criam uma variação no número de pessoas em cada um dos vértices, afetando o contágio. Para analisar a dinâmica do contágio, observa-se que, em cada instante de tempo  $t$ , as pessoas circulando em cada vértice  $i$  se dividem em dois grupos: as que vêm de um vértice vizinho a  $i$  ( $\chi(i)$ ) e as que moram e ficaram em  $i$ . Para o primeiro grupo, para cada vértice vizinho, o número de pessoas vindo para  $i$  obedece às seguintes fórmulas:  $S_{j \rightarrow i}^t = \beta_{j \rightarrow i} x_{j \rightarrow i}^t S_j^t$ ;  $I_{j \rightarrow i}^t = \beta_{j \rightarrow i} x_{j \rightarrow i}^t I_j^t$ ; e  $R_{j \rightarrow i}^t = \beta_{j \rightarrow i} x_{j \rightarrow i}^t R_j^t$ .

Já o segundo grupo consiste das pessoas que não saíram do vértice  $i$  e, desse modo, basta subtrair a porcentagem de pessoas que saiu:  $S_{i \rightarrow i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i \rightarrow j} x_{i \rightarrow j}^t)) S_i^t$ ;  $I_{i \rightarrow i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i \rightarrow j} x_{i \rightarrow j}^t)) I_i^t$ ; e  $R_{i \rightarrow i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i \rightarrow j} x_{i \rightarrow j}^t)) R_i^t$ .

É relevante observar que se uma estrada é bloqueada, a porcentagem das pessoas que tinham intenção de usá-la é acrescentada no vértice onde as pessoas moram. Com isso, o número de pessoas de cada categoria circulando em  $i$  é dado por:  $\dot{S}_i^t = S_{i \rightarrow i}^t + \sum_{j \in \chi(i)} S_{j \rightarrow i}^t$ ;  $\dot{I}_i^t = I_{i \rightarrow i}^t + \sum_{j \in \chi(i)} I_{j \rightarrow i}^t$ ; e  $\dot{R}_i^t = R_{i \rightarrow i}^t + \sum_{j \in \chi(i)} R_{j \rightarrow i}^t$ .

O objetivo é minimizar o número de infectados ao longo do tempo. É necessário considerar para cada vértice e instante o número de pessoas que se recuperam e o número de novos infectados, que é uma razão do número de encontros entre pessoas suscetíveis e infectadas:  $S_i^{t+1} = S_i^t - v \mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t)$ ;  $I_i^{t+1} = I_i^t - \xi I_i^t + v \mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t)$ ;  $R_i^{t+1} = R_i^t + \xi I_i^t$ . As constantes  $v$  e  $\xi$  são a *virulência* e a taxa de recuperação, respectivamente.

Como a porcentagem de infectados das pessoas circulando em um vértice  $i$  é  $\frac{\dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t}$ , tomamos esse valor como a proporção de suscetíveis que têm um encontro com um infectado nesse vértice. O número esperado de encontros para a população do vértice  $i$  é portanto

$$\mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t) = \frac{S_{i \rightarrow i} \dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t} + \sum_{j \in \chi(i)} \frac{S_{i \rightarrow j} \dot{I}_j^t}{\dot{S}_j^t + \dot{I}_j^t + \dot{R}_j^t}.$$

Temos uma limitação sobre o modo como escolhemos bloquear as ruas, entretanto devemos garantir que seja possível acessar todos os bairros, ou seja, o subgrafo induzido pelas ruas liberadas deve ser conexo. Para obter um subgrafo conexo, as seguintes restrições são adicionadas ao modelo:

$$\sum_{(i,j) \in E} f_{i \rightarrow j} + 1 = \sum_{(j,i) \in E} f_{j \rightarrow i} \text{ para cada vértice } i \neq s; \text{ e } \sum_{(s,j) \in E} f_{s \rightarrow j} = n - 1,$$

sendo que  $f_{i \rightarrow j}$  representa um fluxo inteiro através do arco  $(i, j)$  e  $n$  é o número de vértices do grafo. Estas restrições garantem sobre  $f$ , um fluxo  $s$ -arborescente para algum vértice fonte  $s$  [Carvalho et al. 2020].

Note que cada fluxo  $s$ -arborescente induz uma árvore geradora em um grafo quando consideramos como arestas cada par de arcos  $(i, j)$  onde  $f_{i \rightarrow j} > 0$  ou  $f_{j \rightarrow i} > 0$ . Como os arcos são pares, a estrutura do grafo é essencialmente mantida. O vértice  $s$  pode ser escolhido arbitrariamente, como uma consequência do seguinte resultado.

**Teorema:** Seja  $G$  um grafo conexo, e  $T$  uma árvore geradora de  $G$ . Para qualquer vértice  $v$  em  $T$ , existe um fluxo  $v$ -arborescente que induz  $T$ .  $\square$

Para aproximar as restrições de conectividade com as soluções viáveis e para integrar elas com o contágio, foram adicionadas restrições tal que um par de arcos não é bloqueado quando existe fluxo sobre ele: para cada arco  $(i, j)$ , é adicionado  $n x_{i \rightarrow j} \geq f_{i \rightarrow j}$ , ( $n$  é o número total de vértices). Note que isso não implica que o grafo obtido (possivelmente ótimo) seja uma árvore geradora, mas somente que ele seja conexo.

Para resumir, o programa de otimização não-linear deseja minimizar  $\sum_{i \in V} I_i^{t+1}$ , onde  $V$  é o conjunto de vértices do grafo da entrada, sujeito às restrições de circulação das pessoas ( $\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t$  para cada vértice  $i$ ) e às restrições de conectividade em grafos. A não-linearidade está na função objetivo e uma questão importante é saber se é possível linearizar este modelo. A seguir, apresentamos o modelo de otimização inteiro não-linear por completo.

$$\begin{aligned}
 \min \quad & \sum_{i \in V} I_i^{t+1} \\
 \text{sujeito a:} \quad & \dot{S}_i^t = S_{i \rightarrow i}^t + \sum_{j \in \chi(i)} S_{j \rightarrow i}^t, \forall i \in V \\
 & \dot{I}_i^t = I_{i \rightarrow i}^t + \sum_{j \in \chi(i)} I_{j \rightarrow i}^t, \forall i \in V \\
 & \dot{R}_i^t = R_{i \rightarrow i}^t + \sum_{j \in \chi(i)} R_{j \rightarrow i}^t, \forall i \in V \\
 & \sum_{(i,j) \in E} f_{i,j} + 1 = \sum_{(j,i) \in E} f_{j,i}, \forall i \in V \text{ and } i \neq j \\
 & \sum_{(s,j) \in E} f_{s,j} = n - 1 \\
 & nx_{i,j} \geq f_{i,j}, \forall (i,j) \in E \\
 & f_{i,j} \in \{0, \dots, n-1\}, \forall (i,j) \in E \\
 & x_{i,j} \in \{0, 1\}, \forall (i,j) \in E,
 \end{aligned}$$

onde

$$\begin{aligned}
 I_i^{t+1} &= I_i^t - \xi I_i^t + v \mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t); \\
 \mathbb{E}_i^t(\dot{S}_i^t, \dot{I}_i^t, \dot{R}_i^t) &= \frac{S_{i \rightarrow i}^t \dot{I}_i^t}{\dot{S}_i^t + \dot{I}_i^t + \dot{R}_i^t} + \sum_{j \in \chi(i)} \frac{S_{i \rightarrow j}^t \dot{I}_j^t}{\dot{S}_j^t + \dot{I}_j^t + \dot{R}_j^t}; \\
 S_{i \rightarrow i}^t &= (1 - \sum_{j \in \chi(i)} (\beta_{i,j} x_{i,j}^t)) S_i^t; \quad I_{i \rightarrow i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i,j} x_{i,j}^t)) I_i^t; \quad R_{i \rightarrow i}^t = (1 - \sum_{j \in \chi(i)} (\beta_{i,j} x_{i,j}^t)) R_i^t; \\
 S_{j \rightarrow i}^t &= \beta_{j,i} x_{j,i}^t S_j^t; \quad I_{j \rightarrow i}^t = \beta_{j,i} x_{j,i}^t I_j^t; \quad R_{j \rightarrow i}^t = \beta_{j,i} x_{j,i}^t R_j^t.
 \end{aligned}$$

Agora comentamos sobre os resultados computacionais. Para avaliar o modelo, foi construída uma instância baseada em dados reais da cidade de Florianópolis, Santa Catarina. Os dados foram coletados do *Covidômetro*<sup>1</sup> e o censo de 2010 census, de acordo com o banco de dados do IBGE. *Covidômetro* continha dados da COVID-19 referente aos casos confirmados, infectados ativos, recuperados e mortos de cada bairro da cidade. O banco de dados do IBGE foi usado para definir a população total de cada bairro. Para tratar casos inconsistentes de definição de bairro e para simplificar a instância construída, alguns bairros vizinhos, e consequentemente as populações correspondentes, foram agrupados. Para resolver o problema de otimização, as ferramentas SCIP [Bestuzheva et al. 2021] e JuMP [Dunning et al. 2017] foram usadas. SCIP é uma ferramenta de código aberto e com bom desempenho e tem uma licença acadêmica livre. Na SCIP é também possível trabalhar com restrições ou função objetivo não-linear. A implementação completa pode ser encontrada na página <https://github.com/weslyca/TCC>. Todos os experimentos foram desenvolvidos em uma máquina Aspire ES1-572 Acer, com um processador Intel(R) Core(TM) i3-6006u, 2.00GHz e 4GB e memória RAM. Com relação a implementação, foram utilizadas as linguagens Python (versão 3.9.9) e Julia (versão 1.7.0) com JuMP (versão 0.22.1).

Sobre este trabalho houveram

- uma apresentação oral por vídeo no Encontro Latino Americano de Matemática e Aplicações (ELAMAP 2021 - Universidade Federal do ABC - UFABC) com acesso pelas páginas
  - [https://www.youtube.com/watch?v=xnA-rzN3\\_CA&list=PLWxSDYVF6deKzLI3WijbteT-cBeS9iPbi&index=3&t=6s](https://www.youtube.com/watch?v=xnA-rzN3_CA&list=PLWxSDYVF6deKzLI3WijbteT-cBeS9iPbi&index=3&t=6s);
  - <https://sites.google.com/view/elamap2021/comunica%C3%A7%C3%B5es?authuser=0>;
  - <https://www.inf.ufsc.br/~alvaro.junio/proj-docs/elamap-2021.pdf>
- uma publicação de um artigo e uma apresentação oral e presencial no Encontro de Teoria da Computação (CSBC - ETC 2022 - Universidade Federal Fluminense - UFF) com acesso pelas páginas:
  - <https://doi.org/10.5753/etc.2022.223209>
  - <https://www.inf.ufsc.br/~alvaro.junio/proj-docs/etc-2022.pdf>

<sup>1</sup><https://covidometrofloripa.com.br/> - página acessada em 2022 mas agora está descontinuada

- 
- uma publicação de um trabalho de conclusão de curso com acesso pela página
    - <https://repositorio.ufsc.br/handle/123456789/232973>
  - implementações com acesso pela página
    - <https://github.com/weslyca/TCC>

## Um Estudo sobre Árvores Geradoras como Alternativas Conexas para o Controle de Doença em um Grafo de Infecção

Trabalho em conjunto com Pedro G. Caninas e Priscila C. Calegari.

No trabalho anterior, foi desenvolvido um modelo de programação não-linear e inteiro para minimizar o contágio de uma doença em um grafo de infecção. O modelo foi resolvido utilizando ferramentas de otimização. No trabalho que é descrito agora, a ideia é encontrar subgrafos conexos que sirvam como alternativas para diminuir o pico de infectados em um grafo de infecção ao longo do tempo. Em ambos os trabalhos, foi adotado um modelo epidemiológico básico desenvolvido [Franco 2020] que tem como fundamento um modelo epidemiológico sobre um autômato celular [White et al. 2007]. As notações do modelo tomado como base para ambos os trabalhos são diferentes assim como o que representam as (ou o significado das) variáveis SIR do modelo. No trabalho anterior, as variáveis SIR correspondem ao número de pessoas suscetíveis, infectadas e recuperadas em um determinado tempo e que eram obtidas em função dos suscetíveis, infectados e recuperados do tempo anterior. Agora, as variáveis SIR correspondem a limitantes inferiores e superiores.

O modelo epidemiológico sobre um grafo considera de fato a aplicação de um modelo epidemiológico mais fundamental e em cada vértice do grafo. Considera também uma quantidade de pessoas em cada vértice, além de uma certa movimentação das pessoas entre vértices adjacentes. O trabalho de [Tomassini 2006] descreve um modelo sobre *redes de autômato generalizado*, semelhante à ideia de aplicar um modelo epidemiológico sobre um grafo. Um modelo epidemiológico qualquer pode ser aplicado a cada vértice. Na ocasião, foi escolhido o modelo clássico SIR em que as pessoas são classificadas como *suscetíveis, infectadas e recuperadas* (veja este e outros modelos clássicos, por exemplo, em [Hethcote 1989]). Dessa forma, cada pessoa de um vértice está no correspondente grupo SIR. Além disso, as pessoas podem aderir ou não a um *distanciamento social*, uma medida dada por um índice de distanciamento por vértice.

O grupo SIR de pessoas no vértice  $i$  e no tempo  $t$  é dado por  $N_i^t = S_i^t \cup I_i^t \cup R_i^t$ . O subconjunto  $\bar{S}_i^t \subseteq S_i^t$  ( $\tilde{S}_i^t \subseteq S_i^t$ ) destaca as pessoas suscetíveis do vértice  $i$  no tempo  $t$  que atendem (não atendem) ao distanciamento social. Da mesma forma temos isso para os infectados  $\bar{I}_i^t$  ( $\tilde{I}_i^t$ ) e recuperados  $\bar{R}_i^t$  ( $\tilde{R}_i^t$ ). A constante  $\alpha_i^t$  corresponde ao índice de pessoas do vértice  $i$  que atendem ao distanciamento social no tempo  $t$ . Com isso, o número de suscetíveis que atendem e não atendem ao distanciamento é, respectivamente  $|\bar{S}_i^t| = \lfloor \alpha_i^t |S_i^t| \rfloor$  e  $|\tilde{S}_i^t| = |S_i^t| - |\bar{S}_i^t|$ . O mesmo é aplicado para obter os números de infectados e de recuperados que atendem e não atendem ao distanciamento social. Portanto, as pessoas de  $i$  e no tempo  $t$  que atendem ao distanciamento social pertencem ao conjunto  $\bar{N}_i^t = \bar{S}_i^t \cup \bar{I}_i^t \cup \bar{R}_i^t$ ; e as pessoas que não atendem ao distanciamento social pertencem ao conjunto  $\tilde{N}_i^t = \tilde{S}_i^t \cup \tilde{I}_i^t \cup \tilde{R}_i^t$ .

Assim como na seção anterior, a movimentação de pessoas considerada no modelo acontece com pessoas circulando em cada vértice  $i$  do grafo e em cada tempo  $t$ . Essas pessoas são pessoas do próprio vértice  $i$  (de  $\bar{N}_i^t$  e de  $\tilde{N}_i^t$ ) e pessoas de vértices adjacentes de  $i$  que não atendem ao distanciamento social (de  $\bigcup_{j \in \chi(i)} \tilde{N}_j^t$ , sendo  $\chi(i)$  o conjunto dos vértices adjacentes de  $i$ ). Um fator  $\beta_{j \rightarrow i}^t$  é utilizado para representar a proporção de pessoas do vértice  $j$  que vão para o vértice  $i$  sendo  $j \in \chi(i)$ .

O conjunto de suscetíveis que encontram com um infectado no vértice  $i$  e no tempo  $t$  é denotado por  $X_i^t$ . Considerando o distanciamento social, temos  $\bar{X}_i^t$  ( $\tilde{X}_i^t$ ) o conjunto de pessoas  $p \in \bar{S}_i^t$  ( $p \in \tilde{S}_i^t$ ) que encontram com um infectado no vértice  $i$  (em vértices adjacentes a  $i$ ). Note que o modelo considera uma movimentação de pessoas que atendem ao distanciamento social. Tais pessoas precisaram se movimentar por algum motivo. Dessa forma, temos  $X_i^t = \bar{X}_i^t \cup \tilde{X}_i^t$ . Finalmente, o modelo sobre o grafo considerando

o modelo epidemiológico SIR em cada vértice é o seguinte:

$$\begin{aligned} |I_i^{t+1}| &\geq |I_i^t| - \lceil \varepsilon \mathbb{E}[|I_i^t|] \rceil + \lfloor v(\mathbb{E}[|\bar{X}_i^t|] + \mathbb{E}[|\tilde{X}_i^t|]) \rfloor =: |I_i^{t+1}|, \\ |S_i^{t+1}| &\leq |S_i^t| - \lfloor v(\mathbb{E}[|\bar{X}_i^t|] + \mathbb{E}[|\tilde{X}_i^t|]) \rfloor =: |S_i^{t+1}|, \\ |R_i^{t+1}| &\leq |R_i^t| + \lceil \varepsilon \mathbb{E}[|I_i^t|] \rceil =: |R_i^{t+1}|, \end{aligned}$$

sendo  $v$  é uma taxa de virulência,  $\varepsilon$  é a taxa de recuperação; o valor esperado de  $|\bar{X}_i^t|$  definido como  $\mathbb{E}[|\bar{X}_i^t|] = \tilde{S}_i^t \mathbb{P}\{\tilde{Y}_{pi}^t\}$  com  $\tilde{S}_i^t$  sendo o número de pessoas suscetíveis de  $i$  que atendem ao distanciamento social mas que por algum motivo circulam somente em  $i$  e o valor esperado de  $|\tilde{X}_i^t|$  definido como sendo  $\mathbb{E}[|\tilde{X}_i^t|] = \tilde{S}_i^t \mathbb{P}\{\tilde{Y}_{pi \rightarrow i}^t\} + \sum_{j \in \chi(i)} \tilde{S}_j^t \mathbb{P}\{\tilde{Y}_{pi \rightarrow j}^t\}$ , com  $\tilde{S}_i^t$  sendo o número de pessoas suscetíveis de  $i$  que não atendem ao distanciamento social e que circulam em  $i$  e  $\tilde{S}_j^t$  sendo o número de pessoas suscetíveis de  $i$  que não atendem ao distanciamento social e que circulam em um vértice adjacente  $j \in \chi(i)$ . As probabilidades  $\mathbb{P}\{\tilde{Y}_{pi}^t\}$ ,  $\mathbb{P}\{\tilde{Y}_{pi \rightarrow i}^t\}$  e  $\mathbb{P}\{\tilde{Y}_{pi \rightarrow j}^t\}$  são todas iguais ao número de pessoas infectadas circulando em um vértice dividido pelo total de pessoas circulando em tal vértice. Dessa forma, temos para cada tempo e para cada vértice do grafo, limitantes inferiores para o número de infectados e limitantes superiores para os números de suscetíveis e recuperados. Para a execução dos experimentos, os números de infectados, suscetíveis e recuperados são fixados nos limitantes encontrados.

Um modelo epidemiológico pode ser utilizado para estudar e prever a dinâmica de uma doença em uma população. O trabalho de [Barbosa et al. 2021] faz uma análise da pandemia na cidade de Marabá utilizando o modelo SEIR, com dados reais da cidade e da doença (Covid-19). Outros trabalhos, como o [Manrique-Abril et al. 2020], consideram uma abordagem SIR puramente matemática para analisar e prever o desenvolvimento da Covid-19 no território colombiano. O trabalho [Visbal and Pedraza 2020] também utiliza o modelo SIR e realiza uma conexão entre parâmetros do modelo e motivações em torno das medidas de contenção adotadas na cidade de Santa Marta, Colômbia. Os modelos epidemiológicos também podem ser aplicados sobre grafos, como feito no trabalho [Keeling and Eames 2005]. Nele, cada pessoa é representada como um vértice da rede e a proximidade/contato entre pessoas são representadas como arestas, individualizando de fato as pessoas. No presente trabalho, o modelo epidemiológico SIR desenvolvido é aplicado sobre dois grafos, com os vértices representando um conjunto de pessoas que pertencem a regiões adjacentes de duas cidades do Brasil. O objetivo aqui é analisar a dinâmica de uma doença sobre grafos depois de alterar a sua topologia. Os números de pessoas em cada vértice das redes de entrada foram obtidos através de dados oficiais (IBGE e Data.Rio). Apresentamos os resultados experimentais comparando o desenvolvimento de uma doença ao longo do tempo sobre os grafos originais com subgrafos minimamente conexos (árvores). Com base nos resultados experimentais observados, podemos dizer que na grande maioria dos casos, há maior controle sobre o espalhamento de uma doença sobre um grafo conexo quando a topologia do grafo é alterada para uma árvore.

Para os experimentos com o modelo proposto, foram construídas duas instâncias. As instâncias representam de certa forma as cidades de Florianópolis (SC) e Rio de Janeiro (RJ). Em seguida, a construção do grafo de infecção da cidade do Rio de Janeiro é descrita. Para o grafo de infecção da cidade Florianópolis, a construção foi feita de forma similar e descrita no seguinte trabalho de conclusão de curso [Ataide 2022]. No entanto, é importante pontuar que para o caso da cidade de Florianópolis, os vértices correspondem a grupos de bairros vizinhos, simplificando a construção deste grafo de infecção. No total, foram desenvolvidos 16 grupos de bairros e 25 conexões interligando os grupos. Os grupos combinam bairros que estão em regiões próximas uns dos outros e foram assim definidos: Grupo 1: Cachoeira do Bom Jesus, Ingleses, Canasvieiras, Pontas das Canas, Santinho; Grupo 2: Vargem Grande, Rio Vermelho; Grupo 3: Vargem Pequena, Jurerê; Grupo 4: Santo Antônio de Lisboa, Ratoões, Saco Grande, Costa da Lagoa; Grupo 5: Itacorubi, João Paulo, Lagoa da Conceição, Barra da Lagoa; Grupo 6: Balneário, Jardim Atlântico, Estreito, Novo Continente; Grupo 7: Coloninha, Sapé, Monte Cristo, Capoeiras; Grupo 8: Vila Aparecida, Abraão, Coqueiros; Grupo 9: Agrônômica, Prainha, Centro, Monte Serrat; Grupo 10: Trindade, Pantanal, Saco dos Limões; Grupo 11: Rio Tavares, Costeira do Pirajubaé; Grupo 12: Canto da Lagoa, Córrego Grande; Grupo 13: Tapera, Carianos, Fazenda do Rio Tavares, Campeche; Grupo 14: Morro das Pedras, Armação; Grupo 15: Ribeirão da ilha, Alto Ribeirão; Grupo 16: Caieira da Barra do Sul, Pântano do Sul. A Figura 1 (à esquerda) ilustra o desenho do grafo simples.

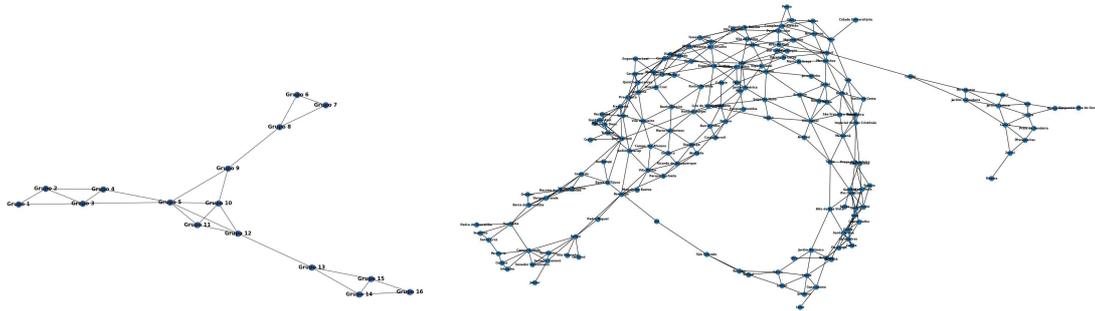


Figura 1: Grafos simples das instâncias de Florianópolis (à esquerda) e do Rio de Janeiro (à direita).

Já para a cidade do Rio de Janeiro, a construção do grafo de infecção não considera essa simplificação. Os vértices do grafos da cidade do Rio de Janeiro representam bairros da cidade, as arestas representam uma vizinhança entre bairros. Na Figura 2, temos o mapa do município do Rio de Janeiro com destaque para os bairros da cidade (números indo-arábicos).

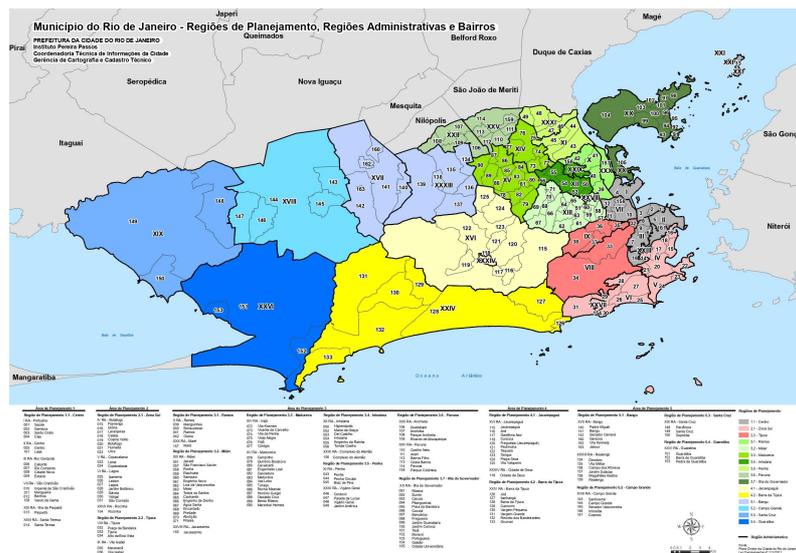


Figura 2: Município do Rio de Janeiro, Fonte: Plano Diretor da Cidade do Rio de Janeiro Lei complementar n<sup>o</sup> 111/2011.

Note que a cidade do Rio está dividida em 16 *regiões de planejamento*. Cada região possui vários bairros. Para esta instância, a construção foi *mais fina*, representando cada bairro em um vértice do grafo. Então, os vértices adjacentes ao vértice correspondente ao bairro 024 (Copacabana) são os vértices 023 (Leme), 020 (Botafogo), 027 (Lagoa) e 025 (Ipanema). Dessa forma, criamos uma lista de adjacências para cada vértice e construímos o grafo simples desta instância que está ilustrado na Figura 1 (à direita).

A população de cada bairro foi obtida a partir de dados oficiais do <https://www.rio.rj.gov.br/web/ipp/who-we-are> Instituto Municipal de Urbanismo Pereira Passos, referência na divulgação de dados geográficos para o planejamento urbano e produção cartográfica do Rio. Este instituto mantém o portal chamado <https://www.data.rio/Data.Rio> que disponibiliza a população do Rio de Janeiro por bairro, extraídos a partir do Censo Demográfico do Brasil de 2010 realizado pelo <https://www.ibge.gov.br/> Instituto Brasileiro de Geografia e Estatística (IBGE). Algumas alterações foram necessárias considerar.

Os dados do Data.Rio não possuem a população dos bairros Lapa, Vila Kennedy e Jabour, pois em 2010 tais bairros não existiam. Por causa disso estes bairros tiveram que ser removidos e as adjacências atualizadas para refletir como os bairros eram em tal ano. Nos nossos experimentos, o bairro Paquetá também não foi considerado por se tratar de uma ilha sem acesso por ruas, apenas por barcas que transportam uma quantidade de pessoas que pode ser no máximo 0,0005% da população total da cidade.

Ainda foram consideradas as seguintes simplificações. Toda aresta  $i - j$  possui fluxo de pessoas de  $i \rightarrow j$  e de  $j \rightarrow i$ . Para o caso do Rio, a quantidade de pessoas do vértice  $i$  que circula no próprio vértice e nos vértices adjacentes é igualmente proporcional ao número total de pessoas de  $i$ , ou seja, fazemos  $\beta_{i \rightarrow j}^t = \frac{1}{|\mathcal{X}(i)|+1}$  para todo vértice  $i$  e todo tempo  $t$ . Para o caso de Florianópolis, cada valor  $\beta_{i \rightarrow j}^t$  foi definido como sendo a proporção entre a população de  $j$  e a soma das populações de  $i$  e de todos os vértices adjacentes a  $i$ . Para ambas as instâncias, o fator de distanciamento social foi fixado em 40% para todo vértice  $i$  e todo tempo  $t$ . As pessoas que atendem ao distanciamento social mas que ainda saem para circular no próprio vértice são as seguintes frações  $\frac{2}{5}$  para os suscetíveis,  $\frac{1}{10}$  para os infectados e  $\frac{3}{5}$  para os recuperados.

A movimentação das pessoas que o modelo considera pode ser mais detalhada agora. A Figura 3 destaca o número de pessoas que estão circulando no Flamengo no dia 30.

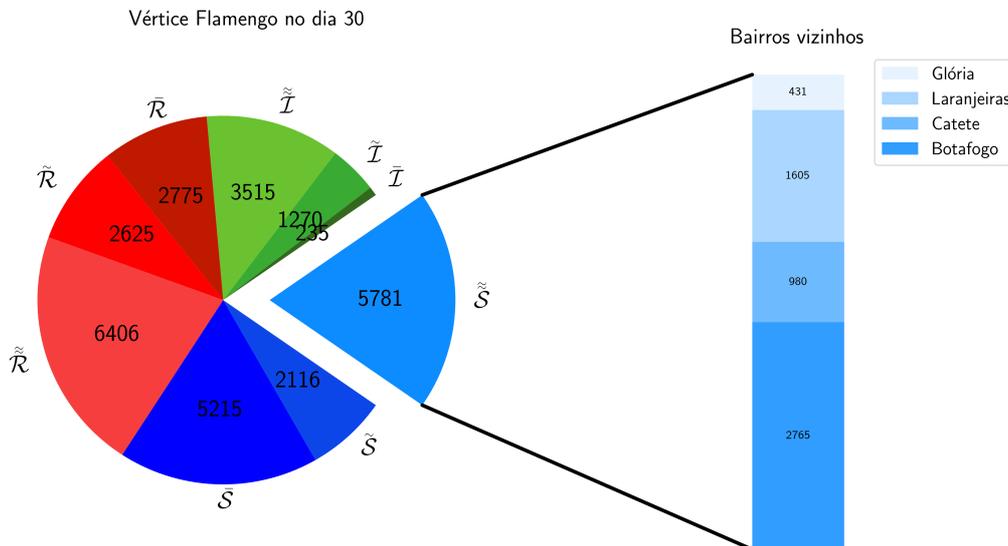


Figura 3: Estão em destaque somente as pessoas que circulam no Flamengo no tempo  $t = 30$ . Os números totais de suscetíveis, infectados e recuperados que atendem ao distanciamento são 11.594, 2.635 e 5.787, respectivamente.

Existem pessoas do próprio Flamengo que atendem ao distanciamento social ( $\bar{S}$ ,  $\bar{I}$  e  $\bar{R}$ ); pessoas do próprio Flamengo que não atendem ao distanciamento social ( $\tilde{S}$ ,  $\tilde{I}$  e  $\tilde{R}$ ); e pessoas de bairros vizinhos ao Flamengo que não atendem ao distanciamento social ( $\check{S}$ ,  $\check{I}$  e  $\check{R}$ ). A figura destaca também o número de pessoas suscetíveis circulando no Flamengo mas pertencentes a bairros vizinhos ao Flamengo. Esse número é obtido através da distribuição de pessoas suscetíveis desses bairros (usando  $\beta_{i \rightarrow j}^t$  com  $j$  igual a Flamengo e  $i$  um bairro vizinho). Em seguida, algumas árvores geradoras do grafo de infecção original são construídas e os resultados são comentados.

A principal ideia é construir subgrafos minimamente conexos envolvendo todos os vértices do grafo original para assim manter a característica de que no grafo obtido ainda há caminhos entre quaisquer vértices do grafo original. Com isso, seria possível movimentar mercadorias por todo o grafo, por exemplo. De fato, o modelo considera uma restrição sobre o movimento das pessoas, mas não sobre o movimento das mercadorias, caso este seja o caso.

Quando temos um subgrafo que é uma árvore contendo todos os vértices do grafo original, então essa árvore é chamada de *geradora* do grafo original [Bondy and Murty 1976]. Utilizamos *árvores geradoras* obtidas através do processo de busca em largura (em profundidade) sobre o grafo original. Tais processos, devolvem *árvores de busca em largura* ou (*em profundidade*). O livro [Dasgupta et al. 2009] é uma boa referência de leitura sobre as buscas em largura (em profundidade) e as características das correspondentes árvores devolvidas pelas buscas. Na Figura 4, temos um exemplo de uma árvore de busca em profundidade considerando o grafo de infecção do Rio de Janeiro.

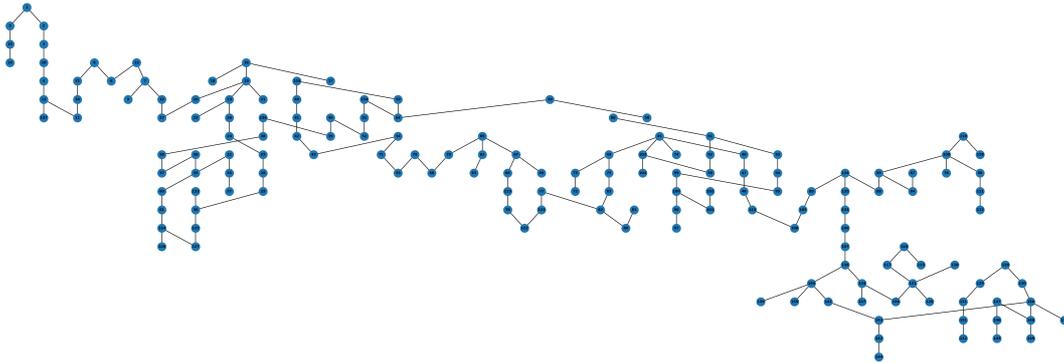


Figura 4: Árvore de busca em profundidade do grafo de infecção do Rio de Janeiro considerando como raiz da árvore o bairro Flamengo (vértice com rótulo 14 localizado na região de planejamento: Zona Sul).

Agora pode-se comparar o comportamento do espalhamento de uma doença sobre o grafo de infecção original e sobre uma árvore geradora, com as mesmas condições iniciais de pessoas infectadas para ambos os casos. No primeiro experimento, observamos os resultados de duas simulações do espalhamento de uma doença considerando infectados somente no bairro Flamengo - Figura 5 (na linha cheia, o experimento considerado foi aplicado sobre o grafo de infecção original da cidade do Rio de Janeiro, enquanto que na linha triangulada, o experimento considerado foi aplicado sobre uma árvore geradora (em largura) com raiz no Flamengo). No início da simulação, haviam 2.502 pessoas infectadas no bairro Flamengo (5% da população do bairro). Note que houve uma certa amortização na curva dos infectados quando comparamos os resultados considerando o grafo original e uma árvore geradora (em largura). Além disso, o número de infectados no grafo original atingiu o pico com 810.106 pessoas, enquanto que na árvore geradora, o pico foi de 527.026 pessoas.

Os resultados iniciais obtidos motivou a equipe deste projeto a verificar o espalhamento de uma doença sobre um grafo considerando inicialmente infectados em cada um dos bairros da cidade. Como existem 159 vértices no grafo do Rio de Janeiro, foram construídas  $159 \times 2$  árvores geradoras (159 árvores de busca em largura e 159 árvores de busca em profundidade - cada bairro sendo uma raiz de exatamente duas árvores). O resultados obtidos estão ilustrados na Figura 6.

Os experimentos realizados indicam que manter o grafo de infecção original proporciona, para quase todos os casos, um pico de pessoas infectadas maior que as topologias das árvores geradoras testadas. As árvores de busca em profundidade possuem grande destaque no que diz respeito ao número de pessoas infectadas no pico, tal número parece convergir para uma constante. No entanto, vale a pena citar que em alguns casos, o grafo de infecção original é a melhor opção de topologia pois a doença não conseguiu se espalhar (a distribuição das pessoas da forma como foi considerada no modelo implicou no número esperado de infectados igual a 0 (zero) para todo vértice logo no início da simulação). Isso acontece quando os infectados estão somente nos bairros Saúde, Cidade Nova, Barra de Guaratiba ou Jacaré, por exemplo. Para poucos casos, temos o grafo original tendo menos infectados no pico que uma árvore de busca em largura, mas não que uma árvore de busca em profundidade (como no caso dos bairros Vista Alegre, Cocotá, Deodoro ou Padre Miguel). Ainda temos o caso de infectados no bairro Vaz Lobo, cujo pico de infectados não é igual a zero para qualquer topologia considerada, mas a melhor opção é o grafo

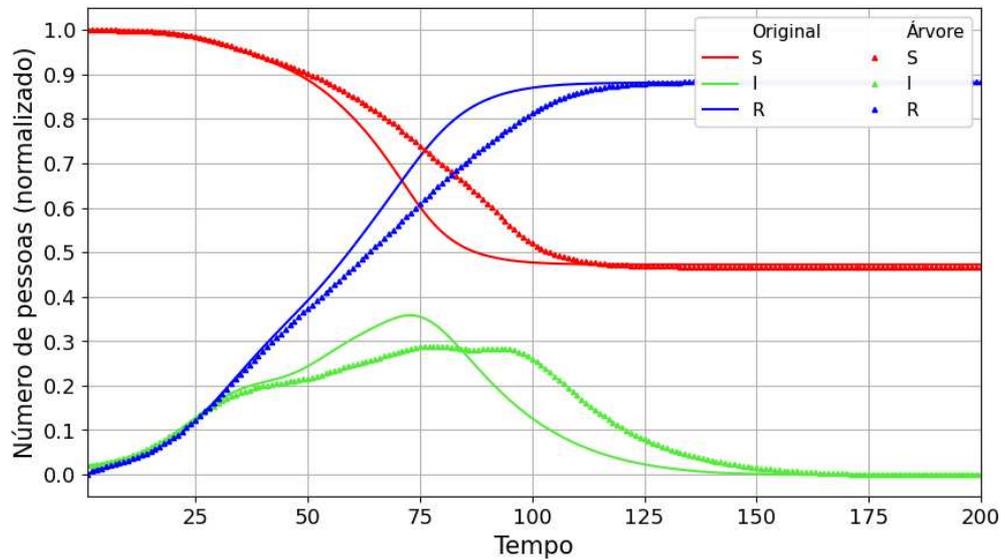


Figura 5: Rio de Janeiro - Resultado da simulação sobre o grafo de infecção original (linha cheia) e sobre uma árvore geradora com raiz no Flamengo (linha triangulada) - a condição inicial considera que 5% do total de pessoas que vivem no bairro Flamengo estão infectadas.

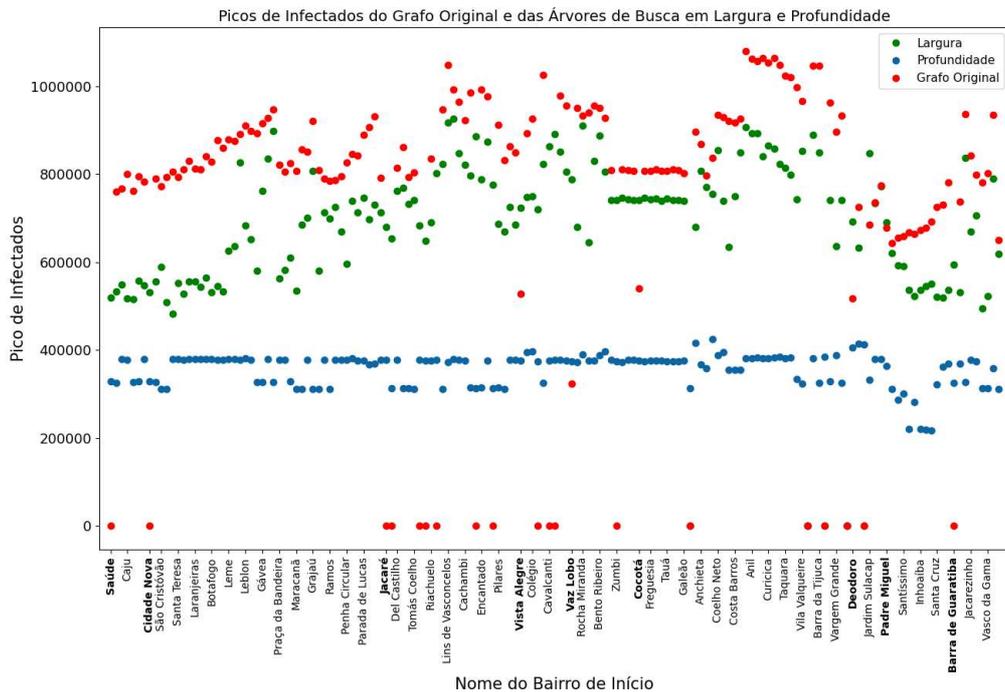


Figura 6: Rio de Janeiro - Observe três curvas para os picos de infectados para cada uma das árvores de busca em largura, em profundidade e para o grafo de infecção original. Alguns bairros da cidade (no eixo  $x$ ) foram descritos e aqueles citados no texto aparecem em negrito.

de infecção original.

Tais experimentos também foram realizados no grafo de infecção da cidade de Florianópolis. Para

este caso, aplicamos o modelo sobre o grafo de infecção original e também sobre árvores de busca em largura e em profundidade - cada grupo de bairros sendo a raiz de uma árvore de busca em largura e em profundidade totalizando  $16 \times 2$  árvores geradoras. Os resultados relacionados ao número de infectados no pico podem ser observados na Figura 7. Vale a pena lembrar que neste caso, são conhecidos os números reais de suscetíveis, infectados, e recuperados de cada vértice em um determinado dia (dados reais). A linha horizontal que cruza esta figura está sobre o número 83.271, o número de infectados no pico considerando o grafo de infecção original. Neste caso, os experimentos nos indicam que há muitas árvores geradoras (daquelas consideradas) cujo o número de infectados no pico é maior que o número de infectados no pico do grafo de infecção original. A maioria das árvores de busca em profundidade possuem pico de infectados maior que o pico de infectados do grafo de infecção original, enquanto que a maioria das árvores de busca em largura possuem pico de infectados menor. Das árvores geradoras consideradas, aquela que obteve o menor número de infectados no pico foi uma árvore de busca em largura com raiz no Grupo 11.

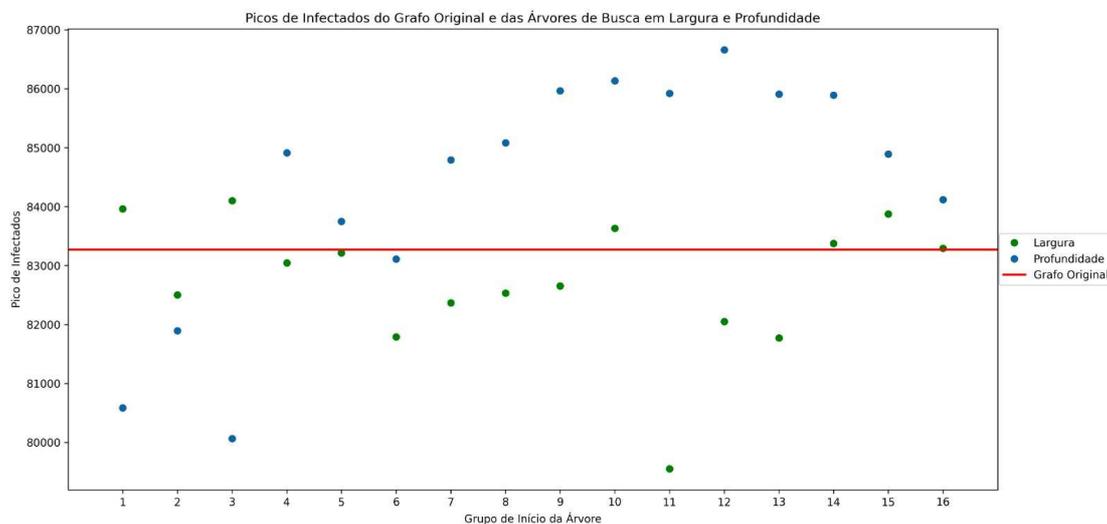


Figura 7: Florianópolis - Observe duas curvas para os picos de infectados para cada uma das árvores de busca em largura e em profundidade, além do pico de infectados considerando o grafo de infecção original.

Os experimentos realizados indicam que, na maioria dos casos, a restrição da topologia nos leva a um menor pico de infecção no grafo. Isso acontece em decorrência da restrição de movimento que essas mudanças topológicas proporcionam. A movimentação das pessoas é dificultada e os experimentos ilustram que o espalhamento da doença no grafo é normalmente reduzido. Com essas restrições, pode ser impossível a movimentação direta de pessoas infectadas entre vértices adjacentes do grafo original. Com isso, a infecção pode levar mais tempo para atingir tais vértices. Vale ressaltar a importância de controlar o espalhamento de uma doença. Uma menor quantidade de pessoas infectadas simultaneamente implica em menor ocupação em hospitais e em UTIs, por exemplo, o que nos leva a possibilidade de tratar eficientemente cada um dos pacientes de acordo com a gravidade dos sintomas.

Este trabalho conta ainda com o desenvolvimento de uma heurística que encontra novas árvores geradoras com o objetivo de ampliar o espaço de busca de árvores geradoras do grafo original. A heurística funciona da seguinte maneira. Dada uma árvore geradora  $T$ , é encontrada uma sequência de *árvores vizinhas* que ou diminuem o número de infectados no pico considerando  $T$  ou este número continua igual. A sequência é iniciada a partir da árvore geradora  $T$  dada (árvores vizinhas são árvores que diferem de uma aresta). A última árvore dessa sequência é um mínimo local segundo a definição de árvores vizinhas. Dessa forma, para alguns casos, obtemos árvores geradoras com um número de infectados no pico ainda menor que anteriormente. Tais resultados podem ser observados em [Caninas et al. 2023].

Sobre este trabalho houveram

- uma apresentação oral e presencial no Encontro Acadêmico de Matemática (Universidade Federal de Santa Catarina - UFSC) com acesso pela página
  - <https://www.inf.ufsc.br/~alvaro.junio/proj-docs/eamat-2023-1.pdf>
- uma publicação de um relatório de iniciação científica (Universidade Federal de Santa Catarina - UFSC) com acesso pela página
  - <https://pibic.sistemas.ufsc.br/pibicFile/file/1073328?tipo=1>
  - <https://www.inf.ufsc.br/~alvaro.junio/proj-docs/ric-2023.pdf>
- uma apresentação oral e presencial no Simpósio de Iniciação Científica (Universidade Federal de Santa Catarina - UFSC) com acesso pelas páginas
  - <https://repositorio.ufsc.br/handle/123456789/250872>
  - <https://www.inf.ufsc.br/~alvaro.junio/proj-docs/sic-2023.pdf>
- uma publicação de um artigo e uma apresentação oral e presencial no Simpósio Brasileiro de Pesquisa Operacional (SBPO 2023 - Universidade Federal de São Paulo - UNIFESP e Instituto Tecnológico de Aeronáutica - ITA) com acesso pelas páginas
  - <https://proceedings.science/sbpo-2023/trabalhos/um-estudo-sobre-arvores-geradoras-como-alternativas-conexas-para-o-controle-de-d?lang=pt-br>
  - <https://www.inf.ufsc.br/~alvaro.junio/proj-docs/sbpo-2023.pdf>
- uma indicação ao 21o. Prêmio Destaque na Iniciação Científica e Tecnológica (Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq)
  - <http://pibic.propesq.ufsc.br/2024/02/09/indicados-ao-21o-premio-destaque-na-iniciacao-cientifica-e-tecnologica/>
- implementações com acesso pela página
  - <https://github.com/Caninas/IC-UFSC-2022-2023>

Em seguida, são iniciadas as discussões dos trabalhos relacionados ao problema geral da alocação de recursos.

## Um Problema de Agenda Escolar no Estado de Santa Catarina

**Trabalho em conjunto com Edmilson Domingues, Luiza B. M. Guimarães e Lucas H. G. Wodtke.<sup>2</sup>**

Até o momento, os principais resultados deste trabalho foram os desenvolvimentos de modelos de programação inteira e de uma heurística (um algoritmo genético) para o problema da agenda escolar. O Lucas H. G. Wodtke desenvolveu um modelo de programação inteira durante o seu trabalho de conclusão de curso. O Edmilson Domingues, também durante o seu trabalho de conclusão de curso, desenvolveu um novo modelo tomando como base o modelo do Lucas. No momento, a Luiza B. M. Guimarães está realizando uma iniciação científica e está desenvolvendo um novo modelo para o problema. Os resultados apresentados nesta seção são os resultados mais consolidados para o problema até o momento.

Este trabalho aborda o problema da agenda escolar, que vem sendo tratado pelo menos desde a década de 60. É válido pensar que tal problema tenha muitas variações, visto que uma agenda escolar para ser implementada pode ter que obedecer a várias regras específicas de uma escola em particular. No entanto, há também regras comuns a qualquer uma dessas variações. Em [Gotlieb 1963], o autor

<sup>2</sup>Agradecimentos aos colegas Gabriel V. Ferrari e Moema L. Cargnin pelas instâncias de escolas de Santa Catarina.

descreve uma solução para o problema como sendo um emparelhamento entre professores e disciplinas sob algumas condições: os horários devem ser viáveis para ambos (professor e disciplina); um horário não pode ser ocupado por duas disciplinas de uma turma, um professor não pode emparelhar com duas disciplinas com horários iguais etc. Neste mesmo trabalho, Gotlieb propõe um método para encontrar uma solução viável para o problema. Já os autores do seguinte trabalho [Csimá and Gotlieb 1964] revisitaram o método de Gotlieb e conjecturaram que tal método revisitado ou encontrava uma solução viável ou não existia qualquer solução. No entanto, o trabalho [Lions 1966] mostrou um contraexemplo. Dessa forma o método de Csimá e Gotlieb pode falhar. Apesar disso, o autor de [Lions 1967] implementou o método de Csimá e Gotlieb, fez uma observação de que tal método parece falhar para poucas instâncias, e aplicou o método a oito escolas de Ontário/Canadá, obtendo soluções viáveis para algumas delas. O problema é importante para a área de Otimização Combinatória, é NP-difícil [Even et al. 1975] e possui muitas contribuições como, por exemplo, [Carter and Laporte 1997], [Daskalaki and Birbas 2005], [Pillay 2014], [Babaei et al. 2015],[Wodtke 2022], [Domingues 2023]. Vale ainda citar que existe uma conferência que recebe trabalhos relacionados ao problema da agenda escolar (PATAT - *International Conference on the Practice and Theory of Automated Timetabling* e que está na sua 14a. edição no ano de 2024). O presente trabalho apresenta duas abordagens para um problema da agenda escolar com regras específicas para escolas do estado de Santa Catarina. A primeira abordagem descreve um modelo de programação inteira e utiliza a ferramenta *Gurobi*. A segunda, apresenta um algoritmo genético que pode encontrar soluções viáveis para o problema, mas também pode falhar. Em seguida, a notação para este trabalho é apresentada. O conjunto  $S_i$  possui os períodos de tempo (ou *slots*) do dia  $i$  da semana ( $i = 2, 3, \dots, 6$  - segunda, terça, ...);  $S_p$  são os *slots* disponíveis do professor  $p$ ;  $S_t$  possui os *slots* da turma  $t$ . Todos os *slots* possuem o mesmo tamanho. O conjunto  $D_t$  possui as disciplinas da turma  $t$ ;  $P_d$  possui os professores habilitados para a disciplina  $d$ ;  $C_p$  são os créditos que o professor  $p$  tem disponível;  $C_d$  são os créditos da disciplina  $d$ . Em seguida, apresentamos um modelo de programação inteira e um algoritmo genético para o problema. Por fim, apresentamos os resultados obtidos quando aplicadas as duas abordagens sobre as instâncias de escolas de Santa Catarina cujo objetivo é encontrar um horário para disciplinas já associadas aos professores e que minimize uma medida que depende de como estão arranjados os professores nos dias da semana. Vamos agora para o modelo de programação inteira para o problema da agenda escolar.

As variáveis binárias do problema são as seguintes:  $x_{tdps} = 1$ , se para a turma  $t$  a disciplina  $d$  é ministrada pelo professor  $p$  no *slot*  $s$ ;  $x_{tdps} = 0$ , caso contrário;  $y_{ps} = 1$ , se ocorre uma *janela* para o professor  $p$  no *slot*  $s$ ;  $y_{ps} = 0$ , caso contrário. Um professor tem uma *janela* quando existe (exatamente) um *slot* vago entre dois já associados a ele. O objetivo é minimizar a soma do número de janelas dos professores. As restrições são as seguintes, em ordem: (1) Cada *slot* de cada turma é ocupado no máximo uma vez por um emparelhamento (professor - disciplina); (2) Cada *slot* de cada professor é ocupado por no máximo uma disciplina de uma turma; (3) Cada disciplina de cada turma e para cada dia da semana, um emparelhamento (professor-disciplina) ocorre no máximo uma vez; (4) Cada professor precisa completar os seus créditos; (5) Cada disciplina de cada turma precisa ser dada de forma completa; (6) Cada janela é contabilizada.

$$\sum_{\forall d \in D_t} \sum_{\forall p \in P_d : s \in S_p} x_{tdps} \leq 1, \forall t \in T \text{ e } \forall s \in S_t; \quad (1)$$

$$\sum_{\forall t \in T : s \in S_t} \sum_{\forall d \in D_t : p \in P_d} x_{tdps} \leq 1, \forall p \in P \text{ e } \forall s \in S_p; \quad (2)$$

$$\sum_{\forall p \in P_d} \sum_{\forall s \in S_i \cap S_p \cap S_t} x_{tdps} \leq 1, \forall i = 2, \dots, 6 \forall t \in T \text{ e } \forall d \in D_t; \quad (3)$$

$$\sum_{\forall t \in T} \sum_{\forall d \in D_t} \sum_{\forall s \in S_p : p \in P_d} x_{tdps} = C_p, \forall p \in P; \quad (4)$$

$$\sum_{\forall p \in P_d} \sum_{\forall s \in S_t \cap S_p} x_{tdps} = C_d, \forall t \in T \text{ e } \forall d \in D_t; \quad (5)$$

$$2x_{tdps-} - 2x_{tdps} + 2x_{tdps+} \geq 3 - M(1 - y_{ps}); 2x_{tdps-} - 2x_{tdps} + 2x_{tdps+} \leq 3 + M(y_{ps}).^3 \quad (6)$$

Em seguida, é apresentado um algoritmo genético que é utilizado como uma heurística para encontrar soluções viáveis para o problema da agenda escolar.

A implementação tenta preencher a agenda de uma escola considerando uma turma por vez. Para dada turma  $t$ , são consideradas uma população inicial com 50 indivíduos construídos ao acaso, ou seja, 50 soluções para o problema, provavelmente todas inviáveis; uma operação de cruzamento (dois indivíduos geram dois novos indivíduos); e uma operação de mutação sob indivíduos gerados a partir de um cruzamento. O *fitness* de cada indivíduo é calculado a partir do atendimento das restrições, com exceção da (6). No algoritmo genético, a restrição (6) foi substituída por uma restrição que fixa o número de janelas que poderiam ocorrer (começando com 0 janelas). A cada  $G$  gerações o valor desse número  $w$  é relaxado para  $w + 1$  janelas. A execução para a turma  $t$  termina quando uma solução viável considerando somente a turma  $t$  é encontrada. A execução para a agenda da escola é finalizada quando temos uma agenda para cada turma da escola.

Os 10 indivíduos com melhor *fitness* de cada geração fazem parte da *elite* desta geração (20%). A operação de *seleção* dos indivíduos para um possível cruzamento é uma *roleta ponderada* sobre os indivíduos da geração. Adicionalmente, a elite é selecionada. Nesta etapa, poderá haver repetição de indivíduos. Então, os indivíduos de um par selecionado irão se cruzar com probabilidade de 80%. O cruzamento gera dois indivíduos. Poderá ocorrer *mutação* em cada um deles.

O cruzamento segue aquele apresentado em [Luger 2004]. Pense nos  $n$  *slots* de dois indivíduos  $p_0$  e  $p_1$  de forma linear. Escolha dois valores distintos ao acaso no intervalo  $[0: n - 1]$  e chame-os de  $k$  e  $k + l$ . Os indivíduos  $p_0 = s_0^0, \dots, s_{n-1}^0$  e  $p_1 = s_0^1, \dots, s_{n-1}^1$  são particionados nas posições  $k$  e  $k + l$ ,  $p_0 = x_0 y_0 z_0$  e  $p_1 = x_1 y_1 z_1$  sendo  $x_i = s_0^i, \dots, s_k^i$ ,  $y_i = s_{k+1}^i, \dots, s_{k+l}^i$  e  $z_i = s_{k+l+1}^i, \dots, s_{n-1}^i$ . Depois, são gerados dois filhos  $f_0 = w_0^0 y_0 w_1^0$  e  $f_1 = w_0^1 y_1 w_1^1$  sendo  $w_0^i$  e  $w_1^i$  preenchidos em ordem (do início para o fim) por  $s_{k+l+1 \bmod 2}^{i+1}, \dots, s_{k+l+n-1 \bmod 2}^{i+1}$  sem repetição sobre  $y_i$  e até completar todos os *slots* de  $f_i$ . Em seguida ilustramos um caso com 9 *slots*:  $p_0 = 1, 9, 2|4, 6, 5, 7|8, 3$ ;  $p_1 = 4, 5, 9|1, 8, 7, 6|2, 3$ ;  $f_0 = 2, 3, 9|4, 6, 5, 7|1, 8$ ; e  $f_1 = 3, 9, 2|1, 8, 7, 6|4, 5$ . Poderá ocorrer mutação em um *gene* (ou seja, um *slot* preenchido), de um filho  $f_i$  com probabilidade de 1%. Caso a mutação ocorra, um outro gene diferente é escolhido ao acaso e o valor preenchido nos *slots* são trocados. Em seguida, apresentamos os resultados até o momento.

Para resolver o modelo de programação inteira utilizamos a versão acadêmica da ferramenta *Gurobi Optimizer* versão 9.1.2. Originalmente, pensamos em utilizar o algoritmo genético desenvolvido em conjunto com o *Gurobi* como uma alternativa para gerar soluções viáveis para o problema inicial, antes da ferramenta iniciar a busca pela solução ótima pela árvore de *Branch & Bound*. No entanto, apesar dos experimentos mostrarem que as soluções viáveis dadas pelo algoritmo genético têm boa qualidade e podem ser utilizadas na prática, tal heurística não se mostrou eficiente em termos do tempo de execução. Isso inviabilizou a utilização em conjunto com o *Gurobi*. Apesar dessa desvantagem, a implementação deste algoritmo genético é livre, ela foi realizada em *Python* versão 3.6 e está disponível<sup>4</sup>. Por outro lado, o *Gurobi* não é uma ferramenta livre e sua utilização depende de uma licença que pode ser acadêmica ou comercial. Todos os experimentos foram executados em uma máquina Intel(R) Core(TM) i5-7200 CPU, 2.50 GHz, 8 GB de memória RAM e sistema operacional Windows - 64 *bits*. No total temos 129 instâncias reais. O *Gurobi* encontrou solução ótima para 121 instâncias. As 8 instâncias restantes precisam ser revisadas pelo cliente pois acreditamos estarem com problemas. Os tempos de execução e os valores das soluções ótimas estão na Figura 8.

Os tempos de execução do algoritmo genético são muito superiores aos do *Gurobi*. No entanto, a qualidade das soluções viáveis encontradas pode ser observada quando comparadas a algumas soluções ótimas. Isso está ilustrado na Figura 9. É importante citar que executamos este algoritmo genético para 42 instâncias. A figura ilustra o valor das 36 soluções viáveis encontradas. Este algoritmo falhou para 6 instâncias.

Para mais detalhes sobre as soluções propostas para o problema da agenda escolar, o leitor pode verificar os trabalhos de conclusão de curso de Lucas H. G. Wodtke [Wodtke 2022] e Edmison Domingues

<sup>3</sup>A constante  $M$  tem valor igual a 10. Além disso,  $s-$  e  $s+$  são os *slots* antecessor e sucessor de  $s$ . A restrição é feita  $\forall t \in T, \forall d \in D_t, \forall p \in P_d$  e  $\forall s \in S_t \cap S_p$ .

<sup>4</sup>[https://github.com/EdmilsonDomingues/TCC\\_Edmilson-Domingues](https://github.com/EdmilsonDomingues/TCC_Edmilson-Domingues)

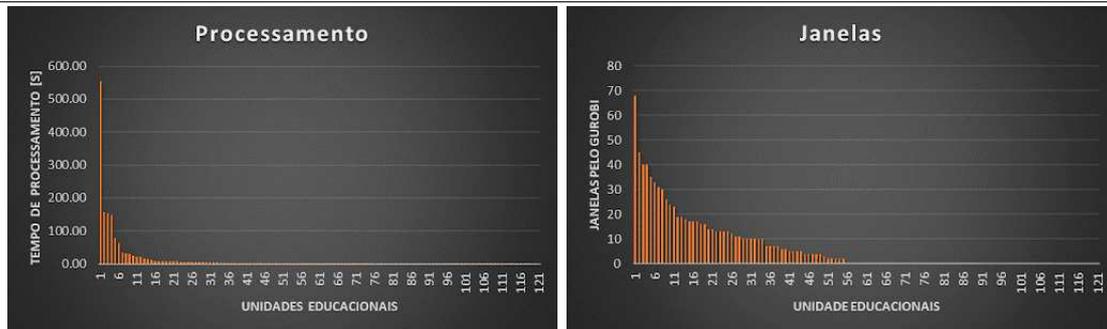


Figura 8: Tempo de execução e número de janelas por unidade educacional

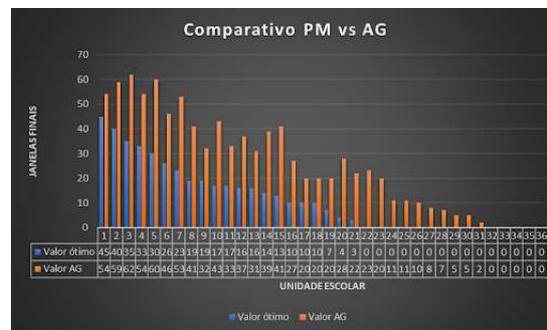


Figura 9: Soluções ótimas versus soluções viáveis do algoritmo genético

[Domingues 2023].

Sobre este trabalho houveram

- uma apresentação oral e presencial no Encontro Acadêmico de Matemática (Universidade Federal de Santa Catarina - UFSC) com acesso pela página
  - <https://www.inf.ufsc.br/~alvaro.junio/proj-docs/eamat-2023-2.pdf>
- duas publicações de trabalhos de conclusão de curso com acesso pela página
  - <https://repositorio.ufsc.br/handle/123456789/237812>
  - <https://repositorio.ufsc.br/handle/123456789/253139>
- implementações com acesso pela página
  - [https://github.com/EdmilsonDomingues/TCC\\_Edmilson-Domingues](https://github.com/EdmilsonDomingues/TCC_Edmilson-Domingues)

Há ainda uma iniciação científica sobre este assunto que está sendo desenvolvida por Luiza B. M. Guimarães.

### Aprimorando o Desempenho de Sistemas Combinando Checkpointing e Particionamento Dinâmico de Estados

Trabalho em conjunto com Henrique dos Santos Goulart e Odorico M. Mendizabal.

O trabalho que é descrito agora está relacionado ao problema de particionar o estado de um sistema com o objetivo de melhorar o seu desempenho através do aumento do número de requisições atendidas ao longo do tempo. O modelo do sistema com particionamento de estados é discutido em seguida. O

estado do sistema  $S$  é um conjunto de variáveis que está particionado em  $n$  subconjuntos disjuntos, cada um chamado de partição. O conjunto de partições é  $\{p_1, p_2, \dots, p_n\}$  e a união dos elementos de cada  $p_i$  é igual a  $S$ , isto é,  $\bigcup_{i=1}^n p_i = S$ . O sistema é paralelo e executa  $n$  *threads* trabalhadoras, sendo a *thread*  $t_i$  a responsável pela execução de requisições envolvendo a partição  $p_i$ . Uma requisição arbitrária  $r$  executa sobre um conjunto de variáveis do sistema. Seja  $K(r)$  o conjunto de variáveis acessadas por  $r$ , e seja  $P(k)$  a partição para a qual a variável  $k$  foi designada. Toda *thread*  $t_i$  tem sua própria fila  $q_i$ , com as requisições que acessam sua partição  $p_i$ . Assim, quando uma requisição  $r$  chega, um escalonador atribui  $r$  na fila de todas as *threads* cuja variáveis são acessadas por  $r$ . Formalmente,  $r$  é colocada na fila  $q_i$  para todo  $i$  tal que  $p_i \in \bigcup_{k \in K(r)} P(k)$ .

Como um exemplo, assuma um sistema cujo estado  $S = \{x, y, w, z\}$  e um conjunto de *threads*  $T = \{t_1, t_2\}$ . A figura 10 representa um possível particionamento com as *threads*  $t_1$  e  $t_2$  sendo responsáveis por acessar/atualizar as variáveis nas partições  $p_1 = \{x, y\}$  e  $p_2 = \{w, z\}$ , respectivamente. A figura também mostra a fila de *threads* depois de receber as requisições de  $r_1$  até  $r_7$ . Como pode ser observado,  $r_1$  acessa  $x$ , e como  $x \in p_1$ ,  $r_1$  é despachada para  $t_1$ , e assim por diante para toda requisição. Requisições são despachadas respeitando a ordem de chegada na fila apropriada. Neste exemplo, as semânticas das requisições são definidas pelos comandos *write*( $k, v$ ) and *swap*( $k_i, k_j$ ), sendo que *write* atualiza a variável dada pela chave  $k$  com um valor  $v$  e *swap* muda mutuamente os valores das variáveis cujas chaves são  $k_i$  e  $k_j$ .

Na figura 10, as requisições  $r_1, r_3, r_5, r_6$ , e  $r_7$  são requisições que envolvem uma única variável, e elas são executadas em ordem, uma a uma, sem sincronização entre as *threads* das partições. A requisição  $r_2$  é uma requisição que envolve duas variáveis, porém, as duas estão em uma mesma partição e portanto, não há necessidade de sincronização do sistema. A requisição  $r_4$  possui duas variáveis em partições diferentes. Neste caso o sistema precisa sincronizar as *threads* para manter o sistema consistente.

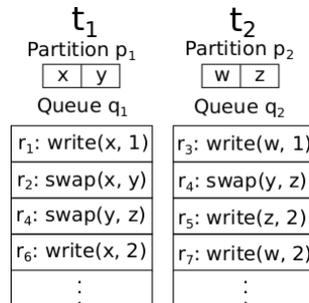


Figura 10: Exemplo de escalonamento de requisições entre as filas de *threads*

Em seguida o problema é definido formalmente. É importante dizer que a quantidade de variáveis utilizadas por requisições de leitura e escrita, possivelmente envolvendo muitas variáveis, é mantida. Este histórico pode ser organizado em um grafo com pesos nas arestas e nos vértices sendo que os vértices representam as variáveis, o número de requisições que usam uma variável é o peso no vértice correspondente ( $c_i$  para vértice  $i$ ), as arestas representam as requisições que utilizam muitas variáveis, e o peso da correspondente aresta dá o número de vezes que as duas variáveis aparecem em uma requisição com muitas variáveis ( $w_{ij}$  para aresta  $ij$ ). Agora, o sistema pode usar este grafo para sugerir, em certos momentos, uma partição de estados do sistema  $P = \{p_1, \dots, p_k\}$ . Aqui, cada  $p_i$  de  $P$  deve estar “balanceada”, ou seja, a soma dos pesos de todos os vértices pertencentes à partição não é tão diferente quando comparada com o peso de uma outra partição  $p_j$  de  $P$ ; uma outra propriedade importante é que a soma das arestas que tem vértices em partes diferentes de  $P$  é mínima. Do ponto de vista da aplicação, o primeiro objetivo ajuda a acelerar o armazenamento paralelo do estado, e o segundo ajuda a reduzir as sincronizações de requisições com muitas variáveis. Este problema é conhecido como o *problema de particionamento balanceado com corte mínimo*. O problema é NP-difícil [Garey et al. 1974], e muitos métodos heurísticos foram propostos, por exemplo, [Karypis and Kumar 1998], [Sanders and Schulz 2013], [Tsourakakis et al. 2014] e [Nishimura and Ugander 2013]. A seguir é descrita uma abordagem de modelagem do problema em

duas fases. A primeira fase encontra uma partição balanceada. A segunda fase encontra um corte mínimo considerando o particionamento balanceado encontrado na primeira fase. O número de partições não varia com o tempo e é dado como uma condição inicial. A modelagem das fases 1 e 2 é descrita em seguida.

**Fase 1.** Considere uma variável inteira  $y$ , que significa uma folga depois de empacotar todos os vértices em partições. Esta variável deve ser tão pequena quanto possível para garantir algum balanceamento. Considere também variáveis binárias  $x_{ip}$  definidas como segue.

$$x_{ip} = \begin{cases} 1, & \text{se o vértice } i \text{ pertence à partição } p, \\ 0, & \text{caso contrário.} \end{cases}$$

O número de partições  $k$  é fixo. Considere um conjunto com  $k$  partições  $P = \{p_1, \dots, p_k\}$  e  $C = \sum_{i \in V} c_i$ . O empacotamento dos vértices em partições evita criar partições muito “pesadas”. O modelo de programação inteira é o seguinte.

$$\begin{aligned} \min & y \\ \text{subject to: } & \sum_{p \in P} x_{ip} = 1, \forall i \in V & (1) \\ & \sum_{i \in V} c_i x_{ip} \leq \frac{C}{k} + y, \forall p \in P & (2) \\ & x_{ip} \in \{0, 1\}, \forall i \in V \text{ e } \forall p \in P & (3) \\ & y \in \mathbb{Z}^+. & (4) \end{aligned}$$

A restrição (1) garante que cada vértice está em exatamente uma partição. A restrição (2) garante que cada partição não é muito “pesada” em uma solução ótima pois a variável  $y$  é minimizada. As restrições (3) e (4) significam, respectivamente, que as variáveis  $x_{ip}$  são binárias e  $y$  é um inteiro.

**Fase 2.** Como o valor da variável  $y$  já é conhecido da fase 1, configura-se o valor para uma variável como sendo  $M = \frac{C}{k} + y$ . Agora, é desejado encontrar um corte mínimo considerando o valor de  $y$ . As variáveis  $x_{ip}$  são mantidas, e as novas variáveis binárias  $z_{ijp}$  são definidas como segue.

$$z_{ijp} = \begin{cases} 1, & \text{se aresta } i - j \text{ tem ambos os vértices } i \text{ e } j \text{ na partição } p, \\ 0, & \text{caso contrário.} \end{cases}$$

$$\begin{aligned} \min & \sum_{i-j \in E} w_{ij} (1 - \sum_{p \in P} z_{ijp}) \\ \text{subject to: } & \sum_{p \in P} x_{ip} = 1, \forall i \in V & (5) \\ & \sum_{i \in V} c_i x_{ip} \leq M, \forall p \in P & (6) \\ & z_{ijp} \leq x_{ip}, \forall i - j \in E \text{ e } \forall p \in P & (7) \\ & z_{ijp} \leq x_{jp}, \forall i - j \in E \text{ e } \forall p \in P & (8) \\ & z_{ijp} \geq \frac{1}{2}(x_{ip} + x_{jp}) - \frac{1}{2}, \forall i - j \in E, \text{ e } \forall p \in P & (9) \\ & x_{ip} \in \{0, 1\}, \forall i \in V \text{ e } \forall p \in P & (10) \\ & z_{ijp} \in \{0, 1\}, \forall i - j \in E. & (11) \end{aligned}$$

As restrições (5) e (6) garantem uma partição balanceada. As restrições (7), (8) e (9) encontram as arestas dentro das partições. As restrições (10) e (11) significam que as variáveis  $x_{ip}$  e  $z_{ijp}$  são binárias. Como a restrição (5) garante cada vértice  $i$  em exatamente uma partição  $p$ , para uma aresta  $i - j$  vale que ou ambos seus vértices estão em uma partição  $p$  (quando  $i$  e  $j$  estão em  $p$ ) ou no máximo um deles está em  $p$ . Quando ambos os vértices estão em  $p$ ,  $z_{ijp}$  é forçado ser 1 pela restrição (9). Caso contrário,  $z_{ijp}$  é forçado ser 0 pelas restrições (7) ou (8). Portanto, dada uma aresta fixa  $i - j$ , a soma  $\sum_{p \in P} z_{ijp}$  é igual a 1 somente quando ambos os vértices  $i$  e  $j$  estão em uma mesma partição  $p$ . Logo, a soma dos pesos das arestas entre as partições pode ser feita como descrita na função objetivo da fase 2.

Este trabalho segue com uma implementação de uma heurística para o problema, experimentos realizados sobre instâncias que simulam aplicações reais (*Yahoo! Cloud Serving Benchmark (YCSB)* [Cooper et al. 2010]), e uma análise dos resultados obtidos. O trabalho [Goulart et al. 2023] pode ser consultado para informações mais detalhadas.

Sobre este trabalho houveram

- uma publicação de um artigo e uma apresentação oral e presencial no Simpósio Internacional em Arquitetura de Computadores e Computação de Alto Desempenho (*International Symposium on Computer Architecture and High Performance Computing* - SBAC-PAD 2023 - Pontifícia Universidade Católica do Rio Grande do Sul - PUCRS) com acesso pelas páginas:
  - <https://doi.org/10.1109/SBAC-PAD59825.2023.00024>
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/sbac-pad-2023.pdf>
- uma publicação de uma dissertação de mestrado
  - <https://repositorio.ufsc.br/handle/123456789/254850>
- implementações com acesso pela página
  - <https://github.com/hensg/kvpaxos>

## Algoritmos para Super-Coloração em Grafos Dirigidos

Trabalho em conjunto com Wesly C. Ataide e Rafael de Santiago.

Dado um grafo dirigido com vértices coloridos, para cada vértice  $v$ , deseja-se obter o maior número de cores encontrado em vértices de um caminho simples começando de  $v$ . Entre as aplicações de interesse na literatura, existe uma na área da Antropologia que investiga a interação entre indivíduos de famílias e/ou clãs de um mesmo povo. Este trabalho projeta e analisa métodos computacionais e algumas propriedades para o problema de *super-coloração* em grafos dirigidos, particularmente em grafos dirigidos acíclicos (DAGs). Como contribuições, foram desenvolvidos dois algoritmos e um modelo de programação inteira para o problema de *super-coloração*. Os algoritmos foram implementados e avaliados em instâncias de povos reais. Este trabalho possui os resultados de um aluno de mestrado que ainda está em execução. Um artigo foi submetido para apreciação no Simpósio Brasileiro de Pesquisa Operacional de 2024 (veja seção **Anexos** no final deste texto). A previsão para a divulgação do resultado do comitê de programa do simpósio é dia 19 de agosto de 2024. Por enquanto, não há nenhuma publicação deste trabalho.

## 6. CONTRIBUIÇÕES DO PROJETO

*Relatar as contribuições do projeto para o avanço da ciência, apropriação dos resultados pela sociedade, impactos econômicos, ambientais e políticas públicas.*

Todas as implementações realizadas neste projeto são de livre acesso e estão disponíveis para utilização. O projeto possui contribuição na formação de recursos humanos. Foram formados três alunos de graduação, um aluno de mestrado e um aluno de iniciação científica. Até setembro de 2024, haverá mais uma aluna de iniciação científica formada; e até dezembro de 2024 mais um aluno de mestrado. Foram desenvolvidos e publicados três trabalhos de conclusão de curso de graduação (dois em Ciência da Computação e um em Sistemas de Informação), uma dissertação de mestrado (em Ciência da Computação) e um relatório de iniciação científica. Foram desenvolvidos um modelo de programação não-linear e inteira e heurísticas (baseada em árvores geradoras) para o problema dos grafos com infecção; foram desenvolvidos dois modelos de programação inteira e uma heurística (algoritmo genético) para o problema da agenda escolar; foi desenvolvido um modelo de programação inteira de duas fases para o problema do balanceamento de grafo com corte mínimo. Foram publicados dois artigos em anais de conferência nacional (ETC e SBPO); um artigo em anais de conferência internacional (SBAC-PAD); foram realizadas sete apresentações dos resultados do projeto (ELAMAP, ETC, SIC, duas no EAMAT, SBPO, e SBAC-PAD).

## 7. EQUIPAMENTOS PREVISTOS

Sim()   Não()   Não se aplica(X)

---

## 8. MENÇÃO DO APOIO DA FAPESC EM PUBLICAÇÕES, DIVULGAÇÃO DE EVENTOS E DEMAIS MATERIAIS PRODUZIDOS:

As menções do apoio da FAPESC foram realizadas em páginas dos artigos publicados nos anais das seguintes conferências:

- Encontro de Teoria da Computação - ETC - 2022
  - <https://doi.org/10.5753/etc.2022.223209>
- Simpósio Brasileiro de Pesquisa Operacional - SBPO - 2023
  - <https://proceedings.science/sbpo-2023/trabalhos/um-estudo-sobre-arvores-geradoras-como-alternativas-conexas-para-o-controle-de-d?lang=pt-br>
- *IEEE International Symposium on Computer Architecture and High-Performance Computing* - SBAC-PAD - 2023
  - <https://doi.org/10.1109/SBAC-PAD59825.2023.00024>
- Relatório de Iniciação Científica - IC - 2023
  - <https://pibic.sistemas.ufsc.br/pibicFile/file/1073328?tipo=1>

As menções do apoio da FAPESC também foram realizadas nas apresentações dos trabalhos nas conferências:

- Encontro Latino Americano de Matemática e Aplicações - ELAMAP - 2021
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/elamap-2021.pdf>
- Encontro de Teoria da Computação - ETC - 2022
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/etc-2022.pdf>
- Simpósio de Iniciação Científica - SIC - 2023
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/sic-2023.pdf>
- Encontro Acadêmico de Matemática - EAMAT - 2023
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/eamat-2023-1.pdf>
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/eamat-2023-2.pdf>
- Simpósio Brasileiro de Pesquisa Operacional - SBPO - 2023
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/sbpo-2023.pdf>
- *IEEE International Symposium on Computer Architecture and High-Performance Computing* - SBAC-PAD 2023
  - <http://www.inf.ufsc.br/~alvaro.junio/proj-docs/sbac-pad-2023.pdf>

---

## Referências

- [Ataide 2022] Ataide, W. C. (2022). Minimização do contágio em uma topologia com Programação Matemática. Trabalho de conclusão de curso, Universidade Federal de Santa Catarina. Acessado: 02-06-2022.
- [Babaei et al. 2015] Babaei, H., Karimpour, J., and Hadidi, A. (2015). A survey of approaches for university course timetabling problem. *Computers & Industrial Engineering*, 86:43–59.
- [Barbosa et al. 2021] Barbosa, W. F., da Costa Moreira, E. B., de Araújo, J. M., Pazin-Filho, A., and Bueno, C. D. F. (2021). Modelo seir para avaliação do comportamento da pandemia de covid-19 em marabá-pa. *Revista de Medicina USP*, 100(4):322–328.
- [Bestuzheva et al. 2021] Bestuzheva, K., Besançon, M., Chen, W.-K., Chmiela, A., Donkiewicz, T., van Doornmalen, J., Eifler, L., Gaul, O., Gamrath, G., Gleixner, A., Gottwald, L., Graczyk, C., Halbig, K., Hoen, A., Hojny, C., van der Hulst, R., Koch, T., Lübbecke, M., Maher, S. J., Matter, F., Mühmer, E., Müller, B., Pfetsch, M. E., Rehfeldt, D., Schlein, S., Schlösser, F., Serrano, F., Shinano, Y., Sofranac, B., Turner, M., Vigerske, S., Wegscheider, F., Wellner, P., Weninger, D., and Witzig, J. (2021). The SCIP Optimization Suite 8.0. Technical report, Optimization Online.
- [Bondy and Murty 1976] Bondy, J. A. and Murty, U. S. R. (1976). *Graph theory with applications*, volume 290. Macmillan London.
- [Caninas et al. 2023] Caninas, P. G., Calegari, P. C., and Franco, Á. (2023). Um estudo sobre árvores geradoras como alternativas conexas para o controle de doença em um grafo de infecção. In *Simpósio Brasileiro de Pesquisa Operacional*. Galoá.
- [Carter and Laporte 1997] Carter, M. W. and Laporte, G. (1997). Recent developments in practical course timetabling. In *International conference on the practice and theory of automated timetabling*, pages 3–19. Springer.
- [Carvalho et al. 2020] Carvalho, C., Costa, J., Sales, C. L., Lopes, R., Maia de Oliveira, A. K., and Nisse, N. (2020). On the characterization of networks with multiple arc-disjoint branching flows. Technical report, UFC ; INRIA ; CNRS ; Université Côte d’Azur ; I3S ; LIRMM ; Université de Montpellier.
- [Chang et al. 2021] Chang, S., Pierson, E., Koh, P. W., Gerardin, J., Redbird, B., and Grusky, D. Leskovec, J. (2021). Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589:82–87.
- [Cooper et al. 2010] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., and Sears, R. (2010). Benchmarking cloud serving systems with ycsb. In *Proceedings of the 1st ACM symposium on Cloud computing*, pages 143–154.
- [Csimá and Gottlieb 1964] Csimá, J. and Gottlieb, C. (1964). Tests on a computer method for constructing school timetables. *Communications of the ACM*, 7(3):160–163.
- [Dasgupta et al. 2009] Dasgupta, S., Papadimitriou, C., and Vazirani, U. (2009). *Algoritmos*. Mc Graw Hill.
- [Daskalaki and Birbas 2005] Daskalaki, S. and Birbas, T. (2005). Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, 160(1):106–120.
- [Domingues 2023] Domingues, E. (2023). Planejamento de agenda escolar fazendo uso de técnicas de programação matemática e algoritmo genético. Repositório UFSC. TCC referente ao Curso de Sistemas de Informação - UFSC.

- 
- [Dunning et al. 2017] Dunning, I., Huchette, J., and Lubin, M. (2017). JuMP: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320.
- [Even et al. 1975] Even, S., Itai, A., and Shamir, A. (1975). On the complexity of time table and multi-commodity flow problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*, pages 184–193. IEEE.
- [Franco 2020] Franco, Á. J. P. (2020). Epidemic model with restricted circulation and social distancing on some network topologies. In *International Conference on Cellular Automata for Research and Industry*, pages 261–264. Springer.
- [Garey et al. 1974] Garey, M. R., Johnson, D. S., and Stockmeyer, L. (1974). Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63.
- [Gotlieb 1963] Gotlieb, C. C. (1963). The construction of class-teacher timetables. In *Proceedings IFIP Congress*, pages 73–77, Amsterdam, Ho. Nort Holland Publishing Co.
- [Goulart et al. 2023] Goulart, H. S., Trombeta, J., Franco, Á., and Mendizabal, O. M. (2023). Achieving enhanced performance combining checkpointing and dynamic state partitioning. In *2023 IEEE 35th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, pages 149–159.
- [Hethcote 1989] Hethcote, H. W. (1989). Three basic epidemiological models. In *Applied Mathematical Ecology*, pages 119–144. Springer.
- [Karypis and Kumar 1998] Karypis, G. and Kumar, V. (1998). A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392.
- [Keeling and Eames 2005] Keeling, M. J. and Eames, K. T. (2005). Networks and epidemic models. *Journal of the royal society interface*, 2(4):295–307.
- [Lions 1966] Lions, J. (1966). A conter-example for Gotlieb’s method for construction of school timetables. *Communications of the ACM*, 9(9):697.
- [Lions 1967] Lions, J. (1967). The Ontario school scheduling program. *The Computer Journal*, 10(1):14–21.
- [Liu et al. 2020] Liu, C., Wu, X., Niu, R., Wu, X., and Fan, R. (2020). A new sair model on complex networks for analysing the 2019 novel coronavirus (covid-19). *Nonlinear Dynamics*, 101(3):1777–1787.
- [Luger 2004] Luger, G. (2004). *Inteligência Artificial: Estruturas e estratégias para a solução de problemas complexos*. Bookman.
- [Manrique-Abril et al. 2020] Manrique-Abril, F. G., Agudelo-Calderon, C. A., González-Chordá, V. M., Gutiérrez-Lesmes, O., Téllez-Piñerez, C. F., and Herrera-Amaya, G. (2020). Modelo sir de la pandemia de covid-19 en colombia. *Revista de Salud Pública*, 22:123–131.
- [Nishimura and Ugander 2013] Nishimura, J. and Ugander, J. (2013). Restreaming graph partitioning: simple versatile algorithms for advanced balancing. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1106–1114.
- [Pillay 2014] Pillay, N. (2014). A survey of school timetabling research. In *Annals of Operations Research*, pages 261–293.
- [Sanders and Schulz 2013] Sanders, P. and Schulz, C. (2013). Think locally, act globally: Highly balanced graph partitioning. In *International Symposium on Experimental Algorithms*, pages 164–175. Springer.

- 
- [Tomassini 2006] Tomassini, M. (2006). Generalized automata networks. In *International Conference on Cellular Automata*, pages 14–28. Springer.
- [Tsourakakis et al. 2014] Tsourakakis, C., Gkantsidis, C., Radunovic, B., and Vojnovic, M. (2014). Fennel: Streaming graph partitioning for massive scale graphs. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 333–342.
- [Visbal and Pedraza 2020] Visbal, J. H. W. and Pedraza, M. C. C. (2020). Aproximación matemática del modelo epidemiológico sir para la comprensión de las medidas de contención contra la covid-19. *Rev. esp. salud pública*.
- [White et al. 2007] White, S. H., Del Rey, A. M., and Sánchez, G. R. (2007). Modeling epidemics using cellular automata. *Applied Mathematics and Computation*, 186(1):193–202.
- [Wodtke 2022] Wodtke, L. H. G. (2022). Otimização na gerência acadêmica utilizando técnicas de programação linear e inteira. Repositório UFSC. TCC referente ao Curso de Sistemas de Informação - UFSC.

## Anexos



**✓ Submissão recebida com sucesso | LVI Simpósio Brasileiro de Pesquisa Operacional**

---

Prezado(a) Álvaro Junio Pereira Franco,

Gostaríamos de confirmar que sua submissão intitulada **Algorithms for Super-coloring in Directed Graphs** foi enviada para ser avaliada para o LVI Simpósio Brasileiro de Pesquisa Operacional.

Todos(as) autores(as) receberão uma notificação sobre o status da submissão por email. Mas você também sempre poderá acompanhar em seu painel na área restrita.

[Ver submissão](#)

Agradecemos a participação, e aproveite para convidar colegas para ampliarmos ainda mais nossa comunidade.

Comitê Científico.  
SBPO 2024

Se o idioma deste email estiver incorreto, [clique aqui](#) para alterá-lo.