



Dissertação de Mestrado

***Framework* baseado em modelos de otimização
para o agendamento de pessoal durante eventos
pandêmicos**

Flávio Oscar Hahn
foh@ic.ufal.br

Orientadores:

Dr. Rian Gabriel Santos Pinheiro
Dr. Bruno Costa e Silva Nogueira

Maceió, Agosto de 2022

Flávio Oscar Hahn

***Framework* baseado em modelos de otimização
para o agendamento de pessoal durante eventos
pandêmicos**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Curso de Mestrado em Modelagem Computacional de Conhecimento do Instituto de Computação da Universidade Federal de Alagoas.

Orientadores:

Dr. Rian Gabriel Santos Pinheiro

Dr. Bruno Costa e Silva Nogueira

Maceió, Agosto de 2022

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Taciana Sousa dos Santos – CRB-4 – 2062

H148f Hahn, Flávio Oscar.

Framework baseado em modelos de otimização para o agendamento de pessoal durante eventos pandêmicos / Flávio Oscar Hahn. – 2022.
41 f. : il. color.

Orientador: Rian Gabriel Santos Pinheiro.

Coorientador: Bruno Costa e Silva Nogueira.

Dissertação (Mestrado em Modelagem Computacional do Conhecimento) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2022.

Bibliografia: f. 37-41.

1. Covid-19 (Pandemia). 2. *Framework* (Programa de computador). 3. Programação Linear Inteira (PLI). I. Título.

CDU: 004.42



UNIVERSIDADE FEDERAL DE ALAGOAS/UFAL
Programa de Pós-Graduação em Informática – PPGI
Instituto de Computação/UFAL
Campus A. C. Simões BR 104-Norte Km 14 BL 12 Tabuleiro do Martins
Maceió/AL - Brasil CEP: 57.072-970 | Telefone: (082) 3214-1401



Folha de Aprovação

FLÁVIO OSCAR HAHN

FRAMEWORK BASEADO EM MODELOS DE OTIMIZAÇÃO PARA O AGENDAMENTO
DE PESSOAL DURANTE EVENTOS PANDÊMICOS

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal de Alagoas e aprovada em 23 de agosto de 2022.

Banca Examinadora:

Prof. Dr. RIAN GABRIEL SANTOS PINHEIRO
UFAL – Instituto de Computação
Orientador

Prof. Dr. BRUNO COSTA E SILVA NOGUEIRA
UFAL – Instituto de Computação
Coorientador

Prof. Dr. ERICK DE ANDRADE BARBOZA
UFAL – Instituto de Computação
Examinador Interno

Prof. Dr. ANAND SUBRAMANIAN
UFPB - Universidade Federal da Paraíba
Examinador Externo

Agradecimentos

Agradeço primeiramente à Deus pelo dom da vida e por ter sua presença em todos os momentos.

À minha esposa, Natália da Silva Hahn, pelo seu amor, confiança e dedicação. Minha amiga, minha conselheira, obrigado por toda a motivação e compreensão nesta fase da minha vida e por deixá-la mais alegre e mais leve. Agradeço por acreditar e me incentivar a realizar meus objetivos;

À minha filha, Maria Luísa H. da Siva Hahn, por todo carinho, amor, gargalhadas, brincadeiras e presepadas que fazem meus dias alegres.

À minha avó, Selma F. Hahn (*in memoriam*) e minha mãe, Ieda Maria Hahn, por toda dedicação e amor por mim;

À minha família por apoiar e incentivar todos os meus projetos, vocês são responsáveis por mais essa conquista;

Aos meus sogros: José Darci B. da Silva e Clair Terezinha da Silva, por serem tão especiais para mim;

Ao meu orientador Prof. Dr. Rian Gabriel Santos Pinheiro, estendendo ao meu coorientador Prof. Dr. Bruno Costa e Silva Nogueira, pela brilhante e incansável orientação e por todo ensinamento que os dois proporcionaram durante minha jornada acadêmica. O conhecimento, amizade, paciência e a dedicação de vocês, foram essenciais para realização deste trabalho, muito obrigado!;

Aos Professores membros da banca examinadora, Prof. Dr. Erick de Andrade Barboza e Prof. Dr. Anand Subramanian pelo tempo dedicado na leitura e contribuição em função de tornar essa obra melhor;

À Universidade Federal de Alagoas e a todos os professores e funcionários por toda assistência;

Aos professores do Instituto de Computação, do Programa de Pós-Graduação em Informática, pela dedicação e conhecimentos passados nas aulas ministradas, que muito contribuíram para a minha formação acadêmica e pessoal;

Aos meus colegas e amigos de mestrado que me ajudaram em muitas dificuldades. Agradeço a todos que direta ou indiretamente contribuíram para a realização desse trabalho.

Resumo

Nos últimos anos, as empresas foram obrigadas a se adaptar às novas diretrizes e estratégias para prevenir e reduzir a transmissão da COVID-19 no local de trabalho. Um dos principais desafios nesta adaptação é gerir eficazmente a jornada de trabalho de forma a reduzir o contato social. Este trabalho apresenta um *framework* de otimização abrangente para planejar automaticamente os horários dos funcionários (equipe) durante eventos de pandemia. O *framework* usa programação linear inteira para definir um conjunto de restrições gerais que podem ser usadas para representar vários tipos de restrições de distanciamento e diferentes funções objetivo. Para usar o *framework*, uma empresa deve simplesmente instanciar um subconjunto dessas restrições com uma função objetivo (de acordo com suas prioridades). O *framework* de agendamento foi testado em três empresas reais diferentes, e os resultados mostram que nossa abordagem é capaz de melhorar o número de trabalhadores presenciais em 15%, atendendo às restrições de distanciamento social das empresas.

Palavras Chave: Pandemia, Programação Inteira, COVID-19, personal scheduling.

Abstract

In the recent years, companies were forced to adapt to the new guidelines and strategies to prevent and reduce transmission of COVID-19 in the workplace. One of the main challenges in this adaptation is to effectively manage the workday schedule in order to reduce social contact. This work presents a comprehensive optimization framework for automatically planning employee (staff) schedules during pandemic events. Our framework uses integer linear programming for defining a set of general constraints that can be used to represent several types of distancing restrictions and different objective functions. To use the framework, a company must simply instantiate a subset of these constraints with an objective function (according to its priorities). We tested our scheduling framework in three different real-life companies, and the results show that our approach is able to improve the number of in-person workers by 15% while attending the companies social distance restrictions.

Keywords: Pandemic, Integer programming, COVID-19, personal scheduling

Lista de Figuras

2.1	Processos de decisão com modelos de otimização	5
2.2	Processo de construção de modelos	6
2.3	Exemplo de problema de PL	8
2.4	Fluxo do algoritmo simplex	11
2.5	Exemplo de execução do algoritmo simplex	12
2.6	Árvore Branch-and-bound	13
5.1	SENAI: rede de contato dos funcionários	24
5.2	Horário de trabalho presencial	27
5.3	SENAC: Rede de contato dos funcionários	28
5.4	Maceió Shopping: Rede de contato dos funcionários	29
5.5	Horário de trabalho presencial	33

Lista de Tabelas

4.1	Tabela de índice	20
4.2	Parâmetros de entrada de dados	20
5.1	Cronograma de trabalho produzido manualmente pelo SENAI Alagoas	25
5.2	Dados para o problema do SENAI	26
5.3	Horário de trabalho gerado pelo nosso <i>framework</i> para o problema SENAI	26
5.4	Dados para o problema do SENAC	29
5.5	Horário de trabalho gerado pelo nosso <i>framework</i> para o problema do SENAC	30
5.6	Horário de trabalho produzido manualmente pelo Maceió Shopping	31
5.7	Dados para o problema do Maceió Shopping	32
5.8	Horário de trabalho gerado pelo nosso <i>framework</i> para o problema do Maceió Shopping	32
5.9	Horário de trabalho gerado pelo nosso <i>framework</i> para o problema do Maceió Shopping	33
5.10	Horário de trabalho gerado pelo nosso <i>framework</i> para o problema do Maceió Shopping	34
5.11	Horário de trabalho gerado pelo nosso <i>framework</i> para o problema do Maceió Shopping	35

Conteúdo

Lista de Figuras	v
Lista de Tabelas	vii
1 Introdução	1
1.1 Motivação e contextualização da pesquisa	1
1.2 Objetivos	2
1.2.1 Objetivo Geral	2
1.2.2 Objetivos Específicos	2
1.3 Estrutura dos capítulos	3
2 Fundamentação Teórica	4
2.1 Problemas de Otimização	4
2.1.1 Classes de Complexidade	5
2.1.2 Processo de Modelagem de Otimização	5
2.2 Programação Linear	7
2.2.1 Programação Linear Inteira	8
2.2.2 Método Simplex	10
2.3 Branch-and-bound	11
2.4 Branch-and-Cut	13
2.5 Pandemia	14
3 Trabalhos Relacionados	15
3.1 Problemas de otimização com aplicações em diferentes situações durante a pandemia	15
3.2 Personal Scheduling	16
3.3 Problemas de Scheduling com aplicações durante a pandemia	17
4 <i>Framework</i> MILP proposto	19
5 Estudo de casos	23
5.1 Serviço Nacional de Aprendizagem Industrial - SENAI	23
5.2 Estudo de caso Serviço Nacional de Aprendizagem Comercial - SENAC	27
5.3 Estudo de caso Maceió Shopping	28
6 Considerações Finais	36
Referências bibliográficas	37

1

Introdução

Neste capítulo, inicialmente é apresentada a motivação e contextualização da presente pesquisa. Em sequência, é demonstrado os objetivos, gerais e específicos. Por último, a estrutura do trabalho é detalhada.

1.1 Motivação e contextualização da pesquisa

A doença conhecida como COVID-19, que é causada pelo coronavírus SARS-COV2, surgiu na cidade de *Wuhan, China*, no final de 2019 e se espalhou pelo mundo no início de 2020. Após a Organização Mundial da Saúde declarar como uma pandemia (Jebril, 2020), os países começaram a fechar suas fronteiras tentando impedir a propagação do vírus. Apesar de algumas pessoas apresentarem sintomas leves e serem pouco afetadas pelo vírus, muitas outras ficam em estado grave e precisam de ventiladores ou tratamento em uma unidade de terapia intensiva (UTI).

Nesse cenário, organizações públicas e privadas de todo o mundo precisam se adequar aos decretos e orientações governamentais, que envolvem procedimentos de higiene, distanciamento de pessoas e protocolos de segurança. De acordo com Coroiu et al. (2020), medidas extensas de distanciamento social, especialmente aquelas que reduzem os contatos sociais em pelo menos 60%, têm potencial para reduzir a transmissão da doença. Assim, as organizações precisam gerenciar o horário de trabalho para reduzir a propagação do vírus e proteger a saúde de seus funcionários. Algumas abordagens nesse sentido incluem a redução da jornada de trabalho, suspensão do contrato de trabalho por tempo determinado, turnos de trabalho diversificados e novos formatos de jornada de trabalho, como home office e teletrabalho.

Em um cenário de pandemia, otimizar os turnos dos funcionários e o horário de trabalho é um problema desafiador enfrentado pelas empresas que precisam manter seu funcionamento, e ao mesmo tempo atender as orientações governamentais. Vários objetivos podem ser considerados, como minimizar o contato social, número de funcionários, custo do serviço ou horas de

home office. Como a complexidade desse problema cresce exponencialmente com o número de trabalhadores, é necessário um mecanismo automático para liberar as empresas do ônus de gerar manualmente esses horários.

Este trabalho propõe um *framework* baseado em *Mixed Integer Linear Programming MILP* para agendamento de horário de trabalho pessoal durante eventos de pandemia. Nosso *framework* considera um conjunto de parâmetros e restrições gerais que podem ser configurados para abranger muitos decretos governamentais, bem como diretrizes para reduzir a propagação do vírus. Aplicamos a proposta em três empresas brasileiras reais e mostramos como nosso *framework* pode ser facilmente adaptado aos diferentes requisitos e objetivos dessas empresas. Os cronogramas gerados pelo *framework* apresentado foram comparados com aqueles atualmente em uso nas empresas. Resultados experimentais mostram que, com baixo esforço computacional, a abordagem é capaz de gerar automaticamente escalas de horas de trabalho que melhoram aquelas produzidas manualmente nessas empresas.

Embora os experimentos realizados por [Zucchi et al. \(2021\)](#), [Guerrero and Guido \(2021\)](#) e [Mehrizi et al. \(2022\)](#) tenham mostrado resultados satisfatórios, ao conhecimento dos autores, este é o único *framework* onde existe a possibilidade de maximizar o agendamento presencial de funcionários que trabalham de forma híbrida, alternando presencial e remoto. Desta forma, as empresas podem garantir o máximo de profissionais atuando presencialmente de forma que a produção local seja garantida. Portanto, a abordagem apresentada pode ajudar empresas que precisam restringir o número de funcionários em um mesmo ambiente, adotar períodos remotos para parte da equipe, garantir um número máximo de horas de trabalho presenciais ou minimizar o número de profissionais por turno de trabalho. Além disso, a abordagem proposta também foi aplicada a três empresas diferentes, enquanto os trabalhos de referência consideram apenas uma empresa.

1.2 Objetivos

1.2.1 Objetivo Geral

Propor um *framework* para gerar agendamentos de horários de trabalho durante situações pandêmicas.

1.2.2 Objetivos Específicos

- Estudar as características específicas de agendamento de trabalhadores de cada empresa;
- Propor modelos matemáticos de Programação Linear Inteira *PLI* para o agendamento dos funcionários nas empresas durante uma pandemia;

- Implementar um *framework* geral que se adapte às especificidades de cada empresa;
- Realizar estudos de caso e comparar a solução proposta com as soluções adotadas pelas empresas;

1.3 Estrutura dos capítulos

Com o fim de alcançar esses objetivos, a estrutura desse trabalho é dividida em seis capítulos, incluindo esta introdução. No Capítulo 2 é apresentada a fundamentação teórica da pesquisa, são apresentados conceitos e técnicas importantes para o entendimento e a compreensão da temática. No Capítulo 3 é abordado todo o referencial bibliográfico do tema, são listados alguns trabalhos relacionados com problemas de otimização durante a pandemia causada pela COVID-19. No Capítulo 4 é apresentado o modelo matemático de otimização em formato de um *framework* para solução dos problemas inerentes ao agendamento de pessoas durante situações pandêmicas. No Capítulo 5, são apresentados os três estudos de caso, juntamente com resultados computacionais obtidos na formulação. O Capítulo 6 apresenta as conclusões finais e aponta trabalhos futuros.

2

Fundamentação Teórica

Visando o entendimento de técnicas empregadas nesta pesquisa, neste capítulo são apresentados os principais conceitos envolvidos em problemas de otimização. Na Seção 2.1, é explanado o que são Problemas de Otimização e alguns temas que fazem parte do contexto, como: Complexidade Computacional e Processo de Modelagem de Problemas. Na Seção 2.2, são definidas algumas técnicas de programação matemática como: Programação Linear e Programação Inteira. Em seguida, na Seção 2.3 é explanado o algoritmo de *Branch-and-Bound*, na Seção 2.4 apresentamos o algoritmo de *Branch-and-cut* e por fim, na Seção 2.5, é descrito a definição de pandemia e as principais pandemias da história, destacando a mais recente causada pela COVID-19.

2.1 Problemas de Otimização

A expressão Pesquisa Operacional (PO) foi utilizada pela primeira vez durante a Segunda Guerra Mundial, quando equipes de pesquisadores procuraram desenvolver métodos para resolver determinados problemas de operações militares (ANDRADE, 2000). Essas equipes de cientistas foram as primeiras na área de PO. Com o final da guerra, o sucesso da PO despertou interesse fora do ambiente militar. No início dos anos 1950, a PO foi introduzida nas diversas organizações dos setores comercial, industrial e governamental (Hillier and Lieberman, 2013). Sua rápida disseminação veio a seguir.

Uma das explicações para o sucesso da PO reside na objetividade das técnicas, instrumentalizadas por meio de modelos que têm a potencialidade de traduzir, de forma clara, objetiva e estruturada, problemas vivenciados no dia a dia das organizações. Outro fator que contribuiu para esse êxito foi a evolução computacional, permitindo um grande volume de processamento de cálculos de problemas complexos normalmente considerados pela PO (Longaray, 2017).

Formalmente, pesquisa operacional pode ser definida como o conjunto de técnicas que recorre ao método científico para auxiliar as pessoas a tomarem decisões. A pesquisa operacional

usa modelos como base para estruturação de decisões (Longaray, 2017). Desta forma, pesquisadores de PO devem ter domínio das características básicas da modelagem do problema de otimização. Os modelos de otimização encontram utilização em problemas nos quais as variáveis podem assumir muitos valores ou variar em intervalos muito amplos. Um modelo de otimização é composto por sua função objetivo e um conjunto de restrições. O resultado ou “solução ótima” encontrada, segundo o critério, é tomada como referência para a decisão real (ANDRADE, 2000). A Figura 2.1 ilustra o processo.

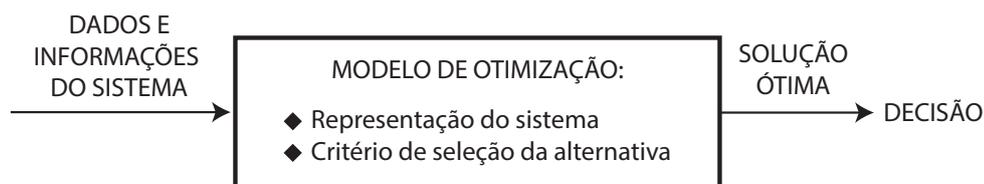


Figura 2.1: Processos de decisão com modelos de otimização (ANDRADE (2000))

2.1.1 Classes de Complexidade

Problemas de otimização podem ser classificados em conjuntos de classe de complexidade. A Complexidade Computacional é um ramo da Ciência cuja principal finalidade é estudar a eficiência de algoritmos utilizados para resolver problemas no computador. Na complexidade computacional, os problemas são separados através do desempenho de seus algoritmos em duas classes, problemas tratáveis ou não.

Alguns problemas de otimização podem ser resolvidos de forma eficiente e em tempos computacionais aceitáveis. Tais problemas são conhecidos como problemas tratáveis. Problemas da classe P, são problemas cujos algoritmos conhecidos fornecem soluções que podem ser obtidas em tempo limitado por uma função polinomial em n (tamanho da entrada) ou seja: $f(n) = O(n^k)$, em que k é uma constante. Um problema x é NP-completo se i) x pertencer a NP; e ii) todo problema y em NP puder ser reduzido a x em tempo polinomial. Já um problema que atende apenas ao item ii) é denominado NP-difícil. É importante observar que não é conhecido um método eficiente (polinomial) para solucionar um problema NP-completo (NP-difícil). Para uma leitura mais aprofundada recomenda-se a consulta de Du and Ko (2011), Garey and Johnson (1979) e Papadimitriou (1994).

2.1.2 Processo de Modelagem de Otimização

A pesquisa operacional usa modelos como aporte científico a fim de estruturar as tomadas de decisões. Desta forma, pressupõe-se que o praticante de PO domine as características básicas da modelagem do problema. Longaray (2017) define modelo como:

... a representação matemática, simbólica ou descritiva, de um conjunto de eventos

físicos, ou aspectos subjetivos, considerados importantes para determinado decisor em um contexto específico.

Goldberg and Luna (2005) sugerem um fluxograma de processos (Figura 2.2) para o estudo e construção de um modelo de otimização.

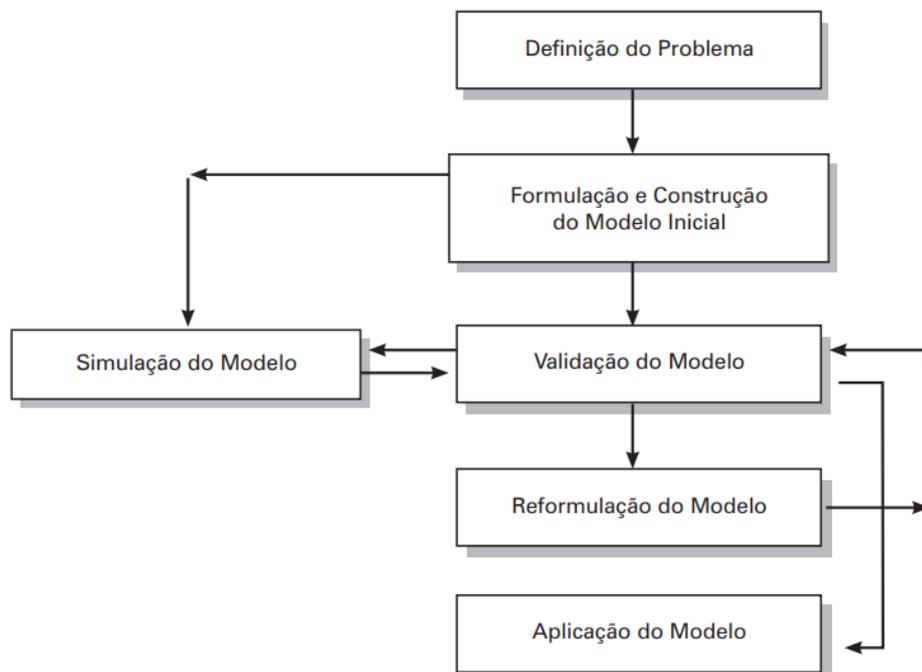


Figura 2.2: Processo de construção de modelos Goldberg and Luna (2005)

Definição do problema: primeira e mais importante fase do processo, onde a compreensão e o entendimento do problema devem ser transcritos de forma clara e objetiva, definindo os principais elementos do problema como objetivos, variáveis de decisão ou controle e níveis de detalhes.

Formulação e construção do modelo inicial: trata-se de adequar o problema definido em uma "fórmula", uma formulação de um modelo de otimização utilizando modelagem matemática, construído através das informações obtidas na definição do problema.

Validação e simulação do modelo: neste momento o modelo proposto é testado, é verificado se os resultados apresentados são satisfeitos em tempo computacional satisfatório. Esta fase está diretamente relacionada com a complexidade do problema, quanto maiores as dimensões do problema, maior é a chance de ocorrerem erros.

Reformulação do modelo: esta fase é aplicada apenas quando o modelo inicial não chegar a um resultado ótimo, ou seja, não atender aos objetivos propostos pelo modelo inicialmente. Neste caso, o modelo poderá sofrer alterações, ser reformulado e testado até atingir os objetivos propostos.

Aplicação do modelo: esta é a última fase, onde será aplicado o modelo em casos reais a fim de obter os resultados propostos inicialmente na definição do problema. Se faz necessário

documentar os resultados e métodos que foram utilizados de forma a permitir facilmente sua replicação por qualquer pessoa que venha atuar no projeto.

Para concluir, este fluxograma apresentado é apenas uma sugestão dos autores, cada especificidade de problema pode e deve ser tratado da maneira que a equipe de otimização sinta-se confortável, desta forma, o fluxograma pode ser visto apenas como um guia para facilitar a condução de estudos em problemas de PO.

2.2 Programação Linear

De acordo com [Rodrigues \(2017\)](#), Programação Linear (PL) é uma técnica de otimização aplicada em sistemas de equações e inequações lineares que representam modelos já projetados. Desta forma, a PL usa um modelo matemático para descrever o problema a ser solucionado. O adjetivo *linear* indica que todas as funções matemáticas apresentadas no modelo são necessariamente funções *lineares*, enquanto *programação*, trata-se de um sinônimo para pesquisa ([Hillier and Lieberman, 2013](#)).

Pode-se entender PL como uma técnica que procura uma solução ótima para um problema de otimização formado por um conjunto de restrições. De modo geral, quando se fala de problemas de otimização como os problemas de PL, significa que uma função precisa ser maximizada ou minimizada.

Pode-se formular um problema de PL, na sua forma geral, com objetivo de facilitar o entendimento como:

$$\text{Otimizar } z = \sum_{j=1}^n c_j x_j \quad (2.1)$$

$$\text{sujeito a: } \sum_{j=1}^n a_{i,j} x_j \geq b_i, i = 1, 2, \dots, m \quad (2.2)$$

$$x_j \geq 0, j = 1, 2, \dots, n \quad (2.3)$$

Em que x_j representa para $j = 1, 2, \dots, n$, as variáveis de decisão. A função linear a ser otimizada na equação (2.1) é denominada *Função Objetivo*, *Função Econômica* ou *Função Critério*. As equações ou inequações (2.2) são chamadas de *Restrições* do problema de PL, enquanto as Restrições (2.3) são denominadas restrição de não negatividade. O termo *otimizar*, utilizado na função (2.1), é utilizado, genericamente, para representar as possibilidades de maximizar ou minimizar a função objetivo. Onde c_j , $a_{i,j}$ e b_i são constantes do problema.

Problemas de PL com até duas variáveis podem ser solucionados com a técnica do método gráfico. Essa técnica consiste em representar em um sistema de eixos ortogonais o conjunto das possíveis soluções, isto é, o conjunto de pontos (x_1, x_2) que obedecem ao grupo de res-

trições impostas pelo sistema em estudo (SILVA et al., 2010). Como exemplos, a Figura 2.3 apresenta a representação gráfica do problema de PL a seguir:

$$\text{maximizar } 11x_1 + 12x_2 \quad (2.4)$$

$$\text{sujeito a } x_1 + 4x_2 \leq 10 \quad (2.5)$$

$$5x_1 + 2x_2 \leq 20 \quad (2.6)$$

$$x_1, x_2 \geq 0 \quad (2.7)$$

A área cinza indica o politopo, o conjunto de soluções viáveis, a chamada região de soluções viáveis onde se encontra o ponto vermelho, a solução ótima da equação. As definições de região viável e solução ótima são apresentadas a seguir.

Definição: O conjunto $C = \{x | Ax \leq b; x \geq 0\}$ denomina-se conjunto de soluções viáveis ou região viável.

Definição: Dado $x^* \in X$; x^* é denominado solução ótima do PPL se $cx^* \geq cx$, para todo $x \in X$.

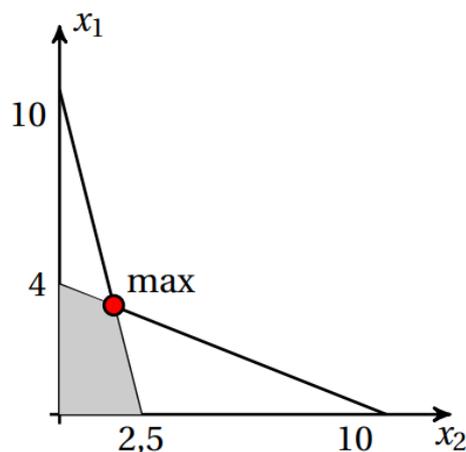


Figura 2.3: Exemplo de problema de PL

Examinando o gráfico, entende-se que x atinge o maior valor na região de soluções sobre o cruzamento das retas. Desta forma, o ponto vermelho é a solução ótimas do modelo. Este procedimento é conhecido como método gráfico para programação linear. Pode ser utilizado para solucionar problemas de PL com duas variáveis de decisão.

2.2.1 Programação Linear Inteira

O problema de Programação Linear Inteira (PLI), a princípio, é estruturado da mesma forma que os de PL com variáveis reais. O que caracteriza a PLI é a presença de ao menos uma

restrição de integralidade (Loesch and Hein, 2009). O que caracteriza restrição de integralidade é a presença de uma variável que assuma valores inteiros, como apresentado na forma geral da PLI:

$$\min \quad z = x_1 - 3x_2 - 4x_3 \quad (2.8)$$

$$\text{sujeito a} \quad 2x_1 + x_2 - x_3 \leq 4 \quad (2.9)$$

$$4x_1 - 3x_2 \leq 2 \quad (2.10)$$

$$3x_1 + 2x_2 + x_3 \leq 3 \quad (2.11)$$

$$x_1 \in \mathbb{R}^+ \quad (2.12)$$

$$x_2, x_3 \in \mathbb{Z}^+ \quad (2.13)$$

De forma similar à PL, o problema restringe x_2 e x_3 a valores inteiros não negativos, enquanto x_1 é um real qualquer não negativo. Exemplo:

$$\text{maximizar} \quad 4x_1 + 6x_2 \quad (2.14)$$

$$\text{sujeito a:} \quad 6x_1 + 6x_2 \leq 36 \quad (2.15)$$

$$2x_1 + 4x_2 \leq 12 \quad (2.16)$$

$$x_1, x_2 \in \mathbb{Z}_+ \quad (2.17)$$

Neste caso, x_1 e x_2 são variáveis de decisão, (36, 12), são vetores das restrições do problema. A restrição de domínio (2.17) indica que cada componente dos vetores x_1 e x_2 é não negativo e inteira.

Às vezes, o problema admite que uma variável não seja inteira, ou seja, um problema misto de programação linear inteira. Como, por exemplo:

$$\text{maximizar} \quad 4x_1 + 6x_2 \quad (2.18)$$

$$\text{sujeito a:} \quad 6x_1 + 6x_2 \leq 36 \quad (2.19)$$

$$2x_1 + 4x_2 \leq 12 \quad (2.20)$$

$$x_1 \geq 0 \quad (2.21)$$

$$x_2 \in \mathbb{Z}_+ \quad (2.22)$$

Neste caso, a restrição de domínio (2.22) indica que cada componente do vetor x_2 é não negativo e inteiro, enquanto (2.21) indica apenas que cada componente do vetor x_1 é não negativo

e pode assumir outros valores.

Existem outras variações de programação matemática como programação binária, quando as variáveis respostas só podem assumir valores binários (0 e 1), como problemas de alocação de funcionários em turno de trabalhos, onde 1 quer dizer presença de funcionário e 0 ausência (Virgillito, 2017). Conteúdo mais abrangente sobre PLI pode ser encontrado em Wolsey (2020) e em Hillier and Lieberman (2013).

2.2.2 Método Simplex

O algoritmo simplex é o mecanismo matemático que serve para resolver problemas de programação linear. Para um problema na forma padrão, o algoritmo simplex caminha de uma solução básica viável (vértice do politopo) para outra, reduzindo o valor da função objetivo até que o ponto ótimo seja alcançado. Para Maculan and Fampa (2006), o algoritmo pode ser definido em três partes:

- **Início:** o algoritmo prepara os dados de entrada;
- **Iteração:** o algoritmo repete diversas vezes o procedimento e faz com que a otimização do modelo seja alcançada;
- **Regra de parada:** o algoritmo avalia se a solução ótima foi obtida, ou se é impossível obtê-la.

O simplex é um algoritmo iterativo, que tem objetivo de solucionar o problema para o modelo de PL, seu fluxo é apresentado na Figura 2.4.

A ideia do método é partir de uma solução básica satisfazendo as restrições e vértices do politopo, isto é, uma solução básica primal viável, passar para outra solução básica primal viável sem que o valor da função objetivo diminua (no caso de maximização). Como o número de soluções básicas é finito, o algoritmo, sob algumas condições, convergirá.

A Figura 2.5 mostra a execução do algoritmo simplex. A partir de uma solução básica viável, o algoritmo realiza mudanças de base — troca de uma variável da base por outra fora da base — como obtêm uma nova solução básica viável com melhor função objetivo. O algoritmo segue até que a condição de otimalidade seja atendida. Note que, o método simplex pode percorrer todos os vértices da região viável. Como a quantidade máxima de vértice é $\binom{n}{m}$, o algoritmo possui uma complexidade de pior caso exponencial. Apesar que, na prática, seu desempenho é superior a algoritmos de complexidade polinomial.

A Figura 2.5 mostra a execução do algoritmo simplex. A partir de uma solução básica viável, o algoritmo realiza mudanças de base, realiza a troca de uma variável da base por outra fora da base até que a condição de otimalidade seja atendida. Para cada nova base é repetido até que uma regra de parada seja verificada (Maculan and Fampa, 2006).

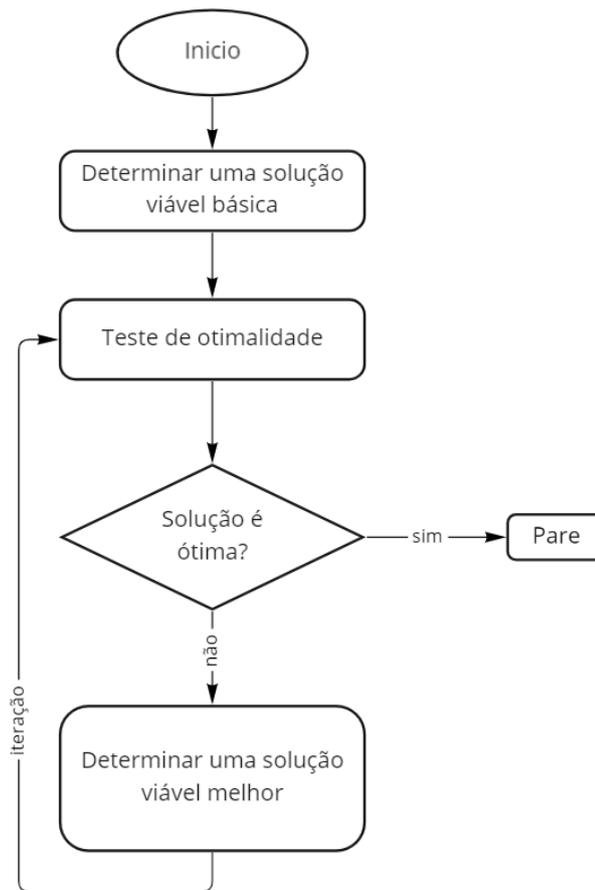


Figura 2.4: Fluxo do algoritmo simplex baseado em [Maculan and Fampa \(2006\)](#)

2.3 Branch-and-bound

Existem muitos problemas de otimização para os quais os métodos de solução são ineficientes. Problemas podem ser dessa natureza porque objetivo ou restrição não são convexos, ou algumas, ou todas as variáveis são restritas a valores discretos ([Lawler and Wood, 1966](#)).

De acordo com [dos Santos et al. \(2003\)](#), o algoritmo de Branch-and-bound é um algoritmo enumerativo, cuja estrutura de resolução baseia-se na construção de uma árvore onde os nós representam os problemas candidatos e os ramos representam as novas restrições que devem ser consideradas. Por intermédio dessa árvore, todas as soluções inteiras da região viável do problema são enumeradas de modo implícito ou explícito, o que garante que todas as soluções ótimas serão encontradas.

Para começar, é necessário aplicar o processo do método de Branch-and-bound para resolver o problema em particular.

$$\max \quad c^T x \quad (2.23)$$

$$\text{s.a.} \quad x \in \mathcal{S} \quad (2.24)$$

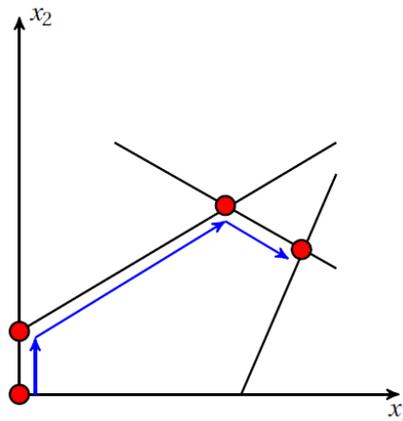


Figura 2.5: Exemplo de execução do algoritmo simplex

Alguns conjuntos de \mathcal{S} são frequentemente estimado pela relaxação linear de sua PL, a ramificação consiste em determinar um vetor α e números β^1, β^2 com $\beta^1 < \beta^2$ tal que cada $x \in \mathcal{S}$ satisfaça qualquer $\alpha^T x \leq \beta^1$ ou $\alpha^T x \geq \beta^2$. Em seguida, o método resolve os dois subproblemas:

$$\max \quad c^T x \quad (2.25)$$

$$\text{s.a.} \quad x \in \mathcal{S} \quad (2.26)$$

$$\alpha^T x \leq \beta^1 \quad (2.27)$$

e

$$\max \quad c^T x \quad (2.28)$$

$$\text{s.a.} \quad x \in \mathcal{S} \quad (2.29)$$

$$\alpha^T x \geq \beta^2 \quad (2.30)$$

separadamente. Em algum momento, os dois subproblemas podem ser divididos, ou não, em outros subproblemas, e assim por diante. Os subproblemas são os vértices, juntos eles formam uma árvore ramificada. O problema original é a raiz da árvore, sendo pai de dois subproblemas, e todos os subproblemas restantes são as folhas da árvore como apresentado na Figura 2.3.

As soluções inviáveis criadas pelo método revelam o limite do valor objetivo buscado. À medida que a solução se torne mais viável, ou seja, maior número de restrições atendidas, o limite do valor objetivo buscado se aproxima mais da solução ótima (Applegate et al., 2006).

Assim, uma interpretação importante que deve ser feita da árvore, é que as folhas de cada ramificação representam soluções potenciais ao ótimo, quando atendem às restrições do problema. Em outras palavras, a solução ótima está dentre as folhas dessa árvore criada. Caso

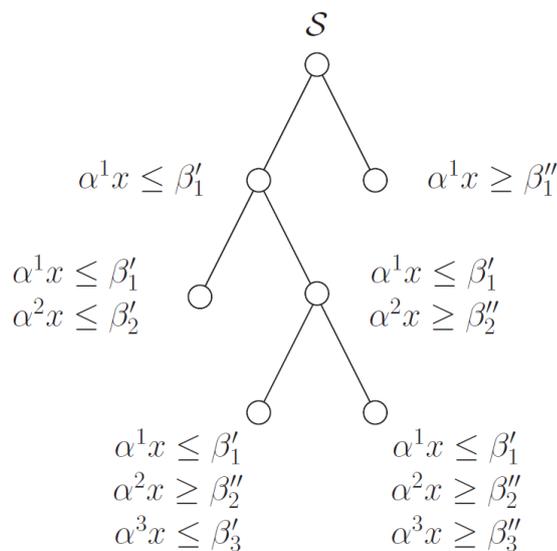


Figura 2.6: Arvore Branch-and-bound (Applegate et al., 2006)

nenhuma folha represente uma solução viável, o domínio de soluções possíveis é vazio.

As soluções inviáveis criadas pelo método revelam o limite do valor objetivo buscado. À medida que o algoritmo itera, o limite do valor objetivo buscado se aproxima mais do valor da solução ótima. Desta forma, é notório que o método Branch-and-Bound é mais robusto computacionalmente quando comparado ao método de Programação Linear. Isso vem do fato que a cada nó da árvore gerada é aplicado um método de otimização linear. Mas a grande variação que o resultado de um simples arredondamento pode provocar em uma solução, revela a necessidade da utilização de métodos como tal (Campêlo et al., 2009).

2.4 Branch-and-Cut

O algoritmo branch and cut é um método exato de programação linear inteira introduzida em Padberg and Rinaldi (1987) e Padberg and Rinaldi (1991) aplicado inicialmente para solução do problema do caixeiro viajante. Nesse algoritmo, a relaxação linear do problema é restringida adicionando um conjunto de desigualdades válidas à formulação (Wolsey and Nemhauser, 1999).

Para (González et al., 2017), o objetivo geral é encontrar famílias de desigualdades válidas, de forma que, definam a envoltória convexa do problema, ao resolver a relaxação linear em cada nó do Branch and Bound, seja possível verificar se a solução ótima encontrada viola alguma dessas desigualdades. Em caso positivo, insere-se uma nova restrição no modelo e a relaxação linear é resolvida novamente, repetindo o processo enquanto o limite inferior seja melhorado.

Desta forma, o método resolve problemas de PL sem a restrição de inteiro usando o algoritmo simplex. Quando uma solução ótima é obtida, e esta solução tem um valor não inteiro

para uma variável que supostamente é inteira, um algoritmo de plano de corte pode ser usado para encontrar outras restrições lineares satisfeitas por todos os pontos inteiros viáveis, mas violadas pela solução fracionária atual. Essas desigualdades podem ser adicionadas a PL, de modo que resolvê-las resultará em uma solução diferente “menos fracionária” (Wolsey, 2020).

Neste ponto, a parte branch-and-bound do algoritmo é iniciada. O problema é dividido em várias versões. Os novos programas lineares são então resolvidos usando o método simplex e o processo se repete. Durante o processo de ramificação e limite, soluções não integrais para relaxações PL servem como limites superiores e soluções integrais servem como limites inferiores. Um nó pode ser removido se um limite superior for inferior a um limite inferior existente. Além disso, ao resolver as relaxações PL, planos de corte adicionais podem ser gerados, que podem ser cortes globais, ou seja, válidos para todas as soluções inteiras viáveis, ou cortes locais, o que significa que eles são satisfeitos por todas as soluções que atendem às restrições laterais da ramificação e da subárvore vinculada atualmente considerada (Gutin and Punnen, 2006).

2.5 Pandemia

Segundo de Rezende (1998) a palavra “pandemia” de origem grega, inicialmente empregada por Platão, referindo-se a qualquer acontecimento capaz de alcançar toda a população, tem em seu conceito moderno, uma epidemia de grandes proporções, que se espalha a vários países e a mais de um continente, sendo pandemia, a disseminação mundial de uma nova doença. Na história, muitas pandemias ameaçaram o mundo. De acordo com Smith et al. (2009), durante o Século XX, três pandemias importantes foram observadas: a gripe espanhola em 1918, a gripe asiática em 1957 e em Hong-Kong a influenza em 1968. Já no Século XXI, de acordo com Bertolini et al. (2016), muitas pessoas foram afetadas pelo vírus SARS em 2003, pela gripe suína que impactou o mundo em 2009 e COVID-19 em 2020.

A pandemia da COVID-19 atingiu todas as populações do mundo, levando a morte mais de 6.407.556 pessoas e contaminando outras 579.092.623, notificadas até 7 de agosto de 2022. No Brasil, são cerca de 34.018.371 casos confirmados, e um total de 606.996 óbitos informados pelo MS (2022) em 7 de agosto de 2022. Tais informações evidenciam fortemente a necessidade de isolamento dos casos, segundo Aquino et al. (2020), a quarentena de contatos e medidas amplas de distanciamento social, principalmente aquelas que reduzem em pelo menos 60% os contatos sociais, têm o potencial de diminuir a transmissão da doença.

3

Trabalhos Relacionados

O cenário de pandemia da COVID resultou em um grande número de novos problemas de otimização, que surgiram com características particulares que os diferenciavam dos modelos utilizados até então. Este capítulo apresenta algumas aplicações de PO em diversos contextos e problemas relacionados à pandemia.

3.1 Problemas de otimização com aplicações em diferentes situações durante a pandemia

No contexto de problemas de roteirização de veículos (VRPs), [Tordecilla et al. \(2021\)](#) projetaram planos de roteirização diários para coletar o número máximo de itens em um tempo limitado, reduzindo assim a exposição dos motoristas ao vírus. [Chen et al. \(2020\)](#) descrevem um VRP multi-viagem multi-veículo para serviço de distribuição conjunta sem contato. [Pacheco and Laguna \(2020\)](#) consideram um VRP para a entrega urgente de protetores faciais durante a pandemia da COVID-19. Na distribuição da cadeia de suprimentos, [Perdana et al. \(2020\)](#) discutem um modelo de otimização para lidar com o impacto da pandemia de COVID-19 com base na rede de abastecimento de alimentos por meio de hubs regionais de alimentos sob incerteza. [Moosavi and Hosseini \(2021\)](#) investigaram um estudo de caso real de interrupção da cadeia de suprimentos após a COVID-19.

[Ambrogio et al. \(2022\)](#) analisam os impactos da COVID-19 na força de trabalho e fornece oportunidade de inovação digital e tecnológica do local de trabalho inspirada nos princípios da inovação digital harmônica (que coloca o bem-estar humano no centro). Para os sistemas de saúde, [Geibinger et al. \(2021\)](#) apresentam um modelo de programação de restrições que inclui a variedade de requisitos necessários para garantir o funcionamento do dia-a-dia de um hospital, bem como as restrições adicionais impostas pela situação de pandemia. Por fim, alguns trabalhos levam em conta o distanciamento social, por exemplo, [Contardo and Costa](#)

(2022) consideram o problema de maximizar o número de pessoas que uma sala de jantar pode acomodar desde que as cadeiras pertencentes às mesas diferentes estejam socialmente distantes.

3.2 Personal Scheduling

Como dito anteriormente, nas últimas décadas, o problema de escalonamento de pessoas têm sido amplamente estudado, de modo geral, esses problemas são classificados como problemas de NP-difíceis. No entanto, hoje é muito diferente daqueles introduzidos por [Dantzig \(1954\)](#) e [Edie \(1954\)](#) na década de 1950, a importância em satisfazer as necessidades dos funcionários ao criar horários de trabalho cresceu.

Um dos primeiros métodos de classificação para o problema de escalonamento de pessoas foi proposto por [Baker \(1976\)](#). De acordo com ele, podem ser distinguidos em três grupos principais: agendamento de turnos, agendamento de dias de folga e agendamento de passeios, sendo uma combinação dos dois primeiros. A seguir, será revisado a literatura mais relevante nesta área.

[Graves \(1981\)](#) apresenta uma classificação ampla para vários problemas de programação, revisando importantes desenvolvimentos teóricos para essas classes de problemas e contrastando a teoria atualmente disponível com a prática de programação da produção, além de destacar áreas problemáticas para as quais há uma discrepância significativa entre a prática e a teoria e para as quais a prática corresponde intimamente à teoria.

[Burgess and Busby \(1992\)](#) apresentam uma visão ampla da programação de turnos e inclui a disponibilidade do trabalhador, o número máximo de trabalhadores que podem começar ao mesmo tempo, a eficiência relativa de um trabalhador, o custo de fazer um cliente esperar, férias anuais fatores, o número esperado de trabalhos que um funcionário pode realizar em cada período, o número de contratados e o número máximo de trabalhos que podem ser realizados em cada período, como parâmetros em seu modelo.

[Dowland \(1998\)](#) apresenta uma solução para o problema da produção de escalas para a equipe de enfermagem de um hospital geral de grande porte, é abordado por meio da busca tabu com oscilação estratégica. Os resultados obtidos buscam garantir que um número suficiente de enfermeiros esteja de plantão em todos os momentos, considerando as preferências individuais e os pedidos de dias de folga de uma maneira em que seja vista para tratar todos os funcionários de forma justa.

A maioria das pesquisas sobre escala de pessoas tem se concentrado em escalas de turnos e escalas de folgas, visando minimizar o custo total, [Ernst et al. \(2004\)](#) apresentam uma revisão do agendamento e escala de pessoas. Sua classificação apresenta o processo de escalação em uma série de módulos: modelagem de demanda, programação de dias de folga, programação de turnos, construção da linha de trabalho, atribuição de tarefas e atribuição de pessoal. Essa

classificação é usada para discutir os principais problemas relacionados ao agendamento de pessoas em diferentes áreas de aplicação.

[Topaloglu and Ozkarahan \(2004\)](#) afirmam que as organizações usam diferentes horários de início de turno, duração de turno, janelas de intervalo diário e padrões de trabalho diários para fornecer flexibilidade. Quando o número de alternativas de flexibilidade aumenta, o desenvolvimento de horários de passeios torna-se mais complexo.

[Burke et al. \(2004\)](#) apresentam um artigo de revisão sobre a programação de escala de trabalhos de enfermeiros. Além de categorizar os papéis conforme o método de solução, restrições e medidas de desempenho, também fornecem informações sobre o período de planejamento, os dados que foram usados, o número de habilidades e sua substituibilidade. Esta abordagem é compatível com o processo de classificação.

O artigo de [Horn et al. \(2007\)](#) fornecem um exemplo de destaque da taxonomia de decisão. Para ajudar a Força de Barcos de Patrulha da Marinha Real Australiana a fazer uso eficiente de uma nova geração de barcos, eles desenvolvem procedimentos de otimização para programar as atividades dos barcos e suas tripulações. As principais tarefas de agendamento são estabelecer horários para todas as atividades e atribuir cada atividade a um barco específico e uma tripulação específica. As tarefas associadas são atribuir cada tripulação a um porto de origem e determinar a localização de cada barco quando não estiver implantado no mar.

[Ovchinnikov and Milner \(2008\)](#) descrevem uma implementação de um modelo de planilha para designar residentes médicos em um horizonte de 1 ano em radiologia na faculdade de medicina da Universidade de Vermont. Eles argumentam que as planilhas são preferíveis aos códigos independentes para problemas de pequeno porte, especialmente quando os profissionais são os usuários finais.

Segundo [Van den Bergh et al. \(2013\)](#), a programação de turnos é o tipo mais simples pelo fato de não haver turnos sobrepostos, implicando no tratamento de cada turno independentemente. No entanto, quando a demanda flutua em pequenos intervalos modifica a duração dos turnos, e uma nova modelagem com turnos sobrepostos torna-se necessária.

Uma vasta literatura que aborda o problema de escalonamento pode ser encontrada, entre outros, [Van den Bergh et al. \(2013\)](#), [Özder et al. \(2020\)](#) e [Erhard et al. \(2018\)](#).

3.3 Problemas de Scheduling com aplicações durante a pandemia

A pandemia causada pelo vírus SARS-COV2 mudou a forma de pensar a escala dos trabalhadores na gestão de pessoas, tornando-a mais flexível, por exemplo, adotando o trabalho remoto e modificando os turnos de trabalho. Ao conhecimento dos autores, apenas três autores abordam o problema do agendamento pessoal na era COVID-19 ([Zucchi et al. \(2021\)](#), [Guerriero and Guido \(2021\)](#) e [Mehrizi et al. \(2022\)](#)).

Zucchi et al. (2021) propõem uma formulação baseada em MILP para gerar o agendamento pessoal durante a pandemia causada pela COVID em um armazém de uma grande distribuidora de medicamentos na Itália. Para resolver o problema, os funcionários são separados em grupos mutuamente exclusivos, trabalhando em turnos diferentes para evitar a transmissão do vírus. O cronograma deve atender a jornada de trabalho contratual dos funcionários, visando minimizar o desvio total entre a quantidade de horas de trabalho contratuais e as horas de trabalho reais realizadas por cada trabalhador. A solução encontrada no estudo de caso supera entre 9% e 37% a solução gerada pela empresa, a depender do cenário avaliado. Testes computacionais com instâncias aleatórias de tamanho maior mostraram soluções de alta qualidade e tempo de CPU aceitável.

Outro trabalho, apresentado por Guerriero and Guido (2021), propõe um modelo de otimização para resolver problemas de agendamento de equipe flexível, levando em consideração requisitos de demanda, responsabilidades pessoais e familiares dos funcionários e medidas anti-Covid-19 ao mesmo tempo. O modelo foi testado em dados reais fornecidos pelo Departamento de Engenharia Mecânica, Energia e Gestão da Universidade da Calábria, Itália. Experimentos computacionais apresentam bom desempenho para garantir o trabalho e a continuidade das atividades, os agendamentos encontrados são preferidos em relação aos construídos à mão porque são obtidos em poucos segundos, as solicitações modificadas podem ser tratadas rapidamente e até mesmo uma fase de negociação é menos demorada.

Mehrizi et al. (2022) propuseram uma estrutura de otimização matemática de dois estágios para agendamento pessoal durante uma pandemia e demonstrou uma aplicação de *framework* para um estudo de caso do departamento de radioterapia. Seu *framework* considera a probabilidade de transmissão de infecção entre clientes e funcionários em potencial, o efeito da interação entre funcionários que trabalham em contato próximo e as características da doença, como o período de incubação durante o qual clientes e funcionários podem ser assintomáticos, mas infecciosos. O modelo de otimização determina padrões de trabalho ideais que minimizam o número esperado de ausências de funcionários devido a interações. No entanto, o modelo não considera as horas trabalhadas por cada funcionário.

Embora os experimentos realizados por Zucchi et al. (2021), Guerriero and Guido (2021) e Mehrizi et al. (2022) tenham mostrado resultados satisfatórios, ao conhecimento dos autores, esta é a única solução onde existe a possibilidade de maximizar o agendamento presencial de funcionários que trabalham de forma híbrida, alternando presencial e remoto. Portanto, nossa abordagem pode ajudar empresas que precisam restringir o número de funcionários em um mesmo ambiente, adotar períodos remotos para parte da equipe, garantir um número máximo de horas de trabalho presenciais ou minimizar o número de profissionais por turno de trabalho. Além disso, ressaltamos que a abordagem proposta também foi aplicada a três empresas diferentes, enquanto os trabalhos de referência consideram apenas uma empresa.



Framework MILP proposto

Neste capítulo, é apresentado uma estrutura baseada em um modelo matemático de otimização para gerar agendamento pessoal em situações de pandemia. Nosso objetivo é formular um conjunto de restrições que possam ser aplicadas em diferentes empresas de acordo com seus propósitos. Na formulação proposta, uma empresa pode executar nosso *framework*: (i) selecionando um subconjunto dessas restrições e (ii) estabelecendo os parâmetros de entrada.

Sejam E , P e T conjuntos representando empregados, períodos de trabalho e tempo, respectivamente. O conjunto E pode ser particionado em $E = F \cup E_r$, onde F representa o conjunto de funcionários que podem trabalhar presencialmente e E_r é o conjunto de funcionários do grupo de risco. Os períodos de trabalho, definidos em P , podem variar de acordo com a empresa, por exemplo, pode ser presencial, remoto, etc. O conjunto T , por outro lado, pode ser definido em termos de dias, semanas ou turnos. Os funcionários são divididos em setores, e o conjunto de setores é definido por S .

As Tabelas 4.1 e 4.2 descrevem os índices das variáveis de decisão e os parâmetros de nossa formulação, respectivamente. As variáveis de decisão são definidas da seguinte forma:

- $x_{p,e}^t$: variável binária que é 1 se o funcionário e trabalha no período p no tempo t , e 0 caso contrário;
- $y_{p,e}^t$: variável inteira que representa o número de horas trabalhadas pessoalmente pelo funcionário no período p no tempo t ;
- $maxWORKS$: representa o número máximo de funcionários permitidos para cada turno de trabalho.

A formulação matemática proposta é definida da seguinte forma:

Tabela 4.1: Tabela de índice

Índice	Descrição
p	Home Office ou índice de períodos presenciais
e	Índice de funcionários
t (também definido como w ou d ou s)	índice de tempo. Pode ser definido de acordo com a preferência da instituição, assumindo os valores de semana, dias ou turnos

Tabela 4.2: Parâmetros de entrada de dados

Parametros	Descrição
T (também definido como W ou D)	Horário definido
E	Conjunto de funcionários
E_r	Representa os funcionários que pertencem ao grupo de risco COVID-19;
P	Conjunto de períodos
S	Conjunto de setores
$max_{authorized}$	Número máximo de funcionários do mesmo setor que podem trabalhar presencialmente sem ultrapassar a capacidade máxima do ambiente
$min_{authorized}$	Número mínimo de funcionários que cada projeto precisa de funcionários trabalhando pessoalmente no mesmo período
$employee_{max}$	Número máximo de funcionários permitidos no mesmo ambiente trabalhando pessoalmente
dif	Diferença máxima entre a carga de trabalho mínima e máxima
$hours_{day}$	Carga de trabalho de um dia útil
$hours_{week}$	Carga de trabalho de uma semana de trabalho
$hours_{month}$	Carga de trabalho de um mês de trabalho
min_w	Carga de trabalho mínima de cada funcionário
max_w	Carga de trabalho máxima de cada funcionário

$$\min/\max \quad f(x, y, maxWORKS) \quad (4.1)$$

$$\text{s.a} \quad \sum_{p \in P} x_{p,e}^t = 1 \quad \forall t \in T, \forall e \in E \quad (4.2)$$

$$\sum_{p \in P} \sum_{t \in T} y_{p,e}^t \geq hours_{month} \quad \forall e \in E \quad (4.3)$$

$$\sum_{e \in S} x_{1,e}^t \leq max_{authorized} \quad \forall S \in S, \forall t \in T \quad (4.4)$$

$$\sum_{e \in S} x_{1,e}^t \geq \min_{authorized} \quad \forall S \in \mathcal{S}, \forall t \in T \quad (4.5)$$

$$x_{1,e}^t = 0 \quad \forall e \in E_r, \forall t \in T \quad (4.6)$$

$$y_{p,e}^t \leq \text{hours}_{day} x_{p,e}^t \quad \forall e \in E, \forall t \in T, \forall p \in P \quad (4.7)$$

$$x_{p,e}^t \leq y_{p,e}^t \quad \forall e \in E, \forall t \in T, \forall p \in P \quad (4.8)$$

$$\sum_{t \in T} y_{1,e}^t \leq \text{hours}_{month} \quad \forall e \in E \setminus E_r, \forall e \in E \quad (4.9)$$

$$\sum_{t \in T} \sum_{p \in P} y_{p,e}^t \geq \text{hours}_{week} \quad \forall e \in E \quad (4.10)$$

$$\sum_{e \in E} x_{1,e}^t \leq \text{employee}_{max} \quad \forall t \in T \quad (4.11)$$

$$\sum_{p \in P} \sum_{t \in T} y_{p,e}^t \geq \text{min}_W \quad \forall e \in E \setminus E_r, \forall e \in E \quad (4.12)$$

$$\sum_{p \in P} \sum_{t \in T} y_{p,e}^t \leq \text{max}_W \quad \forall e \in E \setminus E_r, \forall e \in E \quad (4.13)$$

$$\text{max}_W - \text{min}_W \leq \text{dif} \quad (4.14)$$

$$\sum_{e \in S} x_{p,e}^t \leq \text{max}_{WORKS} \quad \forall p \in P, \forall S \in \mathcal{S}, \forall t \in T \quad (4.15)$$

$$x_{p,e}^t \geq 0 \quad \forall e \in E, \forall t \in T, \forall p \in P \quad (4.16)$$

$$y_{p,e}^t \in \mathbb{Z}_+ \quad \forall e \in E, \forall t \in T, \forall p \in P \quad (4.17)$$

$$\text{max}_{WORKS} \in \mathbb{Z}_+ \quad (4.18)$$

A Função Objetivo (4.1) busca otimizar o critério escolhido pela empresa. Por exemplo, se a empresa estiver interessada em maximizar o número de horas trabalhadas pessoalmente, pode ser formulado como:

$$\max \sum_{t \in T} \sum_{e \in E} y_{1,e}^t \quad (4.19)$$

Observa-se que, na equação (4.19), assume-se que um período $p = 1$ significa um período de trabalho presencial e $p = 0$ um período de trabalho remoto. Como outro exemplo de função objetivo, se a empresa estiver interessada em minimizar o número de funcionários $e \in E$ no mesmo período $p \in P$, isso pode ser expresso como:

$$\min \text{max}_{WORKS} \quad (4.20)$$

As Restrições (4.2) garantem que cada trabalhador $e \in E$ seja designado durante o tempo $t \in T$ para o período $p \in P$, seja presencial onde $p = 1$ ou remoto onde $p = 0$. As restrições (4.3) garantem que cada trabalhador $e \in E$ cumpra uma carga de trabalho $hours_{month}$ em um período $p \in P$ no tempo $t \in T$. As Restrições (4.4) e (4.5) garantem que durante o período presencial onde $p = 1$, o número de trabalhadores $e \in E$ trabalhando no setor $S \in S$ está entre o número máximo ($max_{authorized}$) e mínimo ($min_{authorized}$) de trabalhadores em cada setor. As Restrições (4.6) garantem que os trabalhadores do grupo de risco $e \in E_r$ não trabalharão no período presencial $p \in P$ onde $p = 1$. As Restrições (4.7) garantem que o trabalhador $e \in E$ que trabalha no período $p \in P$, complete um total de horas de trabalho. As Restrições (4.8) garantem que o trabalhador $e \in E$ que trabalha no período $p \in P$ e no tempo $t \in T$, não tenha carga horária igual a 0. As Restrições (4.9) garantem que cada trabalhador $e \in E$ fora do grupo de risco tenha uma carga horária mínima no período presencial onde $p = 1$. As Restrições (4.10) garantem que todos os funcionários tenham sua carga de trabalho total menor ou igual à carga de trabalho do período. As Restrições (4.11) garantem um máximo de trabalhadores $e \in E$ por período $p \in P$ de tempo $t \in T$. As Restrições (4.12) e (4.13) garantem que todo funcionário $e \in E$ fora do grupo de risco tenha sua carga horária entre a máxima max_W e a mínima min_W . A Restrição (4.14) garantem que a diferença entre a carga de trabalho mais alta max_W e a mais baixa min_W deve ser menor ou igual a diferença dif . As Restrições (4.15) garantem que durante o período presencial em que $p \in P$, o número de trabalhadores $e \in E$ trabalhando no setor $S \in S$ seja menor que ou igual ao máximo permitido. Finalmente, as Restrições (4.16 – 4.18) definem os domínios das variáveis.

5

Estudo de casos

Este capítulo relata os experimentos computacionais que foram conduzidos para avaliar o *framework* proposto. Nossa plataforma experimental é composta por um Intel i7-1165G7 @ 2.80GHz com 16GB de RAM, apenas um núcleo foi utilizado para executar os experimentos. O *framework* proposto foi desenvolvido em linguagem Python utilizando a API PuLP e o CBC¹ da COIN-OR como solucionador de programação linear inteira mista. Consideramos três empresas brasileiras do mundo real com diferentes restrições e objetivos como estudos de caso: SENAI², SENAC³ e Maceió Shopping⁴. Comparamos os cronogramas gerados automaticamente pelo nosso *framework* com os produzidos manualmente por essas três empresas.

5.1 Serviço Nacional de Aprendizagem Industrial - SENAI

O Serviço Nacional de Aprendizagem Industrial (SENAI) é um dos mais importantes centros brasileiros de geração e disseminação de conhecimento aplicado para o desenvolvimento industrial. Este estudo de caso tem como foco o departamento de Soluções Digitais, que faz parte do HUB de Inovação do Departamento Regional do SENAI Alagoas. Este departamento é responsável pelo desenvolvimento de software para o segmento educacional e conta com uma equipe de 18 profissionais de diferentes áreas e formações, incluindo analistas de sistemas, designers, desenvolvedores, entre outros. Divididos em equipes de trabalho, em um cenário ideal, todos os trabalhadores compartilham o mesmo ambiente durante a jornada de trabalho. De segunda a sexta-feira, a jornada de trabalho é composta por oito horas diárias e uma hora de intervalo, totalizando quarenta horas semanais de trabalho. Assim, o compartilhamento de recursos como computadores, mesas, cadeiras, além de material de escritório, torna-se comum entre as equipes, como mostra a Figura 5.1.

¹COIN-OR Branch-and-Cut solver: <https://github.com/coin-or/Cbc>

²SENAI Alagoas: <https://al.senai.br/>

³SENAC Alagoas: <https://www.al.senac.br/>

⁴Maceió Shopping: <https://www.maceioshopping.com/>

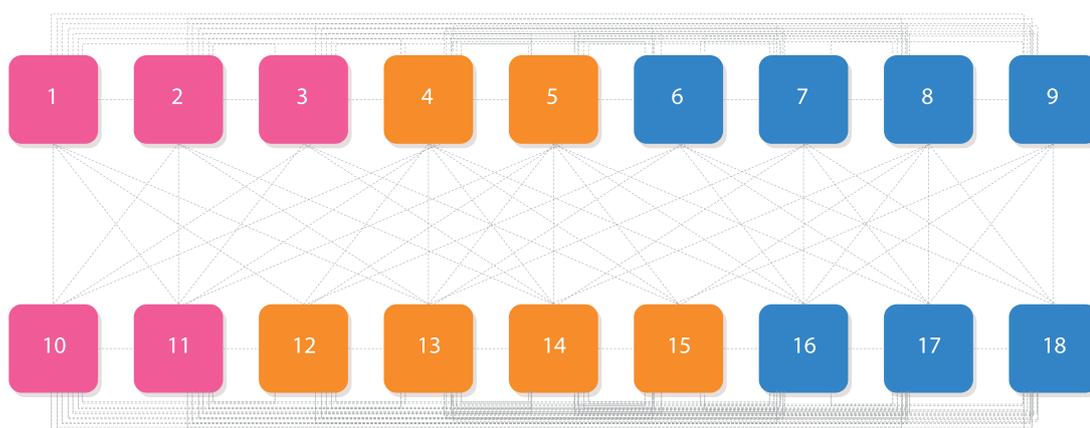


Figura 5.1: SENAI: rede de contato dos funcionários

Figura 5.1 representa a rede de relacionamentos entre os funcionários, de forma que os vértices representam os funcionários, as cores dos vértices suas equipes correspondentes. As arestas indicam que dois funcionários trabalharam pelo menos uma vez em simultâneo, no mesmo ambiente, demonstrando a integração e compartilhamento de recursos entre as equipes e seus projetos. Vale ressaltar que algumas equipes possuem membros em comum, como designers e analistas que fazem parte de mais de um projeto.

Com o surgimento da pandemia causada pela COVID-19, o departamento teve que se adaptar às normas de distanciamento. Para permitir a continuidade das demandas de cada equipe, o SENAI se interessou em maximizar a jornada de trabalho presencial, atendendo às seguintes restrições:

- O ambiente de trabalho tem capacidade máxima de 10 funcionários no local (Restrições (4.2), (4.3), (4.4) e (4.5));
- Deve haver sempre pelo menos um trabalhador de cada projeto trabalhando pessoalmente (Restrições (4.6) e (4.7));
- Cada trabalhador deve trabalhar pelo menos um dia pessoalmente. Sempre que alocado presencialmente, deve cumprir a carga horária total da jornada de trabalho (Restrições (4.9), (4.10) e (4.11));
- Funcionários pertencentes ao grupo de risco devem ser alocados 100% remotamente. No entanto, durante este estudo de caso, nenhum funcionário estava neste grupo (Restrição (4.6));

A Tabela 5.1 mostra a solução adotada pela instituição durante a Pandemia, de forma que uma célula marcada representa o dia em que um funcionário irá trabalhar presencialmente. Essa programação é periódica, ou seja, se repete a cada duas semanas. A solução gerada pela empresa atinge um total de 1360h das 1600h possíveis de horas de trabalho presencial.

Ela divide os 18 trabalhadores em duas equipes, de modo que a cada semana uma equipe trabalha presencialmente e a outra trabalha remotamente.

Tabela 5.1: Cronograma de trabalho produzido manualmente pelo SENAI Alagoas

Empregados	Semana 0	Semana 1
E0		
E1		
E2		
E3		
E4		
E5		
E6		
E7		
E8		
E9		
E10		
E11		
E12		
E13		
E14		
E15		
E16		
E17		

O problema SENAI foi modelado em nosso *framework* da seguinte forma:

$$\max \sum_{w \in W} \sum_{e \in E} y_{1,e}^w \tag{5.1}$$

$$\text{s.a } (4.2)(4.3)(4.4)(4.5)(4.6)(4.7)(4.8)(4.9)(4.10)(4.11)(4.16)(4.17). \tag{5.2}$$

A Tabela 5.2 mostra as configurações de dados para este estudo de caso. Observe que, para este problema, o parâmetro de tempo T em (4.2) foi definido como W , que se refere aos números de semanas.

A Tabela 5.3 mostra os resultados obtidos através do nosso *framework*. Verifica-se que a solução atingiu um total de 1600h das 1600h possíveis de trabalho presencial. Este resultado é cerca de 15% melhor em relação à solução adotada pela empresa. O *framework* obteve essa solução ótima em 0,064 segundo. A Figura 5.2 mostra, para cada trabalhador, o número de horas que ele trabalha presencialmente em um mês de acordo com nossa solução e a solução adotada pela empresa.

Tabela 5.2: Dados para o problema do SENAI

Dados	Valor
$ W $	4
$ E $	18
$ E_R $	0
$ P $	2
$ S $	3
$max_{authorized}$	10
$min_{authorized}$	1
$employee_{max}$	10
dif	-
$hours_{week}$	40
$hours_{month}$	160
$hours_{day}$	8
min_W	-
max_W	-

Tabela 5.3: Horário de trabalho gerado pelo nosso *framework* para o problema SENAI

Empregado	Semana 0	Semana 1	Semana 2	Semana 3
E0	█		█	█
E1		█	█	█
E2	█		█	█
E3	█			
E4		█		█
E5	█			
E6		█	█	█
E7		█	█	
E8		█		█
E9			█	
E10	█	█	█	█
E11	█			█
E12	█			
E13	█		█	
E14		█		
E15	█			█
E16		█		█
E17	█		█	

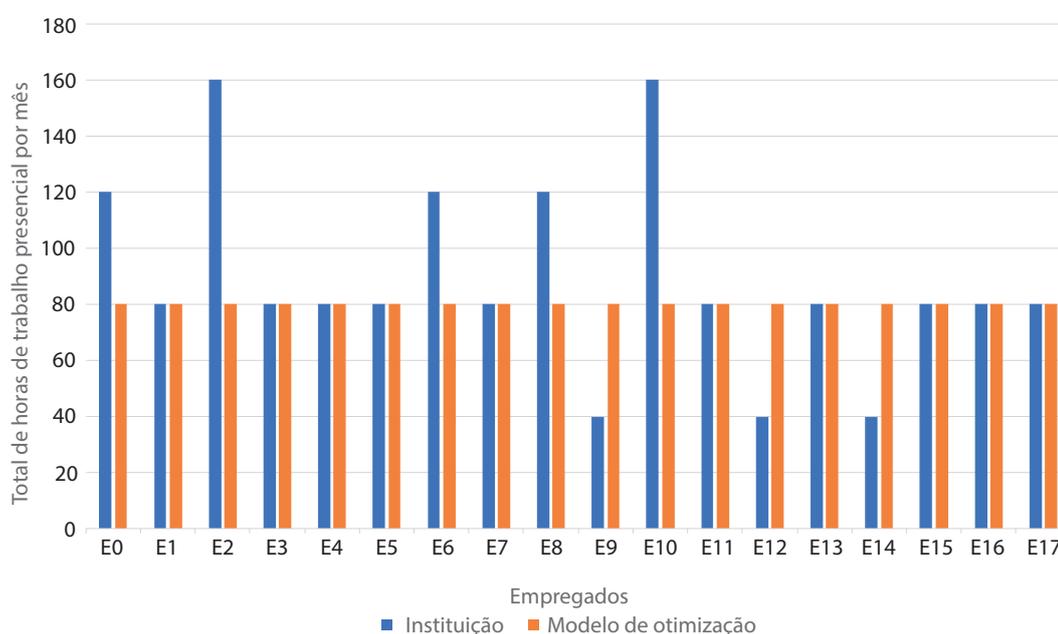


Figura 5.2: Horário de trabalho presencial (nosso *framework* vs solução adotada pelo SENAI)

5.2 Estudo de caso Serviço Nacional de Aprendizagem Comercial - SENAC

Criado em 1946, o SENAC é uma instituição de direito privado subordinada à Confederação Nacional do Comércio (CNC)⁵. É um dos principais agentes de educação profissional voltada para o comércio de bens, serviços e turismo no Brasil. Este estudo de caso considerou o setor administrativo do SENAC Alagoas, que possui um ambiente com 14 funcionários conforme Figura 5.3. A jornada de trabalho é normalmente dividida em dois turnos de 8 horas, distribuídos entre as 8:00 e as 22:00. Um terceiro turno foi adotado durante a pandemia, modificando o horário de trabalho e realocando alguns dos funcionários. Além disso, a empresa optou por não permitir o trabalho remoto para funcionários que não pertencem ao grupo de risco. As restrições adotadas pela empresa foram:

- Cada trabalhador deve ter um turno de trabalho: manhã, tarde ou noite (Restrições (4.2), (4.3) e (4.9));
- Cada trabalhador deve cumprir a carga horária total presencialmente com no máximo 5 trabalhadores por turno, e cada turno deve conter no mínimo 4 trabalhadores (Restrições (4.5), (4.7), (4.8) e (4.15));

⁵Confederação Nacional do Comércio: <https://www.portaldocomercio.org.br/>

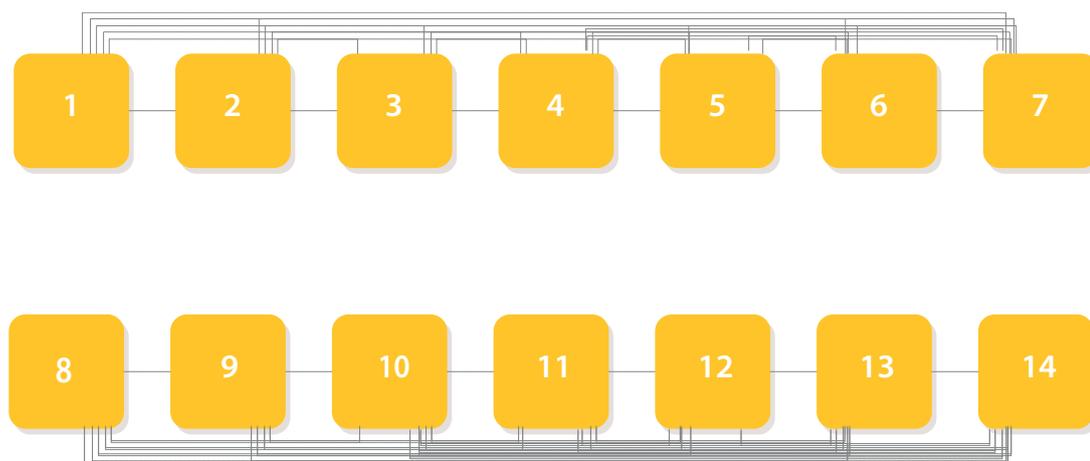


Figura 5.3: SENAC: Rede de contato dos funcionários

- Funcionários pertencentes ao grupo de risco devem ser alocados 100% remotamente. Durante este estudo de caso, nenhum funcionário estava neste grupo (Restrição (4.6));

O objetivo do SENAC foi minimizar o número de funcionários trabalhando no mesmo ambiente para reduzir o contato entre os funcionários. Modelamos este estudo da seguinte forma:

$$\min \max_{WORKS} \quad (5.3)$$

$$\text{s.a. } (4.2)(4.3)(4.5)(4.6)(4.7)(4.8)(4.9)(4.15)(4.17)(4.18). \quad (5.4)$$

Tabela 5.4 e Tabela 5.5 mostram, respectivamente, os dados do modelo e os resultados obtidos através do *framework*. Esse resultado é satisfatório, tendo em vista que reduziu o número de trabalhadores no mesmo ambiente em aproximadamente 37,5% em comparação com a agenda adotada em um cenário sem pandemia. Apesar de o *framework* não ser o responsável direto pela melhoria, pois o aumento de um turno tornou isso possível, o *framework* gerou essa solução em 0,22 segundo, auxiliando no agendamento de forma considerável.

5.3 Estudo de caso Maceió Shopping

Com trinta e dois anos de atuação e localizado em uma das áreas mais importantes da capital alagoana, o Maceió Shopping ocupa posição de destaque no varejo local e regional. Possui mais de 300 pontos de venda, oferecendo diversos produtos e serviços. A administração conta com um total de 30 profissionais responsáveis por sua gestão, dentre os quais 20 atuam no mesmo ambiente. A jornada de trabalho é de 8 horas diárias de segunda a sexta-feira, totalizando 40 horas semanais. O contato entre os membros tornou-se constante e inevitável, como pode ser visto na Figura 5.4.

Tabela 5.4: Dados para o problema do SENAC

Dados	Valor
$ W $	1
$ E $	14
$ E_R $	0
$ P $	3
$ S $	1
$max_{authorized}$	-
$min_{authorized}$	4
$employee_{max}$	5
dif	-
$hours_{week}$	33
$hours_{month}$	132
$hours_{day}$	6.6
min_W	-
max_W	-

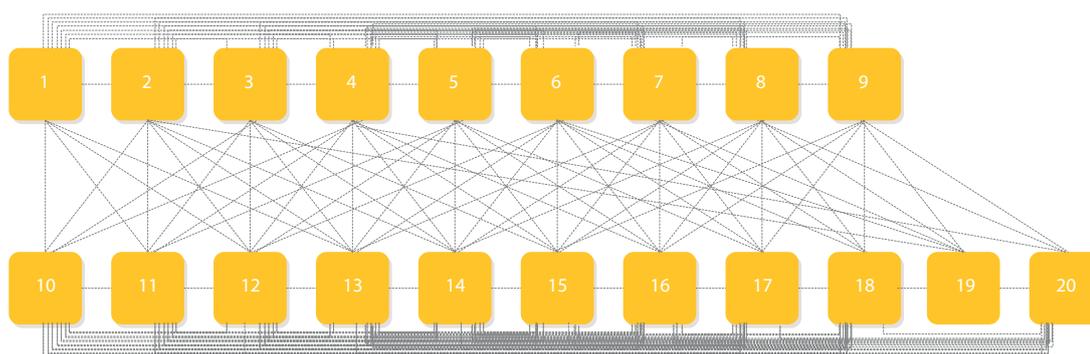


Figura 5.4: Maceió Shopping: Rede de contato dos funcionários

Durante a pandemia, foram adotadas as seguintes restrições:

- Cada funcionário deve ter um período de trabalho presencial ou remoto (Restrições (4.2), (4.3) e (4.4));
- O ambiente de trabalho tem capacidade máxima de 10 funcionários trabalhando presencialmente (Restrições (4.5), (4.6), (4.10) e (4.11));
- A carga horária definida como presencial deve ser efetivamente cumprida compulsoriamente pelo trabalhador (Restrição (4.7))
- Todo trabalhador deve trabalhar presencialmente com exceção dos trabalhadores que pertençam ao grupo de risco, com idade igual ou superior a 60 anos, ou com condições clínicas de risco para o desenvolvimento de complicações, que devem exercer integralmente suas funções remotamente (Constrangimentos (4.8) e (4.9));

Tabela 5.5: Horário de trabalho gerado pelo nosso *framework* para o problema do SENAC

Empregado	Matutino	Vespertino	Noturno
E0	■		
E1	■		
E2	■		
E3			■
E4			■
E5	■		
E6		■	
E7			■
E8			■
E9		■	
E10		■	
E11		■	
E12			■
E13	■		

- A carga horária mensal dos trabalhadores deve estar entre 85 e 160 horas presenciais (Restrições (4.12) e (4.13))
- A diferença entre a maior e a menor carga de trabalho mensal deve ser menor que 30 horas atrás (Restrição (4.14));
- Os 3 funcionários pertencentes ao grupo de risco devem ser alocados 100% remotamente (Restrição (4.6));

A Tabela 5.6 mostra a solução adotada pela instituição durante a pandemia, que se repete a cada duas semanas. Nesta solução, cada pessoa é alocada presencialmente durante dois dias da semana, com uma sexta-feira alternada entre as equipes de trabalho. Outro ponto importante é que a empresa gerou a escala de trabalho sem considerar os trabalhadores do grupo de risco, o que viola uma de suas restrições. A solução da empresa tem um total de 1120 horas presenciais.

Dado que a empresa estava interessada em maximizar o horário de trabalho presencial, este estudo de caso foi modelado da seguinte forma:

$$\max \sum_{d \in D} \sum_{e \in E} y_{1,e}^d \quad (5.5)$$

$$\text{s.a } (4.2)(4.3)(4.6)(4.7)(4.8)(4.9)(4.10)(4.11)(4.12)(4.13)(4.14)(4.16)(4.17). \quad (5.6)$$

No cenário descrito, o parâmetro de tempo T do *framework* é considerado o intervalo de

Tabela 5.6: Horário de trabalho produzido manualmente pelo Maceió Shopping

Empregado	Segunda	Terça	Quarta	Quinta	Sexta
E0	Teal		Teal		Orange
E1	Teal		Teal		Orange
E2	Teal		Teal		Orange
E3	Teal		Teal		Orange
E4	Teal		Teal		Orange
E5	Teal		Teal		Orange
E6	Teal		Teal		Orange
E7	Teal		Teal		Orange
E8	Teal		Teal		Orange
E9	Teal		Teal		Orange
E10		Teal		Teal	Yellow
E11		Teal		Teal	Yellow
E12		Teal		Teal	Yellow
E13		Teal		Teal	Yellow
E14		Teal		Teal	Yellow
E15		Teal		Teal	Yellow
E16		Teal		Teal	Yellow
E17		Teal		Teal	Yellow
E18		Teal		Teal	Yellow
E19		Teal		Teal	Yellow

um dia e é definido como D (número de dias úteis). A Tabela 5.7 mostra as configurações dos dados.

A Figura 5.5 apresenta os resultados obtidos através do modelo matemático apresentado, totalizando 1600 horas presenciais das 1600 horas possíveis. As Tabelas 5.8, 5.9, 5.10 e 5.11 mostram as horas de trabalho semanais geradas como resultado do estudo de caso, distribuídas ao longo de 4 semanas. Pode-se observar que a solução obtida é cerca de 30% melhor em relação à solução gerada pela empresa. Outro ponto a ser observado é que essa solução ótima foi encontrada em apenas 0,47 segundos e, diferentemente da solução adotada pela empresa, ela não viola as restrições do grupo de risco.

Tabela 5.7: Dados para o problema do Maceió Shopping

Dados	Valor
$ D $	20
$ E $	20
$ E_R $	3
$ P $	2
$ S $	-
$max_{authorized}$	10
$min_{authorized}$	8
$employee_{max}$	10
dif	30
$hours_{weekh}$	44
$hours_{month}$	176
$hours_{day}$	8
min_W	85
max_W	160

Tabela 5.8: Horário de trabalho gerado pelo nosso *framework* para o problema do Maceió Shopping: Semana 1

Empregado	Segunda	Terça	Quarta	Quinta	Sexta
E1		■		■	
E2	■				■
E3	■		■	■	■
E4	■		■	■	
E5	■		■	■	■
E6		■	■	■	■
E7				■	■
E8		■		■	
E9					
E10					
E11		■			■
E12	■		■		■
E13	■	■			■
E14	■		■		
E15	■		■	■	
E16					
E17	■		■	■	■
E18		■	■		■
E19	■	■	■		■
E20				■	

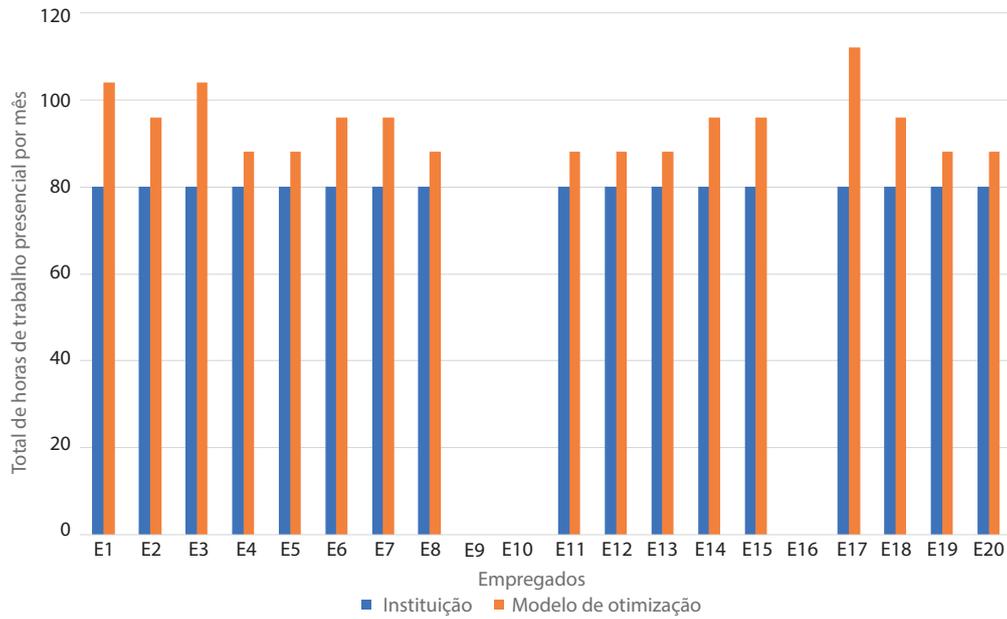


Figura 5.5: Horário de trabalho presencial (nosso *framework* vs solução adotada pelo Maceió Shopping)

Tabela 5.9: Horário de trabalho gerado pelo nosso *framework* para o problema do Maceió Shopping: Semana 2

Empregado	Segunda	Terça	Quarta	Quinta	Sexta
E1	■	■		■	
E2	■	■	■		
E3	■			■	
E4		■	■	■	
E5				■	■
E6		■	■	■	
E7	■	■	■	■	■
E8	■	■	■		■
E9					
E10					
E11		■	■	■	■
E12	■	■	■		■
E13					
E14	■		■	■	■
E15	■			■	■
E16					
E17	■		■	■	■
E18			■	■	■
E19	■	■	■		
E20	■	■	■		■

Tabela 5.10: Horário de trabalho gerado pelo nosso *framework* para o problema do Maceió Shopping: Semana 3

Empregado	Segunda	Terça	Quarta	Quinta	Sexta
E1	█	█	█	█	█
E2	█	█	█	█	█
E3	█	█	█	█	█
E4	█	█	█	█	█
E5	█	█	█	█	█
E6	█	█	█	█	█
E7	█	█	█	█	█
E8	█	█	█	█	█
E9	█	█	█	█	█
E10	█	█	█	█	█
E11	█	█	█	█	█
E12	█	█	█	█	█
E13	█	█	█	█	█
E14	█	█	█	█	█
E15	█	█	█	█	█
E16	█	█	█	█	█
E17	█	█	█	█	█
E18	█	█	█	█	█
E19	█	█	█	█	█
E20	█	█	█	█	█

Tabela 5.11: Horário de trabalho gerado pelo nosso *framework* para o problema do Maceió Shopping: Semana 4

Empregado	Segunda	Terça	Quarta	Quinta	Sexta
E1	█	█		█	
E2	█		█		
E3	█		█	█	█
E4		█			
E5		█	█	█	
E6	█	█	█		█
E7			█	█	
E8		█		█	█
E9					
E10					
E11			█		█
E12	█	█			
E13	█	█	█		█
E14	█	█	█	█	
E15				█	█
E16					
E17	█	█		█	█
E18	█		█		
E19			█	█	█
E20	█	█		█	

6

Considerações Finais

Os surtos de pandemia levantam muitos desafios que podem comprometer a continuidade dos negócios. Recentemente, o coronavírus trouxe alguns problemas para o horário de trabalho pessoal. Esta dissertação apresentou um *framework* baseado em programação matemática para um problema de agendamento pessoal considerando eventos pandêmicos. O *framework* considera um conjunto de parâmetros gerais e restrições MILP que selecionamos para atender às diretrizes para reduzir a propagação de um vírus. A abordagem proposta é aplicada a três empresas brasileiras reais, onde, em duas delas, comparamos nossos resultados com os produzidos manualmente por essas empresas e na terceira, aplicamos o *framework* para gerar a escala de trabalho com a inclusão de mais um turno.

Os resultados experimentais mostraram que, em menos de um segundos, o *framework* proposto conseguiu construir uma solução que melhorou os resultados entre 15% e 20%, em comparação com as soluções geradas pelas empresas. Auxiliou na construção de escalas de trabalho com inclusão de novos turnos. Além de reduzir o risco de contágio em casos de redistribuição de turnos.

Os resultados demonstraram que o *framework* pode ser adequado com sucesso para diferentes tipos de problema. A adaptação permitirá à empresa, através da inserção e remoção de Restrições, modelar o *framework* para adaptar às diferentes situações e conforme a necessidade.

Como trabalho futuro, pode-se considerar a inclusão do fator de risco de contágio na função objetivo. Além disso, tem potencial para considerar a inserção de um fator emocional e as preferências de desempenho de cada colaborador, aplicando mudanças na função objetivo a fim de melhorar o desempenho individual e coletivo da equipe. Pretende-se aumentar o número de empresas estudadas, e serão possivelmente incluídas novas Restrições e variáveis, possibilitando dessa forma, aumentar a flexibilidade do *framework*.

Referências bibliográficas

- G. Ambrogio, L. Filice, F. Longo, and A. Padovano. Workforce and supply chain disruption as a digital and technological innovation opportunity for resilient manufacturing systems in the COVID-19 pandemic. *Computers & Industrial Engineering*, 169:108158, 2022.
DOI [10.1016/j.cie.2022.108158](https://doi.org/10.1016/j.cie.2022.108158).
- Eduardo Leopoldino ANDRADE. Introdução à pesquisa operacional métodos e modelos para análise de decisões. 2. Ed. Rio de Janeiro: LTC-Livros Técnicos e Científicos Editora SA, 2000.
- David Applegate, Robert Bixby, Vašek Chvátal, and William Cook. The traveling salesman problem, 2006.
- Estela ML Aquino, Ismael Henrique Silveira, Julia Moreira Pescarini, Rosana Aquino, Jaime Almeida de Souza-Filho, Aline dos Santos Rocha, Andrea Ferreira, Audêncio Victor, Camila Teixeira, Daiane Borges Machado, et al. Medidas de distanciamento social no controle da pandemia de covid-19: potenciais impactos e desafios no brasil. *Ciência & Saúde Coletiva*, 25:2423–2446, 2020.
- Kenneth R Baker. Workforce allocation in cyclical scheduling problems: A survey. *Journal of the Operational Research Society*, 27(1):155–167, 1976.
- Guido Bertolini, Giovanni Nattino, Martin Langer, M Tavola, Daniele Crespi, M Mondini, C Rossi, C Previtali, J Marshall, Daniele Poole, et al. The role of the intensive care unit in real-time surveillance of emerging pandemics: the italian giviti experience. *Epidemiology & Infection*, 144(2):408–412, 2016.
- William J Burgess and Robert E Busby. Personnel scheduling. *Handbook of industrial engineering*, 81:2155–2169, 1992.
- Edmund K Burke, Patrick De Causmaecker, Greet Vanden Berghe, and Hendrik Van Landeghem. The state of the art of nurse rostering. *Journal of scheduling*, 7(6): 441–499, 2004.

- Manoel Campêlo, Victor A Campos, and Ricardo C Corrêa. Um algoritmo de planos-de-corte para o número cromático fracionário de um grafo. *Pesquisa Operacional*, 29:179–193, 2009.
- Dawei Chen, Shuangli Pan, Qun Chen, and Jiahui Liu. Vehicle routing problem of contactless joint distribution service during covid-19 pandemic. *Transportation Research Interdisciplinary Perspectives*, 8:100233, 2020. ISSN 2590-1982.
DOI <https://doi.org/10.1016/j.trip.2020.100233>. URL <https://www.sciencedirect.com/science/article/pii/S2590198220301445>.
- C. Contardo and L. Costa. On the optimal layout of a dining room in the era of COVID-19 using mathematical optimization. *International Transactions in Operational Research*, 2022.
DOI [10.1111/itor.13139](https://doi.org/10.1111/itor.13139).
- Adina Coroiu, Chelsea Moran, Tavis Campbell, and Alan C Geller. Barriers and facilitators of adherence to social distancing recommendations during covid-19 among a large international sample of adults. *PloS one*, 15(10):e0239795, 2020.
- George B Dantzig. A comment on edie’s “traffic delays at toll booths”. *Journal of the Operations Research Society of America*, 2(3):339–341, 1954.
- Joffre Marcondes de Rezende. Epidemia, endemia, pandemia, epidemiologia. *Revista de Patologia Tropical/Journal of Tropical Pathology*, 27(1), 1998.
- Tercio Alberto dos Santos, Alexandre César Rodrigues da Silva, and Carlos Eduardo da Silva Santos. Branch and bound aplicado no problema de cobertura de funções booleanas. 2003.
- Kathryn A. Dowsland. Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, 106(2):393–407, 1998. ISSN 0377-2217.
DOI [https://doi.org/10.1016/S0377-2217\(97\)00281-6](https://doi.org/10.1016/S0377-2217(97)00281-6). URL <https://www.sciencedirect.com/science/article/pii/S0377221797002816>.
- Ding-Zhu Du and Ker-I Ko. *Theory of computational complexity*, volume 58. John Wiley & Sons, 2011.
- Leslie C Edie. Traffic delays at toll booths. *Journal of the operations research society of America*, 2(2):107–138, 1954.
- Melanie Erhard, Jan Schoenfelder, Andreas Fügener, and Jens O Brunner. State of the art in physician scheduling. *European Journal of Operational Research*, 265(1):1–18, 2018.
- Andreas T Ernst, Houyuan Jiang, Mohan Krishnamoorthy, and David Sier. Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27, 2004.

- Michael R Garey and David S Johnson. Computers and intractability. *A Guide to the*, 1979.
- T. Geibinger, L. Kletzander, M. Krainz, F. Mischek, N. Musliu, and F. Winter. Physician scheduling during a pandemic. In *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 456–465. Springer International Publishing, 2021. DOI [10.1007/978-3-030-78230-6_29](https://doi.org/10.1007/978-3-030-78230-6_29).
- Marco Cesar Goldberg and Henrique Pacca L Luna. *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier, 2005.
- Pedro Henrique González, Av Maracanã, Rio de Janeiro-RJ Maracanã, Glaubos Climaco, Luidi Simonetti, Bruno Salezze Vieira, Glaydston Mattos Ribeiro, Nilo Flávio Rosa Campos Júnior, Carlos Alberto Abramides, and André de Oliveira Nunes. Um algoritmo branch-and-cut para o problema de localização ótima de contadores de tráfego em redes de transporte. *XLIX Simpósio Brasileiro de Pesquisa Operacional*. <https://doi.org/10.13140>, 2017.
- Stephen C Graves. A review of production scheduling. *Operations research*, 29(4):646–675, 1981.
- Francesca Guerriero and Rosita Guido. Modeling a flexible staff scheduling problem in the Era of Covid-19. *Optimization Letters*, July 2021. ISSN 1862-4480. DOI [10.1007/s11590-021-01776-3](https://doi.org/10.1007/s11590-021-01776-3). URL <https://doi.org/10.1007/s11590-021-01776-3>.
- Gregory Gutin and Abraham P Punnen. *The traveling salesman problem and its variations*, volume 12. Springer Science & Business Media, 2006.
- Frederick S Hillier and Gerald J Lieberman. *Introdução à pesquisa operacional*. McGraw Hill Brasil, 2013.
- Mark ET Horn, Houyuan Jiang, and Philip Kilby. Scheduling patrol boats and crews for the royal australian navy. *Journal of the Operational Research Society*, 58(10):1284–1293, 2007.
- Nadia Jebril. World health organization declared a pandemic public health menace: a systematic review of the coronavirus disease 2019 “COVID-19”. *International Journal of Psychosocial Rehabilitation*, 24:2784–2795, 2020.
- Eugene L Lawler and David E Wood. Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719, 1966.
- Cláudio Loesch and Nelson Hein. *Pesquisa operacional: fundamentos e modelos*. Saraiva, 2009.
- André Andrade Longaray. *Introdução à pesquisa operacional*. Saraiva Educação SA, 2017.

- Nelson Maculan and Marcia H Costa Fampa. Otimização linear. *Editora Universidade de Brasilia: Brasilia*, 2006.
- H. Abouee Mehrizi, A. Aminoleslami, J. Darko, E. Osei, and H. Mahmoudzadeh. Staff scheduling during a pandemic: The case of radiation therapy department. *SSRN Electronic Journal*, 2022. DOI [10.2139/ssrn.4104581](https://doi.org/10.2139/ssrn.4104581).
- J. Moosavi and S. Hosseini. Simulation-based assessment of supply chain resilience with consideration of recovery strategies in the COVID-19 pandemic context. *Computers & Industrial Engineering*, 160:107593, 2021. DOI [10.1016/j.cie.2021.107593](https://doi.org/10.1016/j.cie.2021.107593).
- MS. Painel de casos de doença pelo coronavírus 2019 (covid-19) no brasil pelo ministério da saúde, 2022. URL <https://covid.saude.gov.br/>.
- Anton Ovchinnikov and Joseph Milner. Spreadsheet model helps to assign medical residents at the university of vermont's college of medicine. *Interfaces*, 38(4):311–323, 2008.
- Emir Hüseyin Özder, Evrencan Özcan, and Tamer Eren. A systematic literature review for personnel scheduling problems. *International Journal of Information Technology & Decision Making*, 19(06):1695–1735, 2020.
- J. Pacheco and M. Laguna. Vehicle routing for the urgent delivery of face shields during the COVID-19 pandemic. *Journal of Heuristics*, 26(5):619–635, 2020. DOI [10.1007/s10732-020-09456-8](https://doi.org/10.1007/s10732-020-09456-8).
- Manfred Padberg and Giovanni Rinaldi. Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations research letters*, 6(1):1–7, 1987.
- Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM review*, 33(1):60–100, 1991.
- Christos H Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and system Sciences*, 48(3):498–532, 1994.
- T. Perdana, D. Chaerani, A. Luqmanul H. Achmad, and F. Rahayu Hermiatin. Scenarios for handling the impact of COVID-19 based on food supply network through regional food hubs under uncertainty. *Heliyon*, 6(10):e05128, 2020. DOI [10.1016/j.heliyon.2020.e05128](https://doi.org/10.1016/j.heliyon.2020.e05128).
- Rodrigo Rodrigues. *Pesquisa Operacional*. Grupo A, 2017.
- Ermes Medeiros da SILVA, Elio Medeiros SILVA, Valter GONÇALVES, and Afrânio Carlos MUROLO. Pesquisa operacional para os cursos de administração e engenharia. *São Paulo: Atlas*, page 1, 2010.

- Gavin JD Smith, Justin Bahl, Dhanasekaran Vijaykrishna, Jinxia Zhang, Leo LM Poon, Honglin Chen, Robert G Webster, JS Malik Peiris, and Yi Guan. Dating the emergence of pandemic influenza viruses. *Proceedings of the National Academy of Sciences*, 106(28):11709–11712, 2009.
- Seyda Topaloglu and Irem Ozkarahan. An implicit goal programming model for the tour scheduling problem considering the employee work preferences. *Annals of Operations Research*, 128(1):135–158, 2004.
- Rafael D. Tordecilla, Leandro do C. Martins, Miguel Saiz, Pedro J. Copado-Mendez, Javier Panadero, and Angel A. Juan. *Agile Computational Intelligence for Supporting Hospital Logistics During the COVID-19 Crisis*, pages 383–407. Springer International Publishing, Cham, 2021. ISBN 978-3-030-72929-5. DOI 10.1007/978-3-030-72929-5_18. URL https://doi.org/10.1007/978-3-030-72929-5_18.
- Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review. *European journal of operational research*, 226(3):367–385, 2013.
- Salvatore Benito Virgillito. *Pesquisa Operacional*. Saraiva Educação SA, 2017.
- Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.
- Laurence A Wolsey and George L Nemhauser. *Integer and combinatorial optimization*, volume 55. John Wiley & Sons, 1999.
- Giorgio Zucchi, Manuel Iori, and Anand Subramanian. Personnel scheduling during Covid-19 pandemic. *Optimization Letters*, 15(4):1385–1396, June 2021. ISSN 1862-4480. DOI 10.1007/s11590-020-01648-2. URL <https://doi.org/10.1007/s11590-020-01648-2>.