

Universidade Federal de Mato Grosso do Sul

Instituto de Matemática

Programa de Pós-Graduação

Mestrado Profissional em

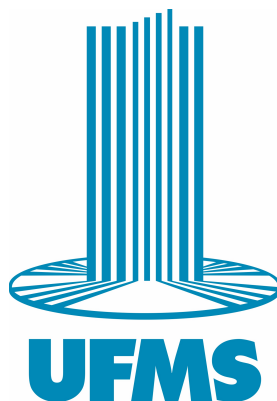
Matemática em Rede Nacional

Lee Jing Xuan

**PROBLEMA DE DIMENSIONAMENTO DE LOTES: UMA
ABORDAGEM VIA RELAXAÇÃO LAGRANGIANA**

Campo Grande - MS

2024



Universidade Federal de Mato Grosso do Sul

Instituto de Matemática

Programa de Pós-Graduação

Mestrado Profissional em

Matemática em Rede Nacional

Lee Jing Xuan

**PROBLEMA DE DIMENSIONAMENTO DE LOTES: UMA
ABORDAGEM VIA RELAXAÇÃO LAGRANGIANA**

Orientadora Prof.^a Dra. Rubia Mara de Oliveira Santos

Dissertação apresentada ao Programa de Pós-Graduação Mestrado Profissional em Matemática em Rede Nacional do Instituto de Matemática da Universidade Federal de Mato Grosso do Sul - INMA/UFMS, como parte dos requisitos para obtenção do título de Mestre.

Campo Grande - MS

2024

PROBLEMA DE DIMENSIONAMENTO DE LOTES: UMA ABORDAGEM VIA RELAXAÇÃO LAGRANGIANA

Lee Jing Xuan

Dissertação submetida ao Programa de Pós-Graduação Mestrado Profissional em Matemática em Rede Nacional do Instituto de Matemática da Universidade Federal de Mato Grosso do Sul - INMA/UFMS, como parte dos requisitos para obtenção do título de Mestre.

Banca examinadora:

Profa. Dra. Rubia Mara de Oliveira Santos (Orientadora)
Universidade Federal de Mato Grosso do Sul - UFMS

Prof. Dr. Willy Alves de Oliveira Soler
Universidade Federal de Mato Grosso do Sul - UFMS

Profa. Dra. Maristela Oliveira dos Santos
Universidade de São Paulo - USP

Campo Grande - MS, 25 de julho de 2024

“Aprender sem pensar é um esforço em vão, mas pensar sem nada aprender é um perigo”

Confúcio

Agradecimentos

Agradeço primeiramente a Deus pela criação de tudo existente neste Universo, por ter concedido a vida e a inteligência ao ser humano, para que, no curso da vida, possamos aprender a evoluir com cada pessoa, assunto e circunstância.

Aos meus pais, Lee Goon Leng e Lim Siew Kio, pelas suas virtudes, ensinamentos, sacrifícios, amor incondicional e atitudes exemplares que me fez trilhar um caminho bom e me tornar o que eu sou hoje. Nunca irei me esquecer que cada conquista da minha vida foi devida a eles.

Ao meu irmão, Lee Yu Xuan, pela sua fraternidade. Por ter compartilhado tantos momentos da vida, sempre fornecendo apoio, incentivo e energias positivas.

À orientadora, Profa. Dra. Rubia Mara de Oliveira Santos, pela sua paciência e apoio dedicado ao longo do meu programa de mestrado, os quais fortaleceram minha determinação e confiança. Admiro sua proatividade e entusiasmo, os quais me inspiraram academicamente e também como pessoa. Sou grato por ter tido a oportunidade de aprender sob sua supervisão.

Ao Prof. Dr. Willy Alves de Oliveira Soler, pela sua excelência profissional e seu apoio constante no desenvolvimento acadêmico e profissional.

A todos os professores do INMA, pelo acolhimento e ensino-aprendizagem que contribuíram para meu crescimento acadêmico e profissional.

Aos colegas de turma, Marcelo Costa Valeriano, Donizeth Jacinto de Souza e Adriana Livi, pela companhia nessa jornada de mestrado, compartilhando momentos de alegria e superando as dificuldades.

À Fundação de Apoio ao Desenvolvimento do Ensino, Ciência e Tecnologia (FUN-DECT) do Estado de Mato Grosso do Sul pelo apoio financeiro e à Universidade Federal de Mato Grosso do Sul (UFMS) pelo incentivo e a oportunidade de trilhar o caminho da pesquisa científica.

Resumo

Um problema de dimensionamento de lotes (PDL) é um problema de planejamento de produção. Dado um horizonte de planejamento discretizado em períodos de tempo, busca-se determinar quando e quantos produtos devem ser produzidos em cada período, visando à minimização dos custos operacionais. O objetivo deste trabalho é resolver um PDL da classe NP-difícil por meio da aplicação da Relaxação Lagrangiana. Essa técnica simplifica o problema ao dualizar restrições, introduzindo penalidades. O desafio é, então, determinar as melhores penalidades que aproximem a solução da otimalidade, utilizando, para isso, o método do subgradiente. Além disso, propõem-se heurísticas de factibilização e melhoria para a obtenção de boas soluções. Por fim, a eficiência da técnica para o problema estudado será avaliada com base em testes realizados com instâncias da literatura, considerando o tempo de resolução, o gap de dualidade e as soluções obtidas pelas abordagens propostas.

Palavras-chave: Problema de Dimensionamento de Lotes, Relaxação Lagrangiana, Método Subgradiente e Heurísticas de Factibilização e Melhoria.

Abstract

A lot-sizing problem (LSP) is a production planning problem. Given a planning horizon discretized into time periods, the goal is to determine when and how many products should be produced in each period, aiming to minimize operational costs. The objective of this work is to solve an NP-hard LSP by applying Lagrangian Relaxation. This technique simplifies the problem by dualizing constraints and introducing penalties. The challenge is to determine the optimal penalties that bring the solution closer to optimality, using the subgradient method. In addition, heuristics for feasibility and improvement are proposed to obtain high-quality solutions. Finally, the efficiency of the technique for the problem under study will be evaluated based on tests performed with instances from the literature, considering the resolution time, duality gap, and the solutions obtained by the proposed approaches.

Key words: Lot Sizing Problem, Lagrangian Relaxation, Subgradient Method and Feasibility and Improvement Heuristics.

Sumário

1	Introdução	1
1.1	Objetivo do Trabalho	6
1.2	Organização do Trabalho	6
2	Problema de Dimensionamento de Lotes	8
2.1	Modelo Matemático	9
2.2	Considerações Finais	13
3	Relaxação Lagrangiana	14
3.1	Construção da Relaxação Lagrangiana para o Modelo Matemático	19
3.1.1	Aplicação na Restrição 2	20
3.1.2	Aplicação na Restrição 5	21
3.1.3	Aplicação na Restrição 2 e 5 Simultaneamente	22
3.2	Método Subgradiente	23
3.2.1	Algoritmo para Relaxar Restrição 2	25
3.2.2	Algoritmo para Relaxar Restrição 5	28
3.2.3	Algoritmo para Relaxar Restrição 2 e 5 Simultaneamente	30
3.3	Considerações Finais	32
4	Heurísticas Desenvolvidas	33
4.1	Heurísticas de Factibilização	35
4.1.1	Relaxação da Restrição 5	35
4.1.2	Relaxação da Restrição 2	40
4.1.3	Relaxação das Restrições 2 e 5 Simultaneamente	43
4.2	Heurística de Melhoria	43
4.3	Considerações Finais	46

5	Desenvolvimentos e Resultados Computacionais	47
5.1	Implementação do Modelo Matemático	48
5.2	Testes com Instâncias da Literatura	50
5.3	Considerações Finais	59
6	Conclusões e Trabalhos Futuros	60

Lista de Tabelas

5.1	Resultados dos Testes das Instâncias Classe A	52
5.2	Resultados dos Testes das Instâncias Classe B	53
5.3	Resultados dos Testes das Instâncias Classe C - Parte 1	55
5.4	Resultados dos Testes das Instâncias Classe C - Parte 2	55
5.5	Resultados dos Testes das Instâncias Classe D - Parte 1	57
5.6	Resultados dos Testes das Instâncias Classe D - Parte 2	57
5.7	Resultados dos Testes das Instâncias Classe E - Parte 1	58
5.8	Resultados dos Testes das Instâncias Classe E - Parte 2	58

Lista de Figuras

3.1	Soluções Ótimas em S e R	17
-----	------------------------------------	----

Capítulo 1

Introdução

Conforme o Art. 13^o do regimento do ProfMat, a obtenção do título de Mestre requer o desenvolvimento de um recurso educacional e uma dissertação de mestrado, os quais descrevem os fundamentos teóricos empregados e os processos que culminaram na criação deste produto e sua aplicação em situações de ensino. Todo o projeto desenvolvido durante o mestrado foi documentado em um artigo aceito para publicação na revista Professor de Matemática Online (PMO). Cumprindo as exigências do Mestrado Profissional em Matemática e considerando o impacto da lógica computacional na prática pedagógica para o ensino da matemática, os resultados demonstraram contribuições significativas para a sala de aula, além de melhorias para o professor de Matemática. Com base na importância da lógica computacional no ensino da matemática, avançamos para suas aplicações em contextos reais, explorando heurísticas e modelos matemáticos para resolver e otimizar diversos problemas, com foco específico nesta dissertação no problema de dimensionamento de lotes.

Dessa forma, o presente trabalho será composto por duas partes principais. A primeira, intitulada **Contribuição na Educação**, discute um projeto voltado para a integração da lógica computacional no ensino da matemática. A segunda, denominada **Contribuição na Matemática**, trata da aplicação da Relaxação Lagrangiana e do método do subgradiente em um PDL, além do desenvolvimento de heurísticas. Inicialmente, serão apresentadas as contribuições na área educacional.

Contribuição na Educação

O processo de ensino e aprendizagem da matemática nas escolas públicas enfrenta desafios crescentes, com um notável aumento de alunos apresentando conhecimentos básicos pouco consolidados e habilidades lógicas subdesenvolvidas. Esta realidade reflete um desânimo considerável dos estudantes em relação à disciplina. Em resposta a esse cenário, propomos uma abordagem temática inspirada em bruxos e magias de Harry Potter para introduzir conceitos de lógica computacional e a linguagem de programação Python. O objetivo é utilizar uma metodologia ativa e lúdica para contribuir na construção de pensamentos matemáticos, lógicos e computacionais. Almeja-se, por fim, promover um maior engajamento dos alunos e aprimorar a comunicação nas aulas, destacando a relevância da lógica computacional no desenvolvimento de competências e habilidades matemáticas.

Nos últimos anos, a linguagem de programação Python tem recebido crescente reconhecimento, contando sua participação no desenvolvimento de software moderno, gerenciamento de infraestrutura, análise de dados, criação de jogos e robótica. Com isso, debates sobre sua importância na educação básica e as práticas de integração no ensino da Matemática tornaram-se assuntos recorrentes na literatura. Por exemplo, a investigação da eficiência da linguagem de programação Python, como recurso didático para formular e resolver problemas matemáticos no contexto da educação básica [20]; a análise das possibilidades do Python na escola pública para Ensino Médio [5]; a linguagem Python como ferramenta no ensino básico [3]; entre outros.

A proposta foi implementada a partir do segundo semestre de 2023, na Escola Estadual Professora Clarinda Mendes de Aquino (CMA), localizada na Avenida Murilo Rolim Júnior, 200 - Jardim Petrópolis, Campo Grande - MS. É uma instituição de educação integral e autoral que fomenta o protagonismo juvenil e valoriza constantemente práticas pedagógicas inovadoras e diversificadas.

No término do projeto, a partir do retorno e comentários de alunos, professores e demais membros da comunidade escolar, constatamos que nosso trabalho foi bem-sucedido e recebeu reconhecimento. O respaldo e estímulo que recebemos nos impulsionam a prosseguir e expandir o projeto, com o objetivo de proporcionar, nos próximos anos, a oportunidade de aprendizado e aprimoramento em conhecimentos computacionais a todos os estudantes. No entanto, é essencial compreender que tanto a lógica computacional quanto a lógica matemática são habilidades e competências que se desenvolvem ao longo do tempo.

Nesse sentido, não se deve esperar uma melhora significativa imediata no desempenho nas avaliações. Contudo, a relevância dessas habilidades no ensino da Matemática é indiscutível. Ao integrar as lógicas de programação no entendimento dos conteúdos matemáticos, os assuntos tornam-se mais coerentes e acessíveis.

Contribuição na Matemática

Na era contemporânea, após grandes eventos como as revoluções industriais e o avanço tecnológico, a economia global tornou-se cada vez mais complexa e interconectada. As empresas enfrentam desafios constantes para otimizar suas operações e manter-se competitivas. Entre esses desafios, destaca-se o problema de dimensionamento de lotes, crucial para a gestão eficiente da produção e dos estoques.

O problema de dimensionamento de lotes (PDL) (em inglês, LSP – Lot Sizing Problem) refere-se à determinação da quantidade ideal de unidades de um produto que deve ser produzida ou adquirida em um determinado intervalo de tempo. Esse problema é central para muitas indústrias, pois influencia diretamente os custos de produção, os níveis de estoque, o atendimento à demanda e a utilização de recursos. Em geral, as indústrias podem utilizar o PDL para traçar planos de produção em seu horizonte de planejamento, visando atender as demandas em todos os períodos, evitando atrasos, e também obter um bom aproveitamento das linhas de produção quando são preparadas e inicializadas em cada período.

Em um cenário onde a variação da demanda, os tempos de produção e as restrições de capacidade são fatores dinâmicos, encontrar a solução ótima para o dimensionamento de lotes exige uma abordagem integrada. Técnicas diversificadas, assim como métodos heurísticos, são frequentemente aplicadas para resolver esse problema, visando a eficiência e a flexibilidade nas operações empresariais.

Um PDL pode possuir diversas caracterizações dependendo dos seus aspectos, e estes influenciam diretamente na estrutura e complexidade computacional do problema. Vassalo [14] definiu os principais aspectos como:

1) Item: Um sistema produtivo é classificado como único-item quando só produz um único produto, e quando há mais tipos para serem desenvolvidos, denomina-se de multi-item.

2) Estágio: Um sistema produtivo é classificado como multiestágio quando os itens produzidos são dependentes um do outro, ou melhor, para a produção de algum determinado produto é necessário de um outro, o qual pode-se chamar de produto componente. Quando um sistema não apresenta dependência entre os produtos, denominamos de monoestágio.

3) Demanda: Um sistema produtivo pode ter itens com demanda estática quando os valores da demanda não alteram conforme o tempo. E nos casos que acontecem as variações, denomina-se de demanda dinâmica. Paralelamente, pode-se ter problemas em que os valores da demanda são conhecidos, os quais chama-se de demanda determinística, e aqueles com demanda conhecidos através de uma distribuição de probabilidade, são problemas com demanda estocástica.

4) Horizonte de planejamento: Um sistema produtivo pode ter um horizonte de planejamento, isto é, o intervalo de tempo no planejamento de produção, finito ou infinito. Além disso, um sistema produtivo pode ser planejado de duas maneiras distintas, sendo discreto quando há uma divisão em intervalos de tempo menores denominados de período, e caso contrário, um problema contínuo.

5) Capacidade: Um sistema produtivo que não há limitação na quantidade de itens produzidos é classificado como não capacitado. E quando existir um limite máximo de produção, o problema é considerado capacitado.

6) Tempo e custo de preparo: Em um sistema produtivo, se o tempo e custo necessário para a preparação de uma máquina depende unicamente do item que será produzido, então o tempo e custo de preparo é considerado simples. Caso o tempo e custo da preparação envolve os itens produzidos antes ou depois, então o tempo e custo é complexo, ou, dependente da sequência.

7) Estoque: Em um sistema produtivo, dado um determinado período, os itens produzidos excedente às demandas nesse intervalo de tempo será armazenada, o que se denomina de estoque. Em geral, haverá um custo associado ao armazenamento desses produtos, ou seja, um custo de estoque.

8) Atraso na entrega: Em algumas indústrias, pode ser que a demanda não atendida em um período seja atendida em períodos posteriores, e assim, quando uma demanda não é atendida no seu período, terá um atraso na entrega.

Em 1913, a fórmula do Lote Econômico de Compra (LEC ou, em inglês, Economic Order Quantity - EOQ) foi introduzido ao mundo pela primeira vez, considerado o modelo primordial envolvendo o PDL. Este considera que o processo produtivo ocorre em apenas um único nível, sem restrições de capacidade, as demandas ocorrem a uma taxa constante e o horizonte de planejamento é infinito [2]. Posteriormente, outros modelos foram criados a partir desse para assimilar cada vez mais as situações reais.

Em 1950, surgiu a Programação de Lotes Econômicos de Produção (em inglês, Economic Lot Scheduling Problem - ELSP) que é um modelo monoestágio, com restrição de capacidade e vários itens, demanda estacionária e os períodos de tempo são contínuos com um horizonte de planejamento infinito [24]. O ELSP é caracterizado por ser um problema pertencente à classe NP-difícil.

Em 1987, Eppen e Martin [8] desenvolveram métodos e técnicas para resolver PDL capacitados com múltiplos itens. Especificamente, suas contribuições incluíram a proposição de técnicas de redefinição de variáveis que permitiam a formulação de modelos matemáticos mais eficientes para problemas complexos de dimensionamento de lotes.

Em 1989, Trigeiro, Thomas e Mcclain [31] abordaram sobre o problema de dimensionamento de lotes capacitado (em inglês, Capacitated Lot Sizing Problem - CLSP), considerando restrições de capacidade de produção da linha, tempos e custos de preparação, que podem variar entre os períodos. A técnica de Relaxação Lagrangiana foi aplicada para simplificar os problemas por meio da dualização das restrições de capacidade. Além disso, para obter soluções factíveis através das soluções duais, foi proposto um procedimento heurístico. Novas abordagens de resolução para CLSP podem ser encontradas em Gramani, Bourjolly e França [19].

Segundo Maes et al. [11], encontrar uma solução factível para problemas de dimensionamento de lotes com capacidade limitada que considere tempo de preparação é NP-completo. Por esta razão, a maioria das técnicas de resolução encontradas na literatura são heurísticas dedicadas a resolver problemas específicos (Billington et al. [22] e Bahl et al. [9]).

Em 1996, Haase [12] propôs o modelo *Capacitated Lotsizing Problem with Sequence Dependent Setup Costs* (CLSD), permitindo a produção de diversos tipos de produtos num mesmo período.

Em 2021, Soler, Santos e Akartunali [30] abordaram PDL com base nos contextos das indústrias alimentícias brasileiras em que os produtos são perecíveis e diversas linhas de produção compartilham do mesmo recurso escassos. Ademais, propuseram heurísticas construtivas do tipo *relax-and-fix* que demonstraram rendimento maior do que o *solver* comercial de alto desempenho quando o tempo máximo de execução é inferior a três horas.

Em 2022, Medeiros, Soler e Queiroz [16], um caso particular do problema proposto por Soler, Santos e Akartunali [30] foi estudado e designado de Problema de Dimensionamento de Lotes considerando Produtos Perecíveis e Linhas de Produção que Compartilham Recursos Escassos (PDL-PPRE). O desempenho computacional do modelo foi analisado por meio da linguagem de programação C++ com auxílio da ferramenta Concert Technology do solver IBM ILOG Cplex 20.1. Os resultados evidenciaram que para instâncias cada vez maiores, o tempo de busca e qualidade das soluções não estavam sendo muito satisfatórias. E assim, os autores indicam uma pendência na investigação de abordagens baseadas na Relaxação Lagrangiana, bem como, procedimentos heurísticos baseadas na formulação matemática do problema.

1.1 Objetivo do Trabalho

Nesta dissertação de mestrado, o modelo desenvolvido será o mesmo estudado por Medeiros, Soler e Queiroz [16], o qual foi adaptado dos modelos propostos por Haase [12] e por Soler, Santos e Akartunali [30]. Além disso, foi utilizada a redefinição de variáveis proposta por Eppen e Martin [8] para a modelagem da perecibilidade dos itens.

O objetivo deste trabalho é resolver a pendência proposta pelos autores do PDL-PPRE. Utilizando a Relaxação Lagrangiana e métodos heurísticos, busca-se obter melhores soluções para as instâncias disponíveis na literatura, especialmente para aquelas de grande porte.

1.2 Organização do Trabalho

Primeiramente, o modelo matemático do PDL-PPRE será apresentado no Capítulo 2. Serão detalhados os conjuntos e parâmetros, a função objetivo e as restrições, para proporcionar uma melhor compreensão do problema.

Em seguida, no Capítulo 3, serão introduzidos os conceitos da Relaxação Lagrangiana. Essa técnica envolve a remoção de um subconjunto de restrições para simplificar o problema, embora com o custo de penalidades por essa simplificação. As restrições de acoplamento são candidatas ideais para aplicação da técnica, dado que são elas que aumentam a complexidade do modelo. E assim, na ausência delas, podem resultar em subproblemas independentes mais fáceis de serem solucionados. Para determinar as melhores penalidades que garantam uma solução próxima da ótima do problema original, será apresentado e utilizado o método subgradiente, uma abordagem iterativa para encontrar penalidades cada vez mais eficazes.

No entanto, o método subgradiente raramente alcança a solução ótima desejada e as soluções obtidas não são factíveis para o problema original. Assim, desenvolvemos heurísticas de factibilização e melhoria para as soluções, as quais serão apresentadas no Capítulo 4.

Com as formulações matemáticas finalizadas, o Capítulo 5 será dedicado ao desenvolvimento computacional. Apresentaremos os algoritmos desenvolvidos, assim como os resultados dos testes com as instâncias da literatura.

Por fim, com base nos resultados obtidos, elaboraremos as conclusões do nosso trabalho e discutiremos os possíveis trabalhos futuros, os quais serão abordados no Capítulo 6.

Capítulo 2

Problema de Dimensionamento de Lotes

A gestão industrial refere-se à administração de processos industriais, abrangendo atividades como o planejamento, a organização, a programação e o controle da produção de bens e serviços, além da gestão de recursos e estoques, entre outras responsabilidades. O profissional de PCP (Planejamento e Controle da Produção), encarregado dessas tarefas, deve buscar executá-las da forma mais eficiente possível para contribuir para o sucesso da fábrica.

A indústria alimentícia no Brasil está passando por um período de notável prosperidade e crescimento. Com um ritmo acelerado de exportações, a incorporação de tecnologias avançadas e diversas melhorias operacionais, o setor tem se mostrado capaz de atender às demandas cada vez mais exigentes dos consumidores. Com base no relatório anual, divulgado no final de 2022, pelo Ministério da Economia mais de 10% de todo o PIB nacional é oriundo da indústria alimentícia [1].

No entanto, apesar dos avanços significativos, a indústria ainda enfrenta uma série de desafios que precisam ser superados para assegurar sua continuidade e prosperidade no futuro. Podemos destacar várias questões, como a perecibilidade dos produtos, a otimização do uso dos recursos para promover a sustentabilidade, e a gestão aprimorada da cadeia de suprimentos e da logística da empresa.

O problema de dimensionamento de lotes estudado neste trabalho foi inspirado no contexto de algumas indústrias alimentícias brasileiras, semelhante ao problema abordado por Soler [28]. Considere um ambiente produtivo com diversas linhas de produção que compartilham os mesmos recursos produtivos, tais como trabalhadores e máquinas, e devido à escassez desses recursos e características do problema, essas linhas de produção

não podem operar todas simultaneamente. As linhas de produção são especializadas, de modo que, para cada item uma única linha é apta para produzi-lo.

Em relação à perecibilidade dos produtos, é essencial garantir que eles sejam estocados de forma adequada para preservar sua qualidade e segurança. Em indústrias alimentícias, a estocagem deve seguir rigorosos padrões de higiene e controle de temperatura para evitar a deterioração dos produtos. Técnicas como o uso de sistemas de controle de temperatura em tempo real e a implementação de práticas de "primeiro a entrar, primeiro a sair"(FIFO) são imprescindíveis para manter a qualidade dos produtos e minimizar o desperdício. Além disso, a organização eficiente dos estoques, com base em métodos como a classificação ABC para priorizar a gestão de itens de maior valor ou demanda, pode otimizar a eficiência operacional e reduzir custos. A correta rotulagem e monitoramento dos prazos de validade também são fundamentais para assegurar a segurança alimentar e a conformidade com regulamentações vigentes.

2.1 Modelo Matemático

Nesta Seção apresenta-se o modelo matemático de otimização inteira mista para representação do PDL-PPRE. O modelo proposto contempla os seguintes aspectos:

1. Multi-item
2. Monoestágio
3. Demanda dinâmica e determinística
4. Horizonte de planejamento finito e discreto
5. Capacitado
6. Tempo e custo de preparo simples
7. Estoque
8. Atraso na entrega

Além dos aspectos apresentados, outras características devem ser salientadas. Primeiramente, mesmo que o sistema produtivo seja monoestágio, ou seja, não há dependência entre os produtos para serem produzidos, mas as múltiplas linhas de produção compartilham dos mesmos recursos escassos. Em segundo lugar, os produtos são perecíveis e não devem sofrer deterioração em estoque, isto é, existe um prazo de validade que deve ser controlado por produto. E por fim, os procedimentos de preparação iniciados num dado período devem ser completamente concluídos nesse período.

Os parâmetros utilizados para a modelagem do problema abordado são apresentados.

T	Conjunto de períodos, indexados por t e p
L	Conjunto de linhas de produção, indexadas por l
J	Conjunto de produtos, indexados por j
K	Conjunto de recursos, indexados por k . Usa-se o índice $k = 0$ para representar os trabalhadores e o índice $k > 0$ para os demais recursos
P_l	Conjunto de produtos que podem ser produzidos na linha l
d_{jt}	Demanda do produto j para o período t
C_{lt}	Capacidade produtiva (em tempo) da linha l para o período t
a_{lj}	Tempo de produção de uma unidade do produto j na linha l
m_{lj}	Quantidade mínima do produto j que deve ser produzida quando a linha l está configurada para produzi-lo
h_j	Custo associado à manutenção de uma unidade do item j em estoque
b_j	Custo unitário referente ao atraso no atendimento da demanda do item j
sc_{lj}	Custo de setup para produção do produto j na linha l
st_{lj}	Tempo de preparação para produção do produto j na linha l
r_{kl}	Quantidade do recurso k necessário para montar na linha l
R_k	Quantidade disponível do recurso k no período t
sl_j	Prazo de validade do produto j
ac_l	Custo de montagem da linha l
γ	Número máximo de produtos que podem ser fabricados em cada linha e período

Para descrever a função objetivo que será apresentada, faz-se necessário o parâmetro auxiliar a seguir.

hb_{jtp} O custo unitário associado à produção item j , durante o período t , para atendimento da demanda referente ao período p .

Observe que quando:

$p > t \rightarrow hb_{jtp}$ representa necessidade de estoque, e assim, é o valor unitário de manter o item j em estoque do período t até o período p

$p < t \rightarrow hb_{jtp}$ representa um atraso, e assim, é o custo unitário relacionado ao atraso de $t-p$ períodos no atendimento de uma demanda pelo produto j .

$$hb_{jtp} = \begin{cases} h_j(p - t) & \text{se } p \geq t \\ b_j(t - p) & \text{se } p < t \end{cases}$$

As variáveis necessárias e utilizadas:

- x_{ljtp} Quantidade de produtos do tipo j produzidos na linha l durante o período t para atender a demanda no período p
- w_{ljt} Igual a 1 se o produto j é produzido na linha l durante o período t e 0, caso contrário
- δ_{lt} Igual a 1 se a linha l está montada durante o período t e 0, caso contrário

A função objetivo:

$$\text{minimizar } \sum_{ljtp} hb_{ljtp} x_{ljtp} + \sum_{ljt} sc_{lj} w_{ljt} + \sum_{lt} ac_l \delta_{lt}$$

O objetivo do modelo matemático é minimizar os custos totais de produção. Esses custos totais são obtidos pela soma dos custos de manutenção em estoque e atraso nos atendimentos das demandas, dos custos de preparação para produção das linhas, e por último, dos custos de montagem das linhas de produção. Além da função objetivo, também serão necessárias as restrições para adequar às caracterizações do problema trabalhado.

Restrição 1: Atendimento das demandas

$$\sum_{p=\max\{1, t-sl_j\}}^T \sum_l x_{ljtp} = d_{jt}, \forall j, t$$

Essa primeira restrição assegura que todas demandas de um período t sejam atendidas, podendo acontecer produtos em estoque e atraso na entrega. Para os casos de atraso de entrega, será necessário pagar o custo associado ao atraso, e quando for para os produtos em estoque, além de cobrir os custos de armazenamento, também deverá ter um cuidado no tempo devido ao prazo de validade. Em outras palavras, se uma demanda de um produto j que deve ser produzido no período t e tem validade sl_j , não pode ser produzido nos períodos anteriores a $t - sl_j$, pois os produtos ficarão deteriorados. E em casos particulares em que o resultado da subtração é zero ou negativo, os produtos podem ser produzidos em qualquer período do horizonte de planejamento.

Restrição 2: Capacidade das linhas de produção

$$\sum_{j \in P_{l,p}} a_{lj} x_{ljtp} + \sum_j st_{lj} w_{ljt} \leq C_{lt}, \forall l, t$$

A segunda restrição garante que o tempo total de produção e de preparação para produção não exceda a capacidade produtiva (em tempo) de uma linha l num período t . Primeira somatório calcula o tempo total usado para a produção de produtos j que podem ser e que foram produzidos na linha l , independente para a demanda de qual período p . Segunda somatório calcula o tempo de preparação para a produção do produto j caso ele foi utilizado na linha l durante o período t . E assim, fixado cada par de l e t , a adição das duas somatórias nunca excederá a capacidade produtiva em tempo.

Restrição 3: A produção só pode ocorrer após a preparação da respectiva linha

$$\sum_p x_{ljtp} \leq \frac{C_{lt}}{a_{lj}} w_{ljt}, \forall l, j \in P_l, t$$

Dado $w_{ljt} = 0$, os itens não poderão ser produzidos devido à falta de preparação, e assim, é trivial que a desigualdade estará em conformidade. Agora quando $w_{ljt} = 1$, deve-se assegurar que a parte direita é maior sempre, e de fato, pode-se chegar em tal conclusão a partir da dedução da desigualdade na restrição 2.

Restrição 4: A produção só pode ocorrer em linhas ativadas

$$\sum_{j \in P_l} w_{ljt} \leq \gamma \delta_{lt}, \forall l, t$$

Quando δ_{lt} é 0, ou seja, quando a linha l não está montada durante o período t , nenhum produto j será produzido na linha l e período t . E quando contrário, todos os produtos que podem ser reproduzidos nessa linha não poderão exceder um limitante superior γ .

Restrição 5: Capacidade dos recursos produtivos necessários para ativação das linhas

$$\sum_l r_{kl} \delta_{lt} \leq R_k, \forall k, t$$

Dado um tipo de recurso k e um período t , seja todas as linhas montadas nesse período, a somatória de todos recursos do tipo k necessário para as linhas de produção não poderá exceder a quantidade disponível desse recurso.

Restrição 6: Controle do prazo de validade

$$x_{ljtp} = 0, \forall l, j, p, t < \max\{1, p - sl_j\}$$

Fixado um produto do tipo j que deve atender uma demanda no período p que tem prazo de validade sl_j . Então, ele só pode ser produzido a partir do período $\max\{1, p - sl_j\}$, sendo assim, todas as variáveis de decisão x_{ljtp} com $t < p - sl_j$ serão iguais a 0.

Restrição 7: Domínio das variáveis de decisão

$$x_{ljtp} \geq 0, \forall l, j, p, t$$

Define uma quantidade maior ou igual a zero de produtos do tipo j produzidos na linha l durante o período t para atender a demanda no período p .

Restrição 8: Domínio das variáveis de decisão

$$w_{ljt} \in \{0, 1\}, \forall l, j, t$$

Igual a 1 se o produto j é produzido na linha l durante o período t e 0, caso contrário.

Restrição 9: Domínio das variáveis de decisão

$$\delta_{lt} \in \{0, 1\} \forall l, t$$

Igual a 1 se a linha l está montada durante o período t e 0, caso contrário.

2.2 Considerações Finais

Neste Capítulo, apresentamos o problema de dimensionamento de lotes estudado e um modelo matemático que representa o contexto do ambiente industrial descrito. Além das características do problema, os parâmetros, a função objetivo e suas restrições também foram contempladas e detalhadas.

Capítulo 3

Relaxação Lagrangiana

Os recentes avanços em informática, com a construção de equipamentos mais rápidos e confiáveis, proporcionaram o desenvolvimento de sistemas mais robustos para programação matemática, permitindo a resolução de problemas de grande porte (Lorena, Pereira e Salomão [15]).

Na teoria da otimização matemática, a dualidade, também conhecida como princípio da dualidade, estabelece que os problemas de otimização podem ser abordados sob duas perspectivas: o problema primordial (ou primal) e o problema dual. A solução do problema dual fornece um limite inferior para a solução do problema primal, que visa a minimização. No entanto, em geral, os valores ótimos dos problemas primal e dual não necessariamente coincidem, resultando em uma diferença conhecida como diferença de dualidade. A seguir, abordaremos sobre os conceitos da Relaxação Lagrangiana de forma resumida conforme apresentado por Arenales, Armentano, Morabito e Yanasse [18].

Considere um problema de otimização linear na forma padrão, que denominamos problema primal:

$$\begin{aligned} \text{Minimizar} \quad & f(x) = c^T x \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

Temos uma função objetivo $f(x)$ que deseja minimizar, formada por combinação linear das variáveis de decisão x , ponderada pelos coeficientes do vetor c^T . E também, uma restrição que para a combinação linear da matriz de coeficientes A com as variáveis de decisão x , que deverá ser igual a vetor de valores constantes b .

Se considerarmos que o vetor b possa sofrer alterações, então a restrição $b - Ax = 0$ não precisa necessariamente ser satisfeita. Podemos, no entanto, analisar como um vetor y que está relacionado a essas alterações. O vetor y pode ser visto como uma medida das alterações no vetor b , de modo que a solução do problema agora pode ser representada como $Ax = b - y$. Vale ressaltar que o sinal de y é irrelevante; em alguns livros de otimização linear, adota-se $+y$ em vez de $-y$, bastando definir $y = Ax - b$.

Agora, suponha que λ_i seja a penalização ou o custo unitário para alterar a i -ésima componente. Dessa forma, $\lambda_i y_i$ é o custo adicional de alterar a i -ésima valor em y_i unidades. Em outras palavras, λ_i representa o custo adicional que você terá que pagar por cada unidade de mudança em y_i . Para aumentar ou diminuir y_i , cada unidade de alteração custará λ_i .

A partir disso, associa-se a um novo problema chamado problema lagrangiano, isto é, para cada $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, resolva:

$$\begin{aligned} \text{Minimizar} \quad & f(x) + \lambda_1 y_1 + \lambda_2 y_2 + \dots + \lambda_m y_m \\ \text{s.a.} \quad & x \geq 0 \end{aligned}$$

A função objetivo do problema lagrangiano é chamada função lagrangiana, e é dada por:

$$L(x, \lambda) = f(x) + \lambda_1 y_1 + \lambda_2 y_2 + \dots + \lambda_m y_m$$

Em que $y = b - Ax$, ou seja,

$$\begin{aligned} L(x, \lambda) &= c^T x + \lambda^T y \\ &= c^T x + \lambda^T (b - Ax) \\ &= (c^T - \lambda^T A)x + \lambda^T b \end{aligned}$$

Seja $A = [a_1, a_2, \dots, a_n]$, em que a_j é a j -ésima coluna da matriz A e $c = (c_1, c_2, \dots, c_n)$, então:

$$(c^T - \lambda^T A) = (c_1 - \lambda^T a_1, c_2 - \lambda^T a_2, \dots, c_n - \lambda^T a_n)$$

De modo que a função lagrangiana pode ser escrita como:

$$L(x_1, \dots, x_n, \lambda) = (c_1 - \lambda^T a_1)x_1 + (c_2 - \lambda^T a_2)x_2 + \dots + (c_n - \lambda^T a_n)x_n + \lambda^T b$$

A função dual é definida por:

$$g(\lambda) = \min_{x \geq 0} \{L(x_1, \dots, x_n, \lambda)\}$$

$$g(\lambda) = \min_{x_1 \geq 0} \{(c_1 - \lambda^T a_1)x_1 + (c_2 - \lambda^T a_2)x_2 + \dots + (c_n - \lambda^T a_n)x_n + \lambda^T b\}$$

$$g(\lambda) = \min_{x_1 \geq 0} \{(c_1 - \lambda^T a_1)x_1\} + \min_{x_2 \geq 0} \{(c_2 - \lambda^T a_2)x_2\} + \dots + \min_{x_n \geq 0} \{(c_n - \lambda^T a_n)x_n\} + \lambda^T b$$

$$g(\lambda) = \lambda^T b + \sum_{j=1}^n \min_{x_j \geq 0} (c_j - \lambda^T a_j)x_j$$

A decomposição anterior na soma de n subproblemas menores é válida, pois as variáveis $x_1, x_2, \dots, x_n$ são independentes entre si.

Agora observe que para cada $j=1, 2, \dots, n$, temos:

$$\min_{x_j \geq 0} (c_j - \lambda^T a_j)x_j = \begin{cases} -\infty, & \text{se } (c_j - \lambda^T a_j) < 0, \quad (x_j \rightarrow \infty) \\ 0, & \text{caso contrário} \quad (x_j = 0) \end{cases}$$

Sempre que a expressão resulta em $-\infty$, temos um limitante inferior inócuo, uma vez que $-\infty$ é um limitante inferior para qualquer número real. E então, para evitar esse tipo de resultado, restringimos λ tal que $(c_j - \lambda^T a_j) \geq 0$. Dessa forma:

$$\min_{x_j \geq 0} (c_j - \lambda^T a_j)x_j = 0$$

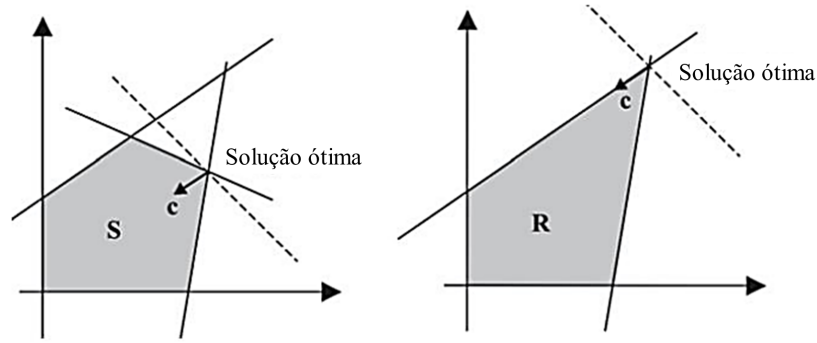
Pois, a solução do problema lagrangiano é dada por:

- Se $c_j - \lambda^T a_j > 0$, então $x_j = 0$ e $(c_j - \lambda^T a_j)x_j = 0$;
- Se $c_j - \lambda^T a_j = 0$, então x_j é qualquer valor não negativo e $(c_j - \lambda^T a_j)x_j = 0$;

A definição da função dual fornece uma desigualdade fundamental da teoria da dualidade, a qual é baseada em uma estratégia de se obter limitantes inferiores (para os problemas de minimização). Esta estratégia é muito comum em otimização e é chamada relaxação.

Suponha que um conjunto R contenha S (R é uma relaxação de S), conforme ilustrado na Figura 3.1.

Figura 3.1: Soluções Ótimas em S e R



Fonte: M. Arenales, V. Armentano, R. Morabito e H. Yanasse (2007)

O mínimo de uma função f em R (conjunto maior) deve ser menor ou igual ao mínimo de f em S . Isso porque, se $x^0 \in S$ é tal que $f(x^0) = \text{mínimo} \{f(x), x \in S\}$, e como $x^0 \in R$ (já que $R \supseteq S$), então x^0 também pode ser um mínimo de f em R ou um outro ponto de R pode ser ainda melhor. Ou seja:

$$\text{Minimizar}\{f(x), x \in R\} \leq \text{Minimizar}\{f(x), x \in S\}, \text{ em que } R \supseteq S$$

Pela definição da função dual e definindo:

$$R = \{x \in \mathbb{R}^n \text{ tal que } x \geq 0\} \supseteq S = \{x \in \mathbb{R}^n \text{ tal que } Ax = b, x \geq 0\}$$

Temos:

$$\begin{aligned} g(\lambda) &= \min_{x \geq 0} c^T x + \lambda^T (b - Ax) \\ &\leq \min_{\{Ax=b, x \geq 0\}} c^T x + \lambda^T (b - Ax) \text{ (O termo } b - Ax \text{ se anula)} \\ &= \min\{c^T x, \text{ sujeito a: } Ax = b, x \geq 0\} \text{ (Problema primal)} \\ &\leq f(x), \text{ para todo } x \text{ tal que } Ax = b, x \geq 0 \end{aligned}$$

Em suma, para todo $\lambda \in \mathbb{R}^m$ e para todo x tal que $Ax = b, x \geq 0$, então $g(\lambda) \leq f(x)$.

Em outras palavras, a função dual $g(\lambda)$ fornece um limitante inferior para a função objetivo primal $f(x)$, para todo x factível. E assim, isso nos segure a encontrar o λ que seria o maior valor dos limitantes inferiores, o que define o problema dual.

O maior limitante inferior para $f(x)$, obtido pela função dual, define o problema dual lagrangiano, ou simplesmente, problema dual, dado por:

$$\text{Maximizar } g(\lambda)$$

$$\lambda \in \mathbb{R}^m$$

As variáveis $(\lambda_1, \lambda_2, \dots, \lambda_m)$ são chamadas variáveis duais.

Considerando a análise anterior do $\min_{x_j \geq 0} (c_j - \lambda^T a_j) x_j$. O maior limitante inferior resulta em $\min_{x_j \geq 0} (c_j - \lambda^T a_j) x_j = 0$. E assim, a função dual pode ser simplificada:

$$g(\lambda) = \lambda^T b + \sum_{j=1}^n \min_{x_j \geq 0} (c_j - \lambda^T a_j) x_j \rightarrow g(\lambda) = \lambda^T b$$

A não-negatividade da variável do problema primal, isto é, $x_j \geq 0$, fez surgir a restrição sobre as variáveis duais $c_j - \lambda^T a_j \geq 0$, ou seja, o problema primal de minimização e o sinal da variável primal contribuíram para que restrições sobre as variáveis duais fossem do tipo $c_j - \lambda^T a_j \geq 0$.

Portanto, se escolhermos λ tal que $c_j - \lambda^T a_j \geq 0$, $j = 1, \dots, n$, então $g(\lambda) = \lambda^T b$. As n desigualdades podem ser escritas em notação matricial:

$$\lambda^T a_1 \leq c_1, \lambda^T a_2 \leq c_2, \dots, \lambda^T a_n \leq c_n \iff \lambda^T A \leq c^T$$

Podemos reescrever a desigualdade acima tomando-se a transposta de ambos os lados:

$$A^T \lambda \leq c$$

Em resumo, se escolhermos as variáveis duais tais que $A^T \lambda \leq c$, então a função dual é dada por $g(\lambda) = \lambda^T b$ (note que $\lambda^T b = \lambda_1 b_1 + \lambda_2 b_2 + \dots + \lambda_m b_m = b^T \lambda$).

De forma genérica, se considerarmos um problema primal:

$$\text{Minimizar } f(x) = c^T x$$

$$\text{s.a. } Ax = b$$

$$x \geq 0$$

Então o problema dual é dado pelo seguinte problema de otimização linear:

$$\text{Maximizar } g(\lambda) = b^T \lambda$$

$$\text{s.a. } A^T \lambda \leq c$$

Os coeficientes da função objetivo dual são dados pelo vetor b do problema primal; o termo independente das restrições duais é dado pelo vetor custo c do problema primal, e a matriz de restrições do problema dual é a transposta da matriz das restrições do problema primal, A^T .

O conjunto de restrições $A^T\lambda \leq c$ é chamado de restrições duais, e todo vetor λ que satisfaça as restrições duais é chamado de solução dual factível.

A Relaxação Lagrangiana é uma técnica de decomposição que funciona de forma semelhante à toda teoria apresentada anteriormente. A diferença é que apenas um subconjunto das restrições do problema será dualizado.

3.1 Construção da Relaxação Lagrangiana para o Modelo Matemático

O modelo matemático apresentado no Capítulo 2, pertence à classe NP-difícil como demonstrado por Medeiros, Soler e Queiroz [16]. Sendo assim, observa-se uma complicação ao resolver instâncias cada vez maiores. O tempo para encontrar soluções ótimas estava cada vez mais distante daquele considerado aceitável.

Dessa forma, a construção da relaxação lagrangiana para esse modelo matemático contribuirá de forma significativa para mais possibilidades de obter soluções viáveis em menos tempo.

As restrições 2 e 5 de acoplamento serão as candidatas principais para aplicar a técnica. A razão para essas escolhas consiste no fato de que elas afetam diretamente na complexidade do problema, pois criam dependência entre as variáveis, dificultando a convergência dos algoritmos de otimização. A expectativa é que ao relaxar elas separadamente ou simultaneamente, as soluções duais sejam encontradas com mais facilidade, e posteriormente, com o método subgradiente, heurísticas de factibilização e melhoria, as soluções de boa qualidade sejam alcançadas.

3.1.1 Aplicação na Restrição 2

A restrição 2 garante que o tempo total de produção e de preparação para produção não exceda a capacidade produtiva (em tempo) de uma linha l num período t .

$$\sum_{j \in P_{l,p}} a_{lj} x_{ljtp} + \sum_j st_{lj} w_{ljt} \leq C_{jt}, \forall l, t$$

Para dualizar essa restrição, introduzimos o vetor de penalidade β_{lt} para integrar a restrição dualizada à função objetivo, constituindo a função lagrangiana $h(\beta)$.

$$\begin{aligned} & \text{minimizar} \\ h(\beta) = & \begin{matrix} x_{ljtp} \geq 0 \\ w_{ljt} \in \{0, 1\} \\ \delta_{lt} \in \{0, 1\} \end{matrix} \sum_{ljtp} hb_{ljtp} x_{ljtp} + \sum_{ljt} sc_{lj} w_{ljt} + \sum_{lt} ac_l \delta_{lt} \\ & - \sum_{lt} \beta_{lt} \left(C_{lt} - \sum_{j \in P_{l,p}} a_{lj} x_{ljtp} - \sum_j st_{lj} w_{ljt} \right) \end{aligned}$$

$$\begin{aligned} & \text{minimizar} \\ h(\beta) = & - \sum_{lt} \beta_{lt} C_{lt} + \begin{matrix} x_{ljtp} \geq 0 \\ w_{ljt} \in \{0, 1\} \\ \delta_{lt} \in \{0, 1\} \end{matrix} \sum_{ljtp} hb_{ljtp} x_{ljtp} + \sum_{ljt} sc_{lj} w_{ljt} + \sum_{lt} ac_l \delta_{lt} \\ & + \sum_{lt} \sum_{j \in P_{l,p}} \beta_{lt} a_{lj} x_{ljtp} + \sum_{lt} \sum_j \beta_{lt} st_{lj} w_{ljt} \end{aligned}$$

Com $\beta_{lt} \geq 0$, para $\forall l, t$

E sujeito à todas restrições do modelo original, exceto a restrição 2 dualizada. O problema dual lagrangiano, que busca determinar o melhor vetor de penalidades para obter a solução ótima do problema, é dado por:

$$\begin{aligned} & \text{Maximizar} \quad h(\beta) \\ & \text{s.a.} \quad \beta_{lt} \geq 0 \end{aligned}$$

3.1.2 Aplicação na Restrição 5

A restrição 5 garante que para cada tipo de recurso k em todo período t , a somatória de todos recursos do tipo k necessário para as linhas de produção não irá exceder a quantidade disponível desse recurso.

$$\sum_l r_{kl} \delta_{lt} \leq R_k, \forall k, t$$

Para dualizar essa restrição, introduzimos o vetor de penalidade λ_{kt} para integrar a restrição dualizada à função objetivo, constituindo a função lagrangiana $g(\lambda)$.

$$g(\lambda) = \begin{matrix} \text{minimizar} \\ x_{ljtp} \geq 0 \\ w_{ljt} \in \{0, 1\} \\ \delta_{lt} \in \{0, 1\} \end{matrix} \sum_{ljtp} h b_{ljtp} x_{ljtp} + \sum_{ljt} s c_{lj} w_{ljt} + \sum_{lt} a c_l \delta_{lt} - \sum_{kt} \lambda_{kt} \left(R_k - \sum_l r_{kl} \delta_{lt} \right)$$

$$g(\lambda) = - \sum_{kt} \lambda_{kt} R_k + \begin{matrix} \text{minimizar} \\ x_{ljtp} \geq 0 \\ w_{ljt} \in \{0, 1\} \\ \delta_{lt} \in \{0, 1\} \end{matrix} \sum_{ljtp} h b_{ljtp} x_{ljtp} + \sum_{ljt} s c_{lj} w_{ljt} + \sum_{lt} a c_l \delta_{lt} + \sum_{kt} \sum_l \lambda_{kt} r_{kl} \delta_{lt}$$

Com $\lambda_{kt} \geq 0$, para $\forall k, t$

E sujeito à todas restrições do modelo original, exceto a restrição 5 dualizada.

O problema dual lagrangiano, que busca determinar o melhor vetor de penalidades para obter a solução ótima do problema, é dado por:

$$\text{Maximizar } g(\lambda)$$

$$\text{s.a. } \lambda_{kt} \geq 0$$

3.1.3 Aplicação na Restrição 2 e 5 Simultaneamente

Agora, para dualizar simultaneamente as restrições 2 e 5, adicionaremos os vetores de penalidades β_{lt} e λ_{kt} para constituir a função lagrangiana $p(\beta, \lambda)$.

$$\begin{aligned}
 & \text{minimizar} \\
 p(\beta, \lambda) = & \begin{matrix} x_{ljtp} \geq 0 \\ w_{ljt} \in \{0, 1\} \\ \delta_{lt} \in \{0, 1\} \end{matrix} \sum_{ljtp} h b_{ljtp} x_{ljtp} + \sum_{ljt} s c_{lj} w_{ljt} + \sum_{lt} a c_l \delta_{lt} \\
 & - \sum_{lt} \beta_{lt} \left(C_{lt} - \sum_{j \in P_{l,p}} a_{lj} x_{ljtp} - \sum_j s t_{lj} w_{ljt} \right) - \sum_{kt} \lambda_{kt} \left(R_k - \sum_l r_{kl} \delta_{lt} \right) \\
 & \text{minimizar} \\
 p(\beta, \lambda) = & - \sum_{lt} \beta_{lt} C_{lt} - \sum_{kt} \lambda_{kt} R_k + \begin{matrix} x_{ljtp} \geq 0 \\ w_{ljt} \in \{0, 1\} \\ \delta_{lt} \in \{0, 1\} \end{matrix} \sum_{ljtp} h b_{ljtp} x_{ljtp} + \sum_{ljt} s c_{lj} w_{ljt} + \sum_{lt} a c_l \delta_{lt} \\
 & + \sum_{lt} \sum_{j \in P_{l,p}} \beta_{lt} a_{lj} x_{ljtp} + \sum_{lt} \sum_j \beta_{lt} s t_{lj} w_{ljt} + \sum_{kt} \sum_l \lambda_{kt} r_{kl} \delta_{lt}
 \end{aligned}$$

Com $\beta_{lt} \geq 0$, para $\forall l, t$

Com $\lambda_{kt} \geq 0$, para $\forall k, t$

E sujeito à todas restrições do modelo original, exceto as restrições 2 e 5 dualizadas.

O problema dual lagrangiano, que busca determinar os melhores vetores de penalidades para obter a solução ótima do problema, é dado por:

$$\begin{aligned}
 & \text{Maximizar} \quad p(\beta, \lambda) \\
 & \text{s.a.} \quad \beta_{lt} \geq 0 \\
 & \quad \quad \lambda_{kt} \geq 0
 \end{aligned}$$

3.2 Método Subgradiente

Os problemas duais lagrangianos construídos anteriormente apresentam uma complexidade elevada, o que dificulta sua implementação simples como um modelo matemático a ser solucionado por métodos convencionais. Uma vez que o problema busca maximizar uma função que contém uma minimização, torna-se incoerente para um solver de otimização processar e gerar resultados. Dessa forma, será necessário utilizar métodos subgradientes para obter soluções.

Na otimização diferenciável, os métodos mais conhecidos são os Métodos Gradientes e o Método de Newton. Embora sejam técnicas clássicas e amplamente utilizadas, elas são inviáveis para o nosso problema, que envolve a minimização de uma função não diferenciável. Contudo, o que se pode fazer é adaptar eles. Por exemplo, os Métodos Gradientes podem ser modificados para funções não diferenciáveis utilizando a direção oposta a um subgradiente como a direção de busca [27]. Os Métodos Subgradientes elaborados por Shor [26] foram os primeiros métodos de otimização não diferenciável.

A parte mais difícil de um método subgradiente está em determinar com precisão o passo em cada iteração, o qual atualiza o vetor de penalidades. Passos muito grande fazem com que o algoritmo divirja, distanciando da solução almejada. Já com passos muito pequeno, a convergência é retardada e pode chegar em uma estagnação, permanecendo em um mínimo local sem convergir para a solução global.

Na literatura, existem diversos estudos em busca de conjecturas mais sofisticadas para determinar esse passo. Apesar de que não exista ainda uma fórmula que promete um tempo exato ou quantidade de iterações necessárias para alcançar a solução ótima, mas pode-se encontrar critérios que garantem a convergência, como apresentada em Wolsey (1998) [13], podemos usar uma série divergente.

- Seja α_i o tamanho do passo na i -ésima iteração, se $\sum_i \alpha_i \rightarrow \infty$ e $\alpha_i \rightarrow 0$ quando $i \rightarrow \infty$, então $g(p^i) \rightarrow g(p^*)$, sendo $g(p^i)$ o valor da função dual Lagrangiano na i -ésima iteração e $g(p^*)$ o valor ótimo do problema dual Lagrangiano.

O critério é bastante relevante para evitar tentativas com um conjunto de passos que não forme uma série divergente. Todavia, mesmo que sua importância seja verificada, mas a convergência pode ser lenta.

Além do mais, outras regras são usualmente adotada.

1. Seja $\alpha_i = (\delta)^i \alpha_0$ para algum $\delta < 1$. Se α_0 (tamanho do passo inicial) e δ forem suficientemente grandes, então $g(p^i) \rightarrow g(p^*)$.
2. Seja $0 < \omega_i < 2$, LS o limitante superior para $g(p^*)$ e s^i o vetor subgradiente, temos:

$$\alpha_i = \omega_i \frac{(LS - g(p^i))}{\|s^i\|^2}$$

Quando $g(p^{i+1}) \leq g(p^i)$ e $\delta < 1$, pode-se realizar $\omega^{i+1} = \delta \omega^i$.

Teoricamente, a partir das regras expostas, a condição para obter o ponto ótimo será quando $s^i = 0$, o que representa que não há mais necessidade de dar passos, já se alcançou a otimalidade da solução.

Entretanto, a condição é raramente atingida. Logo, critérios de parada mais práticos são utilizados:

- **Duração Máxima:** O algoritmo é interrompido após um tempo de execução predefinido $T_{\text{máx}}$.

$$\text{Tempo de execução} \leq T_{\text{máx}}$$

- **Número Máximo de Iterações:** O algoritmo é interrompido após um número máximo de iterações $N_{\text{máx}}$.

$$\text{Número de iterações} \leq N_{\text{máx}}$$

- **Critério de Convergência da Função Objetivo:** O algoritmo para quando a mudança na função objetivo entre iterações consecutivas é menor do que um valor predefinido ϵ_{tol} , indicando que a melhoria na solução está abaixo de um limiar de significância.

$$|f(x^{k+1}) - f(x^k)| \leq \epsilon_{\text{tol}}$$

Aqui, $f(x^k)$ representa o valor da função objetivo na k -ésima iteração, e ϵ é um parâmetro de tolerância pequeno.

3.2.1 Algoritmo para Relaxar Restrição 2

Agora, construiremos o algoritmo para relaxar a restrição 2. Para tanto, utilizamos o comando `def( )` em *Python* para definir os blocos de códigos independentes da função Lagrangiana e das heurísticas de factibilização e de melhoria que serão apresentadas com mais detalhes na próxima seção. Por meio dessa ferramenta, em toda iteração do método subgradiente, poderemos resolver e obter o valor da função Lagrangiana a partir do vetor de penalidades, e também, o resultado das heurísticas junto com os valores pós-soluções das variáveis de decisão.

Já para descrever o método subgradiente, iremos determinar primeiramente os valores do subgradiente S_{lt} .

$$S_{lt} = \sum_{j \in P_{l,p}} a_{lj} \bar{x}_{ljtp} + \sum_j st_{lj} \bar{w}_{ljt} - C_{lt}, \forall l, t$$

Os valores de $\bar{x}_{ljtp}$ e $\bar{w}_{ljt}$ são aqueles obtidos pelo retorno da função lagrangiana. Com isso, atualizaremos os valores das penalidades (β_{lt}) em cada iteração.

$$\beta_{lt}^* = \max\{0, \beta_{lt} + eps S_{lt}\} \quad \forall l, t$$

Os novos β_{lt} são definidos dessa forma, pois não admitem valores negativos e devem seguir na direção do subgradiente em busca de solução ótima. O *eps*, abreviação de *Error Por Step* em inglês, significa erro por passo, ou melhor, o tamanho do passo para atualizar as variáveis de penalidade.

O tamanho do passo na i -ésima iteração será determinado por:

$$eps_i = \omega_i \frac{(LS - h(\beta^i))}{||S_{lt}^i||^2}$$

O LS representa o limitante superior, a melhor solução primal obtida durante as iterações. O $h(\beta^i)$ representa o valor da função lagrangiana na i -ésima iteração. O S_{lt}^i representa o valor do subgradiente na i -ésima iteração. O ω_i é um parâmetro auxiliar que tem valor inicial 2 e atualizamos a cada 5 iterações da seguinte maneira:

$$\omega_i^* = \omega_i \cdot 0.8$$

Em relação ao critério de parada, para as instâncias da classe A, iremos iterar por 180 segundos (3 minutos). Para as instâncias de classe B, delimitamos por 900 segundos (15 minutos). Já para as instâncias da classe C, D e E, estabelecemos um tempo limite de 3600 segundos (1 hora). Essas classes serão detalhadas na seção 5.1.

Algoritmo 1: Método Subgradiente Relaxando Restrição 2

$$\beta_{lt} = 0 \quad \forall l, t$$

$$i = 0$$

$$\omega_i = 2$$

$$LS = 10^{100}$$

$$LI = 0$$

$$TS = \text{Tempo de Solução}$$

Enquanto $TS < 180$ **faça**

$$i = i + 1$$

Cria e reseta dicionários para armazenar valores de x_{ljtp} , w_{ljt} e δ_{lt}

Resolve função lagrangiana $h(\beta^i)$ e obtém os valores de x_{ljtp} , w_{ljt} e δ_{lt}

Criar cópias dos dicionários que armazenam x_{ljtp} , w_{ljt} e δ_{lt} (x_{ljtp}^c , w_{ljt}^c e δ_{lt}^c)

if $h(\beta^i) > LI$

$$LI \leftarrow h(\beta^i)$$

Resolve as heurísticas e obtém os valores de x_{ljtp} , w_{ljt} e δ_{lt}

if $HEURS < LS$

$$LS \leftarrow HEURS$$

Criar cópia do dicionário que armazena β_{lt} (β_{lt}^c)

$$x_{ljtp} \leftarrow x_{ljtp}^c$$

$$w_{ljt} \leftarrow w_{ljt}^c$$

$$\delta_{lt} \leftarrow \delta_{lt}^c$$

Determinando o subgradiente $S_{lt}^i = \sum_{j \in P_{l,p}} a_{lj} x_{ljtp} + \sum_j st_{lj} w_{ljt} - C_{lt}, \forall l, t$

Determinando o passo $eps_i = \omega_i \frac{(LS - h(\beta^i))}{\|S_{lt}^i\|^2}$

if $i \% 5 == 0$

$$\omega_i \leftarrow \omega_i \cdot 0.8$$

Determinando o $GAP = 100 \frac{LS - LI}{LS}$

if $i == 1$

$$GAP_{inicial} = GAP$$

Determinando os β_{lt} para próxima iteração $\beta_{lt}^* = \max\{0, \beta_{lt} + eps_i S_{lt}^i\} \quad \forall l, t$

Retorna i , LS , LI , $GAP_{inicial}$, GAP , TS e β_{lt}^c

O algoritmo 1 apresenta um pseudocódigo para compreender melhor o método subgradiente aplicado nos testes das instâncias da classe A, o qual será semelhante para os testes de outras classes. Para começar, é definido alguns parâmetros iniciais para ajudar nas estruturas de repetição posteriores. Dentre elas, o limitante inferior (LI) é o maior valor obtido pelos resultados da função lagrangiana, ou seja, a melhor solução dual. Já o limitante superior (LS) é o menor valor obtido pelos resultados das heurísticas, ou seja, a melhor solução primal. Sendo assim, seus valores iniciais foram definidos de forma que o LS seja um valor extremamente alto, e o LI basta ser nulo.

Em seguida, temos uma estrutura de repetição, conhecida como *loop* lagrangiano. Nela, inicia-se com a solução da função lagrangiana ($h(\beta^i)$), e depois, das heurísticas ($HEURS$), obtendo os valores das variáveis de decisão em ambos os casos. Destacamos que para atualizar o vetor de penalidades depois, devemos utilizar aquelas obtidas na função lagrangiana, e não a das heurísticas. Entretanto, quando solucionamos $h(\beta^i)$ e $HEURS$, eles retornam e atualizam os valores das variáveis de decisão automaticamente. Assim, foi preciso armazenar cópias dos valores obtidos pela $h(\beta^i)$, e depois de proceder as $HEURS$, substituímos as variáveis obtidas pelas armazenadas.

Para realizar a atualização do parâmetro auxiliar ω_i , utilizamos o comando de operação básica `%` em *Python* que fornece o resto da divisão entre dois números. Assim, quando escrevemos `if i % 5 == 0`, a estrutura condicional só será executada se o i for múltiplo de 5, ou seja, atualizaremos a cada 5 iterações como pretendido.

As informações retornadas são de grande relevância. O i é o número de iterações. O LS e LI são, respectivamente, as melhores soluções primais e duais. O $GAP_{inicial}$ e GAP servem para analisar o aprimoramento da qualidade das soluções ao longo do método. O TS é o tempo de solução que poderá ser superior ao tempo delimitado, pois este só é avaliado no começo de cada iteração. E, por último, o β_{it}^c que é o melhor vetor de penalidades alcançado, aquele que gera o LS .

Nos testes com as instâncias de classe C, D e E, foi necessário limitar o tempo para solucionar a função lagrangiana em 500 segundos e a heurística de melhoria em 300 segundos, uma vez que foi observado uma duração muito elevada para resolvê-las.

3.2.2 Algoritmo para Relaxar Restrição 5

Agora, na construção do algoritmo para relaxar Restrição 5, para descrever o método subgradiente, iremos determinar primeiramente os valores do subgradiente S_{kt} .

$$S_{kt} = \sum_l r_{kl} \bar{\delta}_{lt} - R_k \quad \forall k, t$$

O valor de $\bar{\delta}_{lt}$ é aquele obtido pelo retorno da função lagrangiana $g(\lambda)$. Com isso, atualizaremos os valores das penalidades (λ_{kt}) em cada iteração.

$$\lambda_{kt}^* = \max\{0, \lambda_{kt} + \text{eps} S_{kt}\} \quad \forall k, t$$

E assim, o tamanho do passo (eps) na i -ésima iteração será determinado por:

$$\text{eps}_i = \omega_i \frac{(LS - g(\lambda^i))}{\|S_{kt}^i\|^2}$$

O valor inicial do parâmetro auxiliar w_i que era 2 na seção anterior será agora 0.1, e também, de forma idêntica à relaxação da restrição 2, iremos reduzir em 20% a cada 5 iterações. Esse valor inicial para w_i foi decidido de forma empírica.

Quando o problema é pouco simplificado, notamos que as soluções duais obtidas pelas funções lagrangianas não é tão distante da solução ótima, e assim, o quociente que acompanha o parâmetro auxiliar fornece um número menor. Logo, é recomendável um valor inicial maior para ω_i , pois, caso contrário, os passos eps serão tão pequenos que não haverá aprimoramento significativo nas buscas iterativas. Em contrapartida, se o problema é bastante simplificado, as soluções duais é distante da solução ótima e o quociente irá fornecer um número maior. Logo, é recomendável que o valor inicial para ω_i seja menor, pois, caso contrário, o eps fica tão grande que poderá ultrapassar a solução ótima, e também, observamos que o tempo para solução da função lagrangiana nas próximas iterações é aumentado inesperadamente.

Dessa forma, a escolha de um ω_i com um valor inicial reduzido é porque através de alguns pequenos testes, foi evidenciado que ao relaxar a restrição 5, o problema fica mais fácil em comparação ao seção anterior quando relaxamos a restrição 2.

Em relação ao tempo, decidimos estabelecer as mesmas durações concedidas na subseção anterior. Para classe A, 180 segundos (3 minutos). Para classe B, 900 segundos (15 minutos). Para classe C, D e E, 3600 segundos (1 hora). No algoritmo 2, observe o pseudocódigo para compreender melhor o método subgradiente para relaxar restrição 5.

Algoritmo 2: Método Subgradiente Relaxando Restrição 5

 $\lambda_{kt} = 0 \quad \forall k, t$ $i = 0$ $\omega_i = 0.1$ $LS = 10^{100}$ $LI = 0$ $TS = \text{Tempo de Solução}$ **Enquanto** $TS < 180$ **faça** $i = i + 1$ Cria e reseta dicionários para armazenar valores de x_{ljtp} , w_{ljt} e δ_{lt} Resolve função lagrangiana $g(\lambda^i)$ e obtém os valores de x_{ljtp} , w_{ljt} e δ_{lt} Criar cópias dos dicionários que armazenam x_{ljtp} , w_{ljt} e δ_{lt} (x_{ljtp}^c , w_{ljt}^c e δ_{lt}^c)**if** $g(\lambda^i) > LI$ $LI \leftarrow g(\lambda^i)$ Resolve as heurísticas ($HEURS$) e obtém os valores de x_{ljtp} , w_{ljt} e δ_{lt} **if** $HEURS < LS$ $LS \leftarrow HEURS$ Criar cópia do dicionário que armazena λ_{lt} (λ_{lt}^c) $x_{ljtp} \leftarrow x_{ljtp}^c$ $w_{ljt} \leftarrow w_{ljt}^c$ $\delta_{lt} \leftarrow \delta_{lt}^c$ Determinando o subgradiente $S_{kt}^i = \sum_l r_{kl} \delta_{lt} - R_k \quad \forall k, t$ Determinando o passo $eps_i = \omega_i \frac{(LS - g(\lambda^i))}{\|S_{kt}^i\|^2}$ **if** $i \% 5 == 0$ $\omega_i \leftarrow \omega_i \cdot 0.8$ Determinando o $GAP = 100 \frac{LS - LI}{LS}$ **if** $i == 1$ $GAP_{inicial} = GAP$ Determinando os λ_{kt} para próxima iteração $\lambda_{kt}^* = \max\{0, \lambda_{kt} + eps_i S_{kt}^i\} \quad \forall k, t$ Retorna i , LS , LI , $GAP_{inicial}$, GAP , TS e λ_{kt}^c

3.2.3 Algoritmo para Relaxar Restrição 2 e 5 Simultaneamente

Agora, para construir o algoritmo para relaxar Restrição 2 e 5 simultaneamente, precisamos utilizar ambos os subgradientes S_{lt} e S_{kt} definidos nas seções anteriores.

$$S_{lt} = \sum_{j \in P_{l,p}} a_{lj} \bar{x}_{ljtp} + \sum_j st_{lj} \bar{w}_{ljt} - C_{lt}, \forall l, t$$

$$S_{kt} = \sum_l r_{kl} \bar{\delta}_{lt} - R_k \forall k, t$$

Os valores de $\bar{x}_{ljtp}$, $\bar{w}_{ljt}$ e $\bar{\delta}_{lt}$ são aqueles obtidos pelo retorno da função lagrangiana $p(\beta, \lambda)$. Com isso, atualizaremos os valores das penalidades (β_{lt} e λ_{kt}) em cada iteração.

$$\lambda_{kt}^* = \max\{0, \lambda_{kt} + \text{eps}^\lambda S_{kt}\} \quad \forall k, t$$

$$\beta_{lt}^* = \max\{0, \beta_{lt} + \text{eps}^\beta S_{lt}\} \quad \forall l, t$$

E os tamanhos dos passos (eps^β e eps^λ) na i -ésima iteração serão determinados por:

$$\text{eps}_i^\beta = \omega_i^\beta \frac{(LS - p(\beta^i, \lambda^i))}{\|S_{lt}^i\|^2} \quad \text{eps}_i^\lambda = \omega_i^\lambda \frac{(LS - p(\beta^i, \lambda^i))}{\|S_{kt}^i\|^2}$$

Para melhor comparação dos resultados, utilizaremos as mesmas durações para os testes das instâncias de cada classe. Para classe A, 180 segundos (3 minutos). Para classe B, 900 segundos (15 minutos). Para classe C, D e E, 3600 segundos (1 hora).

Em comparação com as aplicações da Relaxação Lagrangiana somente na restrição 2 e na restrição 5, podemos presumir que ao relaxar 2 e 5 simultaneamente, simplificaremos ainda mais o problema. Logo, a convergência do método subgradiente também será mais lenta, ou seja, torna muito desafiador determinar os melhores vetores de penalidades.

Apesar de que nas instâncias maiores, temos a garantia de que a função lagrangiana seja resolvida em menos tempo, mas a heurística de factibilização precisará de uma duração grande para encontrar respostas. E então, será necessário aplicar novamente a mudança na função objetivo do modelo heurístico, de forma idêntica àquela realizada na seção anterior.

No algoritmo 3, considerando os testes com as instâncias de classe A, apresentamos um pseudocódigo para compreender melhor o método subgradiente para relaxar as restrições 2 e 5 simultaneamente.

Algoritmo 3: Método Subgradiente Relaxando Restrição 2 e 5 Simultâneamente

$$\beta_{lt} = 0 \quad \forall l, t \quad \text{e} \quad \lambda_{kt} = 0 \quad \forall k, t$$

$$i = 0$$

$$\omega_i^\beta = 2 \quad \text{e} \quad \omega_i^\lambda = 0.1$$

$$LS = 10^{100}$$

$$LI = 0$$

$$TS = \text{Tempo de Solução}$$

Enquanto $TS < 180$ **faça**

$$i = i + 1$$

Cria e reseta dicionários para armazenar valores de x_{ljtp} , w_{ljt} e δ_{lt}

Resolve função lagrangiana $p(\beta, \lambda)$ e obtém os valores de x_{ljtp} , w_{ljt} e δ_{lt}

Criar cópias dos dicionários que armazenam x_{ljtp} , w_{ljt} e δ_{lt} (x_{ljtp}^c , w_{ljt}^c e δ_{lt}^c)

if $p(\beta, \lambda) > LI$

$$LI \leftarrow p(\beta, \lambda)$$

Resolve as heurísticas ($HEURS$) e obtém os valores de x_{ljtp} , w_{ljt} e δ_{lt}

if $HEURS < LS$

$$LS \leftarrow HEURS$$

Criar cópia do dicionário que armazena β_{lt} (β_{lt}^c)

Criar cópia do dicionário que armazena λ_{kt} (λ_{kt}^c)

$$x_{ljtp} \leftarrow x_{ljtp}^c ; \quad w_{ljt} \leftarrow w_{ljt}^c \quad \text{e} \quad \delta_{lt} \leftarrow \delta_{lt}^c$$

$$\text{Determinando o subgradiente } S_{lt}^i = \sum_{j \in P_{l,p}} a_{lj} x_{ljtp} + \sum_j st_{lj} w_{ljt} - C_{lt}, \forall l, t$$

$$\text{Determinando o subgradiente } S_{kt}^i = \sum_l r_{kl} \delta_{lt} - R_k \quad \forall k, t$$

$$\text{Determinando os passos } eps_i^\beta = \omega_i^\beta \frac{(LS - p(\beta^i, \lambda^i))}{\|S_{lt}^i\|^2} \quad \text{e} \quad eps_i^\lambda = \omega_i^\lambda \frac{(LS - p(\beta^i, \lambda^i))}{\|S_{kt}^i\|^2}$$

if $i \% 5 == 0$

$$\omega_i^\beta \leftarrow \omega_i^\beta \cdot 0.8$$

$$\omega_i^\lambda \leftarrow \omega_i^\lambda \cdot 0.8$$

$$\text{Determinando o } GAP = 100 \frac{LS - LI}{LS}$$

if $i == 1$

$$GAP_{inicial} = GAP$$

Determinando os β_{lt} para próxima iteração $\beta_{lt}^* = \max\{0, \beta_{lt} + eps_i^\beta S_{lt}^i\} \quad \forall l, t$

Determinando os λ_{kt} para próxima iteração $\lambda_{kt}^* = \max\{0, \lambda_{kt} + eps_i^\lambda S_{kt}^i\} \quad \forall k, t$

Retorna i , LS , LI , $GAP_{inicial}$, GAP , TS , λ_{kt}^c e β_{lt}^c

3.3 Considerações Finais

Neste capítulo, apresentamos inicialmente sobre os conceitos da Relaxação Lagrangiana, e posteriormente, aplicamos a técnica de 3 formas distintas ao modelo matemático que representa o problema estudado. São novas abordagens para o PDL-PPRE. Após a descrição sobre a busca iterativa do método subgradiente para otimizações não diferenciáveis, apresentamos os algoritmos construídos para cada caso de aplicação.

Capítulo 4

Heurísticas Desenvolvidas

As Heurísticas são métodos ou técnicas que procuram encontrar soluções aceitáveis para os problemas complexos de maneira eficiente e prática, mesmo que não assegurem a obtenção da solução ótima. Considerando um espaço muito amplo de soluções, uma boa escolha é dispensar a necessidade da otimalidade, e partir para buscar, dentro do espaço de soluções viáveis, por soluções que sejam próximas ou tão boas quanto as ótimas.

Segundo Zanaki e Evans [32], heurísticas são procedimentos simples, baseados no bom senso, que se supõe ofereçam uma boa solução (ainda que não necessariamente ótima) a problema difíceis, de um modo fácil e rápido.

As heurísticas podem ser classificadas segundo o modo com que buscam e constroem suas soluções. A seguinte classificação é baseada em Silver, Vidal e de Werra [21].

1. Métodos Construtivos: gradualmente adicionam componentes individuais à solução até que se obtenham uma solução factível.
2. Métodos de Decomposição: dividem o problema em subproblemas menores, sendo a saída de um problema a entrada do problema seguinte, de forma que ao resolvê-los, todos obtenha-se uma solução para o problema global (*divide and conquer*).
3. Métodos de Redução: identificam alguma característica que presumivelmente deva aparecer na solução ótima para que desse modo seja possível a simplificação do problema.
4. Manipulação do Modelo: modificam a estrutura do modelo com a finalidade de fazê-lo mais fácil de resolver, deduzindo, a partir de sua solução, a solução do problema original.

5. Métodos de Busca em Vizinhança: partem de uma solução factível inicial (obtida por alguma outra heurística) e mediante alterações nessa solução, passa-se sucessivamente a uma nova solução, de forma iterativa, até que se satisfaça algum critério de parada.

Em 1981, uma aplicação clássica de heurísticas em relaxação lagrangiana é apresentada por Fisher [7], onde são utilizadas heurísticas para melhorar as soluções obtidas na resolução do problema do caixeiro-viajante (em inglês, Travelling Salesman Problem - TSP) com restrições de capacidade. Neste trabalho, heurísticas de busca local são aplicadas para refinar as soluções intermediárias, resultando em uma abordagem eficaz para grandes instâncias do problema.

Em 2005, Santos e Armentano [25] abordaram o problema de dimensionamento de lotes em um sistema de produção multiestágio com máquinas paralelas e capacidades limitadas. Aplicando também a técnica de Relaxação Lagrangiana aliado ao método subgradiente, um método heurístico básico e suas variantes, incluindo estratégias de Busca Tabu, foram propostos.

Em 2018, Soler [28] abordou sobre a relevância das heurísticas lagrangianas. São procedimentos que conseguem determinar, a cada iteração, uma solução factível para o problema original a partir da solução dual incumbente. E assim, o conhecimento de uma solução primal aprimora a atualização do tamanho do passo a ser dado na direção da busca no método subgradiente.

Em 2020, Catelan, Araujo, Fiorotto e de Carvalho [4] abordaram o problema de dimensionamento de lotes com máquinas paralelas flexíveis. Foram criadas três heurísticas baseadas nas ideias de heurísticas *relax-and-fix* e *fix-and-optimize*.

A necessidade dos procedimentos heurísticos no nosso trabalho é motivada pelas soluções duais obtidas em cada iteração do método do subgradiente. Para obter soluções primais viáveis, são necessárias heurísticas de factibilização. E para alcançar resultados mais satisfatórios, são essenciais as heurísticas de melhoria.

As diversas heurísticas desenvolvidas para problemas semelhantes ao modelo estudado neste trabalho servem como base para o desenvolvimento de heurísticas específicas para o nosso problema. Entretanto, devido às características particulares do PDL abordado, as heurísticas apresentadas posteriormente não serão adaptações de formulações existentes, mas sim desenvolvimentos originais especificamente projetados para este problema.

4.1 Heurísticas de Factibilização

4.1.1 Relaxação da Restrição 5

A primeira heurística desenvolvida foi para factibilizar as soluções duais obtidas durante a relaxação da restrição 5.

$$\sum_l r_{kl} \delta_{lt} \leq R_k, \forall k, t$$

Quando dualizamos essa restrição, a função lagrangiana fornecerá soluções duais em que as capacidades de recursos serão excedidas. Dessa forma, um subconjunto de linhas de produção deverá ser fechado para atender novamente a restrição relaxada. Todavia, após fechar linhas de produção, a restrição 1 sobre o atendimento das demandas deixará de ser cumprida. E assim, será necessário abrir outras linhas de produção para atender as demandas, sem descumprir mais qualquer outra restrição.

Agora, a dificuldade reside em como determinar quais são as melhores linhas de produção para fechar, e depois, quais deveríamos abrir. Para superar tal obstáculo, pensamos em criar um simples modelo matemático para fechar o mínimo de linhas de produção possível.

Introduzindo a variável binária γ_{lt}^f , e compreendendo $\bar{\delta}_{lt}$ como variáveis com valores já determinados pela solução dual, o modelo matemático para determinar mínimo de linhas que devem ser fechadas pode ser expresso como:

$$\begin{aligned} & \text{Maximizar} && \sum_{lt} \gamma_{lt}^f \\ \text{s.a} &&& \sum_l r_{kl} \bar{\delta}_{lt} \gamma_{lt}^f \leq R_k, \forall k, t \end{aligned}$$

Com a adição da variável binária γ_{lt}^f na restrição da capacidade de recursos, será decidido quais $\bar{\delta}_{lt}$ deverão ser fechadas. Se γ_{lt}^f for igual a 1, significa que a linha de produção l no período t deverá continuar na situação em que se encontra (seja aberta ou fechada). Caso contrário, representa que $\bar{\delta}_{lt}$ deverá ser fechado. Vale salientar que se $\bar{\delta}_{lt}$ for igual a zero, γ_{lt}^f não será nulo.

A maximização da função objetivo significa que desejamos o mínimo de linhas de produção sendo fechadas. Pois, na forma expressa, a solução trivial seria todos γ_{lt}^f serem iguais a 1. E então, para cumprir a restrição, o modelo buscará evitar zerar variáveis γ_{lt}^f .

Após realizar alguns testes computacionais, o modelo se demonstrou bastante promissor. Entretanto, quando analisamos as linhas de produção que deverão ser abertas para cumprir a restrição 1, chegamos em casos em que eram impossíveis de serem solucionados. Apesar de que o modelo criado seleciona perfeitamente o mínimo de linhas de produção que devem ser fechadas, mas não considera as possibilidades futuras das linhas que poderão ser abertas. Logo, pode haver exemplos em que todas as linhas que produzem um determinado produto j sejam fechadas, tornando naturalmente impossível solucionar o problema subsequente.

No entanto, a partir da ideia da criação do modelo anterior, que se teve o sucesso da criação de um modelo matemático satisfatório para factibilizar as soluções duais ao relaxar a restrição 5, o qual analisa as possibilidades não contempladas anteriormente.

Na criação do novo modelo matemático, utilizaremos os mesmos conjuntos e parâmetros do problema original e as variáveis de decisão $\bar{x}_{ljtp}$, $\bar{w}_{ljt}$ e $\bar{\delta}_{lt}$ já têm seus valores pré-determinados pela solução dual. Ademais, introduzimos as seguintes variáveis de decisão novas:

- x_{ljtp}^f Quantidade de produtos do tipo j produzidos na linha l durante o período t para atender a demanda no período p
- w_{ljt}^f Igual a 1 se o produto j é produzido na linha l durante o período t e 0, caso contrário
- δ_{lt}^f Igual a 1 se a linha l está montada durante o período t e 0, caso contrário
- γ_{lt}^f Igual a 1 se a linha l continuará na mesma situação (aberta ou fechada) no período t e 0, caso contrário

A função objetivo do modelo continuará na ideia de maximizar as variáveis γ_{lt}^f em busca de minimizar a quantidade de linhas de produção que serão fechadas. As restrições 1 a 6 do modelo original serão brevemente modificados. Adicionamos γ_{lt}^f multiplicando as variáveis $\bar{x}_{ljtp}$, $\bar{w}_{ljt}$ e $\bar{\delta}_{lt}$, pois se fecharmos uma linha de produção, as variáveis $\bar{x}_{ljtp}$ e $\bar{w}_{ljt}$ também deverão ser zeradas. Por outro lado, também acrescentamos as novas variáveis x_{ljtp}^f , w_{ljt}^f e δ_{lt}^f para considerar as novas aberturas de linhas de produção, definindo os produtos que devem ser produzidos e as respectivas quantidades para atender todas as demandas.

$$\begin{aligned}
& \text{Maximizar} \quad \sum_{lt} \gamma_{lt}^f \\
\text{s.a} \quad & \sum_{p=\max\{1, t-sl_j\}}^T \sum_l (\bar{x}_{ljpt} \gamma_{lp}^f + x_{ljpt}^f) = d_{jt}, \forall j, t \\
& \sum_{j \in P_l, p} a_{lj} (\bar{x}_{ljtp} \gamma_{lt}^f + x_{ljtp}^f) + \sum_l st_{lj} (\bar{w}_{ljt} \gamma_{lt}^f + w_{ljt}^f) \leq C_{lt}, \forall l, t \\
& \sum_p (\bar{x}_{ljtp} \gamma_{lt}^f + x_{ljtp}^f) \leq \frac{C_{lt}}{a_{lj}} (\bar{w}_{ljt} \gamma_{lt}^f + w_{ljt}^f), \forall l, j \in P_l, t \\
& \sum_{j \in P_l} (\bar{w}_{ljt} \gamma_{lt}^f + w_{ljt}^f) \leq \gamma (\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f), \forall l, t \\
& \sum_l r_{kl} (\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f) \leq R_k, \forall k, t \\
& x_{ljtp}^f = 0, \forall l, j, p, t < \max\{1, p - sl_j\} \\
& x_{ljtp}^f = 0, \forall l, j \notin P_l, p, t \\
& \bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f \leq 1, \forall l, t \\
& \bar{w}_{ljt} \gamma_{lt}^f + w_{ljt}^f \leq 1, \forall l, j, t \\
& x_{ljtp}^f \geq 0, \forall l, j, p, t \\
& w_{ljt}^f \in \{0, 1\}, \forall l, j, t \\
& \delta_{lt}^f \in \{0, 1\}, \forall l, t \\
& \gamma_{lt}^f \in \{0, 1\}, \forall l, t
\end{aligned}$$

As restrições lógicas $\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f \leq 1$ e $\bar{w}_{ljt} \gamma_{lt}^f + w_{ljt}^f \leq 1$ foram construídas para evitar que os valores das três variáveis de decisão presentes nas desigualdades sejam todos iguais a 1. Vamos analisar, por exemplo, a penúltima restrição. Se uma linha de produção já é fechada, pode-se considerar a sua abertura. Se uma linha de produção será fechada, pode-se considerar a sua reabertura. Mas, se uma linha de produção está aberta e não será fechada, não poderemos atribuir valor unitário para δ_{lt}^f , pois isso é como se a linha tivesse aberta duas vezes. O raciocínio é de forma análoga para a última restrição.

Uma observação relevante sobre os casos comentados, é que, apesar de ser permitido, será incoerente fechar uma linha de produção em um período ($\bar{\delta}_{lt} \gamma_{lt}^f = 0$), e depois abrir ela novamente ($\delta_{lt}^f = 1$). Pois, isso contraria a finalidade de maximização da função objetivo do modelo.

Aparentemente, o modelo construído apresenta mais variáveis que o modelo original. Porém, foi observado que o tempo para factibilização era bem pequeno. Como o objetivo é manter o máximo de linhas já abertas, isso implica na maioria das variáveis δ_{lt}^f ser zero, e assim, os x_{lntp}^f e w_{lnt}^f associados também irão assumir o valor nulo.

Uma questão interessante que poderá servir como trabalhos futuros será as outras formas de descrever a função objetivo, aproximando mais de soluções melhores. Devido ao tempo reduzido para realizar o nosso trabalho, apenas deixaremos o esboço de algumas ideias, sem resultados computacionais exatos que comprovem todas suas vantagens ou desvantagens.

Por exemplo, se a função objetivo fosse:

$$\text{Minimizar} \quad \sum_{lntp} hb_{lntp} (\bar{x}_{lntp} \gamma_{lt}^f + x_{lntp}^f) + \sum_{lnt} sc_{lnt} (\bar{w}_{lnt} \gamma_{lt}^f + w_{lnt}^f) + \sum_{lt} ac_{lt} (\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f)$$

Seria praticamente resolver o próprio modelo original de uma forma distinta.

Agora, se a função objetivo minimizasse apenas as novas variáveis, teríamos uma complexidade maior, mas, ao mesmo tempo, mais fácil que o modelo mostrado anteriormente. Nesse caso, a solução trivial seria todas as variáveis de decisão serem zeros, o que implica que nenhuma linha de produção precisou ser fechada. Por outro lado, se um subconjunto de linhas devem ser selecionadas para serem abertas, os custos totais de produção dessas deverão ser minimizadas.

$$\text{Minimizar} \quad \sum_{lntp} hb_{lntp} x_{lntp}^f + \sum_{lnt} sc_{lnt} w_{lnt}^f + \sum_{lt} ac_{lt} \delta_{lt}^f$$

Uma última ideia para reescrever a função objetivo é incrementar um parâmetro auxiliar que relaciona cada linha de produção aos valores das variáveis $\bar{x}_{lntp}$ associados.

O parâmetro auxiliar para cada linha de produção l em cada período t é dado por:

$$p_{lt}^a = \sum_{jp} \bar{x}_{lntp}$$

Uma nova função objetivo poderia ser:

$$\text{Maximizar} \quad \sum_{lt} \gamma_{lt}^f p_{lt}^a$$

Nessa última hipótese, visamos manter a abertura das linhas de produção que tenham mais $\bar{x}_{lntp}$ associados. Porque com base em alguns testes, quando fechamos linhas de produção com muito produto associado, a solução primal obtida é de baixa qualidade. Em outras palavras, quando adicionamos os parâmetros auxiliares p_{lt}^a multiplicando as variáveis γ_{lt}^f , é como se cada linha de produção ganhasse um peso de importância com base na quantidade de produtos que está sendo produzido nela. Assim, ao maximizar a função objetivo, as linhas de produção com mais produtos associados terá uma tendência maior de continuar aberta.

Nos testes com as instâncias maiores, notamos que o tempo para solucionar a heurística de factibilização estava muito elevado. Dessa forma, adaptamos e modificamos a última ideia apresentada para construir uma variação da heurística desenvolvida que viabilize factibilizar as soluções duais em tempo aceitável.

Considere P_m^a um conjunto que armazena 25% das linhas de produção abertas com os valores mais altos obtidos pela somatória dos $\bar{x}_{lntp}$ associados. Então, a função objetivo que maximizava a somatória do γ_{lt}^f abrangendo todas as linhas de produção e períodos pode ser reescrito como:

$$\text{Maximizar} \quad \sum_{(l,t) \in P_m^a} \gamma_{lt}^f$$

Em suma, reduzimos a abrangência da maximização para facilitar a factibilização. E, paralelamente, as linhas de produção priorizadas serão aquelas que tenham mais valores de $\bar{x}_{lntp}$ associados. Vale salientar que não é possível limitar o tempo de solução da heurística de factibilização, pois quando o algoritmo é interrompido, as soluções retornadas não são factíveis. Além disso, o parâmetro auxiliar P_m^a é iterável, ou seja, o seu valor pode ser atualizado a cada iteração.

Por fim, vale lembrar que as soluções e valor da função objetivo obtidas por meio das heurísticas propostas não são os valores de uma solução primal. Para obter esses números corretamente, devemos atualizar os valores das variáveis na maneira correta.

O valor de cada variável $\bar{x}_{ljtp}$ da solução primal será obtido por meio da soma

$$\bar{x}_{ljtp} \gamma_{lt}^f + x_{ljtp}^f$$

O valor de cada variável $\bar{w}_{ljt}$ da solução primal será obtido por meio da soma

$$\bar{w}_{ljt} \gamma_{lt}^f + w_{ljt}^f$$

O valor de cada variável $\bar{\delta}_{lt}$ da solução primal será obtido por meio da soma

$$\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f$$

Após atualizar as variáveis $\bar{x}_{ljtp}$, $\bar{w}_{ljt}$ e $\bar{\delta}_{lt}$, o valor da função objetivo da solução primal será obtido por meio da soma

$$\sum_{ljtp} hb_{ljtp} \bar{x}_{ljtp} + \sum_{ljt} sc_{lj} \bar{w}_{ljt} + \sum_{lt} ac_l \bar{\delta}_{lt}$$

4.1.2 Relaxação da Restrição 2

Agora, para a formulação de uma heurística que factibilize as soluções duais ao relaxar a restrição 2, será construído de forma semelhante da primeira heurística de factibilização.

$$\sum_{j \in P_{l,p}} a_{lj} x_{ljtp} + \sum_j st_{lj} w_{ljt} \leq C_{lt}, \forall l, t$$

Quando dualizamos essa restrição, a função lagrangiana fornecerá soluções duais em que as capacidades das linhas de produção serão excedidas. Dessa forma, subconjuntos de produtos deverão deixar de ser produzidos nas linhas de produção para atender novamente a restrição relaxada.

Teoricamente, poderíamos utilizar a mesma heurística criada anteriormente, soluções primais também poderão ser obtidas. Mas consideramos os casos em que não há necessidade de fechar as linhas, basta reorganizar os produtos que serão produzidos e suas respectivas quantidades.

Por outro lado, existe, de fato, possibilidade de precisar fechar alguma linha e abrir outras. Ao reorganizar as variáveis de decisão dentro de uma linha em um determinado período, pode haver produtos que não podem ser produzidos mais nessa linha, e nem em mais outra linha aberta em outros períodos.

O modelo será composto pelas mesmas variáveis $\bar{x}_{ljtp}$, $\bar{w}_{ljt}$ e $\bar{\delta}_{lt}$ que já tem valores pré-definidos pela solução dual, e também, as variáveis x_{ljtp}^f , w_{ljt}^f e δ_{lt}^f que servem para

definir as linhas que serão abertas, bem como os produtos que serão produzidos nelas e as respectivas quantidades. Além do mais, a variável γ_{lt}^f para decidir as linhas que deverão ser fechadas e uma nova variável de decisão:

α_{ljt}^f Igual a 1 se o produto j continuará na mesma situação (sendo produzido ou não na linha l) no período t e 0, caso contrário

A função objetivo do modelo também será a maximização das variáveis γ_{lt}^f em busca de menos linhas de produção sendo fechadas. Na melhor das hipóteses, nenhuma linha precisa ser fechado. Com a variável de decisão α_{ljt}^f , deixaremos de produzir alguns produtos nas linhas de produção de algum período que descumprir a restrição relaxada, e depois, produzimos elas em outras linhas já abertas. Se realmente houver a necessidade, novas linhas também podem ser abertas para produzir a quantidade requerida.

$$\begin{aligned}
& \text{Maximizar} \quad \sum_{lt} \gamma_{lt}^f \\
& \text{s.a} \quad \sum_{p=\max\{1, t-sl_j\}}^T \sum_l (\bar{x}_{ljpt} \alpha_{ljt}^f + x_{ljpt}^f) = d_{jt}, \forall j, t \\
& \sum_{j \in P_{l,p}} a_{lj} (\bar{x}_{ljtp} \alpha_{ljt}^f + x_{ljtp}^f) + \sum_l st_{lj} (\bar{w}_{ljt} \alpha_{ljt}^f + w_{ljt}^f) \leq C_{lt}, \forall l, t \\
& \sum_p (\bar{x}_{ljtp} \alpha_{ljt}^f + x_{ljtp}^f) \leq \frac{C_{lt}}{a_{lj}} (\bar{w}_{ljt} \alpha_{ljt}^f + w_{ljt}^f), \forall l, j \in P_l, t \\
& \sum_{j \in P_l} (\bar{w}_{ljt} \alpha_{ljt}^f + w_{ljt}^f) \leq \gamma (\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f), \forall l, t \\
& \sum_l r_{kl} (\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f) \leq R_k, \forall k, t \\
& x_{ljtp}^f = 0, \forall l, j, p, t < \max\{1, p - sl_j\} \\
& x_{ljtp}^f = 0, \forall l, j \notin P_l, p, t \\
& \bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f \leq 1, \forall l, t
\end{aligned}$$

$$\bar{w}_{ljt} \alpha_{ljt}^f + w_{ljt}^f \leq 1, \forall l, j, t$$

$$\gamma_{lt}^f - \alpha_{ljt}^f \geq 0, \forall l, j, t$$

$$x_{lntp}^f \geq 0, \forall l, j, p, t$$

$$w_{ljt}^f \in \{0, 1\}, \forall l, j, t$$

$$\delta_{lt}^f \in \{0, 1\}, \forall l, t$$

$$\alpha_{ljt}^f \in \{0, 1\}, \forall l, j, t$$

$$\gamma_{lt}^f \in \{0, 1\}, \forall l, t$$

Em comparação com a primeira heurística criada, a forma que adicionamos as variáveis de decisão γ_{lt}^f e α_{ljt}^f é um pouco diferente. Como cada variável α_{ljt}^f decide se um produto j continuará a ser produzido ou não em uma linha l de um determinado período t , acrescentamos às variáveis $\bar{w}_{ljt}$ e $\bar{x}_{lntp}$. E atribuímos cada variável γ_{lt}^f às variáveis $\bar{\delta}_{lt}$. Devido essa estrutura, a nova restrição foi necessária para garantir que se uma linha de produção vai ser fechada, então, logicamente, nenhum produto poderá ser produzido nela.

Para obter os valores reais da solução primal, procederemos de forma parecida da primeira heurística.

O valor de cada variável $\bar{x}_{lntp}$ da solução primal será obtido por meio da soma

$$\bar{x}_{lntp} \alpha_{ljt}^f + x_{lntp}^f$$

O valor de cada variável $\bar{w}_{ljt}$ da solução primal será obtido por meio da soma

$$\bar{w}_{ljt} \alpha_{ljt}^f + w_{ljt}^f$$

O valor de cada variável $\bar{\delta}_{lt}$ da solução primal será obtido por meio da soma

$$\bar{\delta}_{lt} \gamma_{lt}^f + \delta_{lt}^f$$

Após atualizar as variáveis $\bar{x}_{ljtp}$, $\bar{w}_{ljt}$ e $\bar{\delta}_{lt}$, o valor da função objetivo da solução primal será obtido por meio da soma:

$$\sum_{ljtp} hb_{ljtp} \bar{x}_{ljtp} + \sum_{ljt} sc_{lj} \bar{w}_{ljt} + \sum_{lt} ac_l \bar{\delta}_{lt}$$

4.1.3 Relaxação das Restrições 2 e 5 Simultaneamente

A heurística para factibilizar as soluções duais obtidas pela relaxação das restrições 2 e 5 simultaneamente será idêntida àquela construída para factibilizar as soluções duais obtidas pela relaxação só da restrição 2. Apesar de utilizar a mesma heurística, os resultados obtidos serão completamente distintos.

4.2 Heurística de Melhoria

Após as factibilizações das soluções duais, para aprimorar os resultados das soluções primais alcançados, consideramos o fato de que as heurísticas não priorizaram a minimização dos custos totais, apenas a factibilidade em questão. Logo, pretendemos criar uma heurística de melhoria que tenha como objetivo, um custo total menor, sem comprometer um tempo de solução reduzida.

Observamos que o ponto chave para chegar em valores melhores consistia em determinar com exatidão as linhas de produção que devem ser abertas ou fechadas. Quando as linhas de produção forem determinadas, o problema de determinar as variáveis w_{ljt} e x_{ljtp} é muito mais fácil. Então, a questão de acertar quais produtos devem ser produzidos em cada linha de produção e suas respectivas quantidades é secundária.

Sendo assim, considerando os valores de $\bar{\delta}_{lt}$ obtidas nas soluções primais, existem linhas abertas que deveriam ser fechadas, e também, linhas fechadas que poderiam ser abertas, para aproximar mais da otimalidade. Para tanto, criamos um parâmetro auxiliar que determina um número de linhas que deverão ser avaliados em relação as suas aberturas ou fechamentos. Se, hipoteticamente, avaliássemos todas as linhas, seria o mesmo que resolver o problema original. Seja N_T o número de períodos e N_L o número de linhas de produção, o parâmetro auxiliar D_p é definido por:

$$D_p = (N_T N_L) - 3$$

Definimos o D_p de maneira que sempre será analisado as possibilidades e vantagens de fechar até 3 linhas de produção. Apesar de que não terá um limite para abrir as linhas de produção que estavam fechadas, mas, intuitivamente, como queremos minimizar os custos e os recursos são escassos, haverá uma tendência natural de abrir menos novas linhas de produção.

Para melhor discernimento, introduzimos as variáveis:

- x_{ljtp}^m Quantidade de produtos do tipo j produzidos na linha l durante o período t para atender a demanda no período p
- w_{ljt}^m Igual a 1 se o produto j é produzido na linha l durante o período t e 0, caso contrário
- δ_{lt}^m Igual a 1 se a linha l está montada durante o período t e 0, caso contrário
- γ_{lt}^m Igual a 1 se a linha l continuará na mesma situação (aberta ou fechada) no período t e 0, caso contrário

$$\text{Minimizar } \sum_{l,j,t,p} hb_{ljtp} x_{ljtp}^m + \sum_{l,j,t} sc_{lj} w_{ljt}^m + \sum_{l,t} ac_l (\bar{\bar{\delta}}_{lt} \gamma_{lt}^m + \delta_{lt}^m)$$

$$\text{s.a } \sum_{p=\max\{1,t-sl_j\}}^T \sum_l x_{ljpt}^m = d_{jt}, \forall j, t$$

$$\sum_{j \in P_l, p} a_{lj} x_{ljtp}^m + \sum_l st_{lj} w_{ljt}^m \leq C_{lt}, \forall l, t$$

$$\sum_p x_{ljtp}^m \leq \frac{C_{lt}}{a_{lj}} w_{ljt}^m, \forall l, j \in P_l, t$$

$$\sum_{j \in P_l} w_{ljt}^m \leq \gamma (\bar{\bar{\delta}}_{lt} \gamma_{lt}^m + \delta_{lt}^m), \forall l, t$$

$$\sum_l r_{kl} (\bar{\bar{\delta}}_{lt} \gamma_{lt}^m + \delta_{lt}^m) \leq R_k, \forall k, t$$

$$x_{ljtp}^m = 0, \forall l, j, p, t < \max\{1, p - sl_j\}$$

$$x_{ljtp}^m = 0, \forall l, j \notin P_l, p, t$$

$$\delta_{lt}^m = 0, \text{ se } \bar{\bar{\delta}}_{lt} = 1, \forall l, t$$

$$\gamma_{lt}^m = 1, \text{ se } \bar{\bar{\delta}}_{lt} = 0, \forall l, t$$

$$\sum_{lt} \gamma_{lt}^m \geq D_p$$

$$x_{ljtp}^m \geq 0, \forall l, j, p, t$$

$$w_{ljt}^m \in \{0, 1\}, \forall l, j, t$$

$$\delta_{lt}^m \in \{0, 1\}, \forall l, t$$

$$\gamma_{lt}^m \in \{0, 1\}, \forall l, t$$

No modelo criado, observe que não utilizamos as variáveis $\bar{w}_{ljt}$ e $\bar{x}_{ljtp}$ obtidas pelas soluções primais, pois notamos que a formulação matemática e o rendimento computacional é melhor quando construído dessa forma.

Algumas novas restrições foram adicionadas. Considerando que uma linha de produção já estava aberta ($\bar{\delta}_{lt} = 1$), não será coerente essa ser aberta novamente ($\delta_{lt}^m = 1$), nem mesmo quando γ_{lt}^m decidir o fechamento desta. Em contrapartida, se uma linha de produção já estiver fechada ($\bar{\delta}_{lt} = 0$), não será coerente essa ser fechada novamente ($\gamma_{lt}^m = 0$). Ademais, a restrição que contém o parâmetro auxiliar D_p , define a quantidade máxima de linhas de produção que podem ser fechadas. Vale ressaltar que como não há um número mínimo de linhas de produção que devem ser fechadas, então, caso não haja vantagem em fechar nenhuma linha, será permitido permanecer todos $\bar{\delta}_{lt}$ com os mesmo valores.

Por fim, o valor da solução primal aprimorada é justamente o valor da função objetivo do modelo criado para melhorar os resultados.

Nos testes com as instâncias maiores (classe D e E), considerando a aplicação da relaxação na restrição 2, foi observado que devido ao aumento do número de produtos, a heurística de melhoria não estava apresentando muita eficiência dentro do tempo delimitado. E assim, após alguns testes, o caminho mais vantajoso foi reescrever o parâmetro auxiliar D_p apenas como produto $N_T N_L$, o qual significa que nenhum δ_{lt} será alterado.

Nos testes com as instâncias pequenas (classes A e B), considerando a aplicação da relaxação na restrição 5, foi necessário reescrever o parâmetro auxiliar D_p , subtraindo o produto $N_T N_L$ por 10. Pois como o problema é bastante simplificado, mais linhas de produção devem ser reavaliadas sobre sua abertura e fechamento. Já para as instâncias de grande porte (classes C, D e E), voltamos a utilizar a conjectura original do parâmetro D_p e delimitamos em 600 segundos para resolvê-la. Vale salientar que a heurística de melhoria, diferente da heurística de factibilização, pode ser interrompido durante seu procedimento, pois as soluções continuam sendo factíveis, apenas não serão os melhores.

Os ajustes realizados no caso anterior também serão aplicados na relaxação simultânea das restrições 2 e 5.

As heurísticas desenvolvidas demonstraram, por meio de alguns testes prévios, eficiência em relação ao tempo de solução e resultados promissores. E assim, a expectativa é integrar elas ao método subgradiente, obtendo soluções boas em tempos competitivos comparado aos outros algoritmos, em específico, o *Cplex*, um solucionador (*solver*) de otimização de alto desempenho.

4.3 Considerações Finais

Neste capítulo, abordamos sobre o desenvolvimento das heurísticas. Apesar de existirem muitos algoritmos prontos na literatura, mas devido a complexidade do problema estudado, formulamos modelos heurísticos particulares para cumprir a necessidade de factibilizar e obter soluções primais mais próximos da otimalidade. Além do mais, deixamos sugestões para trabalhos futuros que podem aprimorar ainda mais os resultados.

Capítulo 5

Desenvolvimentos e Resultados Computacionais

O *Python* é uma linguagem de programação que está se tornando uma das mais indicadas e preferidas pelos programadores, uma vez que a sintaxe básica é semelhante à do inglês, facilitando a aprendizagem. Segundo Marcondes [17], com a linguagem Python é possível desenvolver algoritmos matemáticos de cunho simples a exemplo de equações, expressões numéricas, operações matemáticas básicas, até problemas complexos, como derivadas, integral, números complexos, entre outros. Além do mais, possui uma maior produtividade devido à redução de linhas de código em comparação às outras linguagens, e também, há uma grande biblioteca padrão que contém códigos reutilizáveis para quase todas as tarefas. Dessa forma, não é necessário sempre escrever códigos do zero.

Historicamente, a primeira versão do *Python* foi publicada em 1991 pelo programador de computadores holandês, o Guido Van Rossum. Desde então, com a versão 1.0 lançada em 1994, foi atualizando até a versão mais recente 3.12.4, lançada em 6 de junho de 2024. Atualmente, o Python é amplamente utilizado em uma variedade de aplicações, desde desenvolvimento *web* e científico até automação de sistemas e inteligência artificial. Sua crescente popularidade é impulsionada pela sua simplicidade, legibilidade e pela comunidade ativa que suporta seu desenvolvimento contínuo.

Assim, com base em seus pontos positivos, optamos por utilizá-la para o desenvolvimento computacional deste trabalho.

Ademais, instalamos o *IBM ILOG CPLEX Optimization Studio*, versão 22.1.1, um pacote de software de otimização que dispõe de um algoritmo, denominado *Branch-and-*

Cut, que soluciona problemas de otimização linear inteira mista. Além do mais, apresenta solução de análise prescritiva que permite o rápido desenvolvimento e implementação de modelos de otimização de decisão usando programação matemática e de restrição. Este *solver* será integrado no Python para realizar os testes computacionais.

A versão 3.10.9 do *Python* foi instalada por ser compatível com o *Cplex*. Para estabelecer o vínculo desejado, o primeiro passo é adicionar *Python* ao *Windows PATH*, para que possa executar o interpretador *Python*, e iniciar um ambiente de programação virtual ou executar comandos como *pip install* do terminal. O segundo passo é utilizar o prompt de comando e descrever o caminho para chegar na pasta dentro do *IBM ILOG CPLEX Optimization Studio* que contém o arquivo *setup*. O caminho comumente descrito é *C : \ProgramFiles\IBM\ILOG\CPLEX_Studio2211\python*. E por último, basta executar o comando *setup.py install*. Descrevemos os passos anteriores detalhadamente para auxiliar aqueles que tenham interesse em realizar o mesmo vínculo.

Para facilitar a edição de *scripts*, teste, *debugging* e visualização gráfica, adquirimos o ambiente interativo *Spyder* por meio do prompt de comando e a função *pip install* (gerenciador de pacotes). Por outro lado, os testes serão feitos em um computador com processador *Intel(R) Core(TM) i5-6200U*, memória *RAM 8GB*, com sistema operacional de 64 *bits*.

5.1 Implementação do Modelo Matemático

O código desenvolvido em *Python* para o modelo matemático pode ser dividido em 4 partes principais:

- I. Importar e inicializar as bibliotecas
- II. Gerenciar arquivos das instâncias para definir os conjuntos e parâmetros
- III. Definir as variáveis de decisão, a função objetivo e as restrições lógicas.
- IV. Resolver o modelo e exportar informações relevantes

A primeira parte é sobre a importação e inicialização de bibliotecas. A única necessária foi a biblioteca *Model* dentro do *docplex.mp.model*, a qual fornece um vínculo direto do *Python* com o *Cplex*, e também, contém o comando *Model* que é capaz de relacionar as variáveis, função objetivo e as restrições introduzidas posteriormente. Para obter o

acesso à essa biblioteca, basta inserir no prompt de comando *pip install docplex*. O pacote do *Cplex* instalado é imprescindível para seu funcionamento, pois, caso na sua ausência, existirá um limite de 1000 variáveis. Caso exceda o valor, será informado o erro 1016: *Problem size limits exceeded* (Limites de tamanho do problema excedidos).

Na segunda parte, realizamos a leitura das instâncias que estão no formato de documentos *txt*. Nessa parte, transcrevemos os dados contido nas instâncias, de forma que cada tipo de dado seja organizado em conjuntos, e também, associar as variáveis aos seus respectivos parâmetros. Contabiliza-se um total de 20 conjuntos e parâmetros principais no código, sendo todos eles em forma de **lista** (coleção ordenada de valores, separados por vírgula e dentro de colchetes) ou **dicionário** (coleção que guarda valores multidimensionais para cada índice, separado por vírgula e dentro de chaves).

Na terceira parte, introduzimos primeiramente as variáveis de decisão, que podem ser, no contexto do problema tratado, de dois tipos: binária ou inteira. Nessa parte, é importante compreender que a solução do algoritmo desenvolvido é justamente os números que essas variáveis irão assumir, pois são estes os valores buscados que minimizarão os custos totais como pretendido. Depois, acrescentamos a função objetivo e as restrições. Quando criamos as variáveis de decisão, será definida também o seu tipo, bem como os domínios de cada um. Logo as últimas restrições do modelo original foram dispensadas. Na implementação das restrições, vale salientar que há uma nova restrição inserida. A necessidade dela foi devido à conclusão de que o algoritmo não estava tornando nulo os casos em que as variáveis de decisão $X_{l,jtp}$ tinham $j \notin P_l$, e isso estava gerando uma inconsistência para a linguagem Python em específico.

Na última parte, o modelo é solucionado e informações relevantes são exportadas. Principalmente, o tempo de solução, o valor da função objetivo e o *GAP* de dualidade.

Após o sucesso na construção do algoritmo, foi realizado um simples teste com mínimas variáveis (3 períodos, 3 linhas de produção e 4 tipos de produtos). O tempo necessário foi de 6 centésimos de segundo, e a solução ótima foi obtida facilmente.

Os critérios principais que foram considerados para avaliar se a implementação foi sucedida corretamente consiste em observar se toda demanda foi atendida, e se todas as restrições estabelecidas foram cumpridas. Ao verificar que não há mais erros no código, partimos para ajustes e adequações das 100 instâncias propostas em Soler, Santos e Akartunali [30], as quais serão testadas posteriormente.

5.2 Testes com Instâncias da Literatura

Primeiramente, foi necessário identificar a existência de padrões nos dados das instâncias que estava em formato *txt*, como por exemplo, a relação entre o número da linha que uma informação se encontra com a quantidade de período, linhas de produção e/ou tipo de produto. Dessa forma, associam-se expressões algébricas às linhas que se encontram as informações desejadas, independente do quantitativo de variáveis em cada instância.

No início dos testes das instâncias da literatura, notou-se que os dados contidos não eram totalmente coincidentes ao contexto que abordamos, sendo assim, foi necessário realizar ajustes e adequações para que os testes sejam coerentes e válidos.

As variáveis bidimensionais que foram criadas, constavam nas instâncias que eram tridimensionais. E então, para fazer retificações sem alterar excessivamente os valores iniciais da instância, foi aplicado um método semelhante ao cálculo de média aritmética dos valores. De forma mais explícita, seja uma variável x_{ij} bidimensional, porém na instância constam dados apenas para x_{ijk} (com $k \in \{1, \dots, 10\}$) tridimensional. Para retificar essa incompatibilidade, foi feita a somatória $\sum_{k=1}^{10} x_{ijk}$ e, depois, divide-se por 10, obtendo assim o valor utilizado para x_{ij} . Esses ajustes foram aplicados, por exemplo, para as variáveis sc_{lj} (Custo de setup para produção do produto j na linha l) e st_{lj} (Tempo de preparação para produção do produto j na linha l).

No total de 100 instâncias para testes, com a finalidade de comparar melhor a eficiência dos algoritmos e as técnicas aplicadas, separamos elas conforme o número de variáveis de decisão que afeta no grau de complexidade.

A quantidade de variáveis de decisão em cada classe pode ser deduzida com base no número de períodos (t_n), produtos (j_n) e linhas de produção (l_n). Em cada instância contabiliza-se $l_n j_n t_n^2$ variáveis $X_{l_j t_p}$, $l_n j_n t_n$ variáveis $W_{l_j t}$ e $l_n t_n$ variáveis δ_{lt} . Ou seja, para cada caso, o total de variáveis de decisão será $l_n j_n t_n^2 + l_n j_n t_n + l_n t_n$ ($l_n j_n t_n^2$ inteiras e $l_n j_n t_n + l_n t_n$ binárias).

- Classe A - As instâncias de número 401 a 420.

Tem 10 períodos, 7 linhas de produção, 60 horas de capacidade, 45 tipos de produtos, 5 tipos de recursos. Temos no total: $7.45.10^2 + 7.45.10 + 7.10 = 34720$ variáveis de decisão (31500 inteiras e 3220 binárias).

- Classe B - As instâncias de número 421 a 440.

Tem 10 períodos, 10 linhas de produção, 80 horas de capacidade, 80 tipos de produtos, 6 tipos de recursos. Temos no total: $10 \cdot 80 \cdot 10^2 + 10 \cdot 80 \cdot 10 + 10 \cdot 10 = 88100$ variáveis de decisão (80000 inteiras e 8100 binárias).

- Classe C - As instâncias de número 441 a 460.

Tem 14 períodos, 10 linhas de produção, 112 horas de capacidade, 90 tipos de produtos, 6 tipos de recursos. Temos no total: $10 \cdot 90 \cdot 14^2 + 10 \cdot 90 \cdot 14 + 10 \cdot 14 = 189140$ variáveis de decisão (176400 inteiras e 12740 binárias).

- Classe D - As instâncias de número 801 a 820.

Tem 12 períodos, 10 linhas de produção, 96 horas de capacidade, 110 tipos de produtos, 6 tipos de recursos. Temos no total: $10 \cdot 110 \cdot 12^2 + 10 \cdot 110 \cdot 12 + 10 \cdot 12 = 171720$ variáveis de decisão (158400 inteiras e 13320 binárias).

- Classe E - As instâncias de número 821 a 840.

Tem 14 períodos, 10 linhas de produção, 112 horas de capacidade, 110 tipos de produtos, 6 tipos de recursos. Temos no total: $10 \cdot 110 \cdot 14^2 + 10 \cdot 110 \cdot 14 + 10 \cdot 14 = 231140$ variáveis de decisão (215600 inteiras e 15540 binárias).

Nas tabelas apresentadas a seguir, são expostos os valores das funções objetivo (*FO*) e do *GAP* para cada estratégia de solução: apenas usando solucionador *Cplex*, aplicação da relaxação na restrição 2, aplicação da relaxação na restrição 5, e aplicação da relaxação simultânea nas restrições 2 e 5. Os melhores resultados de cada instância, tanto para o *FO* quanto para o *GAP*, são destacados em negrito.

O *GAP*, ou *GAP* de dualidade, é um conceito utilizado em otimização matemática para avaliar a qualidade das soluções obtidas. Ele mede a diferença entre a solução do problema original (primal) e a do problema associado (dual). Quanto mais próximo de zero for o valor do *GAP*, mais próxima a solução está da otimalidade. Um *GAP* igual a zero indica que a solução encontrada é ótima, enquanto valores maiores indicam que a solução ainda pode ser melhorada. Seja *FO* o valor da solução primal e *LI* o melhor limitante inferior ou a maior solução dual, o *GAP*, em porcentagem, pode ser calculado da seguinte maneira:

$$\text{GAP} = \frac{FO - LI}{FO} \times 100$$

Com objetivo de comparar os resultados de forma mais coerente, padronizamos em utilizar o melhor limitante inferior encontrado pelo solucionador *Cplex* para determinar os *GAP's* em cada estratégia de solução.

Na tabela 5.1, apresentamos os resultados dos testes com as instâncias da classe A.

Tabela 5.1: Resultados dos Testes das Instâncias Classe A

Nº	FO_{Cplex}	FO_{R2}	FO_{R5}	FO_{R2eR5}	GAP_{Cplex}	GAP_{R2}	GAP_{R5}	GAP_{R2eR5}
401	61156	61156	61213	61682	0.01%	0.01%	0.10%	0.86%
402	50131	51056	50210	51174	0.01%	1.82%	0.17%	2.05%
403	52757	52757	53187	53621	0.01%	0.01%	0.82%	1.62%
404	65774	67509	66772	66790	0.01%	2.58%	1.50%	1.53%
405	46867	46867	48454	50528	0.01%	0.01%	3.28%	7.25%
406	60283	60283	62170	62674	0.01%	0.01%	3.04%	3.82%
407	48477	48477	49070	49447	0%	0%	1.21%	1.96%
408	51738	51772	51935	55021	0.01%	0.07%	0.39%	5.97%
409	50481	50481	51141	52460	0.01%	0.01%	1.30%	3.78%
410	54865	57775	56074	62723	0.01%	5.04%	2.16%	12.54%
411	59994	59994	62587	60500	0.01%	0.01%	4.15%	0.85%
412	51065	51245	51082	51082	0%	0.35%	0.03%	0.03%
413	56626	56881	58083	58005	0.01%	0.46%	2.52%	2.39%
414	49274	49643	51352	52158	0.01%	0.75%	4.06%	5.54%
415	53246	53246	55076	57014	0.01%	0.01%	3.33%	6.62%
416	47141	47435	47723	48225	0.01%	0.63%	1.23%	2.26%
417	60477	60904	62622	65068	0.01%	0.71%	3.43%	7.06%
418	55808	56017	55914	57888	0.01%	0.38%	0.20%	3.60%
419	52391	52583	52843	53657	0.01%	0.37%	0.86%	2.37%
420	75825	75825	78871	77978	0.01%	0.01%	3.87%	2.77%

Nos testes utilizando apenas o solucionador *Cplex*, o tempo médio para resolver as instâncias da classe A foi de aproximadamente 25 segundos. O *GAP* de dualidade recorrente de 0,01% indica que a solução encontrada pelo algoritmo difere em, no máximo, 0,01% da solução ótima esperada. No entanto, foi observado que em alguns casos, o tempo médio aumentou exponencialmente devido ao método de solução empregado pelo

Cplex. Nas instâncias com maior desvio padrão de tempo, duas mensagens se destacam: *Performing restart* e *Repeating presolve*. Isso indica que o solucionador está reiniciando a resolução do problema e repetindo o processo de pré-solução, que simplifica o problema ao eliminar variáveis ou restrições redundantes, facilitando a sua resolução. Embora os resultados obtidos pelo *Cplex* sejam de ótima qualidade, observa-se um possível obstáculo ao lidar com instâncias maiores. Caso o solucionador não identifique imediatamente o melhor caminho de pré-solução, o tempo necessário para resolver o problema pode aumentar inesperadamente.

Nos testes com as instâncias da classe A, considerando a aplicação da relaxação na restrição 2, podemos observar que, em 45% dos casos, o resultado alcançado (FO_{R2}) é igual ao do *Cplex* (FO_{Cplex}). Já nos restantes dos casos, a diferença média é de 1,23%.

Nos testes com as instâncias da classe A, considerando a aplicação da relaxação na restrição 5, a diferença média dos resultados em relação ao *Cplex* é de 1.93%. Vale salientar que apesar de nenhum dos resultados (FO_{R5}) conseguiu igualar ao do *Cplex*, mas em 25% das instâncias testadas, os valores foram mais satisfatórias do que os da estratégia anterior.

Nos testes com as instâncias da classe A, considerando a aplicação da relaxação simultânea nas restrições 2 e 5, a diferença média dos resultados em relação ao *Cplex* é de 3.98%. De forma geral, o único resultado considerado satisfatório é encontrado no teste da instância 412. Acredita-se que o problema está muito relaxado neste caso, tornando mais difícil alcançar a otimalidade em comparação aos casos anteriores.

Na tabela 5.2, apresentamos os resultados dos testes com as instâncias da classe B.

Tabela 5.2: Resultados dos Testes das Instâncias Classe B

Nº	FO_{Cplex}	FO_{R2}	FO_{R5}	FO_{R2eR5}	GAP_{Cplex}	GAP_{R2}	GAP_{R5}	GAP_{R2eR5}
421	65855	65855	68152	67779	0.01%	0.01%	3.38%	2.85%
422	73170	73297	77728	76753	0.01%	0.18%	5.87%	4.68%
423	71378	71684	80387	73368	0.01%	0.44%	11.22%	2.72%
424	70789	70846	76211	73368	0.01%	0.09%	7.12%	3.52%
425	72633	74568	77036	74381	0.01%	2.60%	5.72%	2.36%
426	81102	85758	90732	81855	0.01%	5.44%	10.62%	0.93%
427	77352	77352	81118	78233	0.01%	0.01%	4.65%	1.14%

Nº	FO_{Cplex}	FO_{R2}	FO_{R5}	FO_{R2eR5}	GAP_{Cplex}	GAP_{R2}	GAP_{R5}	GAP_{R2eR5}
428	64096	64096	67469	65419	0.01%	0.01%	5.01%	2.03%
429	65175	65175	70118	65874	0.01%	0.01%	7.06%	1.07%
430	83516	83932	86831	86786	0.01%	0.51%	3.83%	3.78%
431	67196	69054	69212	68662	0.01%	2.70%	2.92%	2.14%
432	69955	71404	78714	72915	0.01%	2.04%	11.14%	4.07%
433	74747	77047	78605	78375	0.01%	2.99%	4.92%	4.64%
434	76477	79113	80437	80383	0.01%	3.34%	4.93%	4.87%
435	73183	73183	77549	73519	0.01%	0.01%	5.64%	0.47%
436	72003	72245	79895	76334	0.01%	0.34%	9.89%	5.68%
437	71050	73019	75749	74449	0.01%	2.71%	6.21%	4.58%
438	72900	74885	76296	75553	0.01%	2.66%	4.46%	3.52%
439	64810	65186	65286	68667	0.00%	0.58%	0.73%	5.62%
440	69872	70021	72457	72184	0.01%	0.22%	3.58%	3.21%

Nos testes utilizando apenas o solucionador *Cplex*, o tempo médio para resolver as instâncias de classe B foi de aproximadamente 5 minutos, com 3 instâncias chegando a precisar do triplo do tempo médio para resolver. Além do mais, apesar do número de variáveis da classe B ter um pouco mais que o dobro da classe A, mas o tempo necessário para solucionar a maioria das instâncias aumentou 12 vezes.

Nos testes com as instâncias da classe B, considerando a aplicação da relaxação na restrição 2, podemos observar que em 25% dos casos, o resultado alcançado é igual do *Cplex*, houve uma redução em comparação aos testes com as instâncias de classe A. Mas, nos restantes dos casos, temos uma diferença média de 1.832%, o que é uma divergência ainda pequena.

Nos testes com as instâncias da classe B, considerando a aplicação da relaxação na restrição 5, a diferença média dos resultados em relação ao *Cplex* aumentou para 6.404%. Em nenhum dos casos foi obtido um resultado melhor em comparação às estratégias anteriores.

Nos testes com as instâncias da classe B, considerando a aplicação da relaxação simultânea nas restrições 2 e 5, a diferença média dos resultados em relação ao *Cplex* é de 2.84%. Apesar de não ter nenhum valor igual do *Cplex* ou melhor do que os valores na

aplicação da relaxação na restrição 2, mas, em 95% dos casos, superou os valores obtidos na aplicação da relaxação na restrição 5.

Durante os testes iniciais com as instâncias das classes C, D e E, as estratégias que aplicaram a relaxação na restrição 5 e a relaxação simultânea nas restrições 2 e 5 apresentaram resultados distantes da solução ótima, sendo menos competitivas em comparação ao *Cplex*. Assim, decidimos realizar os testes apenas com as 10 primeiras instâncias dessas classes para essas aplicações. E então, nas tabelas a seguir, será dividido em duas partes em cada classe. Na parte 1, são os resultados dos testes com as 10 primeiras instâncias envolvendo todas as estratégias de solução. E na parte 2, serão as 10 instâncias restantes envolvendo apenas os resultados do solucionador *Cplex* e a aplicação da relaxação na restrição 2.

Nas tabelas 5.3 e 5.4, observe os resultados dos testes com as instâncias da classe C.

Tabela 5.3: Resultados dos Testes das Instâncias Classe C - Parte 1

Nº	FO_{Cplex}	FO_{R2}	FO_{R5}	FO_{R2eR5}	GAP_{Cplex}	GAP_{R2}	GAP_{R5}	GAP_{R2eR5}
441	169031	165501	223802	230550	14.79%	12.97%	35.64%	37.53%
442	296511	265480	337811	307935	53.70%	48.28%	59.36%	55.41%
443	215019	169314	242802	207219	38.30%	21.64%	45.36%	35.98%
444	184541	144470	223250	218230	33.55%	15.12%	45.07%	43.81%
445	248821	193795	264195	277378	36.44%	18.39%	40.13%	42.98%
446	190287	181820	200708	225355	30.29%	27.04%	33.91%	41.13%
447	200379	173071	189443	217567	34.85%	24.57%	31.09%	40.00%
448	335474	203329	292170	284320	57.04%	29.12%	50.67%	49.31%
449	249488	186630	277704	242866	39.49%	19.11%	45.64%	37.84%
450	180026	177558	223786	228834	24.75%	23.70%	39.46%	40.80%

Tabela 5.4: Resultados dos Testes das Instâncias Classe C - Parte 2

Nº	FO_{Cplex}	FO_{R2}	GAP_{Cplex}	GAP_{R2}
451	216043	166473	39.45%	27.28%
452	195706	158425	32.06%	21.18%
453	159492	138240	26.47%	19.54%

Nº	FO_{Cplex}	FO_{R2}	GAP_{Cplex}	GAP_{R2}
454	164206	136726	25.82%	19.25%
455	148386	144167	17.35%	15.76%
456	207995	172346	33.46%	26.49%
457	193390	172579	31.19%	29.02%
458	207417	171565	31.17%	23.25%
459	159521	154965	13.67%	12.52%
460	299742	227306	52.17%	29.38%

Na estratégia de solução utilizando apenas o solucionador *Cplex*, observou-se que, para as instâncias das classes C, D e E, o tempo de solução se estendia por horas, sem garantir a obtenção de soluções ótimas. Por isso, foi definido um limite de 1 hora para a solução das instâncias, com o objetivo de obter o valor da função objetivo e o *GAP* de dualidade alcançados, mesmo que não sejam ótimos. Além disso, conforme descrito nas subseções 3.2.1, 3.2.2 e 3.2.3, o limite de 1 hora também será aplicado às outras estratégias de solução, visando uma melhor análise e comparação dos resultados.

Nos testes com as instâncias da classe C, considerando a aplicação da relaxação na restrição 2, podemos observar que o rendimento computacional superou o *Cplex* em todos os casos. A diferença média dos resultados é de 15.45%, e em número, é de aproximadamente 38469. Em relação ao *GAP*, a diferença média é de 8.95%.

Nos testes com as instâncias da classe C, considerando a aplicação da relaxação na restrição 5, dos resultados das 10 instâncias testadas, 20% foram melhores que os valores obtidos pelo *Cplex*, mas nenhum deles foi mais satisfatório que os resultados da estratégia anterior de aplicar a relaxação na restrição 2.

Nos testes com as instâncias da classe C, considerando a aplicação da relaxação simultânea nas restrições 2 e 5, 30% dos resultados das 10 instâncias testadas foram melhores que os valores obtidos pelo *Cplex* e 50% foram melhores que os valores obtidos na estratégia anterior. Contudo, nenhum deles foi mais satisfatório que os resultados da estratégia de aplicar a relaxação na restrição 2.

Nas tabelas 5.5 e 5.6, observe os resultados dos testes com as instâncias da classe D.

Tabela 5.5: Resultados dos Testes das Instâncias Classe D - Parte 1

Nº	FO_{Cplex}	FO_{R2}	FO_{R5}	FO_{R2eR5}	GAP_{Cplex}	GAP_{R2}	GAP_{R5}	GAP_{R2eR5}
801	216587	187657	202878	241305	44.28%	35.59%	40.36%	49.79%
802	466948	253001	275420	296245	71.69%	47.62%	51.72%	55.18%
803	186919	148856	207183	195510	33.50%	16.34%	39.78%	56.87%
804	169237	160254	184332	192042	29.15%	25.50%	35.32%	37.79%
805	162342	177228	208519	209341	21.89%	28.60%	39.22%	39.38%
806	182806	181446	187398	201842	33.37%	33.04%	35.19%	39.97%
807	180481	189240	199918	232905	28.37%	31.74%	35.31%	44.56%
808	183619	162778	201955	205290	31.31%	22.40%	37.50%	38.50%
809	245859	192737	240745	236355	45.76%	30.63%	44.43%	43.36%
810	223889	191133	220345	240958	44.93%	35.09%	43.66%	48.61%

Tabela 5.6: Resultados dos Testes das Instâncias Classe D - Parte 2

Nº	FO_{Cplex}	FO_{R2}	GAP_{Cplex}	GAP_{R2}
811	195929	189578	22.98%	20.43%
812	196774	175517	36.20%	28.50%
813	303022	185399	57.97%	31.17%
814	420927	222392	69.71%	42.73%
815	259111	227732	48.27%	40.73%
816	398265	187368	68.68%	33.33%
817	314408	157733	57.89%	16.00%
818	369269	183537	68.83%	37.47%
819	225065	160728	41.03%	17.51%
820	160790	156680	23.65%	21.66%

Nos testes com as instâncias da classe D, considerando a aplicação da relaxação na restrição 2, podemos observar que em apenas duas instâncias, Inst805 e Inst807, o resultado não superou a do *Cplex*. Em geral, a diferença média dos valores da função objetivo foi de 15.01%, e em número, é de aproximadamente 51294. Em relação ao *GAP*, a diferença média é de 10.37%.

Nos testes com as instâncias da classe D, considerando a aplicação da relaxação na restrição 5, 40% dos resultados das 10 instâncias testadas foram melhores que os valores obtidos pelo *Cplex*, mas nenhum deles foi mais satisfatório que os resultados da estratégia anterior de aplicar a relaxação na restrição 2.

Nos testes com as instâncias da classe D, considerando a aplicação da relaxação nas restrições 2 e 5, 20% dos resultados das 10 instâncias testadas foram melhores que os valores obtidos pelo *Cplex* e 10% foram melhores que os valores obtidos na estratégia anterior. Contudo, nenhum deles foi mais satisfatório que os resultados da estratégia de aplicar a relaxação na restrição 2.

Nas tabelas 5.7 e 5.8, observe os resultados dos testes com as instâncias da classe E.

Tabela 5.7: Resultados dos Testes das Instâncias Classe E - Parte 1

Nº	FO_{Cplex}	FO_{R2}	FO_{R5}	FO_{R2eR5}	GAP_{Cplex}	GAP_{R2}	GAP_{R5}	GAP_{R2eR5}
821	261891	191300	220427	248329	45.39%	25.00%	34.91%	42.14%
822	247753	189263	221777	203209	45.16%	28.63%	39.05%	33.49%
823	351457	245608	391240	279856	56.89%	38.27%	61.16%	45.77%
824	222151	194680	266360	241418	34.01%	24.91%	45.02%	39.36%
825	611069	224490	258959	245056	75.69%	33.98%	42.72%	39.54%
826	318612	217094	254557	256643	55.21%	34.34%	44.00%	44.49%
827	1053444	222019	268753	305514	84.21%	25.22%	38.08%	45.69%
828	265843	217648	218957	234673	50.01%	38.92%	39.30%	43.43%
829	420558	201618	350850	308255	64.59%	26.12%	57.55%	51.72%
830	225365	214709	299747	264448	45.37%	42.43%	58.76%	53.28%

Tabela 5.8: Resultados dos Testes das Instâncias Classe E - Parte 2

Nº	FO_{Cplex}	FO_{R2}	GAP_{Cplex}	GAP_{R2}
831	351823	224925	59.90%	37.04%
832	204344	204159	29.74%	29.73%
833	265447	192859	46.22%	26.23%
834	234884	222499	40.00%	36.59%
835	252999	187016	49.07%	31.16%

Nº	FO_{Cplex}	FO_{R2}	GAP_{Cplex}	GAP_{R2}
836	362075	225570	62.96%	40.31%
837	519958	283562	69.86%	44.91%
838	226075	183002	37.44%	22.52%
839	323601	165446	61.70%	25.40%
840	279728	245636	38.77%	30.30%

Nos testes com as instâncias da classe E, considerando a aplicação da relaxação na restrição 2, podemos observar que superamos novamente o *Cplex* em todos os casos. A diferença média é de 28.54%, e em número, é de aproximadamente 183213. Em relação ao *GAP*, a diferença média é de 15.99%.

Nos testes com as instâncias da classe E, considerando a aplicação da relaxação na restrição 5, 70% dos resultados das 10 instâncias testadas foram melhores que os valores obtidos pelo *Cplex*, mas nenhum deles foi mais satisfatório que os resultados da estratégia anterior de aplicar a relaxação na restrição 2.

Nos testes com as instâncias da classe E, considerando a aplicação da relaxação simultânea nas restrições 2 e 5, 80% dos resultados das 10 instâncias testadas foram melhores que os valores obtidos pelo *Cplex* e 60% foram melhores que os valores obtidos na estratégia anterior. Contudo, nenhum deles foi mais satisfatório que os resultados da estratégia de aplicar a relaxação na restrição 2.

5.3 Considerações Finais

Neste capítulo, apresentamos inicialmente os detalhes da implementação do modelo matemático na linguagem de programação Python. Em seguida, realizamos testes com 100 instâncias da literatura e analisamos os resultados comparando cada caso de aplicação da Relaxação Lagrangiana. Por fim, observamos que os resultados obtidos com a relaxação da restrição 2 foram os mais satisfatórios. Nas instâncias pequenas, a qualidade das soluções diferiu pouco, e, nas instâncias maiores, superou em quase todos os casos o solver *Cplex*.

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho, estudamos um problema de dimensionamento de lotes de classe NP-difícil. Implementamos o modelo matemático na linguagem de programação *Python*, integrado com o *Cplex*, um *solver* de alto desempenho. Durante os testes com instâncias de tamanhos crescentes, observamos um aumento inesperado no tempo de solução. Para lidar com isso, optamos por aplicar a Relaxação Lagrangiana, iterar usando o método subgradiente, e criar heurísticas de factibilização e melhoria, visando obter boas soluções em menos tempo.

Com base nos testes e resultados computacionais, concluímos que a técnica demonstrou seu potencial por reduzir o tempo necessário para solucionar o problema abordado. Além disso, a qualidade das soluções encontradas para instâncias de grande porte pode ser superior às aquelas obtidas pelo *Cplex* dentro do tempo limite de uma hora.

Relaxando apenas a restrição 2, o problema é simplificado o suficiente para reduzir o tempo de solução a uma duração aceitável e encontrar soluções satisfatórias. Nesse caso, observamos que as soluções duais geradas pela função Lagrangiana são valores supostamente próximos da solução ótima, e o tempo para torná-las factíveis pela heurística foi bem curto. Assim, a busca iterativa do método subgradiente mostrou-se eficiente, mesmo com um número mínimo de iterações.

Ao relaxar apenas a restrição 5, ou as restrições 2 e 5 simultaneamente, o problema é bastante simplificado. No entanto, mesmo que o tempo de solução tenha se tornado muito curto, as soluções ficaram distantes da otimalidade. Nesses casos, observamos que as soluções duais eram difíceis de serem factibilizadas pela heurística, e, mesmo aumentando o número de iterações no método subgradiente, os resultados não foram promissores.

Este trabalho é de grande relevância para a exploração do uso da Relaxação Lagrangiana e resolver o problema apresentado. Embora os resultados sejam bastante positivos, há muitas possibilidades de melhorias em trabalhos futuros.

Em relação ao tempo de solução da função Lagrangiana e das heurísticas de factibilização e melhoria, observa-se que, para instâncias maiores, é frequentemente necessário impor um limite de tempo para a resolução. No entanto, em cada caso, é essencial priorizar a qualidade da resposta de um desses procedimentos. Isso envolve ponderar as vantagens de obter uma solução dual mais próxima da ótima, encontrar uma solução factível ou aprimorar a solução já factível. É fundamental analisar qual dessas abordagens proporcionará os melhores resultados.

Em relação às heurísticas, há duas linhas principais para estudo futuro. Primeiramente, investigar possíveis alterações na estrutura dos modelos construídos e explorar diferentes formas de formular a função objetivo, conforme discutido na seção 4.1.1. Em segundo lugar, aplicar heurísticas clássicas da literatura e comparar os resultados obtidos com essas abordagens.

Em relação à aplicação da Relaxação Lagrangiana, embora as restrições de acoplamento sejam as principais responsáveis pela complexidade que leva à sua relaxação, pode haver potencial para explorar a relaxação de outras restrições. É possível que diferentes combinações de restrições possam oferecer soluções mais eficazes e melhorar o desempenho da técnica.

Referências Bibliográficas

- [1] Altus. *Desafios da Indústria Alimentícia*. 2024. Disponível em: <https://www.altus.com.br/post/553/desafios-da-industria-alimenticia>. Acesso em: 17 ago. 2024.
- [2] A. S. F. Fabrício, L. A. F. Cabral, e A. Subramanian, *Problema de Dimensionamento de Lotes e Seqüenciamento da Produção: Um Estudo de Caso em uma Indústria de Bebidas em ENEGEP 2007 - XXVII Encontro Nacional de Engenharia de Produção*, Foz do Iguaçu, PR, Brasil, 9-11 de outubro de 2007.
- [3] Borges, Brunno de Oliveira. *A Linguagem Python como Ferramenta no Ensino Básico*. Uberaba-MG: Universidade Federal do Triângulo Mineiro, 2021. Mestrado Profissional em Matemática.
- [4] Catelan, M. C. F., de Araujo, S. A., Fiorotto, D. J. e de Carvalho, D. M. (2020). Heurísticas para o Problema de Dimensionamento de Lotes com Máquinas Paralelas Flexíveis. *Tendências em Matemática Aplicada e Computacional*, 21(2), 313-337. © 2020 Sociedade Brasileira de Matemática Aplicada e Computacional. doi: 10.5540/tema.2020.021.02.0003130313
- [5] Costa, A. C. M., Graça, R. J. S., Mota, C. V. A., Franco, A. A., Muniz, V. H. J., Maia, L. L., & Liese, T. M. (2017). Python: Será que é possível numa Escola Pública de Ensino Médio? In Anais do XXIII Workshop de Informática na Escola (WIE 2017) e VI Congresso Brasileiro de Informática na Educação (CBIE 2017). DOI: 10.5753/cbie.wie.2017.255.
- [6] Dantzig, G. B. e Wolfe, P., "Decomposition Principle for Linear Programs," *Operations Research*, vol. 8, no. 1, pp. 101-111, 1960.

- [7] Fisher, M. L. (1981). The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, 27(1), 1-18.
- [8] G. D. Eppen and R. K. Martin. Solving multi-item capacitated lot-sizing problems using variable redefinition. *Operations Research*, [S. 1.], v.35, n.6, p. 832-848, 1987. ISSN: 0030-364X. DOI: 10.1287/opre.35.6.832.
- [9] H. C. Bahl, L. P. Ritzman and J. N. D. Gupta. 1987. Determining lot sizes and resource Requirements: A Review. *Operations Research*, 35(3), 329-345.
- [10] H. C. Gomes. *Relaxação Lagrangiana com Fixação de Variáveis Aplicada ao Problema de Sequenciamento em uma Máquina*. Universidade Federal de Minas Gerais, Programa de Pós-Graduação em Engenharia de Produção, Departamento de Engenharia de Produção / Escola de Engenharia, Belo Horizonte, Agosto 2008.
- [11] J. Maes, J. O. McClain e L. N. Van Wassenhove. 1991. Multilevel capacitated lot-sizing: Complexity and LP-based heuristic. *European Journal of Operational Research*, 53, 131-148.
- [12] K. Haase. Capacitated lot-sizing with sequence dependent setup costs. *Operations-Research-Spektrum*, [S. 1.], v. 18, n.1, p. 51-59, 1996. ISSN: 0171-6468. DOI: 10.1007/BF01539882.
- [13] L. A. Wolsey, *Integer Programming*, New York: Wiley, 1998.
- [14] L. G. Vassalo e A. R. L. Oliveira, *Problema de Dimensionamento de Lotes*, Iniciação Científica - UNICAMP, 2018.
- [15] Luiz A. N. Lorena, Marcos A. Pereira, and Silvely N. A. Salomão, “A Relaxação Lagrangeana/Surrogate e o Método de Geração de Colunas: Novos Limitantes e Novas Colunas,” Repositório Institucional da UNESP, 2003. Disponível em: <http://dx.doi.org/10.1590/S0101-74382003000100004> e <http://hdl.handle.net/11449/211309>.
- [16] M. A. Medeiros, W. O. Soler e A. F. L. Queiroz, *Um modelo matemático para um problema de planejamento de produção em indústrias alimentícias*, Revista Principia, 2022.

- [17] G. A. B. Marcondes, *Matemática com Python – Um guia prático*, São Paulo, 2018.
- [18] M. Arenales, V. Armentano, R. Morabito e H. Yanasse, *Pesquisa Operacional*, 5th ed., Rio de Janeiro: Elsevier, 2007. ISBN: 8535214543.
- [19] M. C. N. Gramani, J. M. Bourjolly , P. M. França.2002. Problema de dimensionamento de lotes capacitado considerando tempos de preparação - novas abordagens de resolução. In *XXXIV Simpósio Brasileiro de Pesquisa Operacional, 8 a 11 de novembro de 2002, Rio de Janeiro, RJ*.
- [20] Morais, D. C. (2021). *A Linguagem de Programação Python para o Ensino da Matemática*. Florianópolis: Universidade do Sul de Santa Catarina.
- [21] Papalambros, P. Y., & Wilde, D. J. (2000). *A Tutorial on Heuristic Methods*. Technical Report. Technical University of Denmark. Available at: <https://orbit.dtu.dk/en/publications/a-tutorial-on-heuristic-methods>
- [22] P. J. Billington, J. O. McClain e L. J. Thomas. 1983. Mathematical programming approaches to capacity MRP systems: Review, formulation and problem reduction. *Management Science*, 29(10), 1126-1141.
- [23] R. Magari dos Santos, F. C. Fenerich, *Gestão de Estoques para Alimentos Perecíveis: Um Estudo de Modelos Aplicáveis e Levantamento das Variáveis para Parametrização – Parte 2*, 29º Encontro Anual de Iniciação Científica e 9º Encontro Atual de Iniciação Científica Junior, Universidade Estadual de Maringá, PR, 2020. <http://www.eaic.uem.br/eaic2020/anais/artigos/3925.pdf>
- [24] S. A. Araújo, *Modelos e Métodos para o Planejamento e Programação da Produção Aplicados ao Setor de Fundição*. Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação, São Carlos, 2003.
- [25] Santos, M. O. e Armentano, V. A. (2005). O problema de dimensionamento de lotes em máquinas paralelas com restrições de capacidade em sistemas multiestágios. In *XXXVII Simpósio Brasileiro de Pesquisa Operacional, 27 a 30 de setembro, Gramado, RS. Pesquisa Operacional e o Desenvolvimento Sustentável*.

- [26] Shor, N.Z. (1985). *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag.
- [27] Souza, T. C. A. (2008). *Métodos Subgradientes em Otimização Convexa Não Diferenciável*. Dissertação de Mestrado, Universidade Federal de Juiz de Fora, Programa de Pós-Graduação em Modelagem Computacional, Juiz de Fora, MG, Brasil.
- [28] Soler, W. A. de O. (2018). *Análise, proposição e solução de modelos para o problema integrado de dimensionamento de lotes e sequenciamento de produção*. Tese de doutorado, Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo. Disponível em: https://teses.usp.br/teses/disponiveis/55/55134/tde-05022018-161825/publico/WillyAlvesdeOliveiraSoler_revisada.pdf
- [29] S. Q. Santos. *Uma Heurística Lagrangiana para o Problema de Ajuste de Fluxo de Caixa e Sequenciamento de Projetos com Recursos Limitados*. João Monlevade, MG, 2018.
- [30] W. A. O. Soler, M. O. Santos e K. Akartunalı. MIP approaches for a lot sizing and scheduling problem on multiple production lines with scarce resources, temporary workstations, and perishable products. *Journal of the Operational Research Society*, [S. 1.], v. 72, n. 8, p. 1691-1706, 2021. ISSN: 0160-5682. DOI: 10.1080/01605682.2019.1640588.
- [31] W. W. Trigeiro, L. J. Thomas and J. O. McClain. Capacitated lot sizing with setup times. *Management Science*, [S. 1.], v. 35, n. 3, p. 353-366, 1989. ISSN: 0025-1909. DOI:10.1287/mnsc.35.3.353.
- [32] Zanaki, D., & Evans, J. (1981). An inventory model with multiple time periods and stochastic demand. *International Journal of Operations & Production Management*, 11(5), 84–91.