



**UNICAMP**

UNIVERSIDADE ESTADUAL DE  
CAMPINAS

Instituto de Matemática, Estatística e  
Computação Científica

GABRIEL GAZZINELLI GUIMARÃES

**Formulações matemáticas para o problema de  
corte de estoque com limitação no número  
máximo de pilhas abertas**

Campinas

2021

Gabriel Gazzinelli Guimarães

**Formulações matemáticas para o problema de corte de  
estoque com limitação no número máximo de pilhas  
abertas**

Dissertação apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Matemática Aplicada.

Orientadora: Kelly Cristina Poldi

Este trabalho corresponde à versão final da Dissertação defendida pelo aluno Gabriel Gazzinelli Guimarães e orientada pela Profa. Dra. Kelly Cristina Poldi.

Campinas

2021

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca do Instituto de Matemática, Estatística e Computação Científica  
Ana Regina Machado - CRB 8/5467

G947f Guimarães, Gabriel Gazzinelli, 1993-  
Formulações matemáticas para o problema de corte de estoque com  
limitação no número máximo de pilhas abertas / Gabriel Gazzinelli Guimarães.  
– Campinas, SP : [s.n.], 2021.

Orientador: Kelly Cristina Poldi.

Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de  
Matemática, Estatística e Computação Científica.

1. Problema de corte de estoque. 2. Pilhas abertas. 3. Sequenciamento de  
padrões de corte. 4. Problema integrado. 5. Formulação matemática. I. Poldi,  
Kelly Cristina, 1979-. II. Universidade Estadual de Campinas. Instituto de  
Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Mathematical formulations for the cutting stock with limited open  
stacks problem

**Palavras-chave em inglês:**

Cutting stock problem

Open stacks

Cutting pattern sequencing

Integrated problem

Mathematical formulation

**Área de concentração:** Matemática Aplicada

**Titulação:** Mestre em Matemática Aplicada

**Banca examinadora:**

Kelly Cristina Poldi [Orientador]

Maria José Pinto

Horacio Hideki Yanasse

**Data de defesa:** 12-03-2021

**Programa de Pós-Graduação:** Matemática Aplicada

**Identificação e informações acadêmicas do(a) aluno(a)**

- ORCID do autor: <https://orcid.org/0000-0002-5773-5273>

- Currículo Lattes do autor: <http://lattes.cnpq.br/6084481362927084>

**Dissertação de Mestrado defendida em 12 de março de 2021 e aprovada  
pela banca examinadora composta pelos Profs. Drs.**

**Prof(a). Dr(a). KELLY CRISTINA POLDI**

**Prof(a). Dr(a). MARIA JOSÉ PINTO**

**Prof(a). Dr(a). HORACIO HIDEKI YANASSE**

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

# Agradecimentos

À minha orientadora, Dra Kelly Cristina Poldi por todo o suporte dado durante a elaboração deste trabalho, pela disponibilidade e os importantes ensinamentos dados, tanto em âmbito profissional quanto pessoal.

A todas as amigadas construídas durante a minha estadia na UNICAMP, em especial ao Arthur, Gabriel, Thales, Joyce, Marcelo, Felipe, Ricardo Douglas, Luiza, Fernando, e Gabriella. E, também, aos meus amigos de Varginha: Lucas Bazzanella, Ivan, Ricardo, Tomás e Maria Clara.

A toda minha família, por toda ajuda oferecida, em especial a minha avó Teresa, por todo o carinho dispensado a mim, aos meus pais, Ênio e Patricia, por todo apoio dado, aos meus irmãos Clara e Lucas, à minha avó Vânia e ao meu avô Paulo, a quem dedico este trabalho, por sempre me encorajar e apoiar, à quem me espelho como pessoa e profissional.

A todos os funcionários da UNICAMP que contribuíram com este trabalho direta ou indiretamente.

À Rita, por todo amor e cumplicidade.

Ao Dr Horácio Yanasse e Dra Maria José Pinto por todas as importantes sugestões dadas.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

# Resumo

O Problema de Corte de Estoque (PCE) é um problema de otimização que busca determinar como um conjunto de objetos deve ser cortado de forma a satisfazer uma demanda de itens de tamanho específico. O objetivo do PCE é tipicamente reduzir o desperdício de material ou minimizar um custo associado aos objetos cortados. A forma específica na qual um objeto é cortado é denominada por padrão de corte. Dependendo do contexto industrial, a ordem na qual os padrões de corte são executados pode influenciar no processo, neste caso, devemos considerar o problema de sequenciamento dos padrões de corte para minimização do número máximo de pilhas abertas denominado, em inglês, *Minimization of Open Stacks Problem (MOSP)*. Enquanto o PCE visa determinar a frequência na qual os padrões de corte devem ser executados, o MOSP se preocupa em encontrar a ordem de execução dos padrões de corte. O MOSP lida com a situação na qual os itens resultantes da execução de um padrão de corte são empilhados de forma que itens do mesmo tipo pertençam à mesma pilha. Caso todos os padrões de corte contendo um tipo de item específico já tenham sido executados, a pilha referente a este item é denominada uma pilha fechada. O objetivo do MOSP é minimizar o número máximo de pilhas abertas simultaneamente. Apesar da relação entre os dois problemas, eles são frequentemente resolvidos de forma independente na literatura. A primeira formulação matemática para o problema integrado foi proposta em 2004 e, desde então, o problema foi pouco estudado. Motivados pela escassez de modelos e sua importância em aplicações reais propomos, neste trabalho, novas formulações para o problema integrado, assim como melhorias para um dos modelos existentes na literatura. Com o intuito de avaliar a qualidade dos modelos propostos, testes computacionais foram efetuados, objetivando-se determinar a qualidade da relaxação linear de cada formulação e o tempo computacional gasto para encontrar uma solução ótima para instâncias unidimensionais e bidimensionais, comparando diferentes modelos. Os testes computacionais contribuíram para a validade dos modelos e para mostrar sua eficiência na resolução de instâncias com diferentes características. Em particular, a metodologia proposta neste trabalho apresentou melhores resultados para instâncias nas quais o comprimento dos itens é pequeno ou nas quais há uma forte limitação no número máximo de pilhas que podem ser abertas.

**Palavras chave:** problema de corte de estoque, pilhas abertas, sequenciamento de padrões de corte, problema integrado, formulação matemática.

# Abstract

The Cutting Stock Problem (CSP) is an optimization problem that aims to determine the best way in which a set of objects must be cut in order to fulfill a demand for specific size items. The objective of the CSP is typically to reduce waste of material or to minimize costs associated with the objects. The specific way in which an object is cut is called a cutting pattern. Depending on the industrial context, the order in which the cutting patterns are carried out can influence the process. In this case, we must consider the Minimization of Open Stacks Problem (MOSP). While the CSP aims to determine the frequency in which the cutting patterns should be performed, MOSP is concerned with finding the order of execution of the cutting patterns. The MOSP deals with the situation in which the items resulting from the execution of a cut are stacked so that items of the same type belong to the same stack. In case all the cutting patterns containing a specific item type have already been executed, the stack referring to this item is said to be a closed stack. The objective of MOSP is to minimize the maximum number of simultaneously open stacks. Despite the relationship between the two problems, they are often solved independently in the literature. The first mathematical formulation for the integrated problem was proposed in 2004, since then, the problem has been little studied. Motivated by the scarcity of models and the importance in practical applications, in this research, we propose new formulations for the integrated problem, as well as improvements for one of the existing models in the literature. In order to assess the quality of the proposed models, computational tests were performed, aiming to determine the quality of the linear relaxation of each formulation and the time spent to find an optimal solution in instances for both one-dimensional and two-dimensional cases, comparing different models. The computational tests contributed to the validity of the models and to show their efficiency in solving instances with different characteristics. In particular, the methodology proposed in this work showed better results for instances where the length of the items is small or where there is a strong limitation on the maximum number of stacks that can be opened.

**Keywords:** cutting stock problem, open stacks, cutting pattern sequencing, integrated problem, mathematical formulation.

# Lista de ilustrações

Figura 1	– Exemplo de padrão de corte unidimensional. Em (a) temos os objetos a serem cortados, em (b), quatro tipos de itens demandados e, em (c), exemplos de três padrões de corte possíveis. . . . .	26
Figura 2	– 27 padrões de corte encontrados a partir da solução do exemplo apresentado em Exemplo I. Os diversos tipos de itens estão representados segundo o código de cores presente no canto inferior direito da Figura 2. . . . .	28
Figura 3	– Exemplo de padrão de corte bidimensional guilhotinado 2-estágios. A figura ilustra o processo de corte no qual, no primeiro estágio é realizado um corte guilhotinado horizontal e, no segundo estágio, um corte guilhotinado vertical. À direita, estão apresentados os itens finais. . . . .	29
Figura 4	– Exemplo de padrão de corte bidimensional guilhotinado 2-estágios inexato. A figura ilustra a execução dos dois estágios sendo o primeiro um corte guilhotinado horizontal e o segundo um corte guilhotinado vertical. Após a execução dos dois cortes são necessários cortes adicionais para obter os itens requeridos. A parte em preto é referente à "rebarba". À direita se encontram os itens finais após a remoção da "rebarba". . . . .	30
Figura 5	– Padrões de corte que compõe uma solução ótima do exemplo dado em Exemplo II. Os diversos tipos de itens estão representados segundo o código de cores presente no lado direito da Figura 5. . . . .	32
Figura 6	– Fluxograma contendo as etapas do método de geração de colunas. . . . .	33
Figura 7	– Formação de pilhas a partir do sequenciamento dos padrões de corte utilizando um exemplo genérico. A ilustração contém, no seu lado esquerdo, três padrões de corte distintos e, na direita, as pilhas formadas pelos itens contidos nestes padrões. Quando o primeiro objeto é cortado, duas pilhas são abertas, estas são representadas no canto superior direito da figura. Ao cortar o segundo objeto, de acordo com o padrão de corte ilustrado, é necessário abrir uma nova pilha, já que este padrão de corte contém um item de tipo distinto aos que já estão empilhados. O mesmo ocorre com a execução do terceiro padrão de corte, finalizando o processo com quatro pilhas abertas. . . . .	37
Figura 8	– Representação em grafo do modelo de Laporte, Salazar-González e Semet (2004). Está ilustrado um caso no qual há cinco tarefas a serem processadas e o vértice 0 representa a tarefa fictícia. Em (a) temos o grafo completo, ou seja, a primeira e última tarefa a ser executada é a fictícia. À exceção dessa, qualquer tarefa pode suceder ou anteceder qualquer tarefa na ordem de execução. Em (b) temos um circuito completo no grafo com começo e fim no vértice 0. Este circuito representa o caso no qual as tarefas são executadas na sequência 3-1-4-2-5. . . . .	43



Figura 9 – Padrões de corte encontrados em uma solução ótima do Exemplo I, sujeito à restrição de que o número máximo de pilhas abertas seja menor ou igual a dois. Os diversos tipos de itens estão representados segundo o código de cores presente no canto direito da figura. . . . .	58
Figura 10 – Representa o caso no qual há um total de cinco padrões de corte disponíveis para resolução do problema, e apenas 3 destes são utilizados na solução ilustrada. Os padrões não utilizados estão representados em vermelho, enquanto os demais em azul claro, o vértice 0 representa o padrão de corte fictício. Em (a) temos o grafo completo excluindo os arcos que tem conexão com os vértices 2 e 4, ou seja, o primeiro e o último padrão de corte a ser executado é o fictício e os vértices 2 e 4 não devem ser executados em nenhuma ordem, fora isso, qualquer padrão de corte pode suceder ou anteceder qualquer padrão de corte na ordem de execução. Em (b) temos um circuito no grafo que percorre todos os vértices com exceção dos vértices 2 e 4 com começo e fim no vértice 0. Este circuito representa a solução na qual os padrões de corte foram utilizados na ordem 1-5-3 e os padrões de corte 2 e 4 não foram utilizados nesta solução. . . . .	71
Figura 11 – Comparação entre a quantidade de padrões de corte factíveis total e a quantidade de padrões de corte dominantes em função da limitação no número máximo de pilhas abertas considerando os dados da primeira instância da Classe 3. . . . .	97
Figura 12 – Quantidade de padrões de corte factíveis em função da limitação no número máximo de pilhas abertas considerando os dados da primeira instância das Classes 1, 3 e 5. . . . .	104

# Lista de tabelas

Tabela 1	– Abertura de pilhas a partir do sequenciamento dos padrões de corte da Figura 5, já a ordem na qual estes padrões de corte são executados é dada pela sequência $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ . Na primeira coluna, temos a ordem de execução dos padrões de corte. Na segunda coluna, temos quais pilhas estão abertas após a execução do padrão de corte $e$ , na terceira, o total de pilhas abertas. . . . .	37
Tabela 2	– Abertura de pilhas a partir do sequenciamento dos padrões de corte da Figura 5, já a ordem na qual estes padrões de corte são executados é dada pela sequência $3 \rightarrow 6 \rightarrow 5 \rightarrow 1 \rightarrow 4 \rightarrow 2$ . Na primeira coluna, temos a ordem de execução dos padrões de corte. Na segunda coluna, temos quais pilhas estão abertas após a execução do padrão de corte $e$ , na terceira, o total de pilhas abertas. . . . .	38
Tabela 3	– Principais notações utilizadas nos modelos apresentados nesta dissertação. . . . .	40
Tabela 4	– Comparação entre a quantidade de restrições presentes nos conjuntos de restrições propostos em Dantzig, Fulkerson e Johnson (1954) e Miller, Tucker e Zemlin (1960) em função de $N$ , com $N$ variando de 10 até 20. . . . .	56
Tabela 5	– Abertura de pilhas a partir do sequenciamento dos padrões de corte, os padrões de corte presentes na tabela podem ser encontrados na Figura 9, já a ordem na qual estes padrões de corte são executados é dada pela sequência $2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 1$ . Na primeira coluna temos os padrões de corte executados, na segunda coluna temos quais pilhas estão abertas após a execução do padrão de corte $e$ , na terceira, o total de pilhas abertas. . . . .	58
Tabela 6	– Comparação entre o número de variáveis presentes nos modelos YLI, LGSRV e CI2. A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 3, 5 e 6 propostas nesta dissertação. . . . .	89
Tabela 7	– Comparação entre o número de restrições presentes nos modelos YLI, LGSRV e CI2. A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 3, 4 e 6 propostas nesta dissertação. . . . .	94
Tabela 8	– Qualidade da relaxação linear dos modelos YLI, LGSRV e CI2. Os dados contidos na tabela são referente às 10 instâncias das Classes 3, 5 e 6 apresentadas na Seção 6.1. . . . .	95

Tabela 9	– Média de tempo gasto para resolver as 10 instâncias de cada uma das 6 classes propostas, nesta dissertação, para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 2. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ * ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . .	98
Tabela 10	– Resultados em detalhes para as classes propostas para o CS-LOSP unidimensional parcialmente resolvidas utilizando os modelos YLI e LGSRV. A primeira coluna indica a classe, a segunda indica o tempo em segundos para se obter uma solução ótima, a terceira quais instâncias foram resolvidas dentro do tempo limite (7200s) e, a última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando $C = 2$ . . . . .	99
Tabela 11	– Comparação entre o tempo gasto e o número de instâncias resolvidas, em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das Classes 3, 4, 5 e 6, considerando $C = 2$ . . . . .	100
Tabela 12	– Média de tempo gasto para resolver as 10 instâncias de cada uma das 6 classes, propostas nesta dissertação, para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 3. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ * ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . .	100
Tabela 13	– Resultados em detalhes para as classes propostas para o CS-LOSP unidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando $C = 3$ . . . . .	101
Tabela 14	– Comparação entre o tempo gasto e o número de instâncias resolvidas em função da demanda requerida, do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das Classes 3, 4, 5 e 6 utilizando $C = 3$ . . . . .	102

Tabela 15 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 6 classes, propostas nesta dissertação, para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 4. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ * ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . .	102
Tabela 16 – Resultados em detalhes para as classes propostas para o CS-LOSP unidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando $C = 4$ . . . . .	103
Tabela 17 – Comparação entre o tempo gasto e número de instâncias resolvidas em função da demanda requerida, do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das Classes 3, 4, 5 e 6 utilizando $C = 4$ . . . . .	103
Tabela 18 – Número de instâncias resolvidas utilizando os modelos YLI, LGSRV e CI2 em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos. Nesta tabela consideramos os três valores de $C$ utilizados nos testes. . . . .	106
Tabela 19 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 4 classes para o CS-LOSP bidimensional propostas nesta dissertação, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 2. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ * ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . .	110
Tabela 20 – Resultados em detalhes para as classes propostas para o CS-LOSP bidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicada a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando $C = 2$ . . . . .	111

Tabela 21 – Comparação entre o tempo gasto e número de instâncias resolvidas em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das classes para o CS-LOSP bidimensional utilizando $C = 2$ . . . . .	112
Tabela 22 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 4 classes, propostas nesta dissertação, para o CS-LOSP bidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 3. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ * ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . .	112
Tabela 23 – Resultados em detalhes para as classes propostas para o CS-LOSP bidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando $C = 3$ . . . . .	113
Tabela 24 – Comparação entre o tempo gasto e número de instâncias resolvidas em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das classes elaboradas para o CS-LOSP bidimensional utilizando $C = 3$ . . . . .	113
Tabela 25 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 4 classes, propostas nesta dissertação, para o CS-LOSP bidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 4. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ * ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . .	114

Tabela 26	– Resultados em detalhes para as classes propostas para o CS-LOSP bidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima; na terceira coluna, quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando $C = 4$ . . . . .	114
Tabela 27	– Comparação entre o tempo gasto e número de instâncias resolvidas em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das classes propostas para o CS-LOSP bidimensional utilizando $C = 4$ . . . . .	115
Tabela 28	– Número de instâncias resolvidas utilizando os modelos YLI, LGSRV e CI2 em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos. Nesta tabela consideramos os três valores de $C$ utilizados nos testes. . . . .	116
Tabela 29	– Comparação entre o número de variáveis presentes nos modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 1, 3 e 5 propostas neste trabalho. . . . .	139
Tabela 30	– Comparação entre o número de restrições presentes nos modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 1, 3 e 5 propostas neste trabalho. . . . .	143
Tabela 31	– Média de tempo gasto para resolver as 10 instâncias das Classes 1, 2, 3 e 4, propostas nesta dissertação para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 2. Nesta tabela, o caractere “–” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . .	144
Tabela 32	– Resultados em detalhes para as classes parcialmente resolvidas utilizando os modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando $C = 2$ . . . . .	145

- Tabela 33 – Média de tempo gasto para resolver as 10 instâncias das Classes 1, 2, 3 e 4, propostas nesta dissertação para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 3. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ \* ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . . 146
- Tabela 34 – Resultados em detalhes para as classes parcialmente resolvidas utilizando os modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 3$ . 147
- Tabela 35 – Média de tempo gasto para resolver as 10 instâncias das Classes 1, 2, 3 e 4, propostas nesta dissertação para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 4. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ \* ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas. . . . . 148
- Tabela 36 – Resultados em detalhes para as classes parcialmente resolvidas utilizando os modelos YLI, YLI-Modificado, CI2 e CI2-Modificado. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 4$ . 149
- Tabela 37 – Número de instâncias resolvidas utilizando os modelos YLI, YLI-Modificado, CI2 e CI2-Modificado em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos. Nesta tabela consideramos os três valores de  $C$  utilizados nos testes. . . . . 151

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>18</b>
<b>2</b>	<b>O PROBLEMA DE CORTE DE ESTOQUE</b>	<b>22</b>
2.1	Formulação de Kantorovich (1960)	22
2.2	Formulação de Gilmore e Gomory (1961,1963,1965) para o Problema de Corte de Estoque	24
2.2.1	Padrões de corte para o PCE unidimensional	25
2.2.2	Padrões de corte para o PCE bidimensional	29
2.2.3	Modelagem matemática proposta em Gilmore e Gomory (1961, 1963, 1965) para o PCE	30
2.2.4	O método de Geração de Colunas	32
<b>3</b>	<b>O PROBLEMA DE SEQUENCIAMENTO DOS PADRÕES DE CORTE PARA MINIMIZAÇÃO DO NÚMERO MÁXIMO DE PILHAS ABERTAS</b>	<b>36</b>
3.1	O problema de minimização de troca de ferramentas	38
3.2	Notações	39
3.3	Formulação de Yanasse (1997) para o MOSP	41
3.4	Formulação de Pinto (2004) para o MOSP	43
3.5	Formulações de Catanzaro et al. (2015) para o MTSP	46
3.6	Restrições de eliminação de subciclos	55
<b>4</b>	<b>REVISÃO DA LITERATURA SOBRE O PROBLEMA INTEGRADO</b>	<b>57</b>
4.1	Formulação de Yanasse e Lamosa (2007)	59
4.2	Formulação de Arbib, Marinelli e Ventura (2016)	62
4.3	Considerações sobre o problema integrado	68
<b>5</b>	<b>FORMULAÇÕES PROPOSTAS PARA O PROBLEMA INTEGRADO</b>	<b>70</b>
5.1	Formulação Proposta baseada no modelo de Laporte, Salazar-González e Semet (2004)	70
5.1.1	Aplicação das melhorias propostas em Laporte, Salazar-González e Semet (2004)	75
5.2	Formulação proposta baseada nos modelos de Catanzaro, Golveia e Labbé (2015)	76
5.3	Formulação proposta baseada no modelo de Kantorovich (1960)	81
<b>6</b>	<b>RESULTADOS COMPUTACIONAIS</b>	<b>85</b>



6.1	Conjunto de dados para o CS-LOSP unidimensional . . . . .	86
6.2	Quantidade de variáveis . . . . .	87
6.3	Quantidade de restrições . . . . .	89
6.4	Análise da qualidade da relaxação linear . . . . .	94
6.5	Resultados para as instâncias do CS-LOSP unidimensional . . . . .	95
6.6	Resultados para as instâncias do CS-LOSP bidimensional . . . . .	108
6.7	Conjunto de dados para o CS-LOSP bidimensional . . . . .	108
6.8	Resultados para as instâncias do CS-LOSP bidimensionais. . . . .	109
7	MELHORIAS . . . . .	118
7.1	Yanasse e Lamosa Integrado-Modificado . . . . .	130
7.2	Catanzaro Integrado II-Modificado . . . . .	134
7.3	Quantidade de variáveis . . . . .	137
7.4	Quantidade de restrições . . . . .	139
7.5	Resultados computacionais . . . . .	143
8	CONSIDERAÇÕES FINAIS . . . . .	153
8.1	Perspectivas de trabalho . . . . .	155
	REFERÊNCIAS . . . . .	156

# 1 Introdução

O Problema de Corte de Estoque (PCE) é alvo de estudo de diversos pesquisadores. O interesse neste problema em específico se dá devido a sua importante aplicação prática em diversos setores industriais, além de ser interessante do ponto de vista teórico, uma vez que representa um problema de otimização combinatória de difícil resolução. O PCE visa otimizar o corte de peças maiores em itens de tamanho menor, pretendendo satisfazer uma demanda de itens preestabelecida enquanto busca-se, ao mesmo tempo, atender a algum objetivo, como por exemplo, reduzir o desperdício de material ou um custo associado aos objetos cortados. A forma específica na qual um objeto é cortado é denominada por padrão de corte.

Do nosso conhecimento, o primeiro estudo que temos reportado na literatura sobre o PCE, data de 1939, quando o matemático e economista Leonid Vitaliyevich Kantorovich identificou em seu artigo, [Kantorovich \(1960\)](#), a necessidade de se otimizar o corte de objetos em peças menores. O artigo original foi publicado em russo, em 1939, e posteriormente foi republicado, em inglês, na revista *Management Science*, em 1960. Desde então, o problema tem sido extensivamente pesquisado e diversos modelos foram propostos visando resolver o PCE de maneira eficiente.

Dentre as formulações propostas para a resolução do PCE, uma das mais conhecidas e utilizadas é a de [Gilmore e Gomory\(1961, 1963, 1965\)](#). Nesta modelagem, supõe-se que todos os padrões de corte possíveis são conhecidos, dados os parâmetros do problema. Dependendo do tamanho da instância do problema, a cardinalidade deste conjunto pode ser excessivamente grande, dificultando a resolução do problema em tempo razoável. Devido a inviabilidade de se trabalhar com todos os padrões de corte diretamente, nos artigos [Gilmore e Gomory\(1961, 1963, 1965\)](#) é proposto a utilização de um método de geração de colunas. Este método tem como estratégia selecionar os padrões de corte de maior utilidade para resolver o PCE sem precisar utilizar todos os padrões de corte diretamente.

A alocação dos itens obtidos a partir da cortagem dos objetos representa um desafio para a produção industrial. Motivado por esta dificuldade, surgiu o problema de sequenciamento dos padrões de corte para minimização do número máximo de pilhas abertas, em inglês, *Minimization of Open Stacks Problem*(MOSP). Uma estratégia frequentemente utilizada para manejar o armazenamento dos itens consiste em, ao realizar o corte dos objetos, dispor os itens obtidos em pilhas, separando cada tipo distinto de item em uma pilha diferente. No momento em que não há mais nenhum padrão de corte contendo um tipo de item específico a ser executado, a pilha referente a este tipo de item pode

ser fechada, caso contrário, a pilha deste item é considerada aberta. O MOSP considera este contexto, objetivando-se determinar em qual ordem os padrões de corte devem ser executados de maneira a minimizar o número máximo de pilhas abertas simultaneamente. Os modelos para o MOSP estudados neste trabalho derivam do Problema de Minimização de Trocas de Ferramentas, em inglês, *Minimization of Tool Switches Problem* (MTSP). Devido a relação entre os dois problemas, o MTSP também é abordado na dissertação.

O MTSP está frequentemente associado ao uso de máquinas flexíveis. Em [Browne et al. \(1984\)](#) os autores definem a flexibilidade de uma máquina como a facilidade que a máquina tem em realizar alterações necessárias com o intuito de gerar um determinado conjunto de peças. Para os autores, a flexibilidade de máquina está associada ao tempo gasto para realizar trocas de ferramentas, ou substituir ferramentas quebradas ou desgastadas. Outros autores definem a flexibilidade de uma máquina de forma diferente, em [Mohamed, Youssef e Huq \(2001\)](#) outras definições para a flexibilidade de máquinas são abordadas. A incorporação de máquinas flexíveis na cadeia de produção industrial oferece a importante vantagem de aumentar a adaptabilidade do sistema, permitindo que diferentes tarefas sejam executadas por uma mesma máquina. O MTSP considera uma máquina flexível que possui uma limitação no número de ferramentas que podem ser utilizadas ao mesmo tempo. Esta máquina deve ser usada para processar um conjunto de tarefas, que exigem um conjunto particular de ferramentas para sua execução, podendo ser necessário realizar trocas das ferramentas para processar todas as tarefas. As trocas de ferramentas demandam tempo e, portanto, o MTSP busca minimizar o número de trocas de ferramentas com o intuito de evitar atrasos no processamento das tarefas.

Em [Yanasse \(1997\)](#), o autor estuda a relação entre MTSP e o MOSP e propõe uma formulação para o MOSP baseada na formulação proposta por [Tang e Denardo \(1988\)](#) para o MTSP. Em [Pinto \(2004\)](#), outra formulação para o MOSP é apresentada e tal formulação utiliza a mesma estratégia de [Yanasse \(1997\)](#). O modelo proposto [Pinto \(2004\)](#) baseia-se na formulação proposta em [Laporte, Salazar-González e Semet \(2004\)](#) para o MTSP.

O problema de corte de estoque e o problema de minimização de pilhas abertas podem ser resolvidos em sequência. Ou seja, em primeiro lugar, uma solução para o PCE é encontrada. Esta solução fornece um conjunto de padrões de corte que é usado para satisfazer a demanda do PCE, enquanto otimiza algum objetivo específico. Feito isso, o conjunto de padrões de corte obtido é ordenado de forma a reduzir ao máximo o número de pilhas abertas simultaneamente através da resolução do MOSP. Entretanto, resolver os problemas de forma independente nem sempre apresenta bons resultados. Em [Arbib, Marinelli e Ventura \(2016\)](#) são apresentadas duas desvantagens da abordagem de resolução independente, sendo estas:

- Resolver os dois problemas de maneira independente não é o mesmo que encontrar

uma solução que minimize a abertura de pilhas ao mesmo tempo em que reduz o desperdício de material. A qualidade da solução obtida a partir da resolução do MOSP depende fortemente do conjunto de padrões de corte encontrado ao se resolver o PCE. É possível que existam diversos conjuntos de padrões de corte que possuam o mesmo valor de função objetivo para o PCE, mas que apresentem resultados distintos para o MOSP. Neste caso, resolver os problemas de forma independente não permite determinar o conjunto de padrões de corte que otimize os dois objetivos pretendidos.

- Caso haja um limite no número máximo de pilhas que podem ser abertas, então a solução do problema de corte de estoque pode não apresentar um sequenciamento factível.

Em [Pinto \(2004\)](#) foi proposto um modelo para resolução integrada do PCE e MOSP. O modelo construído pela autora é uma composição da formulação proposta em [Gilmore e Gomory\(1961, 1963, 1965\)](#) e o modelo elaborado em [Yanasse \(1997\)](#). A formulação resultante foi posteriormente publicada em [Yanasse e Lamosa \(2007\)](#). Nestes trabalhos, os autores adicionam ao problema de corte de estoque a restrição de que o número máximo de pilhas abertas durante o sequenciamento dos padrões de corte não pode ultrapassar um limite predefinido, este problema é conhecido na literatura como *Cutting Stock with Limited Open Stacks Problem (CS-LOSP)*.

A quantidade de variáveis e restrições presentes no modelo proposto em [Pinto \(2004\)](#) é dependente da quantidade de tipos distintos de itens e do número total de padrões de corte. Mesmo para instâncias relativamente pequenas, a quantidade de padrões de corte pode ser demasiadamente grande, de forma a inviabilizar a resolução do problema. Esta dificuldade é parcialmente contornada em [Arbib, Marinelli e Ventura \(2016\)](#), que propõem um modelo com um número significativamente menor de restrições. Enquanto em [Pinto \(2004\)](#) o número de restrições crescia exponencialmente em função da quantidade de tipos distintos de item, o modelo elaborado em [Arbib, Marinelli e Ventura \(2016\)](#) cresce de forma quadrática. Apesar de reduzir drasticamente o número de restrições, este modelo não foi capaz de fazer o mesmo em relação à quantidade de variáveis, sendo assim, o modelo ainda se torna excessivamente grande para instâncias maiores. Outra vantagem do modelo é a possibilidade de resolver a relaxação linear da formulação utilizando métodos de geração de colunas.

Desta forma, pode ser observado que o CS-LOSP é um problema pouco abordado na literatura e os modelos existentes apresentam resultados insatisfatórios para instâncias maiores, inviabilizando a determinação de uma solução ótima para alguns problemas. Neste trabalho, novas formulações para o CS-LOSP são propostas realizando um procedimento similar ao elaborado em [Pinto \(2004\)](#). Os modelos desenvolvidos neste trabalho baseiam-se nos modelos de [Gilmore e Gomory\(1961, 1963, 1965\)](#) e [Kantorovich \(1960\)](#) para lidar

com a parte do problema referente ao problema de corte de estoque, já para tratar do sequenciamento dos padrões de corte são utilizados os modelos baseados nas formulações propostas em [Tang e Denardo \(1988\)](#) e em [Laporte, Salazar-González e Semet \(2004\)](#) para o MTSP, propostas por [Pinto \(2004\)](#) e [Yanasse \(1997\)](#), respectivamente, além de novas formulações para o MOSP, baseadas nos modelos construídos em [Catanzaro, Golveia e Labbé \(2015\)](#).

Além de propor novas formulações, nesta dissertação são sugeridas algumas melhorias, inclusive para o modelo proposto em [Yanasse e Lamosa \(2007\)](#). Estas melhorias têm como principal objetivo reduzir o número de variáveis e restrições presentes no modelo.

A eficiência das formulações propostas nesta dissertação é comparada com o modelo apresentado em [Yanasse e Lamosa \(2007\)](#). Para determinar a qualidade dos modelos três fatores são avaliados, sendo estes: a qualidade da relaxação linear, a quantidade de variáveis e restrições e o tempo exigido para encontrar uma solução ótima de um conjunto de problemas. A limitação no número máximo de pilhas que podem ser abertas é um parâmetro de grande importância no CS-LOSP, podendo alterar significativamente a quantidade de material desperdiçado, ou o tempo computacional exigido para se encontrar uma solução ótima, seja  $C$  este limite. Nesta dissertação, buscamos determinar como o parâmetro  $C$  interfere no tempo computacional gasto para se encontrar uma solução ótima, assim como seu impacto na eficiência dos modelos, desta maneira, três valores de  $C$  são utilizados nos testes computacionais. Os valores adotados para  $C$  foram:  $C = 2$ ,  $C = 3$  e  $C = 4$ .

Para avaliar a qualidade dos modelos, assim como a validade das melhorias propostas, dois conjuntos de dados são propostos para efetuar os testes computacionais. O primeiro contém instâncias unidimensionais e, o segundo, instâncias bidimensionais. Na prática, o processo de corte pode estar ligado a estações de descarregamento automático. Sendo assim, existe um número finito (e relativamente pequeno) de estações de descarregamento, o que torna o problema de minimização de pilhas abertas extremamente importante.

## 2 O problema de corte de estoque

O Problema de Corte de Estoque (PCE) ou *Cutting Stock Problem* (CSP), em inglês, é um problema de otimização no qual peças maiores (objetos) são cortadas de forma a satisfazer a demanda de um conjunto de itens de tamanhos específicos. Existem diversas possibilidades para a função objetivo a ser otimizada, como minimizar o custo dos objetos cortados ou o desperdício de material. Há um grande interesse da esfera industrial no problema devido a sua aplicação em uma grande variedade de setores.

O PCE pode ser classificado de acordo com a dimensão do problema. Problemas nos quais os objetos a serem cortados e os itens requeridos só possuem uma dimensão relevante no processo de corte são ditos unidimensionais. Caso o problema possua duas dimensões relevantes ele é dito bidimensional e, em caso de ter três dimensões relevantes, é dito tridimensional. Alguns exemplos de problemas de corte de estoque unidimensionais incluem o corte de bobinas de papel ou aço. Já o problema de corte de estoque bidimensional está frequentemente associado ao corte de placas de madeira ou vidro.

A primeira formulação para o problema que se tem notícia foi estabelecida em [Kantorovich \(1960\)](#), desde então, o problema tem sido objeto de estudo de diversos autores. Neste capítulo, são revisadas em detalhes as formulações propostas em [Kantorovich \(1960\)](#) e [Gilmore e Gomory \(1961, 1963, 1965\)](#), além destas, também é revisto o problema de construção de padrões de cortes, dado sua relação com o PCE, especialmente quando trabalhamos com a formulação de [Gilmore e Gomory \(1961, 1963, 1965\)](#).

Outras formulações para o PCE incluem as propostas em [Wolsey \(1977\)](#), [Carvalho \(1999\)](#) e [Macedo, Alves e Carvalho \(2010\)](#). Os modelos propostos em [Wolsey \(1977\)](#) e [Carvalho \(1999\)](#) se baseiam em um problema de fluxo em redes para construir um modelo para o PCE. O modelo proposto em [Carvalho \(1999\)](#) aplica-se exclusivamente a PCEs unidimensionais, mas foi estendido para o caso bidimensional em [Macedo, Alves e Carvalho \(2010\)](#).

### 2.1 Formulação de Kantorovich (1960)

A primeira formulação da qual se tem conhecimento na literatura para o PCE foi desenvolvida em [Kantorovich \(1960\)](#), no qual o corte é realizado sobre cada objeto separadamente. Desta forma mesmo padrões de corte idênticos são considerados individualmente. Para valores altos de demanda, tratar cada objeto isoladamente aumenta consideravelmente a complexidade computacional do problema. Considere:

**Índices:**

$N$  : quantidade de objetos disponíveis;

$M$  : quantidade de itens distintos.

**Parâmetros:**

$L$  : comprimento dos objetos;

$l_i$  : comprimento do item do tipo  $i$ ;

$d_i$  : a demanda do item do tipo  $i$ .

**Variáveis de decisão:**

$$y_j = \begin{cases} 1, & \text{se o objeto } j \text{ foi utilizado,} \\ 0, & \text{caso contrário;} \end{cases}$$

$x_{ij}$  : número de itens do tipo  $i$  que são obtidos ao se cortar o objeto  $j$ .

O modelo de [Kantorovich \(1960\)](#) é dado então por:

(Kantorovich)

$$\min \sum_{j=1}^N y_j \tag{2.1}$$

$$\text{sujeito a: } \sum_{j=1}^N x_{ij} \geq d_i, \quad i = 1, \dots, M, \tag{2.2}$$

$$\sum_{i=1}^M l_i x_{ij} \leq L y_j, \quad j = 1, \dots, N, \tag{2.3}$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, N, \tag{2.4}$$

$$x_{ij} \in Z^+, \quad i = 1, \dots, M, j = 1, \dots, N. \tag{2.5}$$

A função objetivo (2.1) minimiza a quantidade total de objetos utilizados. As restrições (2.2) são relativas ao atendimento da demanda dos itens. O conjunto de restrições (2.3) garante que a soma do comprimento de todos os itens obtidos ao se cortar o objeto não ultrapasse o comprimento do objeto, além disso, garante que, caso o objeto  $j$  não seja utilizado, então, nenhum item pode ser obtido a partir dele. As restrições (2.4) e (2.5) são relativas ao domínio das variáveis.

Uma solução ótima para o problema (2.2)-(2.5) determina quantos objetos devem ser cortados ( $y_j$ ) e como tais os objetos serão cortados ( $x_{ij}$ ). Inicialmente, podemos determinar uma estimativa para o número de objetos necessários para resolver o PCE. Assim, seja  $N$  um valor suficientemente grande para que possa atender a demanda de todos os itens. Note que  $l_i \leq L, i = 1, \dots, M$ , do contrário, o item não poderia ser obtido, impossibilitando a satisfação da demanda. Sendo assim, cada objeto é capaz de fornecer ao menos um item de qualquer tipo que seja. Partindo desse pressuposto podemos tomar

$$N = \sum_{i=1}^M d_i.$$

Esta é uma estimativa grosseira para o número de objetos necessários para a resolução do PCE, observe que, caso os itens sejam muito menores que os objetos, a quantidade de itens que pode ser obtida ao cortar um objeto é muito maior que um. O valor de  $N$  está relacionado com a quantidade de variáveis no problema, logo, é interessante minimizar o valor de  $N$ . É possível obter uma estimativa mais refinada para  $N$  observando que a partir do corte de um objeto de comprimento  $L$  podem ser obtidos, no máximo,  $\lfloor L/l_i \rfloor$  itens do tipo  $i$  e, portanto, a demanda deste tipo de item pode ser atendida utilizando  $\lceil d_i / \lfloor L/l_i \rfloor \rceil$  objetos. Repetindo este raciocínio para todo  $i$  de 1 até  $M$ , encontramos uma melhor estimativa para  $N$ :

$$N = \sum_{i=1}^M \left\lceil \frac{d_i}{\lfloor L/l_i \rfloor} \right\rceil \quad (2.6)$$

A formulação de Kantorovich foi proposta para problemas de corte nos quais todos os objetos a serem cortados têm mesmo comprimento  $L$  e o objetivo do problema é de minimizar a quantidade de objetos usados. Entretanto, ela pode ser estendida para tratar casos com diferentes comprimentos de objetos.

## 2.2 Formulação de Gilmore e Gomory (1961,1963,1965) para o Problema de Corte de Estoque

O modelo proposto em Gilmore e Gomory (1961, 1963, 1965) é um dos mais conhecidos na literatura, enquanto na formulação de Kantorovich (1960), cada objeto



era cortado individualmente, o modelo de Gilmore e Gomory (1961, 1963, 1965) visa encontrar uma frequência na qual os padrões de corte são executados. Por outro lado, a construção deste modelo supõe o conhecimento prévio de todos os padrões de corte. Para alguns tamanhos de problemas, pode ser difícil trabalhar com todos os padrões de corte explicitamente. Assim, em Gilmore e Gomory (1961, 1963, 1965) os autores utilizam um método de geração de colunas, no qual somente os padrões de corte que melhoram a solução são determinados, iterativamente. Esta estratégia garante a otimalidade da solução encontrada.

Para melhor entender o funcionamento desta formulação esta seção é dividida em 3 etapas. Na primeira, o foco está na obtenção dos padrões de corte, uma vez que estes são essenciais na modelagem do problema. Nesta etapa é detalhado como obter os padrões de corte para os PCEs unidimensional e bidimensional. Na etapa seguinte, o modelo de Gilmore e Gomory (1961, 1963, 1965) é revisado e, por último, é apresentado o método de geração de colunas aplicado ao problema.

### 2.2.1 Padrões de corte para o PCE unidimensional

A forma específica na qual um objeto é cortado visando satisfazer a demanda dos itens é conhecida como padrão de corte. Quando apenas uma dimensão é relevante na solução do problema, a única restrição que deve ser considerada é referente às restrições físicas de comprimento do objeto. Considere:

#### Índices:

$M$  : quantidade de itens de tipos distintos;

#### Parâmetros:

$l_i$  : comprimento do item do tipo  $i$ ,  $i = 1, \dots, M$ ;

$L$  : comprimento do objeto.

#### Variáveis de decisão:

$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_M)^t$  : representa um padrão de corte. A  $i$ -ésima coordenada do vetor corresponde à quantidade de itens do tipo  $i$  que está presente no padrão de corte.

Um vetor  $\alpha$  é um padrão de corte unidimensional se e somente se satisfaz as seguintes restrições:

$$\sum_{i=1}^M l_i \alpha_i \leq L, \quad (2.7)$$

$$\alpha_i \in \mathbb{Z}^+, \quad i = 1, \dots, M. \quad (2.8)$$

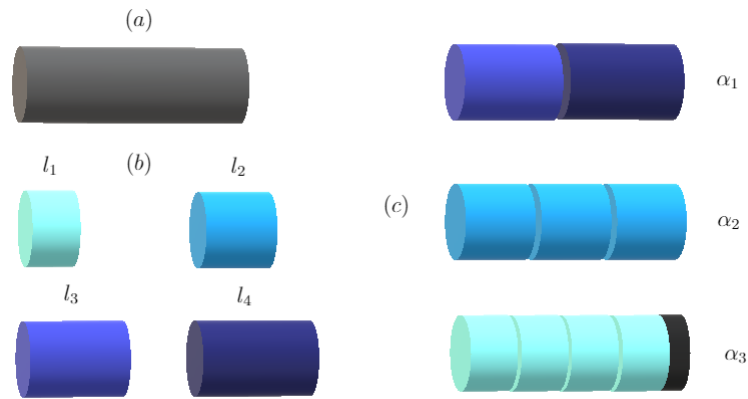


Figura 1 – Exemplo de padrão de corte unidimensional. Em (a) temos os objetos a serem cortados, em (b), quatro tipos de itens demandados e, em (c), exemplos de três padrões de corte possíveis.

Os padrões de corte unidimensionais podem ser determinados através da resolução de um Problema da Mochila. A ideia do problema é modelar matematicamente a situação na qual uma mochila com capacidade limitada deve ser preenchida com objetos que possuem pesos e valores distintos. A única restrição do problema é que a soma dos pesos dos objetos que são colocados na mochila não devem ultrapassar sua capacidade máxima. O objetivo é de maximizar o valor total dos objetos alocados à mochila respeitando a restrição de capacidade. Considere então:

### Índices:

$M$  : quantidade de tipos distintos de itens.

### Parâmetros:

$l_i$  : peso/comprimento do item do tipo  $i$ ;

$L$  : capacidade máxima da mochila/objeto;

$\pi_i$  : valor de utilidade do item do tipo  $i$ .

### Variáveis de decisão:

$\alpha_i$  : quantidade do item do tipo  $i$  que é inserido na mochila.

O modelo para o Problema da Mochila é dado então por:

$$\max \sum_{i=1}^M \pi_i \alpha_i \quad (2.9)$$

$$\text{sujeito a: } \sum_{i=1}^M l_i \alpha_i \leq L, \quad (2.10)$$

$$\alpha_i \in Z^+, \quad i = 1, \dots, M. \quad (2.11)$$

A função objetivo (2.9) maximiza a utilidade dos itens alocados à mochila, a restrição (2.10) garante que a limitação na capacidade da mochila seja respeitada, e as restrições (2.11) são referentes ao domínio das variáveis. O Problema da Mochila é estudado com mais detalhes no livro [Martello e Toth \(1990\)](#).

Observe que as restrições (2.10) e (2.11) são idênticas às restrições (2.7) e (2.8), referentes ao corte dos padrões de corte, sendo assim, toda solução do Problema da Mochila determina um padrão de corte, portanto, basta variar os valores de  $\pi$  para obter todos os padrões possíveis. Para ilustrar esta relação, apresentamos um exemplo do PCE com alguns padrões de corte.

**Exemplo:** Considere um objeto de comprimento  $L = 6,5$  e suponha que temos seis itens distintos com comprimentos  $l_1 = 1$ ,  $l_2 = 1,5$ ,  $l_3 = 2$ ,  $l_4 = 2,7$ ,  $l_5 = 3,2$ ,  $l_6 = 4$ . O Problema da Mochila pode ser utilizado para encontrar as maneiras distintas nas quais o objeto pode ser cortado visando obter os itens supracitados. Inserindo os parâmetros do exemplo no Problema da Mochila temos:

(Exemplo I)

$$\begin{aligned} & \max \sum_{i=1}^6 \pi_i \alpha_i \\ \text{sujeito a: } & \alpha_1 + 1,5\alpha_2 + 2\alpha_3 + 2,7\alpha_4 + 3,2\alpha_5 + 4\alpha_6 \leq 6,5, \\ & \alpha_i \in \mathbb{Z}^+, \quad i = 1, \dots, 6. \end{aligned}$$

Variando os valores de  $\pi_i$ , podemos encontrar padrões de corte distintos. Por exemplo para  $\pi_1 = 1$  e  $\pi_i = 0, i \neq 1$ , a solução ótima equivale a cortar o objeto de forma a obter 6 itens do tipo 1. Assim encontramos o padrão de corte  $\alpha^{(1)} = (6, 0, 0, 0, 0, 0)^t$ . Enquanto para  $\pi_1 = \pi_2 = \pi_6 = 1$  e  $\pi_3 = \pi_4 = \pi_5 = 0$ , o padrão de corte obtido é  $\alpha^{(2)} = (1, 1, 0, 0, 0, 1)^t$ . Resolvendo o problema para diversos valores de  $\pi_i$ , 27 padrões de corte são encontrados para este problema. Estes padrões de corte são apresentados na [Figura 2](#):

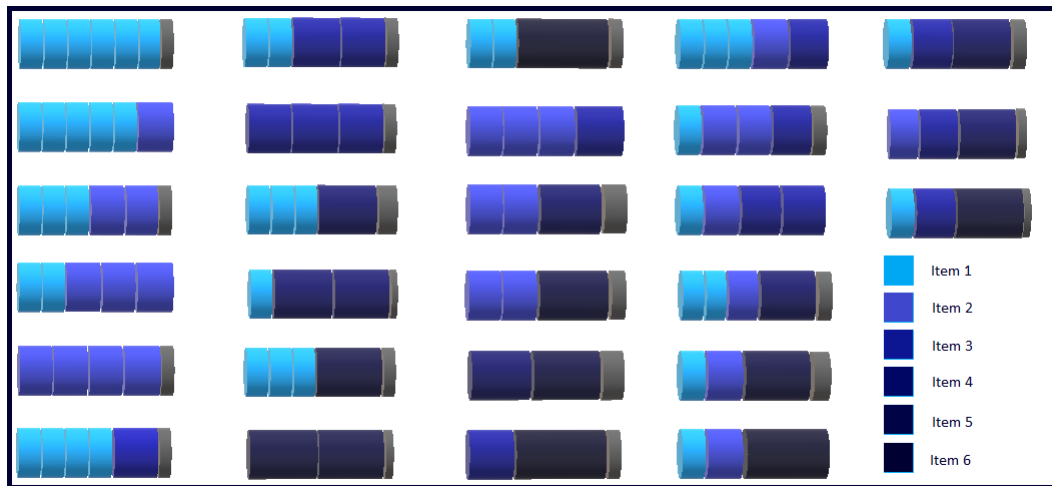


Figura 2 – 27 padrões de corte encontrados a partir da solução do exemplo apresentado em [Exemplo I](#). Os diversos tipos de itens estão representados segundo o código de cores presente no canto inferior direito da [Figura 2](#).

Note que mesmo para um número pequeno de itens a quantidade de padrões de corte é relativamente alta e este número pode se tornar muito maior dependendo dos parâmetros do problema. Para problemas reais, considerar todos os padrões de corte explicitamente pode ser inviável e, com isto, outras estratégias devem ser utilizadas para contornar o problema. Em [Gilmore e Gomory \(1961, 1963, 1965\)](#) é proposto um método de geração de colunas, conforme será apresentado na Subseção [2.2.4](#), para lidar com o grande número de padrões de corte presentes em um PCE.

## 2.2.2 Padrões de corte para o PCE bidimensional

O caso bidimensional é semelhante ao caso unidimensional estudado na Seção 2.2.1, com a diferença que, neste caso, tanto o objeto a ser cortado, quanto os itens que se deseja obter, possuem duas dimensões relevantes, tipicamente comprimento e largura. Apesar de ser uma mudança aparentemente simples, alguns cuidados adicionais devem ser tomados, uma vez que neste caso não apenas as restrições físicas do problema devem ser observadas, mas também restrições referentes ao tipo de corte executado se tornam importantes para a resolução do problema. Em função destas alterações não é possível utilizar os mesmos métodos de solução apresentados na Seção 2.2.1. A construção de padrões de corte bidimensionais é consideravelmente mais trabalhosa, quando comparada ao caso unidimensional.

Ao realizar um corte pode-se exigir que este seja feito de um extremo a outro do objeto, sem ser interrompido. Esta imposição aparece na prática devido a características da máquina que realiza o corte. Este tipo de corte é conhecido como corte guilhotinado. No caso de objetos retangulares um corte guilhotinado é dito ortogonal se este é ortogonal a um dos lados do retângulo que está sendo cortado. Um padrão de corte obtido através de cortes guilhotinados é um padrão de corte guilhotinado.

Em muitas aplicações práticas, um padrão de corte guilhotinado é dividido em estágios. Em cada estágio, os cortes são realizados na mesma direção, sobre o objeto inteiro ou sobre pedaços dele obtidos a partir de cortes realizados no estágio anterior. Mudar a direção dos cortes pode ser dispendioso pois é preciso girar a máquina ou o objeto consumindo tempo. Sendo assim, a estratégia de se limitar o número de estágios pode ser empregada para otimizar o processo. Esta restrição reduz a quantidade de padrões de corte considerados no problema, por outro lado, a quantidade de material desperdiçado pode aumentar. Quando o número de estágios está limitado a  $k$ , o corte guilhotinado é chamado de *k-estagiado*.

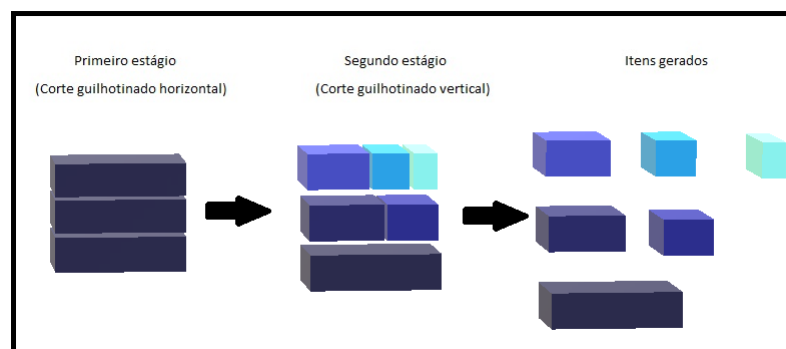


Figura 3 – Exemplo de padrão de corte bidimensional guilhotinado 2-estágios. A figura ilustra o processo de corte no qual, no primeiro estágio é realizado um corte guilhotinado horizontal e, no segundo estágio, um corte guilhotinado vertical. À direita, estão apresentados os itens finais.

Nem sempre é possível obter os itens requeridos a partir da execução dos dois cortes (horizontal e vertical). Caso exista a necessidade de executar cortes adicionais sobre os itens obtidos após a execução dos 2-estágios então o padrão de corte guilhotinado 2-estágios é denominado de inexato, caso contrário, o padrão de corte é denominado de exato. A sobra de material resultante dos cortes adicionais que não é utilizada para obtenção de nenhum novo item é chamada de "rebarba". Um padrão de corte 2-estágios exato é apresentado na [Figura 3](#) enquanto na [Figura 4](#), a seguir, é apresentado um padrão de corte 2-estágios inexato:

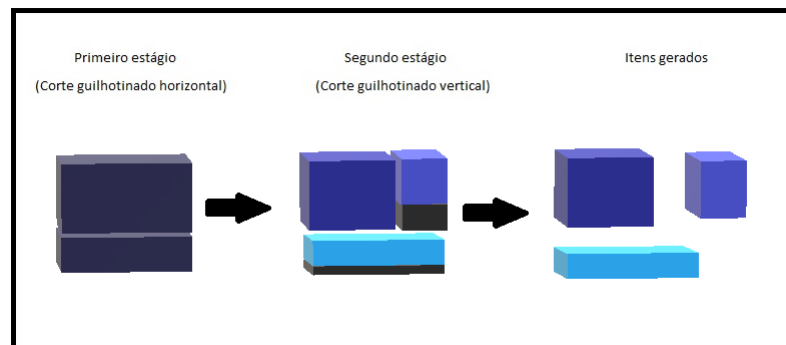


Figura 4 – Exemplo de padrão de corte bidimensional guilhotinado 2-estágios inexato. A figura ilustra a execução dos dois estágios sendo o primeiro um corte guilhotinado horizontal e o segundo um corte guilhotinado vertical. Após a execução dos dois cortes são necessários cortes adicionais para obter os itens requeridos. A parte em preto é referente à "rebarba". À direita se encontram os itens finais após a remoção da "rebarba".

### 2.2.3 Modelagem matemática proposta em Gilmore e Gomory (1961, 1963, 1965) para o PCE

Nas seções anteriores, o foco estava no problema de construção de padrões de corte, utilizados na modelagem do problema. Agora, apresentamos o modelo de [Gilmore e Gomory \(1961, 1963, 1965\)](#), utilizado para determinar a quantidade de vezes que cada padrão de corte deve ser utilizado para atender a demanda requerida. Considere:

#### Índices:

$N$  : quantidade de padrões de corte encontrados;

$M$  : quantidade de itens distintos.

#### Parâmetros:

$\alpha^{(j)} = (\alpha_1^{(j)}, \alpha_2^{(j)}, \dots, \alpha_M^{(j)})^t$  : representa o  $j$ -ésimo padrão de corte. A  $i$ -ésima coordenada do vetor corresponde à quantidade de itens do tipo  $i$  que está presente no padrão de corte;

$c_j$  : custo de se cortar um objeto utilizando o padrão de corte  $j$ ;

$d_i$  : demanda do item do tipo  $i$ .

### Variáveis de decisão:

$p_j$  : quantidade de vezes que o padrão de corte  $j$  é utilizado.

O modelo de [Gilmore e Gomory \(1961, 1963, 1965\)](#) é dado então por:

$$\min \sum_{j=1}^N c_j p_j \quad (2.12)$$

$$\text{sujeito a: } \sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i = 1, \dots, M, \quad (2.13)$$

$$p_j \in Z^+ \quad j = 1, \dots, N. \quad (2.14)$$

A função objetivo (2.12) visa minimizar o custo total associado à execução dos padrões de corte, as restrições (2.13) garantem que a demanda dos itens seja atendida e, o conjunto de restrições (2.14), é referente ao domínio das variáveis.

A presença de variáveis inteiras dificulta a resolução do problema. Além disso, mesmo para um número relativamente pequeno de itens, o número de padrões de corte pode ser muito grande, dificultando a resolução do problema de forma exata. Para lidar com a dificuldade do problema, Gilmore e Gomory sugerem primeiramente relaxar a condição de integralidade das variáveis e, em seguida, aplicar uma técnica de geração de colunas para resolver o problema.

Retomando o exemplo dado no ([Exemplo I](#)) podemos formular um problema de corte a partir dos 27 padrões de corte obtidos anteriormente e apresentados na [Figura 2](#). Seja  $\mathbf{d} = (131, 114, 33, 16, 14, 96)$ , a demanda requerida dos itens de 1 até 6,  $\mathbf{A}$ , uma matriz contendo os vetores  $\alpha_j$  e  $\mathbf{p}$ , um vetor contendo as variáveis  $p_j$ . O PCE a ser resolvido é

dado, então, por:

(Exemplo II)

$$\min \sum_{j=1}^N c_j p_j$$

sujeito a:  $\mathbf{A}\mathbf{p} \geq \mathbf{d}$ ,

$$p_j \in \mathbb{Z}^+.$$

Este exemplo tem como solução:  $p_1 = 3, p_2 = 7, p_3 = 1, p_4 = 8, p_5 = 96, p_6 = 15$  e o restante igual a zero, sendo os padrões de corte associados:  $\alpha^{(1)} = (5, 1, 0, 0, 0, 0)^t$ ,  $\alpha^{(2)} = (0, 0, 0, 0, 2, 0)^t$ ,  $\alpha^{(3)} = (0, 0, 3, 0, 0, 0)^t$ ,  $\alpha^{(4)} = (1, 0, 0, 2, 0, 0)^t$ ,  $\alpha^{(5)} = (1, 1, 0, 0, 0, 1)^t$ ,  $\alpha^{(6)} = (1, 1, 2, 0, 0, 0)^t$ . Estes padrões de corte são apresentados na [Figura 5](#).

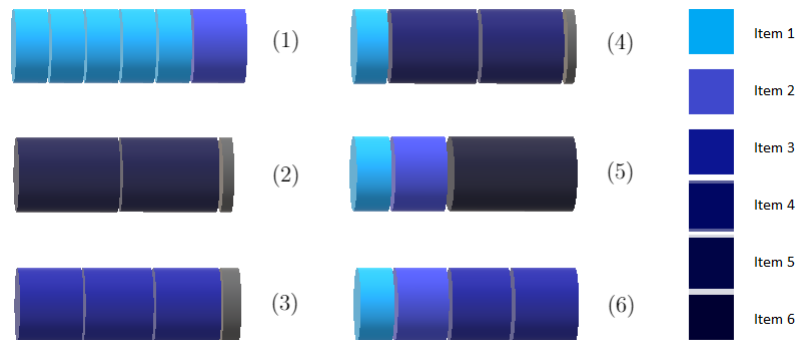


Figura 5 – Padrões de corte que compõem uma solução ótima do exemplo dado em [Exemplo II](#). Os diversos tipos de itens estão representados segundo o código de cores presente no lado direito da [Figura 5](#).

Note que dos 27 padrões de corte factíveis somente seis deles foram utilizados na solução final.

### 2.2.4 O método de Geração de Colunas

O método de geração de colunas pode ser aplicado para resolver a relaxação linear do PCE, de forma a obter um limitante inferior para o problema inteiro. A ideia deste método consiste em dividir o PCE em dois problemas, o problema mestre e o subproblema. O problema mestre equivale ao PCE iniciado com um subconjunto de padrões de corte factíveis, de forma a possuir ao menos uma solução factível. Este problema é resolvido e, em seguida, as variáveis duais do problema mestre são utilizadas para construir o subproblema.



O subproblema é, então, utilizado para verificar se existe algum padrão de corte (coluna) que melhore a qualidade da solução quando inserido no problema mestre.

Se não houver nenhum padrão de corte com tal propriedade, uma solução ótima atual do problema mestre é também solução ótima do PCE original. Caso contrário, a coluna encontrada através do subproblema é inserida no problema mestre e o processo é repetido. As etapas do método de geração de colunas são ilustradas no fluxograma da Figura 6.

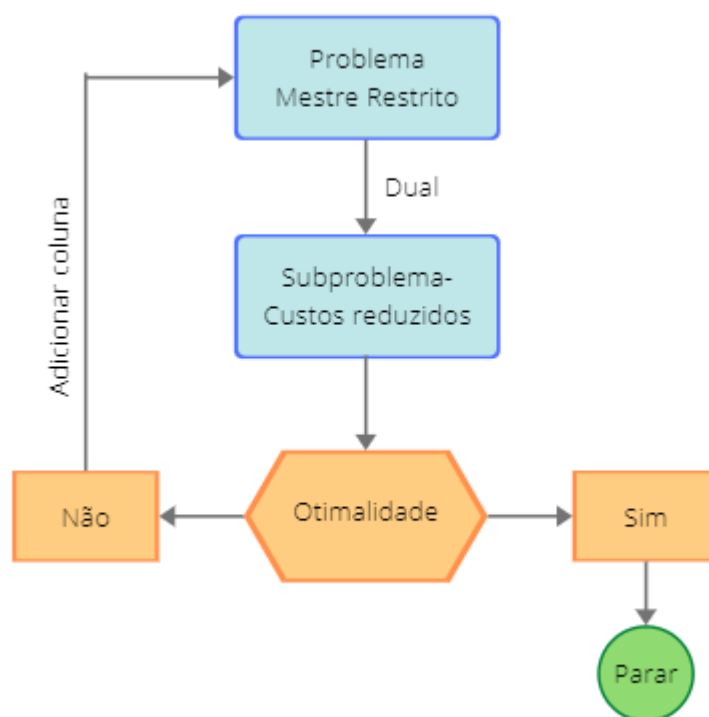


Figura 6 – Fluxograma contendo as etapas do método de geração de colunas.

**Problema Mestre Restrito (PMR):** Seja  $J$  o conjunto de todos os padrões de corte presentes em um problema de corte de estoque. Dependendo do problema, o conjunto  $J$  pode ser difícil de ser determinado. Além disso, em alguns casos, a cardinalidade de  $J$  pode ser excessivamente grande, dificultando consideravelmente a resolução do problema. Para superar estes obstáculos, o Problema Mestre Restrito trabalha com um subconjunto de  $J$ , tal que seja possível encontrar ao menos uma solução factível para o PCE a partir deste conjunto. Excluindo esta diferença, a formulação para o PMR se assemelha muito à formulação para o PCE dada na Seção 2.2.3. Considerando  $J_{PMR} \subset J$  como o conjunto de padrões de corte do problema mestre, o PMR é dado por:

$$\begin{aligned}
& \min \sum_{j \in J_{PMR}} c_j p_j \\
\text{sujeito a: } & \sum_{j \in J_{PMR}} \alpha_i^{(j)} p_j \geq d_i, & i = 1, \dots, M, \\
& p_j \geq 0, & j \in J_{PMR}.
\end{aligned}$$

**Subproblema:** O subproblema e o problema mestre restrito são problemas interligados. Primeiro devemos resolver o PMR e, então, o valor das variáveis duais é usado para se resolver o subproblema e determinar se existe alguma coluna atrativa para ser inserida no PMR. Conforme pode ser visto em [Bazaraa Mokhtar S. e Sherali \(2011\)](#), em um problema de otimização linear, se todos os custos reduzidos são maiores ou iguais a zero, então a solução é ótima. Sendo assim, dada uma solução factível qualquer, se:

$$c_k - \pi^t a_k \geq 0 \quad \forall k \rightarrow \text{esta solução encontrada é ótima.}$$

O subproblema tem como objetivo encontrar algum padrão de corte que, quando inserido no problema mestre, melhore a qualidade de sua solução ótima. A estratégia adotada é de encontrar a coluna  $\hat{a}_k$  que minimize os custos reduzidos. Caso  $c_k - \pi^t \hat{a}_k \geq 0, \forall k$  então a solução ótima foi obtida. Caso contrário, inserimos a coluna  $\hat{a}_k$  no problema mestre e repetimos o procedimento. A função objetivo do subproblema é dada então por:

$$\min \{c_k - \pi^t a_k\}.$$

Note que  $a_k$  deve representar um padrão de corte válido para o PCE, portanto, as colunas  $a_k$  estão sujeitas às restrições que definem um padrão de corte. Por exemplo, para o caso unidimensional, estas restrições são dadas por (2.7). Logo o subproblema é dado por:

$$\min c_k - \pi^t a_k \tag{2.15}$$

Sujeito a:

$$a_k \text{ é um padrão de corte,}$$

Se o objetivo do problema é de minimizar a quantidade de objetos utilizados de forma a satisfazer a demanda requerida, então  $c_k = 1, \forall k$ , logo, (2.15) pode ser substituído por:

$$1 - \max \{\pi^t a_k\},$$

e, portanto, o subproblema pode ser reescrito como:

$$\max \pi^t a_k$$

Sujeito a:

$a_k$  é um padrão de corte.

Neste capítulo definimos o PCE e revisamos o modelo de [Gilmore e Gomory \(1961, 1963, 1965\)](#) e o modelo de [Kantorovich \(1960\)](#), ambos propostos para a solução do PCE. Também foi revisado o método de geração de colunas aplicado ao modelo de [Gilmore e Gomory \(1961, 1963, 1965\)](#). No próximo capítulo, abordamos o problema de minimização de pilhas abertas.

### 3 O problema de sequenciamento dos padrões de corte para minimização do número máximo de pilhas abertas

O foco deste capítulo está em revisar o problema de sequenciamento dos padrões de corte para minimização do número máximo de pilhas abertas em, inglês, *Minimization of Open Stacks Problem* (MOSP). Em [Yanasse \(1997\)](#) o autor demonstra que o MOSP e o problema de minimização de troca de ferramentas (MTSP) são problemas relacionados e propõe uma formulação para o MOSP a partir de um modelo elaborado para o MTSP.

Neste capítulo, são revisadas as formulações propostas em [Yanasse \(1997\)](#) e [Pinto \(2004\)](#). Em [Yanasse \(1997\)](#), o autor se baseia na formulação de [Tang e Denardo \(1988\)](#), proposta para o MTSP, para construir um modelo para o MOSP. De forma similar, o modelo proposto em [Pinto \(2004\)](#) se baseia na formulação de [Laporte, Salazar-González e Semet \(2004\)](#) para elaborar um outro modelo para o MOSP.

Os três modelos estudados no capítulo derivam de formulações para o MTSP, devido à conexão entre os dois problemas, o problema de minimização de troca de ferramentas também é estudado, juntamente com a sua relação com o MOSP.

Na prática, muitas vezes o espaço para armazenamento intermediário de itens cortados ao redor da máquina de corte pode ser limitado. Pode ocorrer também que, após o corte do objeto em itens, o maquinário automático descarregue os itens cortados em esteiras ou em paletes que também são limitados. Assim, é extremamente importante buscar um sequenciamento dos padrões de corte que possa viabilizar a saída otimizada dos itens. Consideramos que somente itens idênticos podem pertencer a uma mesma pilha. Supomos também que uma pilha só é fechada após o corte de todos os padrões de corte que possuem o tipo de item empilhado. Desta forma, denominamos esse problema por problema de sequenciamento dos padrões de corte para minimização de pilhas abertas. Na [Figura 7](#) é ilustrada a formação de pilhas após os três padrões de corte ilustrados serem cortados em sequencia, resultando num total de quatro pilhas abertas. Na [Figura 7](#) nenhuma pilha é removida durante a execução dos padrões de corte.

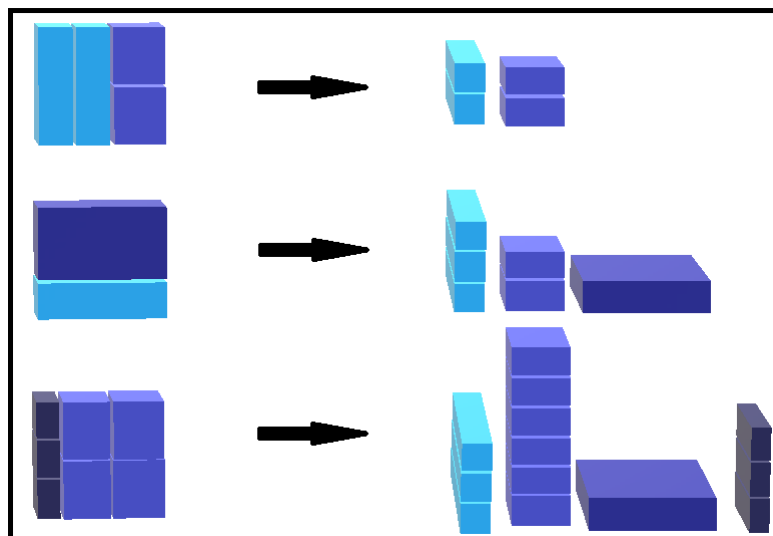


Figura 7 – Formação de pilhas a partir do sequenciamento dos padrões de corte utilizando um exemplo genérico. A ilustração contém, no seu lado esquerdo, três padrões de corte distintos e, na direita, as pilhas formadas pelos itens contidos nestes padrões. Quando o primeiro objeto é cortado, duas pilhas são abertas, estas são representadas no canto superior direito da figura. Ao cortar o segundo objeto, de acordo com o padrão de corte ilustrado, é necessário abrir uma nova pilha, já que este padrão de corte contém um item de tipo distinto aos que já estão empilhados. O mesmo ocorre com a execução do terceiro padrão de corte, finalizando o processo com quatro pilhas abertas. .

Como já foi discutido anteriormente, os padrões de corte a serem sequenciados são fornecidos pela solução de um problema de corte de estoque. Tome então os padrões de corte presentes na Figura 5 e a sequência  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ , ou seja, executar os padrões de corte na ordem em que foram apresentados, sem fazer nenhum esforço adicional para se determinar um sequenciamento que otimize o número máximo de pilhas abertas. Na Tabela 1 é ilustrada a quantidade de pilhas abertas em cada etapa do sequenciamento:

Sequência $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$		
Padrão de corte executado	Pilhas abertas	Total de pilhas abertas
1	1-2	2
2	1-2-5	3
3	1-2-3	3
4	1-2-3-4	4
5	1-2-3-6	4
6	1-2-3	3

Tabela 1 – Abertura de pilhas a partir do sequenciamento dos padrões de corte da Figura 5, já a ordem na qual estes padrões de corte são executados é dada pela sequência  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6$ . Na primeira coluna, temos a ordem de execução dos padrões de corte. Na segunda coluna, temos quais pilhas estão abertas após a execução do padrão de corte e, na terceira, o total de pilhas abertas.

Logo o número máximo de pilhas abertas ao utilizar a sequência  $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

→ 5 → 6 é 4. Note que após executar o terceiro padrão de corte a demanda do item do tipo 5 é satisfeita, logo a pilha referente a este item é fechada e pode ser retirada. De forma similar, após executar o quarto padrão de corte a pilha referente ao item do tipo 4 é fechada. De forma análoga, consideremos a sequência **3 → 6 → 5 → 1 → 4 → 2** que está apresentada na [Tabela 2](#):

Sequência <b>3 → 6 → 5 → 1 → 4 → 2</b>		
Padrão de corte executado	Pilhas abertas	Total de pilhas abertas
3	3	1
6	1-2-3	3
5	1-2-6	3
1	1-2	2
4	1-4	2
2	5	1

Tabela 2 – Abertura de pilhas a partir do sequenciamento dos padrões de corte da [Figura 5](#), já a ordem na qual estes padrões de corte são executados é dada pela sequência **3 → 6 → 5 → 1 → 4 → 2**. Na primeira coluna, temos a ordem de execução dos padrões de corte. Na segunda coluna, temos quais pilhas estão abertas após a execução do padrão de corte e, na terceira, o total de pilhas abertas.

Como pode ser visto na [Tabela 2](#), o valor máximo de pilhas abertas é 3 quando consideramos a sequência **3 → 6 → 5 → 1 → 4 → 2**. Ou seja, ao modificarmos a sequência na qual os padrões de corte são executados a quantidade máxima de pilhas abertas pode ser reduzida. Desta forma, o objetivo do MOSP é determinar uma sequência dos padrões que minimize o total de pilhas abertas.

### 3.1 O problema de minimização de troca de ferramentas

Assim como o MOSP, o problema de minimização de troca de ferramentas, em inglês, *Minimization of Tool Switches Problem* (MTSP) também visa encontrar um sequenciamento ótimo. A diferença entre os dois problemas é que ao invés de sequenciar os padrões de corte, o problema de minimização de troca de ferramentas busca determinar a melhor sequência para executar um conjunto de tarefas, cada uma delas necessitando de um conjunto específico de ferramentas para seu processamento. A motivação deste problema vem do fato que a máquina tem uma limitação no que se refere à quantidade de ferramentas que podem ser utilizadas simultaneamente. Sendo assim, se faz necessária a troca das ferramentas que estão na máquina para processar todas as tarefas. Estas trocas não são desejáveis pois, toda vez que uma ferramenta precisa ser retirada ou inserida na máquina, o processo tem de ser paralisado, inviabilizando temporariamente a execução

das tarefas e, conseqüentemente, atrasando o processamento das tarefas. Desta forma, o problema busca ordenar a execução das tarefas de forma a reduzir ao máximo a troca de ferramentas realizadas no decorrer do processo.

Em [Beezao \(2016\)](#) são discutidas duas variações do MTSP. A primeira delas busca sequenciar um conjunto de tarefas com tempo específico de processamento e que requer um conjunto particular de ferramentas para sua execução. Enquanto no MTSP todas as tarefas são realizadas em uma única máquina, aqui são utilizadas  $p$  máquinas idênticas e paralelas. Esse problema é conhecido, em inglês, por *Identical Parallel Machines Problem with Tooling Constraints* (IPMTC). Este problema é discutido em detalhes em [Beezao et al. \(2017\)](#).

A outra variante abordada em [Beezao et al. \(2017\)](#) trata do problema de escalonamento de tarefas *job shop* com troca de ferramentas, do inglês, *Job Shop Scheduling Problem with Tooling Constraints* (JSSPTC). O problema foi proposto por [Widmer \(1991\)](#) e busca sequenciar um conjunto de tarefas em um sistema flexível de manufatura, conforme descrito em [Zeballos \(2010\)](#), com  $p$  máquinas em um ambiente *job shop*. O objetivo do problema é minimizar o tempo total gasto para realizar todas as tarefas.

## 3.2 Notações

Para auxiliar na apresentação dos modelos abordados nesta dissertação, usamos a notação apresentada na [Tabela 3](#) que contém principais notações utilizadas no decorrer do texto.

Notações	
$C$	Capacidade de ferramentas que a máquina suporta simultaneamente no contexto do MTSP e, no contexto do MOSP, $C$ se refere ao limite na quantidade máxima de pilhas que podem estar abertas simultaneamente.
$M$	Quantidade total de ferramentas que são necessárias para processar todas as tarefas no contexto do MTSP. Já no contexto do MOSP, $M$ se refere ao número de itens de tipos distintos.
$T$	Conjunto de ferramentas necessárias para processar todas as tarefas no contexto do MTSP e conjunto de itens demandados no contexto do MOSP.
$t$	Elemento do conjunto $T$ , que se refere à uma ferramenta no contexto do MTSP e a um item no MOSP.
$T_i$	Conjunto de ferramentas necessárias para processar a tarefa $i$ no contexto do MTSP e conjunto de itens gerados ao executar o padrão de corte $i$ , no contexto do MOSP.
$J$	Conjunto de tarefas para o MTSP e conjunto de padrões de corte para o MOSP.
$J'$	Conjunto de tarefas incluindo a tarefa fictícia no contexto do MTSP e conjunto de padrões de corte incluindo o padrão de corte fictício, para o MOSP.
$J_c$	Conjunto de tarefas que precisam de exatamente $C$ ferramentas para seu processamento no contexto do MTSP. Enquanto no contexto do MOSP, $J_c$ representa um padrão de corte que gera $C$ itens.
$J_t$	Conjunto de tarefas que necessitam da ferramenta $t$ para seu processamento no contexto do MTSP. Enquanto no contexto do MOSP, $J_t$ representa um conjunto de padrões de corte que geram o item $t$ .
$j$	Elemento do conjunto $J$ , se refere à uma tarefa no contexto do MTSP e a um padrão de corte no MOSP.
$K$	Conjunto de posições nas quais as tarefas são executadas para o MTSP e conjunto de posições nas quais os padrões de corte são utilizados para o MOSP.
$k$	Posição na qual uma tarefa é executada para o MTSP e posição na qual um padrão de corte é utilizado para o MOSP.
$S$	Subconjunto próprio de $J$ .
$N$	Número total de tarefas quando consideramos o MTSP e quantidade de padrões de corte distintos quando consideramos o MOSP.

Tabela 3 – Principais notações utilizadas nos modelos apresentados nesta dissertação.



### 3.3 Formulação de Yanasse (1997) para o MOSP

De acordo com Yanasse (1997), uma solução ótima do MTSP representa um sequenciamento de padrões de corte que abre no máximo  $C$  pilhas se e somente se o valor da solução do problema de trocas de ferramenta for igual a  $M - C$ . Desta forma, podemos considerar as ferramentas como itens e as tarefas como os padrões de corte obtendo uma formulação para o MOSP a partir de um modelo formulado para o MTSP. Yanasse usou em seu artigo o modelo de Tang e Denardo (1988) para este fim. Nesta seção, a formulação proposta em Tang e Denardo (1988) para o MTSP e o modelo elaborado em Yanasse (1997) para o MOSP são revisados. Considere:

**Variáveis de decisão:**

$$z_{jt} = \begin{cases} 1, & \text{se a ferramenta } t \text{ é inserida na máquina para o processamento da} \\ & j\text{-ésima tarefa,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$w_{ij} = \begin{cases} 1, & \text{se a tarefa } i \text{ é a } j\text{-ésima a ser processada,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{jt} = \begin{cases} 1, & \text{se a ferramenta } t \text{ está na máquina durante o processamento da} \\ & j\text{-ésima tarefa,} \\ 0, & \text{caso contrário.} \end{cases}$$

A formulação de Tang e Denardo (1988) para o MTSP é dada por:

(Tang e Denardo)

$$\min \sum_{j=2}^N \sum_{t \in T} z_{jt} \tag{3.1}$$

$$\text{sujeito a: } \sum_{i=1}^N w_{ij} = 1, \quad j = 1, \dots, N \tag{3.2}$$

$$\sum_{j=1}^N w_{ij} = 1, \quad i = 1, \dots, N \tag{3.3}$$

$$w_{ij} \leq y_{jt}, \quad i \in J_t, j = 1, \dots, N, t \in T \tag{3.4}$$

$$\sum_{t \in T} y_{jt} \leq C, \quad j = 1, \dots, N \tag{3.5}$$

$$y_{jt} - y_{(j-1)t} \leq z_{jt}, \quad j = 2, \dots, N, t \in T \tag{3.6}$$

$$w_{ij} \in \{0, 1\}, \quad i = 1, \dots, N, j = 1, \dots, N \quad (3.7)$$

$$y_{jt} \in \{0, 1\}, \quad j = 1, \dots, N, t \in T \quad (3.8)$$

$$z_{jt} \in \{0, 1\}, \quad j = 2, \dots, N, t \in T. \quad (3.9)$$

A função objetivo (3.1) minimiza o número total de trocas de ferramentas realizadas. Os conjuntos de restrições (3.2) e (3.3) garantem que todas as tarefas sejam processadas e que as tarefas sejam processadas uma por vez. O conjunto de restrições (3.4) obriga que todas as ferramentas necessárias para processar a tarefa  $i$  estejam na máquina durante a sua execução. As restrições (3.5) impedem que o limite da capacidade da máquina seja ultrapassado e as restrições (3.6) asseguram a contabilização das trocas de ferramentas na variável  $z_{jt}$  sempre que necessário. As restrições (3.7), (3.8) e (3.9) definem o domínio das variáveis.

O modelo (Tang e Denardo) é utilizado em Yanasse (1997) para construir uma formulação para o MOSP. Neste artigo, o autor altera a interpretação das variáveis e parâmetros, de forma que as ferramentas passem a ser itens e as tarefas se tornem padrões de corte, ao observar que, caso o valor da função objetivo em uma solução ótima seja igual a  $M - C$ , ou seja:

$$\sum_{j=2}^N \sum_{t \in T} z_{jt} = M - C.$$

Uma solução ótima do MTSP representa um sequenciamento de padrões de corte que abre no máximo  $C$  pilhas. Desta maneira, em Yanasse (1997), é adicionada ao modelo de Tang e Denardo (1988) a restrição:

$$\sum_{j=2}^N \sum_{t \in T} z_{jt} = M - C.$$

Esta restrição garante que uma solução ótima do problema forneça um sequenciamento no qual, no máximo,  $C$  pilhas sejam abertas. A função objetivo da formulação (Tang e Denardo) é então modificada de forma a minimizar  $C$  e, desta forma, é obtido o modelo de Yanasse (1997) para o problema de minimização de pilhas abertas:

$$\begin{aligned} & \min C \\ \text{sujeito a: } & \sum_{j=2}^N \sum_{t \in T} z_{jt} = M - C, \end{aligned}$$

E às restrições (3.2)-(3.9).

### 3.4 Formulação de Pinto (2004) para o MOSP

Em Pinto (2004) a autora se baseia na formulação de Laporte, Salazar-González e Semet (2004) para construir uma formulação para o MOSP, utilizando o mesmo procedimento proposto por Yanasse (1997) e apresentado na seção anterior. Nesta seção, é revisado o modelo de Laporte, Salazar-González e Semet (2004) para o MTSP, assim como sua extensão para o MOSP, proposta em Pinto (2004).

O modelo proposto em Laporte, Salazar-González e Semet (2004) se baseia no problema do caixeiro viajante, no qual as tarefas são representadas por vértices em um grafo. A ordem na qual percorremos os vértices em um *tour* equivale ao sequenciamento das tarefas. Para o funcionamento do modelo é estabelecida uma tarefa fictícia que representa o começo e o fim das operações.

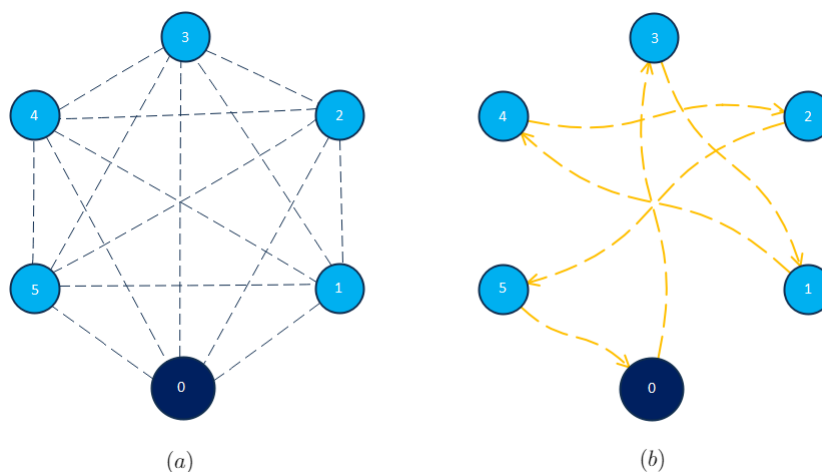


Figura 8 – Representação em grafo do modelo de Laporte, Salazar-González e Semet (2004). Está ilustrado um caso no qual há cinco tarefas a serem processadas e o vértice 0 representa a tarefa fictícia. Em (a) temos o grafo completo, ou seja, a primeira e última tarefa a ser executada é a fictícia. À exceção dessa, qualquer tarefa pode suceder ou anteceder qualquer tarefa na ordem de execução. Em (b) temos um circuito completo no grafo com começo e fim no vértice 0. Este circuito representa o caso no qual as tarefas são executadas na sequência 3-1-4-2-5.

Considere:

**Variáveis de decisão:**

$$z_{it} = \begin{cases} 1, & \text{se a ferramenta } t \text{ é inserida na máquina para o processamento da tarefa } i, \\ 0, & \text{caso contrário;} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{se o processamento da tarefa } j \text{ ocorre imediatamente após o processamento} \\ & \text{da tarefa } i, \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{it} = \begin{cases} 1, & \text{se a ferramenta } t \text{ está na máquina durante o processamento da } i\text{-ésima tarefa,} \\ 0, & \text{caso contrário.} \end{cases}$$

O modelo de Laporte, Salazar-González e Semet (2004) para o MTSP é dado então por:

(Laporte, Salazar-González e Semet)

$$\min \sum_{i \in J} \sum_{t \in T_i} z_{it} \quad (3.10)$$

$$\text{sujeito a: } \sum_{\substack{j \in J' \\ j \neq i}} x_{ij} = 1, \quad i \in J' \quad (3.11)$$

$$\sum_{\substack{i \in J' \\ i \neq j}} x_{ij} = 1, \quad j \in J' \quad (3.12)$$

$$y_{it} = 1, \quad i \in J, t \in T_i \quad (3.13)$$

$$\sum_{t \in T} y_{it} \leq C, \quad i \in J \quad (3.14)$$

$$x_{ij} + y_{jt} - y_{it} \leq z_{jt} + 1, \quad i \in J', j \in J, t \in T \quad (3.15)$$

$$\sum_{i, j \in S} x_{ij} \leq |S| - 1, \quad S \subset J', 2 \leq |S| \leq |J'| - 1 \quad (3.16)$$

$$z_{it} = 0, \quad i \in J, t \in T \setminus T_i \quad (3.17)$$

$$x_{ij} \in \{0, 1\}, \quad i \in J', j \in J' \quad (3.18)$$

$$y_{it} \in \{0, 1\}, \quad i \in J, t \in T \quad (3.19)$$

$$z_{it} \in \{0, 1\}, \quad i \in J, t \in T. \quad (3.20)$$

A função objetivo (3.10) minimiza o número total de trocas de ferramentas realizadas. As restrições (3.11) garantem que uma, e apenas uma, tarefa seja processada imediatamente após a tarefa  $i$ , para todo  $i \in J'$ . O conjunto de restrições (3.12) faz com que para cada  $j \in J'$  exista uma, e apenas uma, tarefa  $i$  que anteceda o processamento da tarefa  $j$ . As restrições (3.13) garantem que todas as ferramentas necessárias para processar a tarefa  $i$  estejam na máquina durante sua execução. O conjunto de restrições (3.14) impede que o limite da capacidade da máquina seja ultrapassado. O conjunto de restrições (3.15) assegura a contagem da troca de ferramentas. Note que se a tarefa  $j$  ocorre imediatamente após a tarefa  $i$ , então  $x_{ij} = 1$ . Logo, se a ferramenta está alocada na máquina no instante  $j$  e não estava no instante  $i$ , temos  $y_{jt} = 1$  e  $y_{it} = 0$ , fazendo com que  $z_{it} = 1$  se  $y_{jt} = 0$  ou  $x_{ij} = 0$ , então a restrição é trivialmente satisfeita. As restrições (3.16) eliminam a formação de subciclos durante a execução das tarefas conforme pode ser visto em Dantzig, Fulkerson e Johnson (1954). As restrições (3.17) impedem que uma ferramenta seja colocada na máquina se ela não é requerida para processar a próxima tarefa. As restrições (3.18), (3.19) e (3.20) são relativas ao domínio das variáveis.

De maneira análoga ao que foi feito em Yanasse (1997), em Pinto (2004) o modelo de Laporte, Salazar-González e Semet (2004) é utilizado para construir uma formulação para o MOSP. O modelo proposto em Pinto (2004) para o MOSP é dado por:

$$\begin{aligned} & \min C \\ & \text{sujeito a: } \sum_{i \in J} \sum_{t \in T} z_{it} \leq M - C, \\ & \text{E às restrições (3.11)-(3.20).} \end{aligned}$$

Em conjunto com o modelo (Laporte, Salazar-González e Semet), em Laporte, Salazar-González e Semet (2004) foi elaborado um conjunto de restrições que visa melhorar a qualidade da relaxação linear do modelo (Laporte, Salazar-González e Semet). A primeira delas se refere ao caso especial no qual uma tarefa exige exatamente  $C$  ferramentas para seu processamento. Seja  $J_c$  o conjunto de tarefas com esta propriedade,  $i \in J_c$  e  $j$  sua sucessora, então o número de trocas entre  $i$  e  $j$  equivale ao número de ferramentas necessárias para o processamento de  $j$  que não são necessárias para o processamento de  $i$ , ou ainda,  $|T_j \setminus T_i|$ . Podemos então, substituir (3.10) por:

$$\min \sum_{i=1}^N \sum_{t=1}^M z_{it} + \sum_{i \in J_c} \sum_{j \neq i} |T_j \setminus T_i| \tag{3.21}$$

e eliminar as restrições (3.17) para os índices  $\{(i, j, t); i \in J_c, j \neq i \text{ e } t \in T_j\}$ . Como a função objetivo é de minimização e, para este conjunto de índices, as variáveis  $z_{it}$  não estão sujeitas à nenhuma restrição além das restrições de domínio dadas por (3.20),  $z_{it}$  sempre

assume o menor valor possível, no caso, recebe o valor 0. Desta forma, podemos manter o somatório com todos os índices na função objetivo. Outras restrições que melhoram o valor da relaxação linear são:

$$\sum_{i \in J_t \setminus \{j\}} x_{ij} + z_{it} \leq 1, \quad t \in T, j \in J. \quad (3.22)$$

No conjunto de restrições (3.22), se a tarefa  $j$  é processada após a tarefa  $i$  então  $x_{ij} = 1$  e, dessa forma, a restrição garante  $z_{it} = 0$ . Note que  $i \in J_t$  e, portanto, requer a ferramenta  $t$  para seu processamento, ou seja, a restrição garante que, neste caso, a ferramenta permanece na máquina após a execução da tarefa  $i$  e, portanto, não precisa ser inserida. Se a tarefa  $j$  não ocorre imediatamente após uma tarefa  $i$  que exige a ferramenta  $t$ , então não há garantias que a ferramenta está no magazine da máquina e, portanto, pode ser necessária sua inserção, ou seja,  $z_{it}$  é livre pra assumir qualquer valor.

$$\sum_{t \in T_i \setminus T_j} y_{jt} \geq (C - |T_j|)x_{ij}, \quad i \in J_c, j \in J : |T_j| < C. \quad (3.23)$$

O conjunto de restrições (3.23) garante que, caso a tarefa  $j$  seja processada após a tarefa  $i$ , o magazine da máquina seja preenchido com as ferramentas necessárias para a execução da tarefa  $j$  e herde da tarefa anterior as ferramentas que são necessárias para processar a tarefa  $i$ , mas que não são necessárias para processar a tarefa  $j$ .

$$\sum_{t \in T_j} z_{jt} \geq \sum_{\substack{i \in J \setminus \{j\} \\ |T_i| \neq C}} l_{ij} x_{ij}, \quad j \in J, . \quad (3.24)$$

Para o conjunto de restrições (3.24), considerando  $l_{ij} = \max\{0, |T_i \cup T_j|\}$ , se o processamento da tarefa  $j$  é feito imediatamente após o processamento da tarefa  $i$ ,  $l_{ij}$  fornece um limitante inferior para a quantidade de trocas de ferramentas executadas entre as tarefas  $i$  e  $j$ .

### 3.5 Formulações de Catanzaro et al. (2015) para o MTSP

Amparado pelo modelo de Laporte, Salazar-González e Semet (2004), no artigo Catanzaro, Golveia e Labbé (2015) os autores propõem uma nova formulação para o MTSP. No artigo citado, o modelo é construído de forma gradual, três novas formulações são propostas, cada uma delas derivando da anterior. A partir da implementação sucessiva de melhorias nas construções teóricas, os novos modelos são gerados.

O primeiro modelo parte da construção de Laporte, Salazar-González e Semet (2004) e modifica as variáveis presentes no modelo. Em Catanzaro, Golveia e Labbé (2015) é demonstrado que o modelo obtido é mais forte que a formulação de Laporte,

Salazar-González e Semet (2004). Considere:

**Variáveis de decisão:**

$$x_{ij} = \begin{cases} 1, & \text{se a execução da tarefa } j \text{ ocorre imediatamente após a execução da} \\ & \text{tarefa } i, \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{ij}^t = \begin{cases} 1, & \text{se a ferramenta } t \text{ permanece na máquina durante a transição da} \\ & \text{tarefa } i \text{ para a tarefa } j, \\ 0, & \text{caso contrário;} \end{cases}$$

$$z_{ij}^t = \begin{cases} 1, & \text{se a ferramenta } t \text{ é inserida na máquina na transição da tarefa } i \\ & \text{para a tarefa } j, \\ 0, & \text{caso contrário.} \end{cases}$$

O modelo de Catanzaro et al. I para o MTSP é dado, então, por:

(Catanzaro et al. I)

$$\min \sum_{\substack{i,j \in J' \\ i \neq j}} \sum_{t \in T_j} z_{ij}^t \quad (3.25)$$

$$\text{sujeito a: } \sum_{j \in J' \setminus \{i\}} x_{ij} = 1, \quad i \in J' \quad (3.26)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = 1, \quad j \in J' \quad (3.27)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad S \subset J', 2 \leq |S| \leq |J'| - 1 \quad (3.28)$$

$$\sum_{i \in J' \setminus \{j\}} (y_{ij}^t + z_{ij}^t) = 1, \quad j \in J', t \in T_j \quad (3.29)$$

$$y_{ij}^t + z_{ij}^t \leq x_{ij}, \quad i, j \in J', i \neq j, t \in T : t \notin T_i, t \in T_j \quad (3.30)$$

$$\sum_{t \in T} (y_{ij}^t + z_{ij}^t) \leq Cx_{ij}, \quad i, j \in J', i \neq j \quad (3.31)$$

$$\sum_{j \in J' \setminus \{i\}} (y_{ji}^t + z_{ji}^t) \geq \sum_{j \in J' \setminus \{i\}} y_{ij}^t, \quad i \in J', t \in T \quad (3.32)$$

$$y_{ij}^t \geq x_{ij}, \quad i, j \in J', i \neq j, t \in T_i \cap T_j \quad (3.33)$$

$$z_{ij}^t = 0, \quad i, j \in J', t \in T \setminus T_j \quad (3.34)$$

$$y_{0j}^t = 0, \quad i \in J, t \in T \setminus T_i \quad (3.35)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in J', i \neq j \quad (3.36)$$

$$y_{ij}^t \in \{0, 1\}, \quad i, j \in J', i \neq j, t \in T \quad (3.37)$$

$$z_{ij}^t \in \{0, 1\}, \quad i, j \in J', i \neq j, t \in T. \quad (3.38)$$

As restrições (3.26)-(3.28) são equivalentes às restrições (3.11), (3.12) e (3.16) do modelo proposto em Laporte, Salazar-González e Semet (2004). Além disso, ambos os modelos compartilham a mesma função objetivo (3.25). As restrições (3.29) garantem que, caso a tarefa  $j$  seja executada imediatamente após a execução da tarefa  $i$ , as ferramentas necessárias para o processamento da tarefa  $j$  sejam herdadas do processamento da tarefa  $i$ , ou, então, que a ferramenta seja adicionada na transição da tarefa  $i$  para a  $j$ . O conjunto de restrições (3.30) impõe que, caso uma ferramenta  $t$  seja necessária para o processamento de uma tarefa  $j$ , então a ferramenta  $t$  deve permanecer na máquina após o processamento da tarefa  $i$  ou ser inserida na máquina durante esta transição, somente se a tarefa  $j$  sucede a tarefa  $i$  em um sequenciamento ótimo. As restrições (3.31) são similares às restrições de capacidade dos modelos Tang e Denardo e Laporte, Salazar-González e Semet. Caso uma ferramenta  $t$  seja carregada na transição da tarefa  $i$  para a tarefa  $j$ , as restrições (3.32) obrigam que a ferramenta  $t$  seja adicionada durante a transição de alguma tarefa  $k$  para a tarefa  $i$ , ou carregada durante esta transição. O conjunto de restrições (3.33) assegura que se  $t \in T_i \cap T_j$  e o processamento da tarefa  $j$  é feito imediatamente após o processamento da tarefa  $i$ , a ferramenta  $t$  deve ser carregada nesta transição. O conjunto de restrições (3.34) é equivalente a (3.17) e, por último, temos que as restrições (3.35) levam em conta a transição da tarefa fictícia para a primeira tarefa. As demais restrições são referentes ao domínio das variáveis. É demonstrado em Catanzaro, Golveia e Labbé (2015) que o valor da relaxação linear deste modelo é maior ou igual a de Laporte, Salazar-González e Semet (2004).

Nos modelos propostos em Tang e Denardo (1988) e Laporte, Salazar-González e Semet (2004), a inserção de ferramentas para iniciar o processo não é contabilizada, ou seja, parte-se do pressuposto que o magazine da máquina já está carregado com, no máximo,  $C$  ferramentas antes de se iniciar a execução das tarefas. O mesmo não ocorre para as formulações propostas em Catanzaro, Golveia e Labbé (2015), neste artigo os



autores consideram que o processo é iniciado com o magazine da máquina vazio e, portanto, estes modelos contabilizam a inserção das primeiras ferramentas na máquina.

Com o intuito de reduzir o tamanho do modelo [Catanzaro et al. I](#), os autores propõem três novos conjuntos de restrições:

$$y_{ij}^t + z_{ij}^t = x_{ij}, \quad i, j \in J' : i \neq j, t \in T : t \notin T_i, t \in T_j \quad (3.39)$$

$$y_{ij}^t = x_{ij}, \quad i, j \in J' : i \neq j, t \in T : t \in T_i \cap T_j \quad (3.40)$$

$$y_{ij}^t \leq x_{ij}, \quad i, j \in J' : i \neq j, t \in T : t \notin T_j. \quad (3.41)$$

Caso em um sequenciamento ótimo, a tarefa  $j$  seja realizada imediatamente após a execução da tarefa  $i$ , o conjunto de restrições (3.39) obriga que todas as ferramentas necessárias para o processamento de  $j$  mas não de  $i$ , sejam inseridas na máquina ou sejam carregadas durante essa transição. As restrições (3.40) asseguram que, se em um sequenciamento ótimo, a tarefa  $j$  é realizada imediatamente após a execução da tarefa  $i$ , as ferramentas  $t \in T_i \cap T_j$  devem ser carregadas durante a transição. As restrições (3.41) permitem que uma ferramenta  $t \notin T_j$  seja carregada na transição da tarefa  $i$  para a tarefa  $j$ , caso no sequenciamento ótimo a tarefa  $j$  seja realizada imediatamente após a execução da tarefa  $i$ .

O conjunto de restrições (3.39) pode ser usado para eliminar as variáveis  $z_{ij}^t$ , já as restrições (3.40) podem ser utilizadas para eliminar as variáveis  $y_{ij}^t$  quando  $t \in T_i \cap T_j$ . Logo, somente as variáveis  $y_{ij}^t$  definidas para  $i, j \in J'$  tais que  $i \neq j$  e, para todo  $t$ , tal que  $t \in T_i$  e  $t \notin T_j$  ou  $t \notin T_i$ , devem ser consideradas. De fato, para todo  $j \in J'$  e  $t \in T_j$ , o conjunto de restrições (3.29) pode ser modificado da seguinte forma:

$$1 = \sum_{i \in J' \setminus \{j\}} (y_{ij}^t + z_{ij}^t), \quad (3.42)$$

Como para  $t \in T_i$ , temos  $z_{ij}^t = 0$ , as restrições (3.42) podem ser reescritas como:

$$1 = \sum_{j \in J' \setminus \{j\}} y_{ij}^t + \sum_{\substack{j \in J' \setminus \{j\} \\ t \notin T_i}} z_{ij}^t,$$

Utilizando as restrições (3.39) temos:

$$1 = \sum_{j \in J' \setminus \{j\}} y_{ij}^t + \sum_{\substack{j \in J' \setminus \{j\} \\ t \notin T_i}} (x_{ij} - y_{ij}^t) = \sum_{j \in J' \setminus \{j\}} y_{ij}^t - \sum_{\substack{j \in J' \setminus \{j\} \\ t \notin T_i}} y_{ij}^t + \sum_{\substack{j \in J' \setminus \{j\} \\ t \notin T_i}} x_{ij} = \sum_{\substack{j \in J' \setminus \{j\} \\ t \in T_i}} y_{ij}^t + \sum_{\substack{j \in J' \setminus \{j\} \\ t \notin T_i}} x_{ij},$$

Utilizando as restrições (3.40) temos:

$$1 = \sum_{\substack{j \in J' \setminus \{j\} \\ t \in T_i}} x_{ij} + \sum_{\substack{j \in J' \setminus \{j\} \\ t \notin T_i}} x_{ij} = \sum_{j \in J' \setminus \{i\}} x_{ij},$$

ou seja as restrições (3.29) se tornam redundantes.

Considere agora as restrições (3.31) para todo  $i, j \in J'$ , ou seja:

$$\sum_{t \in T} y_{ij}^t + \sum_{t \in T} z_{ij}^t \leq Cx_{ij}.$$

O conjunto de restrições (3.34) garantem que se  $t \in T \setminus T_j$  então  $z_{ij}^t = 0$ . Além disso, para  $t \in T_i$  temos  $z_{ij}^t = 0$ . Sendo assim, para todo  $i, j \in J'$ , as restrições (3.31) podem ser reescritas como:

$$\sum_{t \in T} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} z_{ij}^t \leq Cx_{ij}.$$

Utilizando as restrições (3.39):

$$\sum_{t \in T} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} (x_{ij} - y_{ij}^t) \leq Cx_{ij}, \quad (3.43)$$

o conjunto  $T$  pode ser dividido em 4 subconjuntos, sendo estes:

- $T_i \cap T_j$ .
- $T_j \setminus T_i$ .
- $T_i \setminus T_j$ .
- $T \setminus (T_i \cup T_j)$ .

E, portanto, o termo  $\sum_{t \in T} y_{ij}^t$  pode ser reescrito como:

$$\sum_{\substack{t \in T \\ t \in T_i, t \in T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \in T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \notin T_j}} y_{ij}^t,$$

já o termo  $\sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} (x_{ij} - y_{ij}^t)$  pode ser reescrito como:

$$\sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} x_{ij} - \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} y_{ij}^t,$$

Desta forma, o conjunto de restrições (3.43) pode ser reescrito da seguinte forma:

$$\sum_{\substack{t \in T \\ t \in T_i, t \in T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \in T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} x_{ij} - \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} y_{ij}^t \leq Cx_{ij},$$

ou, equivalentemente:

$$\sum_{\substack{t \in T \\ t \in T_i, t \in T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \in T_i, t \notin T_j}} y_{ij}^t + \left( \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} y_{ij}^t - \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} y_{ij}^t \right) + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} x_{ij} \leq Cx_{ij},$$

que pode ser reescrito como:

$$\sum_{\substack{t \in T \\ t \in T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \in T_i, t \in T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} x_{ij} \leq Cx_{ij}.$$

Utilizando as restrições (3.40), temos:

$$\sum_{\substack{t \in T \\ t \in T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \in T_i, t \in T_j}} x_{ij} + \sum_{\substack{t \in T \\ t \notin T_i, t \in T_j}} x_{ij} \leq Cx_{ij},$$

ou, equivalentemente:

$$\sum_{\substack{t \in T \\ t \in T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \in T_j}} x_{ij} \leq Cx_{ij},$$

que, por sua vez, pode ser reescrito como:

$$\sum_{\substack{t \in T \\ t \in T_i, t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T \\ t \notin T_i, t \notin T_j}} y_{ij}^t \leq (C - |T_j|)x_{ij}.$$

Por último, considere as restrições (3.32) para  $t \in T_i$ :

$$\sum_{j \in J' \setminus \{i\}} y_{ji}^t + \sum_{\substack{j \in J' \setminus \{i\} \\ t \notin T_j}} z_{ji}^t \geq \sum_{j \in J' \setminus \{i\}} y_{ij}^t.$$

Aplicando (3.39), temos:

$$\sum_{j \in J' \setminus \{i\}} y_{ji}^t + \sum_{\substack{j \in J' \setminus \{i\} \\ t \notin T_j}} (x_{ji} - y_{ji}^t) \geq \sum_{j \in J' \setminus \{i\}} y_{ij}^t,$$

que pode ser reescrito como:

$$\sum_{\substack{j \in J' \setminus \{i\} \\ t \in T_j}} y_{ji}^t + \sum_{\substack{k \in J' \setminus \{i\} \\ t \notin T_j}} x_{ji} \geq \sum_{j \in J' \setminus \{i\}} y_{ij}^t,$$

Utilizando (3.40) e (3.26), obtemos:

$$1 = \sum_{\substack{j \in J' \setminus \{i\} \\ t \in T_j}} x_{ji} + \sum_{\substack{j \in J' \setminus \{i\} \\ t \notin T_j}} x_{ji} \geq \sum_{j \in J' \setminus \{j\}} y_{ij}^t.$$

E, portanto, o conjunto de restrições (3.32) se torna redundante. Para  $i \notin T_i$  as variáveis  $z_{ki}^t$  não estão definidas e, portanto, as restrições podem ser reduzidas da seguinte forma:

$$\sum_{j \in J' \setminus \{i\}} y_{ji}^t \geq \sum_{j \in J' \setminus \{i\}} y_{ij}^t.$$

Desta maneira, a formulação [Catanzaro et al. I](#) pode ser reduzida e, assim, obtemos a formulação [Catanzaro et al. II](#):

(Catanzaro et al. II)

$$\min \sum_{j \in J} |T_j| x_{0j} + \sum_{\substack{i, j \in J \\ i \neq j}} \sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \quad (3.44)$$

$$\text{sujeito a: } \sum_{j \in J' \setminus \{i\}} x_{ij} = 1, \quad i \in J' \quad (3.45)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = 1, \quad j \in J' \quad (3.46)$$

$$\sum_{\substack{i, j \in S \\ i \neq j}} x_{ij} \leq |S| - 1, \quad \forall S \subset J', 2 \leq |S| \leq |J'| - 1 \quad (3.47)$$

$$\sum_{k \in J' \setminus \{i\}} y_{ki}^t - \sum_{j \in J' \setminus \{i\}} y_{ij}^t \geq 0, \quad i \in J', t \in T \setminus T_i \quad (3.48)$$

$$\sum_{\substack{t \in T: \\ t \in T_i \\ t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T: \\ t \notin T_i \\ t \notin T_j}} y_{ij}^t \leq (C - |T_j|) x_{ij}, \quad i, j \in J', i \neq j \quad (3.49)$$

$$y_{ij}^t \leq x_{ij}, \quad i, j \in J', i \neq j, t \in T \setminus T_j \quad (3.50)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in J', i \neq j \quad (3.51)$$

$$y_{ij}^t \in \{0, 1\}, \quad i, j \in J', i \neq j, t \in (T \setminus T_i) \cup (T_i \setminus T_j). \quad (3.52)$$

As restrições (3.45)-(3.47) são idênticas às restrições (3.26)-(3.28). A função objetivo, dada por (3.25), visa minimizar o número de trocas de ferramentas. As restrições (3.48) impõem que se a ferramenta  $t$  é carregada na transição da tarefa  $i$  para alguma tarefa  $j$  então esta ferramenta deve ter sido carregada ou inserida na transição de alguma tarefa  $k$  para a tarefa  $i$ ,  $i \neq k$ . As restrições (3.49) impedem que o limite máximo no número de pilhas abertas seja ultrapassado. O conjunto de restrições (3.50) garante que uma ferramenta que não é necessária para o processamento da tarefa  $j$  só possa ser carregada na transição da execução da tarefa  $i$  para a execução da tarefa  $j$  se a tarefa  $j$  é executada imediatamente após a execução da tarefa  $i$ . As restrições (3.51) e (3.52) são referentes ao domínio das variáveis.

**Definição:** Um bloco-0 com respeito a uma ferramenta  $t$  é uma sequência de tarefas que começa pela tarefa  $i$  e termina com a tarefa  $j$ , na qual, a ferramenta  $t$  é necessária para o processamento das tarefas  $i$  e  $j$  e, não para as demais.

A formulação [Catanzaro et al. II](#) tem a interessante propriedade de minimizar a ocorrência de blocos-0. A importância desta propriedade para o problema de minimização de troca de ferramentas pode ser vista em mais detalhes em [Crama, Oerlemans e Spieksma \(1994\)](#). As relaxações lineares das formulações [Catanzaro et al. I](#) e [Catanzaro et al. II](#) apresentam a mesma solução ótima. Para fortalecer a relaxação linear do modelo, [Catanzaro, Golveia e Labbé \(2015\)](#) sugere a inserção de duas desigualdades válidas, sendo elas:

$$\sum_{i \in J \setminus \{j,k\}} y_{ij}^t \geq y_{jk}^t,$$

$$\sum_{i \in J \setminus S} \sum_{j \in S \setminus \{k\}} y_{ij}^t \geq \sum_{i \in S \setminus \{k\}} y_{ik}^t, \quad \forall S \subset J, k \in S, t \in T_k : J_t \cap S = \{k\}.$$

Considere  $G_{ij}^t$  o subgrafo cujo o conjunto de vértices é dado por  $V_{ij}^t = (J_t)^c \cup \{i, j\}$ , para todo  $i, j \in J, t \in T_i \cap T_j$  e, o conjunto de arcos por  $A_{ij}^t$ , que representa o subconjunto de arcos com destino em elementos de  $V_{ij}^t$ , excluindo o arco  $(i, j)$  e os arcos entrando em  $i$  e os arcos saindo de  $j$ . Um caminho em  $G_{ij}^t$  pode ser usado para representar a sequência que define um bloco-0. Seja a variável de decisão  $f_{il,kl}^t$  igual a 1, se o vértice  $k$  precede o vértice  $l$  em um caminho em  $G_{ij}^t$  e 0, caso contrário. É possível estabelecer as seguintes relações entre as variáveis  $y_{kl}^t$  e  $f_{il,kl}^t$ :

$$y_{kl}^t = \sum_{i \in J_t} f_{il,kl}^t, \quad t \in T, k \in (J_t)^c, l \in J_t$$

$$y_{kl}^t = \sum_{j \in J_t} f_{kj,kl}^t, \quad t \in T, k \in J_t, l \in (J_t)^c$$

$$y_{kl}^t = \sum_{\substack{i,j \in J_t \\ i \neq j}} f_{ij,kl}^t, \quad t \in T, k, l \in (J_t)^c : k \neq l.$$

O modelo final estabelecido em [Catanzaro, Golveia e Labbé \(2015\)](#) é obtido substituindo-se as variáveis  $y_{kl}^t$  por  $f_{il,kl}^t$ :

(Catanzaro et al. III)

$$\min \sum_{j \in J} |T_j| x_{0j} + \sum_{\substack{i,j \in J \\ i \neq j \\ t \in T_j \setminus T_i}} (x_{ij} - \sum_{\substack{k \in J: \\ k \neq j \\ t \in T_k}} f_{kj,ij}^t) \quad (3.53)$$

$$\text{sujeito a: } \sum_{j \in J' \setminus \{i\}} x_{ij} = 1, \quad i \in J' \quad (3.54)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = 1, \quad j \in J' \quad (3.55)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad S \subset J', 2 \leq |S| \leq |J'| - 1 \quad (3.56)$$

$$\sum_{\substack{l \in V_{ij}^t \\ l \neq k}} f_{il,kl}^t - \sum_{\substack{l \in V_{ij}^t \\ l \neq k}} f_{il,lk}^t \geq 0, \quad i, j \in J : i \neq j,$$

$$t \in T_i \cap T_j, k \in (J_t)^c \quad (3.57)$$

$$\sum_{i \in J_t} f_{il,kl}^t \leq x_{kl}, \quad t \in T, k \in (J_t)^c, l \in J_t \quad (3.58)$$

$$\sum_{j \in J_t} f_{kj,kl}^t \leq x_{kl}, \quad t \in T, k \in J_t, l \in (J_t)^c \quad (3.59)$$

$$\sum_{\substack{i,j \in J_t \\ i \neq j}} f_{ij,kl}^t \leq x_{kl}, \quad t \in T, k, l \in (J_t)^c : k \neq l \quad (3.60)$$

$$\sum_{\substack{t \in T: \\ t \in T_k \\ t \notin T_l}} \sum_{\substack{j \in J_t \\ j \neq k}} f_{kj,kl}^t + \sum_{\substack{t \in T: \\ t \notin T_k \\ t \notin T_l}} \sum_{\substack{i,j \in J_t \\ i \neq j}} f_{ij,kl}^t \leq (C - |T_l|) x_{kl}, \quad k, l \in J, : k \neq l \quad (3.61)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in J', i \neq j \quad (3.62)$$

$$f_{ij,kl}^t \geq 0, \quad t \in T, i, j \in J :$$

$$i \neq j, k, l \in V_{ij}^t :$$

$$k \neq l, \{k, l\} \neq \{i, j\} \quad (3.63)$$

A função objetivo (3.53) minimiza o número total de trocas de ferramentas realizadas, as restrições (3.54) garantem que uma, e apenas uma, tarefa seja processada após a tarefa  $i$ , para todo  $i \in J'$ . As restrições (3.55) fazem com que para cada  $j \in J'$  exista uma, e apenas uma, tarefa  $i$  que anteceda o processamento da tarefa  $j$ . O conjunto de restrições (3.56) elimina a formação de subciclos durante a execução das tarefas conforme pode ser visto em [Dantzig, Fulkerson e Johnson \(1954\)](#). As restrições (3.57) permitem a

existência de um caminho (bloco-0) em  $V_{ij}^t$  se tal caminho está presente em uma solução ótima do problema. As restrições (3.58)-(3.60) permitem que a ferramenta  $t$  permaneça na máquina durante o processamento das tarefas  $k$  e  $l$ , sendo estas pertencentes a um bloco-0, somente se  $l$  é executada após a execução de  $k$ . O conjunto de restrições (3.61) impede que o limite da capacidade da máquina seja ultrapassado e as restrições (3.62)-(3.63) definem o domínio das variáveis.

### 3.6 Restrições de eliminação de subciclos

Para evitar a formação de subciclos, tanto o modelo de Laporte, Salazar-González e Semet (2004) quanto os modelos de Catanzaro, Golveia e Labbé (2015) utilizam o conjunto de restrições proposto em Dantzig, Fulkerson e Johnson (1954). Seja  $N$  o total de tarefas a serem processadas no contexto do MTSP e o número de padrões de corte no contexto do MOSP. Então o conjunto de restrições de subciclos proposto em Dantzig, Fulkerson e Johnson (1954) é dado por (3.64):

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, S \subset J', 2 \leq |S| \leq N - 1 \quad (3.64)$$

O número de restrições presentes neste conjunto cresce exponencialmente em função de  $N$ . De fato, há um total de  $2^N - (N + 2)$  restrições neste conjunto e, dependendo do valor de  $N$ , esta quantidade pode se tornar excessivamente grande, impossibilitando a resolução do problema. O conjunto de restrições proposto em Dantzig, Fulkerson e Johnson (1954) pode inviabilizar a resolução do problema em tempo razoável. Neste caso, podemos utilizar outro conjunto de restrições que impeça a formação de subciclos. Nos Capítulos 5, 6 e 7 o conjunto de restrições proposto em Miller, Tucker e Zemlin (1960) é utilizado no lugar das restrições propostas em Dantzig, Fulkerson e Johnson (1954).

O conjunto de restrições proposto em Miller, Tucker e Zemlin (1960) requer a criação de  $N - 1$  variáveis inteiras. Considerando  $u_i$  como as variáveis auxiliares necessárias para impedir a formação de subciclos, em Miller, Tucker e Zemlin (1960) são propostas as seguintes restrições:

$$1 \leq u_i \leq N, \quad i = 2, \dots, N \quad (3.65)$$

$$u_i - u_j + Nx_{ij} \leq N - 1, \quad i = 2, \dots, N, j = 2, \dots, N \quad (3.66)$$

$$u_i \in Z^+, \quad i = 2, \dots, N. \quad (3.67)$$

Temos então um total de  $2(N - 1)$  restrições em (3.65),  $(N - 1)^2$  restrições em (3.66) e  $N - 1$  restrições em (3.67). Sendo assim, há  $(N - 1)^2 + 3(N - 1)$  restrições no total. Na Tabela 4 a quantidade de restrições de cada um dos conjuntos é comparada em função de  $N$ .

$N$	Número de restrições	
	Dantzig, Fulkerson e Johnson (1954)	Miller, Tucker e Zemlin (1960)
10	$101,20 \times 10$	108
11	$203,50 \times 10$	130
12	$408,20 \times 10$	154
13	$817,70 \times 10$	180
14	$163,68 \times 10^2$	208
15	$327,51 \times 10^2$	238
16	$655,18 \times 10^2$	270
17	$131,05 \times 10^3$	304
18	$262,12 \times 10^2$	340
19	$524,27 \times 10^2$	378
20	$104,86 \times 10^4$	418

Tabela 4 – Comparação entre a quantidade de restrições presentes nos conjuntos de restrições propostos em Dantzig, Fulkerson e Johnson (1954) e Miller, Tucker e Zemlin (1960) em função de  $N$ , com  $N$  variando de 10 até 20.

Portanto, apesar das restrições de Miller, Tucker e Zemlin (1960) exigirem a criação de novas variáveis inteiras, isso é compensado pelo fato do número de restrições para esse conjunto ser consideravelmente menor do que o número de restrições para o conjunto proposto em Dantzig, Fulkerson e Johnson (1954). Para o caso no qual  $N$  representa a quantidade de padrões de corte a serem sequenciados,  $N$  pode ser demasiadamente grande para se utilizar as restrições propostas em Dantzig, Fulkerson e Johnson (1954).



## 4 Revisão da literatura sobre o problema integrado

O problema de corte de estoque e o problema de sequenciamento dos padrões de corte para minimização do número máximo de pilhas abertas podem ser combinados em um único problema. Em [Yanasse e Lamosa \(2007\)](#), é construído um modelo para o CS-LOSP, no qual o problema de corte de estoque é resolvido sujeito a uma limitação no número máximo de pilhas que podem ser abertas.

Neste capítulo, o *CS-LOSP* é discutido em detalhes, demonstrando as vantagens de se resolver o PCE e o MOSP de forma integrada. O modelo apresentado em [Yanasse e Lamosa \(2007\)](#) é revisado. Esse modelo se baseia nas formulações de [Gilmore e Gomory \(1961, 1963, 1965\)](#) para lidar com a parte relativa ao problema de corte de estoque e, no modelo proposto em [Yanasse \(1997\)](#), para tratar da parte do problema relativa ao sequenciamento de padrões de corte.

Como foi visto no Capítulo 3, o problema de minimização de pilhas abertas busca sequenciar um conjunto de padrões de corte. Essa abordagem pode gerar uma série de dificuldades como, por exemplo, caso exista uma limitação no número máximo de pilhas que podem ser abertas e a solução obtida através do PCE não apresentar um sequenciamento factível.

Para fins de comparação, vamos retomar o [Exemplo I](#), desta vez acrescentando a restrição de que no máximo duas pilhas possam ser abertas durante o sequenciamento dos padrões de corte. É fácil notar que os padrões de corte (5) e (6) apresentados na [Figura 5](#) não podem ser usados, pois sua execução abre três pilhas. Logo é esperado que o conjunto de padrões de corte obtidos com esta nova restrição seja diferente do apresentado na [Figura 5](#), o que, de fato, ocorre. Os padrões de corte para a solução do CS-LOSP com o limite de duas pilhas estão contidos na [Figura 9](#).

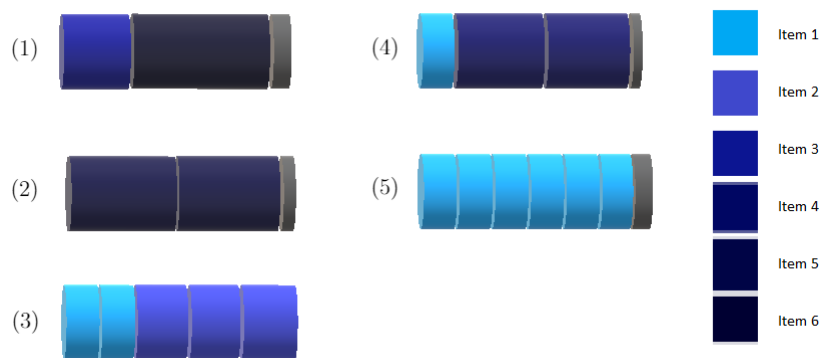


Figura 9 – Padrões de corte encontrados em uma solução ótima do Exemplo I, sujeito à restrição de que o número máximo de pilhas abertas seja menor ou igual a dois. Os diversos tipos de itens estão representados segundo o código de cores presente no canto direito da figura.

O problema integrado visa determinar não só a frequência na qual os padrões de corte são executados mas também a sequência na qual estes devem ser efetuados. A sequência encontrada em uma solução ótima para o Exemplo I é dada por  $2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 1$ . Na Tabela 5 é apresentado o total de pilhas abertas nesta solução.

Sequência $2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 1$		
Padrão de corte executado	Itens nas pilhas abertas	Total de pilhas abertas
2	5	1
4	1-4	2
5	1	1
3	1-3	2
1	3-6	2

Tabela 5 – Abertura de pilhas a partir do sequenciamento dos padrões de corte, os padrões de corte presentes na tabela podem ser encontrados na Figura 9, já a ordem na qual estes padrões de corte são executados é dada pela sequência  $2 \rightarrow 4 \rightarrow 5 \rightarrow 3 \rightarrow 1$ . Na primeira coluna temos os padrões de corte executados, na segunda coluna temos quais pilhas estão abertas após a execução do padrão de corte e, na terceira, o total de pilhas abertas.

O número máximo de pilhas abertas é igual a dois, satisfazendo então a restrição sobre o número máximo de pilhas abertas. A inclusão deste conjunto de restrições limita o conjunto de soluções possíveis, sendo assim, o valor da função objetivo sempre será maior ou igual à do problema de corte. Tipicamente, quão menor for a quantia de pilhas que podem ser abertas, maior é o valor da função objetivo.

Todos os modelos para o CS-LOSP apresentados neste trabalho tem como objetivo minimizar o número de objetos utilizados. Ou seja, duas hipóteses foram consideradas, a primeira destas é que os objetos a serem cortados possuem sempre o mesmo

comprimento  $L$  e a segunda é que o custo de execução dos padrões de corte, dado por  $c_j$ , é igual a 1 para todo  $j$ . Desta maneira,  $c_j$  é omitido dos modelos.

## 4.1 Formulação de Yanasse e Lamosa (2007)

No MOSP, os padrões a serem sequenciados são dados do problema, enquanto no CS-LOSP os padrões de corte que são usados dependem da solução do PCE. Logo, nem todos devem ser considerados no sequenciamento. Para lidar com essa dificuldade, em Yanasse e Lamosa (2007) foi proposto o uso de variáveis binárias  $v_j$ , que são iguais a 1 se e somente se o padrão de corte  $j$  é utilizado na solução do problema e 0, caso contrário. Seja  $K$  uma constante positiva e  $p_j$  uma variável de decisão que determina a quantidade de vezes que o padrão de corte  $j$  é utilizado. Esta variável é definida no Capítulo 2 e utilizada na formulação de Gilmore e Gomory (1961, 1963, 1965). Desta forma,  $v_j$  pode ser modelada matematicamente segundo as restrições:

$$v_j \leq p_j, \quad j = 1, \dots, N \quad (4.1)$$

$$Kv_j \geq p_j, \quad j = 1, \dots, N \quad (4.2)$$

$$v_j \in \{0, 1\}, \quad j = 1, \dots, N \quad (4.3)$$

As restrições (4.1) garantem que, se o padrão de corte  $j$  não for utilizado ( $p_j = 0$ ), então  $v_j = 0$ , já o conjunto de restrições (4.2) garante que, se  $p_j > 0$ , então  $v_j = 1$ . Note que  $K$  deve ser suficientemente grande para garantir o funcionamento dessas restrições. Nos testes computacionais (Capítulos 6 e 7) foi utilizado

$$K = \sum_{i=1}^M d_i. \quad (4.4)$$

Esta estimativa é justificada em (2.1). Note que existem outras estimativas válidas para  $K$  como, por exemplo, a estimativa para  $M$  dada em (2.6), ou  $K = \max_{i=1, \dots, M} \{d_i\}$ . Testes computacionais preliminares executados neste trabalho sugerem que o valor de  $K$ , desde que válido, não possui grande impacto no tempo exigido para se determinar uma solução ótima de um problema. Desta maneira, optamos pela estimativa mais simples, dada por (4.4). As especificações do computador e softwares utilizados no trabalho se encontram na Seção 6.1.

Por último, as restrições (4.3) definem o domínio das variáveis. As variáveis  $v_j$ ,  $j = 1, \dots, N$  são usadas para garantir que as pilhas só sejam consideradas caso o padrão de corte de fato seja utilizado. Para isso as restrições (3.4) são substituídas por:

$$w_{ij} + (v_j - 1) \leq y_{jt}, \quad i \in J_t, j = 1, \dots, N, t \in T. \quad (4.5)$$

Note então que, caso  $v_j$  seja igual a zero, a restrição (4.5) é trivialmente satisfeita, ou seja, as pilhas abertas por esse padrão de corte serão desprezadas. Considere:

**Índices:**

$M$ : quantidade de tipos de itens;

$N$ : quantidade de padrões de corte.

**Parâmetros:**

$\alpha^{(j)} = (\alpha_1^{(j)}, \alpha_2^{(j)}, \dots, \alpha_M^{(j)})^t$  : representa o  $j$ -ésimo padrão de corte. A  $i$ -ésima coordenada do vetor corresponde à quantidade de itens do tipo  $i$  que está presente no padrão de corte;

$K$ : constante dada por (4.4);

$d_i$  : demanda do item do tipo  $i$ .

**Variáveis de decisão:**

$p_j$  : quantidade de vezes que o padrão de corte  $j$  é utilizado;

$z_{it}$  : variável binária utilizada para contabilizar o número de pilhas abertas. No contexto do MTSP,  $z_{it}$  é igual a 1 se e somente se a ferramenta  $t$  é inserida na máquina para o processamento da tarefa  $i$ ;

$$w_{ij} = \begin{cases} 1, & \text{se o padrão de corte } i \text{ é o } j\text{-ésimo a ser processado,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{it} = \begin{cases} 1, & \text{se o item } t \text{ é obtido ao utilizar o padrão de corte } i, \\ 0, & \text{caso contrário;} \end{cases}$$

$$v_j = \begin{cases} 1, & \text{se o padrão de corte } j \text{ é utilizado na solução do problema,} \\ 0, & \text{caso contrário.} \end{cases}$$

O CS-LOSP formulado em Yanasse (1997) é dado por:

(YLI)

$$\min \sum_{j=1}^N p_j \quad (4.6)$$

$$\text{sujeito a: } \sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i = 1, \dots, M \quad (4.7)$$

$$\sum_{i=2}^N \sum_{t \in T} z_{it} \leq M - C, \quad (4.8)$$

$$v_j \leq p_j, \quad j = 1, \dots, N \quad (4.9)$$

$$K v_j \geq p_j, \quad j = 1, \dots, N \quad (4.10)$$

$$\sum_{i=1}^N w_{ij} = 1, \quad j = 1, \dots, N \quad (4.11)$$

$$\sum_{j=1}^N w_{ij} = 1, \quad i = 1, \dots, N \quad (4.12)$$

$$w_{ij} + (v_j - 1) \leq y_{jt}, \quad i \in J_t, j = 1, \dots, N, t \in T \quad (4.13)$$

$$\sum_{t \in T} y_{jt} \leq C, \quad j = 1, \dots, N \quad (4.14)$$

$$y_{jt} - y_{(j-1)t} \leq z_{jt}, \quad i = 2, \dots, N, t \in T \quad (4.15)$$

$$p_j \in Z^+, \quad j = 1, \dots, N \quad (4.16)$$

$$v_j \in \{0, 1\}, \quad j = 1, \dots, N \quad (4.17)$$

$$w_{ij} \in \{0, 1\}, \quad i = 1, \dots, N, j = 1, \dots, N \quad (4.18)$$

$$y_{jt} \in \{0, 1\}, \quad j = 1, \dots, N, t \in T \quad (4.19)$$

$$z_{jt} \in \{0, 1\}, \quad j = 1, \dots, N, t \in T. \quad (4.20)$$

A função objetivo (4.6) é similar à considerada em (2.12). Esta função objetivo minimiza o desperdício de material. De forma análoga, as restrições (4.7) equivalem ao conjunto de restrições (2.13), para garantir que a demanda requerida seja satisfeita. O conjunto de restrições (4.8) garante que o número máximo de pilhas abertas seja igual a  $C$ , este conjunto de restrições é utilizado para transformar um MTSP em um MOSP conforme foi discutido na Seção 3.3. As restrições (4.9) e (4.10) garantem que  $v_j = 0$  se e somente se  $p_j = 0$ . Os conjuntos de restrições (4.11) e (4.12) garantem que todos os padrões de corte sejam executados e que seja executado um padrão de corte por vez. O conjunto de restrições (4.13) garante que, caso o padrão de corte  $i$  seja utilizado, então todas as pilhas abertas devem ser levadas em conta, do contrário, a restrição se torna trivial. As restrições (4.14) impedem que o limite da capacidade de pilhas abertas seja

ultrapassado. O conjunto de restrições (4.15) assegura a contabilização do número de pilhas abertas através das variáveis  $z_{kt}$  sempre que necessário. As restrições (4.16)-(4.20) definem o domínio das variáveis.

Uma das vantagens do modelo YLI é que seu uso é compatível com problemas de quaisquer dimensões. Uma vez gerado o conjunto de padrões de corte factíveis do problema, seja esse conjunto referente a um problema unidimensional, bidimensional ou tridimensional, a formulação pode ser utilizada sem que seja necessária nenhuma alteração em suas variáveis ou restrições. Embora o modelo se mantenha o mesmo para problemas de dimensões distintas, não é razoável assumir que a dimensão do problema não impacte o tempo necessário para se determinar uma solução ótima. No Capítulo 6, testes computacionais unidimensionais e bidimensionais são realizados com o intuito de averiguar a relevância que a dimensão do problema tem no tempo exigido para se determinar uma solução ótima.

## 4.2 Formulação de Arbib, Marinelli e Ventura (2016)

Em Arbib, Marinelli e Ventura (2016) os autores propõem um novo modelo para o CS-LOSP unidimensional, no qual a quantidade de restrições cresce quadraticamente em função da quantidade de tipos distintos de itens. Além disso, a relaxação linear da formulação proposta pode ser resolvida a partir do método de geração de colunas.

Os autores utilizaram a formulação de Kantorovich (1960) para construir um modelo para o CS-LOSP. Em seguida, este modelo foi modificado de maneira a melhorar sua eficiência. Nesta seção, os dois modelos são apresentados. A primeira formulação proposta em Arbib, Marinelli e Ventura (2016) se baseia em Kantorovich (1960) e exige a criação de novas variáveis. Seja então:

$$q_{ij} = \begin{cases} 1, & \text{se a } i\text{-ésima pilha é aberta em algum instante } \leq j, \\ 0, & \text{caso contrário;} \end{cases}$$

$$r_{ij} = \begin{cases} 1, & \text{se a } i\text{-ésima pilha é fechada em algum instante } \leq j, \\ 0, & \text{caso contrário.} \end{cases}$$

As variáveis  $q_{ij}$  e  $r_{ij}$  são usadas para garantir que o número máximo de pilhas abertas não supere o limite preestabelecido. Os demais parâmetros e variáveis presentes nos modelos são idênticos aos apresentados em Kantorovich (1960). Considere:

**Índices:**

$M$  : quantidade de itens de tipos distintos;

$N$  : quantidade de objetos disponíveis, o valor de  $N$  é estimado usando (2.6).

**Parâmetros:**

$L$  : comprimento dos objetos;

$U_i$  : limitante superior para a quantidade de itens de tipo  $i$  que podem ser obtidos a partir do corte de um objeto de comprimento  $L$ ;

$d_i$  : demanda do item do tipo  $i$ ;

$l_i$  : comprimento do item do tipo  $i$ .

**Variáveis de decisão:**

$x_{ij}$  : número de itens do tipo  $i$  que são obtidos ao cortar o objeto  $j$ ;

$$y_j = \begin{cases} 1, & \text{se o objeto } j \text{ for utilizado,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$q_{ij} = \begin{cases} 1, & \text{se a } i\text{-ésima pilha é aberta em algum instante } \leq j, \\ 0, & \text{caso contrário;} \end{cases}$$

$$r_{ij} = \begin{cases} 1, & \text{se a } i\text{-ésima pilha é fechada em algum instante } \leq j, \\ 0, & \text{caso contrário.} \end{cases}$$

O modelo de [Arbib, Marinelli e Ventura \(2016\)](#) para o CS-LOSP é dado por:

(Arbib, Marinelli e Ventura-1)

$$\min \sum_{j=1}^N y_j \quad (4.21)$$

$$\text{sujeito a: } \sum_{j=1}^N x_{ij} = d_i, \quad i = 1, \dots, M \quad (4.22)$$

$$\sum_{i=1}^M l_i x_{ij} \leq Ly_j, \quad j = 1, \dots, N \quad (4.23)$$

$$x_{ij} \leq U_i(q_{ij} - r_{ij}), \quad i = 1, \dots, M, j = 1, \dots, N \quad (4.24)$$

$$q_{ij} \leq q_{i,j+1}, \quad i = 1, \dots, M, j = 1, \dots, N - 1 \quad (4.25)$$

$$r_{ij} \leq r_{i,j+1}, \quad i = 1, \dots, M, j = 1, \dots, N - 1 \quad (4.26)$$

$$\sum_{j=1}^N (q_{ij} - r_{ij}) \leq C, \quad i = 1, \dots, M \quad (4.27)$$

$$y_j \in \{0, 1\}, \quad j = 1, \dots, N \quad (4.28)$$

$$x_{ij} \in Z^+, \quad i = 1, \dots, M, j = 1, \dots, N \quad (4.29)$$

$$q_{ij} \in \{0, 1\}, \quad i = 1, \dots, M, j = 1, \dots, N \quad (4.30)$$

$$r_{ij} \in \{0, 1\}, \quad i = 1, \dots, M, j = 1, \dots, N. \quad (4.31)$$

A função objetivo (4.21) minimiza a quantidade total de objetos utilizados. As restrições (4.22) são relativas ao atendimento da demanda dos itens. O conjunto de restrições (4.23) garante que a soma do comprimento de todos os itens obtidos ao se cortar o objeto não ultrapasse o comprimento do objeto. O conjunto de restrições (4.24) garante que as pilhas abertas e fechadas sejam contabilizadas. Note que estas restrições impedem que uma pilha que ainda não foi aberta seja fechada. As restrições (4.25) e (4.26) impedem que o número de pilhas abertas ou fechadas decresça ao longo do tempo. As restrições (4.27) impedem que o limite da capacidade de pilhas abertas seja ultrapassado. As demais restrições são referentes ao domínio das variáveis.

O número de variáveis  $q_{ij}$  e  $r_{ij}$  pode ser reduzido. Em [Arbib, Marinelli e Ventura \(2016\)](#) os autores dividem a sequência de corte em  $n$  intervalos de tempo, de forma que, em cada intervalo de tempo  $h$ , é executado um conjunto de padrões de corte. Em [Arbib, Marinelli e Ventura \(2016\)](#) é utilizado  $n = M - C + 1$ . Em [Arbib, Marinelli e Ventura \(2016\)](#) este valor para  $n$  é justificado em detalhes, assim como, a forma de se determinar o conjunto de padrões de corte que deve ser executado em cada intervalo  $h$ . Considere:

### Índices:

$n$  : quantidade de intervalos de tempo distintos.



**Parâmetros:**

$J_h$  : conjunto contendo os padrões de corte executados no intervalo de tempo  $h$ .

**Variáveis de decisão:**

$$q_{ih} = \begin{cases} 1, & \text{se a } i\text{-ésima pilha é aberta em algum instante anterior ao } h\text{-ésimo} \\ & \text{instante de tempo,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$r_{ih} = \begin{cases} 1, & \text{se a } i\text{-ésima pilha é fechada em algum instante anterior ao } h\text{-ésimo} \\ & \text{instante de tempo,} \\ 0, & \text{caso contrário.} \end{cases}$$

Então o conjunto de restrições (4.24) pode ser substituído por:

$$\sum_{j \in J_h} x_{ij} \leq d_i(q_{ih} - r_{ih}), \quad i = 1, \dots, M, h = 1, \dots, n. \quad (4.32)$$

A construção do próximo modelo proposto em [Arbib, Marinelli e Ventura \(2016\)](#) depende da definição do conceito de *Track*, considere:

**Índices:**

$K$  : conjunto de índices referentes aos padrões de corte.

**Parâmetros:**

$\alpha^{(j)} = (\alpha_1^{(j)}, \alpha_2^{(j)}, \dots, \alpha_M^{(j)})^t$  : representa o  $j$ -ésimo padrão de corte. A  $i$ -ésima coordenada do vetor corresponde à quantidade de itens do tipo  $i$  que está presente no padrão de corte;

$P = \{\alpha^{(j)} \mid j \in K\}$  : conjunto de padrões de corte unidimensionais;

$C$  : limitante para o número máximo de pilhas abertas.

**Definição:** Seja  $T$  uma matriz 0-1 com  $M$  linhas não nulas.  $T$  é uma *track* para  $P$  se para qualquer  $k \in K$  existe uma coluna  $h$ , pertencente à matriz  $T$ , tal que:

$$\alpha_i^{(k)} > 0 \implies t_{ih} > 0, \quad i = 1, \dots, N.$$

É equivalente dizer que  $T$  é uma *track* para  $P$  e  $P$  é suportado por  $T$ .

**Definição:** Seja  $\omega(T)$  o maior número de elementos não nulos de uma coluna de uma *Track*  $T$ . Seja  $t_{ij}$  o elemento da matriz  $T$  correspondente à linha  $i$  e à coluna  $j$ . Dizemos que  $T$  é factível se  $T$  possui as propriedades:

- Para  $j < k$ ,  $i \leq M$ , se  $t_{ij} = t_{ik} = 1$  então  $t_{ih} = 1$ , para todo  $j < h < k$ . Esta propriedade é conhecida, em inglês, por *consecutive one property (C1P)*.
- $\omega(T) \leq C$ .

Em [Arbib, Marinelli e Ventura \(2016\)](#) os autores demonstram que um conjunto de padrões de corte possui um sequenciamento tal qual o número máximo de pilhas abertas é igual ou inferior a  $C$  se e somente se este conjunto de padrões de corte é suportado por uma *Track* factível. Com base nesta propriedade, o CS-LOSP unidimensional pode ser reformulado como o problema de determinar um conjunto de padrões de corte  $P$  que atenda a demanda ao mesmo tempo em que minimiza o desperdício de material e que seja suportado por uma *Track* factível.

Seja  $A = [\alpha^{(1)} \ \alpha^{(2)} \ \dots, \ \alpha^{(|K|)}]$  uma matriz contendo todas as soluções factíveis de

$$\sum_{i=1}^N l_i \alpha_i^{(k)} \leq L, \quad \alpha_i^{(k)} \in Z^+.$$

A matriz  $A$  possui  $M$  linhas e  $|K|$  colunas e os vetores  $\alpha_i^{(k)}$  possuem  $M$  elementos cada. Utilizando  $\lambda_j^k \in \{0, 1\}$  para selecionar a  $k$ -ésima coluna de  $A$ , seja  $\mathbf{x}_j$  o padrão de corte aplicado ao  $j$ -ésimo objeto, este padrão de corte pode ser reescrito como:

$$\mathbf{x}_j = \sum_{k=1}^{|K|} \lambda_j^k \alpha^{(k)}, \quad (4.33)$$

com  $\sum_k \lambda_j^k = 1$ ,  $\lambda_j^k \geq 0$  e inteiro. Substituindo  $x_{ij}$  por (4.33), em (4.22), (4.23), (4.24) e (4.29) e considerando  $p_k = \sum_j \lambda_j^k$ , obtemos o modelo de [Gilmore e Gomory \(1961, 1963, 1965\)](#):

$$\begin{aligned}
& \min \sum_{j=1}^{|K|} p_j & (4.34) \\
\text{sujeito a: } & \sum_{j=1}^{|K|} \alpha_i^{(j)} p_j = d_i, & i = 1, \dots, M \\
& p_j \in Z^+, & j = 1, \dots, |K|.
\end{aligned}$$

Seja  $T$  uma *Track* factível com dimensões  $M \times n$ . Considere o conjunto  $K_h$  contendo os índices dos padrões de corte  $\alpha^{(k)}$  suportados pela  $h$ -ésima coluna de  $T$ . Seja  $p_h^k$  o comprimento da sequência de padrões de corte pertencentes a  $K_h$ . Substituindo  $p_k$  por  $p_h^k$  em (4.34) e utilizando as demais restrições do modelo (Arbib, Marinelli e Ventura-1), obtemos o modelo melhorado:

(Arbib, Marinelli e Ventura-2)

$$\min \sum_{h=1}^n \sum_{k \in K_h} p_h^k \quad (4.35)$$

$$\text{sujeito a: } \sum_{h=1}^n \sum_{k \in K_h} \alpha_i^{(k)} p_h^k = d_i, \quad i = 1, \dots, M \quad (4.36)$$

$$\sum_{k \in K_h} \alpha_i^{(k)} p_h^k \leq d_i (q_{ih} - r_{ih}), \quad i = 1, \dots, M, h = 1, \dots, n \quad (4.37)$$

$$\sum_{i=1}^M (q_{ih} - r_{ih}) \leq C, \quad h = 1, \dots, n, \quad (4.38)$$

$$q_{ih} \leq q_{i,h+1}, \quad i = 1, \dots, M, h = 1, \dots, n-1 \quad (4.39)$$

$$r_{ih} \leq r_{i,h+1}, \quad i = 1, \dots, M, h = 1, \dots, n-1 \quad (4.40)$$

$$p_h^k \in Z^+, \quad h = 1, \dots, n, k = 1, \dots, |K| \quad (4.41)$$

$$q_{ih} \in \{0, 1\}, \quad i = 1, \dots, M, h = 1, \dots, n \quad (4.42)$$

$$r_{ih} \in \{0, 1\}, \quad i = 1, \dots, M, h = 1, \dots, n. \quad (4.43)$$

O modelo Arbib, Marinelli e Ventura-2 equivale à decomposição de Dantzig-Wolfe da formulação (Arbib, Marinelli e Ventura-1). Os autores mostram em Arbib, Marinelli e Ventura (2016) que  $n = M - C + 1$  e, conseqüentemente, o número de restrições presentes nos conjuntos (4.36)-(4.43) é da ordem de  $N^2$ , ou seja, o número de restrições presentes nesta formulação é significativamente menor que a quantidade de restrições presentes nas demais formulações apresentadas.

### 4.3 Considerações sobre o problema integrado

Os modelos apresentados neste capítulo permitem resolver os problemas de corte de estoque e de sequenciamento dos padrões de corte de forma integrada. A possibilidade de limitar o número máximo de pilhas abertas a partir do sequenciamento dos padrões de corte é essencial para aplicações nas quais existam restrições de espaço de armazenamento.

Os experimentos computacionais realizados em [Pinto \(2004\)](#) demonstraram que a utilização do modelo (YLI) apresenta dificuldades para resolver problemas de forma exata. Mesmo para instâncias contendo um número pequeno de itens de tipos distintos. Como alternativa à aplicação do modelo de forma direta, em [Pinto \(2004\)](#) é feita uma decomposição do CS-LOSP na qual as restrições que conectam o PCE e o MOSP são relaxadas.

Dada a dificuldade de se encontrar uma solução exata utilizando o modelo (YLI), abordagens heurísticas podem ser indicadas visando reduzir a complexidade computacional do problema integrado. Em [Pileggi \(2003\)](#) são propostas 3 abordagens heurísticas para resolver o problema em tempo razoável. As 3 heurísticas propostas em [Pileggi \(2003\)](#) podem ser resumidas da seguinte forma:

- Abordagem I: Os problemas de corte de estoque e de sequenciamento dos padrões de corte são resolvidos de forma alternada. Primeiramente, o PCE é resolvido e, em seguida, o MOSP é resolvido utilizando como parâmetros os padrões de corte obtidos a partir da solução do PCE, obtida anteriormente. Caso o número de pilhas abertas supere o limite preestabelecido, um novo conjunto de padrões de corte deve ser gerado. Em [Pileggi \(2003\)](#) são propostas 3 variantes desta abordagem.
- Abordagem II: Um procedimento iterativo é utilizado para se resolver o PCE e o MOSP simultaneamente. O procedimento consiste em, a cada iteração, utilizar uma heurística construtiva gulosa para gerar os padrões de corte considerando a restrição no número máximo de pilhas que podem ser abertas.
- Abordagem III: O método simplex é utilizado em conjunto com um procedimento de geração de colunas pra resolver o PCE e o MOSP. É utilizada uma heurística proposta em [Yuen \(1995\)](#) para restringir o número máximo de pilhas abertas durante a geração de cada coluna.

O modelo de [Arbib, Marinelli e Ventura \(2016\)](#) representa um importante avanço no estado da arte do problema integrado. A utilização deste modelo permitiu a solução de problemas com 20 itens de tipos distintos, cuja solução era desconhecida. O sucesso da formulação ([Arbib, Marinelli e Ventura-2](#)) se deve principalmente a dois fatores. O primeiro destes é uma redução significativa no número de restrições presentes no modelo

enquanto que, na formulação (YLI), o número de restrições cresce exponencialmente em função de  $M$ , o número de restrições presentes no modelo (Arbib, Marinelli e Ventura-2) cresce quadraticamente em função de  $M$ . Já o segundo é o fato da relaxação linear desta formulação poder ser resolvida utilizando o método de geração de colunas. As formulações propostas em Arbib, Marinelli e Ventura (2016) foram elaboradas apenas para o CS-LOSP unidimensional, porém, acreditamos ser possível estender o modelo Arbib, Marinelli e Ventura-2 para dimensões maiores.

## 5 Formulações propostas para o problema integrado

O problema de corte de estoque com limitação no número máximo de pilhas abertas é um problema recente. Do nosso conhecimento, na literatura só existem três formulações para o problema, a formulação proposta em [Yanasse e Lamosa \(2007\)](#) e os modelos construídos em [Arbib, Marinelli e Ventura \(2016\)](#).

Neste capítulo são propostas novas formulações para o CS-LOSP. Para a construção dos modelos são utilizadas as formulações propostas em [Gilmore e Gomory\(1961, 1963, 1965\)](#) e [Kantorovich \(1960\)](#) para lidar com a parte do problema referente ao PCE e os modelos propostos em [Tang e Denardo \(1988\)](#), [Laporte, Salazar-González e Semet \(2004\)](#) e [Catanzaro, Golveia e Labbé \(2015\)](#) para lidar com a parte do problema referente ao MOSP.

### 5.1 Formulação Proposta baseada no modelo de Laporte, Salazar-González e Semet (2004)

Tomando como base as formulações de [Gilmore e Gomory\(1961, 1963, 1965\)](#) e [Laporte, Salazar-González e Semet \(2004\)](#) é construída uma formulação para o CS-LOSP. A formulação requer a criação de variáveis adicionais  $v_j$  de forma similar ao que foi feito no Capítulo 4. Considere então:

$$\begin{aligned} v_j &\leq p_j, & j \in J \\ K v_j &\geq p_j, & j \in J \\ v_j &\in \{0, 1\}, & j \in J. \end{aligned}$$

O modelo proposto busca remover completamente o padrão de corte do modelo caso ele não seja utilizado, a formulação resultante foi nomeada Laporte Salazar-González e Semet com remoção de vértices (**LGSRV**). Como os padrões de corte são simbolizados por vértices em um grafo completo, remover o padrão de corte equivale a retirar os arcos que estão conectados ao padrão de corte não utilizado. A representação em grafo da remoção de vértices é semelhante à representação presente na [Figura 8](#), porém, com a diferença que a representação da [Figura 10](#) versa sobre o CS-LOSP, enquanto a da [Figura 8](#) ilustrava um modelo proposto para resolver o MTSP, sendo assim, os vértices neste grafo simbolizam os padrões de corte do problema e não as tarefas a serem realizadas, como foi visto na [Figura 8](#).

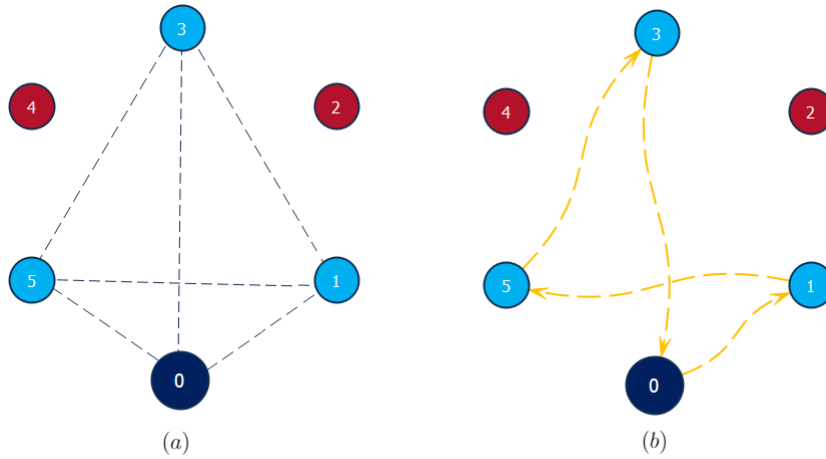


Figura 10 – Representa o caso no qual há um total de cinco padrões de corte disponíveis para resolução do problema, e apenas 3 destes são utilizados na solução ilustrada. Os padrões não utilizados estão representados em vermelho, enquanto os demais em azul claro, o vértice 0 representa o padrão de corte fictício. Em (a) temos o grafo completo excluindo os arcos que tem conexão com os vértices 2 e 4, ou seja, o primeiro e o último padrão de corte a ser executado é o fictício e os vértices 2 e 4 não devem ser executados em nenhuma ordem, fora isso, qualquer padrão de corte pode suceder ou anteceder qualquer padrão de corte na ordem de execução. Em (b) temos um circuito no grafo que percorre todos os vértices com exceção dos vértices 2 e 4 com começo e fim no vértice 0. Este circuito representa a solução na qual os padrões de corte foram utilizados na ordem 1-5-3 e os padrões de corte 2 e 4 não foram utilizados nesta solução.

Para impedir que padrões de corte não utilizados estejam presentes no sequenciamento, as restrições (3.11) e (3.12) devem ser modificadas. Para isto, considere:

$$\sum_{j \in J' \setminus \{i\}} x_{ij} = v_i, \quad i \in J' \quad (5.1)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = v_j, \quad j \in J' \quad (5.2)$$

$$v_1 = 1. \quad (5.3)$$

O conjunto de restrições (5.1) e (5.2) garantem que, caso o padrão de corte  $j$  não seja utilizado, este padrão de corte não estará presente no sequenciamento dos padrões, do contrário, as restrições (5.1) e (5.2) equivalem às restrições (3.11) e (3.12). Considerando o padrão de corte fictício, como o padrão de corte 1, a restrição (5.3) garante que o padrão de corte fictício, que representa o começo e fim das operações, seja utilizado. Considere:

**Índices:**

$M$ : quantidade de tipos de itens;

$N$ : quantidade de padrões de corte.

### Parâmetros:

$\alpha^{(j)} = (\alpha_1^{(j)}, \alpha_2^{(j)}, \dots, \alpha_M^{(j)})^t$ : representa o  $j$ -ésimo padrão de corte. A  $i$ -ésima coordenada do vetor corresponde à quantidade de itens do tipo  $i$  que está presente no padrão de corte;

$K$ : constante dada por (4.4);

$d_i$ : demanda do item do tipo  $i$ .

### Variáveis de decisão:

$p_j$ : quantidade de vezes que o padrão de corte  $j$  é utilizado;

$z_{it}$ : variável binária utilizada para contabilizar o número de pilhas abertas. A quantidade de pilhas abertas é menor ou igual a  $C$  se  $\sum_{i \in J} \sum_{t \in T} z_{it} \leq M - C$ ;

$$x_{ij} = \begin{cases} 1, & \text{se o padrão de corte } j \text{ é executado imediatamente após a} \\ & \text{execução do padrão de corte } i, \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{it} = \begin{cases} 1, & \text{se o item do tipo } t \text{ é obtido ao utilizar o padrão de corte } i, \\ 0, & \text{caso contrário;} \end{cases}$$

$$v_j = \begin{cases} 1, & \text{se o padrão de corte } j \text{ é utilizado na solução do problema,} \\ 0, & \text{caso contrário.} \end{cases}$$

$u_i$ : variável auxiliar usada para impedir a formação de subciclos, seguindo o procedimento proposto em [Miller, Tucker e Zemlin \(1960\)](#).

O modelo resultante é chamado de Laporte, Salazar-González e Semet com remoção de vértices ([LGSRV](#)) e se encontra a seguir:



(LGSRV)

$$\min \sum_{j=1}^N p_j \quad (5.4)$$

$$\text{sujeito a: } \sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i \in T \quad (5.5)$$

$$\sum_{i \in J} \sum_{t \in T} z_{it} \leq M - C, \quad (5.6)$$

$$v_j \leq p_j, \quad j \in J \quad (5.7)$$

$$K v_j \geq p_j, \quad j \in J \quad (5.8)$$

$$v_1 = 1, \quad (5.9)$$

$$\sum_{j \in J' \setminus \{i\}} x_{ij} = v_i, \quad i \in J' \quad (5.10)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = v_j, \quad j \in J' \quad (5.11)$$

$$y_{it} = v_i, \quad i \in J, t \in T_i \quad (5.12)$$

$$\sum_{t \in T} y_{it} \leq C, \quad i \in J \quad (5.13)$$

$$x_{ij} + y_{jt} - y_{it} \leq z_{it} + 1, \quad i \in J', j \in J, t \in T \quad (5.14)$$

$$1 \leq u_i \leq N, \quad i \in J \quad (5.15)$$

$$u_i - u_j + N x_{ij} \leq N - 1, \quad i, j \in J \quad (5.16)$$

$$z_{it} = 0, \quad i \in J, t \in T \setminus T_i \quad (5.17)$$

$$v_j \in \{0, 1\}, \quad j \in J' \quad (5.18)$$

$$x_{ij} \in \{0, 1\}, \quad i \in J', j \in J' \quad (5.19)$$

$$y_{it} \in \{0, 1\}, \quad i \in J, t \in T \quad (5.20)$$

$$z_{it} \in \{0, 1\}, \quad i \in J, t \in T \quad (5.21)$$

$$p_j \in Z^+, \quad j \in J' \quad (5.22)$$

$$u_j \in Z^+, \quad j \in J. \quad (5.23)$$

A função objetivo (5.4) é similar à considerada em (2.12). Considerando objetos de mesmo comprimento, esta função objetivo minimiza o desperdício de material. De forma análoga, as restrições (5.5) equivalem ao conjunto de restrições (2.13) que garantem

o atendimento da demanda. O conjunto de restrições (5.6) garante que o número de pilhas abertas seja, no máximo,  $C$ . Este conjunto de restrições é utilizado para transformar um MTSP em um MOSP conforme foi discutido na Seção 3.3. As restrições (5.7) e (5.8) garantem que  $v_j = 0$ , se e somente se  $p_j = 0$ . Note que, uma vez que as restrições (5.7) e (5.8) só estão definidas para  $j \in J$ , a variável  $p_1$  é livre para assumir qualquer valor. Além disso, como a função objetivo busca minimizar a soma das variáveis  $p_j$ , o valor de  $p_1$  sempre será igual a zero, ou seja, o padrão de corte fictício não interfere na contagem de objetos necessários para satisfazer a demanda.

A restrição (5.9) garante que o padrão de corte fictício seja utilizado. As restrições (5.10) garantem que, se  $v_j = 0$ , então, a execução do padrão de corte  $j$  não sucede a execução de nenhum padrão de corte e, se  $v_j = 1$ , o conjunto de restrições (5.10) assegura que um, e apenas um, padrão de corte seja executado imediatamente após a execução do padrão de corte  $j$ , para todo  $j \in J'$ . As restrições (5.11) garantem que, se  $v_j = 0$ , então não há nenhum padrão de corte que anteceda a execução do padrão de corte de tipo  $j$  e, se  $v_j = 1$ , o conjunto de restrições (5.11) assegura que para cada  $j \in J'$  exista um, e apenas um, padrão de corte que anteceda o processamento do padrão de corte  $j$ . O conjunto de restrições (5.12) garante que, se o padrão de corte  $j$  é utilizado, então todas as pilhas abertas devem ser levadas em conta, do contrário, a restrição se torna trivial. As restrições (5.13) impedem que o limite da capacidade de pilhas abertas seja ultrapassado, o conjunto de restrições (5.14) assegura a contabilização das pilhas abertas na variável  $z_{it}$ , sempre que necessário. As restrições (5.15) e (5.16), propostas em Miller, Tucker e Zemlin (1960), impedem a formação de subciclos e são apresentadas em mais detalhes na Seção 3.6.

O conjunto de restrições (5.17) foi elaborado em Laporte, Salazar-González e Semet (2004) para o MTSP. Neste contexto, o conjunto de restrições impede que uma ferramenta seja colocada na máquina se ela não é requerida para processar a próxima tarefa. As restrições (5.18)-(5.23) são relativas ao domínio das variáveis.

O conjunto de restrições (5.15) utiliza o número total de padrões de corte factíveis, dado por  $N$ . Este conjunto de restrições pode ser melhorado se considerarmos apenas os padrões de corte que foram de fato utilizados, basta então substituir  $N$  por  $\sum_{j \in J} v_j$ , resultando em:

$$1 \leq u_i \leq \sum_{j \in J} v_j, \quad i = 2, \dots, N. \quad (5.24)$$

Dado que, na restrição (5.16),  $N$  multiplica a variável  $x_{ij}$ , a mesma modificação não pode ser efetuada, pois o modelo deixaria de ser linear o que representa uma grande desvantagem.

### 5.1.1 Aplicação das melhorias propostas em Laporte, Salazar-González e Semet (2004)

Na Seção 3.4 foram estudadas algumas modificações no modelo de Laporte, Salazar-González e Semet (2004) de forma a melhorar sua relaxação linear. O propósito desta seção é de analisar tais melhorias no contexto do problema integrado. De forma muito próxima ao que foi feito em (3.21) podemos substituir (5.6) por:

$$\sum_{j=1}^N \sum_{i=1}^M z_{it} + \sum_{i \in J_c} \sum_{j \neq i} |T_j \setminus T_i| \leq M - C.$$

Note que as alterações feitas para integrar o MOSP ao PCE não interferem nessa restrição. Já as restrições (3.22), (3.23) e (3.24) possuem as variáveis  $x_{ij}$  e em (5.10) e (5.11) foram efetuadas alterações que permitem que existam padrões de corte  $f$  que não são utilizados em nenhum momento, ou seja  $x_{fj} = x_{jf} = 0, \forall j \in J'$ . Seja  $F \subset J$  o conjunto de todos os padrões de corte que não são utilizados. Considere o conjunto de restrições (3.22) no contexto do CS-LOSP. Para  $j \in J \setminus F$  as restrições (3.22) podem ser reescritas como:

$$\sum_{\substack{i \in J_t \setminus \{j\} \\ i \in J \setminus F}} x_{ij} + z_{it} \leq 1, \quad t \in T. \quad (5.25)$$

O conjunto de restrições é similar ao apresentado em (3.22), ou seja, se o padrão de corte  $j$  é executado imediatamente após o padrão de corte  $i$  então  $x_{ij} = 1$  e dessa forma a restrição garante  $z_{it} = 0$ . Note que  $i \in J_t$  e, portanto, o item do tipo  $t$  é gerado ao executar o padrão de corte  $i$ . Sendo assim, as restrições garantem que nenhuma pilha é aberta. Se o padrão de corte  $j$  não é executado imediatamente após um padrão de corte  $i$  que gera o item do tipo  $t$ , então não há garantias que a pilha referente ao item do tipo  $t$  já está aberta, ou seja,  $z_{it}$  é livre para assumir qualquer valor. Se  $j \in F$  as restrições (3.22) podem ser reescritas como:

$$z_{it} \leq 1, \quad t \in T. \quad (5.26)$$

Como  $z_{it}$  é uma variável binária, as restrições (5.26) são trivialmente satisfeitas. De forma similar, as restrições (3.23) também podem ser reescritas. Para  $i, j \in J \setminus F$  o conjunto de restrições (3.23) pode ser reescrito como:

$$\sum_{t \in T_i \setminus T_j} y_{jt} \geq (C - |T_j|)x_{ij}, \quad i \in J_c, j \in J : |T_j| < C. \quad (5.27)$$

Portanto, as restrições (5.27) são idênticas às restrições (3.23) para  $i, j \in J \setminus F$ . Caso  $i \in F$  ou  $j \in F$  então o conjunto de restrições (3.23) pode ser reescrito como:

$$\sum_{t \in T_i \setminus T_j} y_{jt} \geq 0, \quad i \in J_c, j \in J : |T_j| < C. \quad (5.28)$$

Dado que  $y_{it}$  é uma variável binária, o conjunto de restrições (5.28) é trivialmente satisfeito. Por último, podem ser reescritas as restrições (3.24). Para  $j \in J \setminus F$ , o conjunto de restrições (3.24) pode ser reescrito como:

$$\sum_{t \in T_j} z_{jt} \geq \sum_{\substack{i \in J \setminus \{j\} \\ |T_i| \neq C \\ i \notin F}} s_{ij} x_{ij}, \quad j \in J, \quad (5.29)$$

considerando  $s_{ij} = \max\{0, |T_i \cup T_j|\}$ . Portanto, para  $j \in J \setminus F$  o conjunto de restrições é similar ao apresentado em (3.24), ou seja, se o padrão de corte  $i$  antecede o padrão de corte  $j$ ,  $s_{ij}$  fornece um limitante inferior para a quantidade de novas pilhas abertas entre os padrões de corte  $i$  e  $j$ . Para  $j \in F$  o conjunto de restrições (3.24) pode ser reescrito como:

$$\sum_{t \in T_j} z_{jt} \geq 0, \quad j \in J. \quad (5.30)$$

Uma vez que  $z_{jt}$  é uma variável binária, o conjunto de restrições (5.30) é trivialmente satisfeito. Sendo assim, podemos concluir que as modificações efetuadas em (5.10) e (5.11) não interferem na aplicação das melhorias. Desta forma, as melhorias apresentadas em (3.21), (3.22), (3.23) e (3.24) podem ser inseridas no modelo LGSRV sem nenhum problema.

## 5.2 Formulação proposta baseada nos modelos de Catanzaro, Golveia e Labbé (2015)

Tanto o modelo de Laporte, Salazar-González e Semet (2004) quanto o de Tang e Denardo (1988) possuem em comum um conjunto de restrições que garante que as ferramentas necessárias para o processamento de uma tarefa estejam na máquina durante sua execução. Para o modelo de Tang e Denardo (1988) estas restrições são dadas por (3.4) e, para Laporte, Salazar-González e Semet (2004), as restrições são dadas por (3.13). Quando o MTSP é transformado em um MOSP, o significado deste conjunto de restrições se altera e passa a significar que todas as pilhas abertas sejam levadas em conta quando um padrão de corte é utilizado. A estratégia de remoção de pilhas consiste em modificar este conjunto de restrições de forma que somente se o padrão de corte é utilizado, as pilhas são consideradas e, caso contrário, o conjunto de restrições se torne trivial.

Em Catanzaro, Golveia e Labbé (2015), a partir das modificações feitas no modelo de Laporte, Salazar-González e Semet (2004), as restrições (3.13) se tornam redundantes e, portanto, são retiradas do modelo. Desta maneira, não foi possível utilizar a

estratégia de remoção de pilhas e o modelo integrado baseado na formulação de [Catanzaro, Golveia e Labbé \(2015\)](#) utiliza a estratégia de remoção de vértices.

Em conjunto com a estratégia de remoção de vértices, adicionamos ao modelo o seguinte conjunto de restrições:

$$\sum_{\substack{i,j \in J \\ i \neq j}} \sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \leq M - C. \quad (5.31)$$

O conjunto de restrições (5.31) é similar aos conjuntos de restrições (4.8) e (5.6) usados nos modelos [YLI](#) e [LGSRV](#), respectivamente. Este conjunto de restrições é utilizado para transformar um MTSP em um MOSP conforme foi discutido na Seção 3.3 e é essencial para impedir que o número máximo de pilhas abertas supere o limitante dado por  $C$ . Ao utilizar a estratégia de remoção de vértices, caso um padrão de corte  $f$  não seja utilizado, então temos  $x_{fj} = x_{jf=0}$  para todo  $j \in J'$ . Esta modificação afeta o conjunto de restrições (5.31), permitindo que existam índices  $i, j$  e  $t$  tais que:

$$\sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \leq 0. \quad (5.32)$$

No contexto do MTSP, isso representa um decrescimento no número de trocas de ferramentas durante a transição da execução da tarefa  $i$  para a tarefa  $j$ . Naturalmente, tal decrescimento viola as regras que regem o MTSP e, portanto, deve ser eliminado para garantir que o modelo reflita de fato o problema real. Para tal, adicionamos ao modelo o seguinte conjunto de restrições:

$$\sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \geq 0, \quad i, j \in J, i \neq j, \quad (5.33)$$

e assim garantimos que a contagem de trocas de ferramentas seja correta no contexto do MTSP e, conseqüentemente, eliminamos sequenciamentos infactíveis para o CS-LOSP.

Nesta Seção, uma formulação para o CS-LOSP é proposta baseada no modelo [Catanzaro et al. II](#). Para fins de notação, o modelo baseado em [Catanzaro et al. II](#) é denominado de [Catanzaro et al. Integrado 2 \(CI2\)](#). O modelo é apresentado a seguir:

### Índices:

$M$ : quantidade de itens de tipos distintos;

$N$ : quantidade de padrões de corte.

**Parâmetros:**

$\alpha^{(j)} = (\alpha_1^{(j)}, \alpha_2^{(j)}, \dots, \alpha_M^{(j)})^t$  : representa o  $j$ -ésimo padrão de corte. A  $i$ -ésima coordenada do vetor corresponde à quantidade de itens do tipo  $i$  que está presente no padrão de corte;

$K$ : constante dada por (4.4);

$d_i$  : demanda do item do tipo  $i$ .

**Variáveis de decisão:**

$p_j$  : quantidade de vezes que o padrão de corte  $j$  foi utilizado;

$y_{ij}^t$  : variável binária utilizada na contabilização do número de pilhas abertas;

$$x_{ij} = \begin{cases} 1, & \text{se a execução do padrão de corte } j \text{ é feita imediatamente após a} \\ & \text{execução do padrão de corte } i \\ 0, & \text{caso contrário;} \end{cases}$$

$$v_j = \begin{cases} 1, & \text{se o padrão de corte } j \text{ é utilizado na solução do problema,} \\ 0, & \text{caso contrário;} \end{cases}$$

$u_i$  : variável auxiliar usada para impedir a formação de subciclos, conforme o procedimento proposto em [Miller, Tucker e Zemlin \(1960\)](#).

(CI2)

$$\min \sum_{j=1}^N p_j \quad (5.34)$$

$$\text{sujeito a: } \sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i \in T \quad (5.35)$$

$$\sum_{\substack{i,j \in J \\ i \neq j}} \sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \leq M - C, \quad (5.36)$$

$$\sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \geq 0, \quad i, j \in J, i \neq j \quad (5.37)$$

$$v_j \leq p_j, \quad j \in J \quad (5.38)$$

$$K v_j \geq p_j, \quad j \in J \quad (5.39)$$

$$v_1 = 1, \quad (5.40)$$

$$\sum_{j \in J' \setminus \{i\}} x_{ij} = v_i, \quad i \in J' \quad (5.41)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = v_j, \quad j \in J' \quad (5.42)$$

$$1 \leq u_i \leq \sum_{j \in J} v_j, \quad i \in J \quad (5.43)$$

$$u_i - u_j + N x_{ij} \leq N - 1, \quad i, j \in J, i \neq j \quad (5.44)$$

$$\sum_{k \in J' \setminus \{i\}} y_{ki}^t - \sum_{j \in J' \setminus \{i\}} y_{ij}^t \geq 0, \quad i \in J', t \in T \setminus T_i \quad (5.45)$$

$$\sum_{\substack{t \in T: \\ t \in T_i \\ t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T: \\ t \notin T_i \\ t \notin T_j}} y_{ij}^t \leq (C - |T_j|) x_{ij}, \quad i, j \in J', i \neq j \quad (5.46)$$

$$y_{ij}^t \leq x_{ij}, \quad i, j \in J', i \neq j, t \in T \setminus T_j \quad (5.47)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in J', i \neq j \quad (5.48)$$

$$y_{ij}^t \in \{0, 1\}, \quad i, j \in J', i \neq j, \quad t \in (T \setminus T_i) \cup (T_i \setminus T_j) \quad (5.49)$$

$$v_j \in \{0, 1\}, \quad j \in J', \quad (5.50)$$

$$p_j \in Z^+, \quad j \in J' \quad (5.51)$$

$$u_j \in Z^+, \quad j \in J. \quad (5.52)$$

A função objetivo (5.34) é similar à considerada em (2.12). Esta função objetivo minimiza o desperdício de material. De forma análoga, as restrições (5.35) equivalem ao conjunto de restrições (2.13) que garantem o atendimento da demanda. O conjunto de restrições (5.36) garante que o número máximo de pilhas abertas seja igual a  $C$  e é utilizado para transformar um MTSP em um MOSP conforme foi discutido na Seção 3.3. O conjunto

de restrições (5.37) elimina a presença de sequenciamentos inactiváveis. Este conjunto foi proposto em (5.33). As restrições (5.38) e (5.39) garantem que  $v_j = 0$  se e somente se  $p_j = 0$ . Note que, uma vez que as restrições (5.38) e (5.39) só estão definidas para  $j \in J$ ,  $p_1$  é livre para assumir qualquer valor. Como a função objetivo busca minimizar a soma das variáveis  $p_i$ , o valor de  $p_1$  sempre será igual a zero, ou seja, o padrão de corte fictício não interfere na contagem de objetos necessários para satisfazer a demanda.

A restrição (5.40) garante que o padrão de corte fictício seja utilizado. As restrições (5.41) garantem que, se  $v_i = 0$ , então a execução do padrão de corte de tipo  $i$  não sucede a execução de nenhum padrão de corte e, se  $v_i = 1$ , o conjunto de restrições (5.41) assegura que um, e apenas um, padrão de corte seja executado imediatamente após a execução do padrão de corte  $i$ , para todo  $i \in J'$ . As restrições (5.42) garantem que, se  $v_j = 0$ , então não há nenhum padrão de corte que anteceda a execução do padrão de corte de tipo  $j$  e, caso  $v_j = 1$ , o conjunto de restrições (5.42) assegura que para cada  $j \in J'$  exista um, e apenas um, padrão de corte  $i$  que anteceda o processamento do padrão de corte  $j$ . O conjunto de restrições (5.43) se baseia nas restrições propostas em Miller, Tucker e Zemlin (1960). Este conjunto de restrições é apresentado em (5.24) e, juntamente com as restrições (5.44), impedem a formação de subciclos. As restrições (5.44) foram propostas em Miller, Tucker e Zemlin (1960) e são apresentadas em mais detalhes na Seção 3.6. O conjunto de restrições (5.46) garante que o limite máximo no número de pilhas abertas não seja ultrapassado.

As restrições (5.45) e (5.47) foram elaboradas em Catanzaro, Golveia e Labbé (2015) para o MTSP. Neste contexto, as restrições (5.45) impõem que se a ferramenta  $t$  é carregada na transição da tarefa  $i$  para alguma tarefa  $j$ , então, esta ferramenta deve ter sido carregada ou inserida na transição de alguma tarefa  $k$  para a tarefa  $i$ ,  $i \neq k$ . O conjunto de restrições (5.47) permite que uma ferramenta  $t$  pertencente a  $T \setminus T_j$  seja carregada na transição das tarefas  $i$  e  $j$ , se o processamento da tarefa  $j$  ocorre logo após o processamento da tarefa  $i$  em uma solução ótima. Como foi demonstrado em Yanasse (1997), uma solução ótima do MTSP que realiza um total de  $M - C$  trocas de ferramentas representa um sequenciamento para o MOSP no qual, no máximo,  $C$  pilhas são abertas simultaneamente, quando consideramos as ferramentas como itens e as tarefas como padrões de corte. Embora os conjuntos de restrições (5.45) e (5.47) não possuam uma interpretação bem definida para o contexto do CS-LOSP, elas são essenciais para garantir o funcionamento do modelo, uma vez que, em conjunto com as restrições (5.36), asseguram que a limitação no número máximo de pilhas que podem ser abertas não seja ultrapassado. O conjunto de restrições (5.46) impede que o limite da capacidade da máquina seja ultrapassado. As restrições (5.48)-(5.52) definem o domínio das variáveis.

Conforme foi discutido na Seção 3.5, os modelos propostos em Tang e Denardo (1988) e Laporte, Salazar-González e Semet (2004) não contabilizam a inserção de ferra-



mentas para iniciar o processo, enquanto aqueles propostos em [Catanzaro, Golveia e Labbé \(2015\)](#) sim. Para garantir que todas as formulações modelem a mesma situação, todas as formulações apresentadas no Capítulo 4 seguem a abordagem de [Tang e Denardo \(1988\)](#) e [Laporte, Salazar-González e Semet \(2004\)](#), sendo assim, as formulações de [Catanzaro, Golveia e Labbé \(2015\)](#) foram levemente alteradas antes de serem integradas ao PCE. Estas modificações se encontram nas restrições (5.36). O conjunto de restrições (5.36) se baseia na função objetivo do modelo ([Catanzaro et al. II](#)). Os termos  $\sum_{j \in J} |T_j| x_{0j}$  são responsáveis por contabilizar a inserção de ferramentas para iniciar o processo. Para garantir que o modelo tenha a mesma interpretação das formulações [Tang e Denardo \(1988\)](#) e [Laporte, Salazar-González e Semet \(2004\)](#), estes termos devem ser eliminados, conforme foi feito em (5.36).

Acreditamos que as demais formulações propostas em [Catanzaro, Golveia e Labbé \(2015\)](#) também podem ser utilizadas para a elaboração de modelos para o CS-LOSP. Dentre as formulações construídas em [Catanzaro, Golveia e Labbé \(2015\)](#), a formulação [Catanzaro et al. III](#) é a que parece mais promissora para este fim. Como proposta futura, pretendemos utilizar o modelo [Catanzaro et al. III](#) em conjunto com as melhorias propostas no Capítulo 7 para construir um novo modelo para o CS-LOSP.

### 5.3 Formulação proposta baseada no modelo de Kantorovich (1960)

Os modelos anteriores tem em comum o fato de se basearem em [Gilmore e Gomory \(1961, 1963, 1965\)](#) para tratar da parte do problema referente ao PCE, a diferença entre eles se encontra no modelo usado para restringir o número máximo de pilhas que podem ser abertas.

Para fins de comparação, construímos um modelo para o problema integrado que se baseia na formulação de [Kantorovich \(1960\)](#), para a parte do problema relativa ao PCE e, em [Yanasse \(1997\)](#), para garantir que o número máximo de pilhas abertas não ultrapasse seu limite. Para a construção do problema integrado, é considerado que cada objeto  $j$  equivale a um padrão de corte distinto, desta forma, é preciso levar em conta as pilhas que são abertas ao se cortar os padrões de corte. Para modelar esta situação podemos nos aproveitar das variáveis  $x_{tj}$ . De forma similar ao que foi feito em [Yanasse e Lamosa \(2007\)](#) considere as variáveis binárias  $v_{tj}$ , que são iguais a 1 se e somente se o item do tipo  $t$  é obtido ao se cortar o objeto  $j$  (ou padrão de corte  $j$ ). Logo  $v_{tj}$  pode ser modelada matematicamente segundo as restrições:

$$\begin{aligned} v_{tj} &\leq x_{tj}, & j = 1, \dots, N, t = 1, \dots, M, \\ Kv_{tj} &\geq x_{tj}, & j = 1, \dots, N, t = 1, \dots, M, \\ v_{tj} &\in \{0, 1\}, & j = 1, \dots, N, t = 1, \dots, M. \end{aligned}$$

O primeiro conjunto de restrições garante que se o item do tipo  $t$  não é obtido a partir do objeto  $j$  ( $x_{tj} = 0$ ) então  $v_{tj} = 0$ . Já o segundo garante que se  $x_{tj} > 0$  então  $v_{tj} = 1$  (note que  $K$  deve ser suficientemente grande para garantir o funcionamento desse conjunto de restrições). Por último, o terceiro conjunto de restrições assegura que  $v_{tj}$  é uma variável binária. Este conjunto de variáveis é então usado para garantir que a pilha referente ao item do tipo  $t$  só seja considerada, ao cortar o objeto  $j$ , se o padrão de corte  $j$  fornece o item do tipo  $t$ . Para isso, as restrições (3.4) são substituídas por:

$$w_{jk} + (v_{tj} - 1) \leq y_{kt}, \quad k = 1, \dots, M, j = 1, \dots, M, t = 1, \dots, N.$$

Note então que, caso  $v_{tj}$  seja igual a zero, a restrição acima é trivialmente satisfeita, ou seja, o item do tipo  $t$  não é levado em conta no empilhamento durante o corte deste objeto. Considere então:

### Índices:

$N$  : quantidade de objetos disponíveis;

$M$  : quantidade de tipos de itens distintos.

### Parâmetros:

$L$  : comprimento dos objetos  $j$ ;

$l_t$  : comprimento do item do tipo  $t$ ;

$d_t$  : demanda do item do tipo  $t$ .

### Variáveis de decisão:

$x_{tj}$  : número de itens tipo  $t$  que são obtidos ao se cortar o objeto  $j$ ;

$$p_j = \begin{cases} 1, & \text{se o objeto } j \text{ é utilizado,} \\ 0, & \text{caso contrário;} \end{cases}$$

$z_{kt}$  : variável binária utilizada na contabilização do número de pilhas abertas;

$$w_{jk} = \begin{cases} 1, & \text{se o padrão de corte } j \text{ é o } k\text{-ésimo a ser processado,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{kt} = \begin{cases} 1, & \text{se o item do tipo } t \text{ é gerado a partir da execução do } k\text{-ésimo} \\ & \text{padrão de corte,} \\ 0, & \text{caso contrário.} \end{cases}$$

$$v_{tj} = \begin{cases} 1, & \text{se o item do tipo } t \text{ é obtido a partir do objeto } j \text{ e o objeto } j \text{ é} \\ & \text{utilizado na solução do problema,} \\ 0, & \text{caso contrário.} \end{cases}$$

O modelo resultante, denominado por **Kantorovich Integrado**, é dado por:

(Kantorovich Integrado)

$$\min \sum_{j=1}^N p_j \tag{5.53}$$

$$\text{sujeito a: } \sum_{j=1}^N x_{tj} \geq d_t, \quad t = 1, \dots, M \tag{5.54}$$

$$\sum_{t=1}^M l_t x_{tj} \leq L p_j, \quad j = 1, \dots, N \tag{5.55}$$

$$\sum_{k=2}^N \sum_{t=1}^M z_{kt} \leq M - C, \tag{5.56}$$

$$v_{tj} \leq x_{tj}, \quad t = 1, \dots, M, j = 1, \dots, N, \tag{5.57}$$

$$K v_{tj} \geq x_{tj}, \quad t = 1, \dots, M, j = 1, \dots, N, \tag{5.58}$$

$$\sum_{j=1}^N w_{jk} = 1, \quad k = 1, \dots, N, \tag{5.59}$$

$$\sum_{k=1}^N w_{jk} = 1, \quad j = 1, \dots, N, \tag{5.60}$$

$$w_{jk} + (v_{tk} - 1) \leq y_{kt}, \quad k = 1, \dots, N, j = 1, \dots, N, t = 1, \dots, M, \tag{5.61}$$

$$\sum_{t=1}^M y_{kt} \leq C, \quad k = 1, \dots, N, \tag{5.62}$$

$$y_{kt} - y_{(k-1)t} \leq z_{kt}, \quad k = 2, \dots, N, t = 1, \dots, M, \quad (5.63)$$

$$p_j \in Z^+, \quad j = 1, \dots, N, \quad (5.64)$$

$$w_{jk} \in \{0, 1\}, \quad j = 1, \dots, N, k = 1, \dots, N, \quad (5.65)$$

$$y_{kt} \in \{0, 1\}, \quad k = 1, \dots, N, t = 1, \dots, M, \quad (5.66)$$

$$z_{kt} \in \{0, 1\}, \quad k = 1, \dots, N, t = 1, \dots, M, \quad (5.67)$$

$$x_{tj} \in Z^+, \quad t = 1, \dots, M, j = 1, \dots, N, \quad (5.68)$$

$$v_{tj} \in \{0, 1\}, \quad t = 1, \dots, M, j = 1, \dots, N. \quad (5.69)$$

A função objetivo (5.53) minimiza a quantidade total de objetos utilizados. As restrições (5.54) são relativas ao atendimento da demanda dos itens. O conjunto de restrições (5.55) garante que a soma do comprimento de todos os itens obtidos ao se cortar o objeto não ultrapasse o comprimento do objeto. A função objetivo (5.53) é equivalente a função objetivo (2.1). De forma análoga, as restrições (5.54) e (5.55) são equivalentes às restrições (2.2)-(2.3). O conjunto de restrições (5.56) garante que o número máximo de pilhas abertas seja igual a  $C$ . Este conjunto de restrições é utilizado para transformar um MTSP em um MOSP conforme foi discutido na Seção 3.3. As restrições (5.57) e (5.58) garantem que  $v_{tj} = 0$  se e somente se  $x_{tj} = 0$ . Os conjuntos de restrições (5.59) e (5.60) garantem que todos os padrões de corte sejam executados e que seja executado um padrão de corte por vez. O conjunto de restrições (5.61) garante que, caso o padrão de corte  $j$  forneça o item do tipo  $t$ , então este item deve ser levado em conta na abertura de pilhas ao se executar o padrão de corte  $j$ , do contrário, a restrição se torna trivial. O conjunto de restrições (5.62) impede que o limite da capacidade de pilhas abertas seja ultrapassado. O conjunto de restrições (5.63) assegura a contabilização das pilhas abertas na variável  $z_{ki}$ , sempre que necessário. As restrições (5.64)-(5.69) são relativas ao domínio das variáveis.

Diferentemente dos demais modelos apresentados neste capítulo, o modelo de [Kantorovich Integrado](#) não exige a determinação do conjunto de padrões de corte factíveis. Apesar desta vantagem, os resultados computacionais para este modelo foram insatisfatórios para instâncias nas quais a demanda é grande, como será visto no Capítulo 6.

A formulação [Kantorovich Integrado](#) é baseada no modelo proposto em [Kantorovich \(1960\)](#) para tratar da parte do problema referente ao corte dos objetos e, portanto, só é válida para problemas unidimensionais. O mesmo ocorre para a formulação [Arbib, Marinelli e Ventura-1](#). Embora em [Arbib, Marinelli e Ventura \(2016\)](#) o modelo [Arbib, Marinelli e Ventura-2](#) seja elaborado considerando apenas padrões de corte unidimensionais, acreditamos ser possível estender o modelo para problemas bidimensionais. Com exceção dos modelos [Kantorovich Integrado](#), [Arbib, Marinelli e Ventura-1](#) e [Arbib, Marinelli e Ventura-2](#), as demais formulações apresentadas nesta dissertação são válidas para problemas de quaisquer dimensões.

## 6 Resultados computacionais

Uma estratégia para averiguar a qualidade dos modelos propostos consiste em estudar a qualidade da relaxação linear e a quantidade de variáveis e restrições de cada modelo. E, então, efetuar testes computacionais, com a intenção de comparar o tempo computacional gasto para encontrar uma solução ótima de um conjunto de problemas.

Além de testar a qualidade da relaxação linear dos modelos, neste capítulo visamos analisar o impacto que restringir o número máximo de pilhas abertas tem no problema, seja na quantidade de material que é desperdiçado ou no tempo necessário para solucionar os problemas. Em todos os modelos, o número máximo de pilhas que podem ser abertas é representado pelo parâmetro  $C$ . Esta constante representa um importante parâmetro no problema podendo alterar significativamente a complexidade computacional do problema e o valor da solução ótima. O valor de  $C$  pode variar de 1 até  $M$ , sendo  $M$  a quantidade de itens distintos do problema. Dependendo do valor de  $M$ , a quantidade de valores possíveis para  $C$  pode se tornar demasiadamente grande, de forma que realizar os testes para cada valor possível de  $C$  é inviável. Em [Arbib, Marinelli e Ventura \(2016\)](#) os autores realizam testes computacionais utilizando  $C = 2$ ,  $C = 3$  e  $C = 4$ . Os autores argumentam no artigo que os valores relevantes de  $C$  são tipicamente pequenos, para as instâncias nas quais foi considerado  $C = 4$ , só houve diferença entre o valor da função objetivo encontrado a partir da solução do CS-LOSP e o valor da função objetivo encontrado a partir da solução do PCE para uma instância. Como será visto neste capítulo, os resultados encontrados nesta dissertação são similares aos apresentados em [Arbib, Marinelli e Ventura \(2016\)](#), só houve diferença no valor da função objetivo ao aumentar o valor de  $C$  de três para quatro em uma única instância. Desta maneira, valores de  $C$  superiores a 4 não são considerados, além disso, para  $C = 1$  o problema se torna trivial, uma vez que apenas os padrões de corte homogêneos podem ser utilizados. Sendo assim três valores de  $C$  são considerados nos testes, sendo eles:  $C = 2$ ,  $C = 3$  e  $C = 4$ . Com isso, podemos avaliar o impacto do parâmetro  $C$ . Na execução dos testes computacionais foi considerado  $c_j = 1, \forall j \in J'$ , ou seja, a função objetivo dos problemas considera a minimização de objetos utilizados.

Testes computacionais preliminares mostraram que o modelo [Kantorovich Integrado](#) apresenta significativa dificuldade em resolver instâncias nas quais o valor da demanda requerida de itens é alta. No final da Seção 6.5, a performance do modelo [Kantorovich Integrado](#) é discutida em mais detalhes.

## 6.1 Conjunto de dados para o CS-LOSP unidimensional

Nesta dissertação é proposto um conjunto de dados para comparar a qualidade da relaxação linear dos modelos [YLI](#), [LGSRV](#) e [CI2](#), assim como, o tempo exigido por cada modelo para se determinar uma solução ótima, quando consideramos um problema unidimensional. Seja:

$L$  : comprimento do objeto a ser cortado;

$m$  : número de tipos distintos de itens;

$l$  : intervalo no qual o comprimento dos itens está contido;

$d$  : intervalo no qual a demanda requerida para cada tipo de item está contida.

Para todas as instâncias geradas, a demanda requerida  $d$  foi gerada aleatoriamente no intervalo  $[100, 400]$  e o comprimento do objeto a ser cortado  $L$  é igual a 1000. Três intervalos  $l$  foram utilizados para gerar as instâncias, sendo estes,  $l = [100, 200]$ ,  $l = [200, 500]$  e  $l = [300, 800]$ . O comprimento dos itens requeridos para cada instância foi gerado aleatoriamente no intervalo sem repetição, ou seja, itens de tipos distintos tem comprimentos distintos. A quantidade de tipos distintos de itens presentes em uma instância é 5, 8, 10 ou 20. O conjunto de dados foi dividido em 6 classes, sendo estas:

- Classe 1:  $m = 5$ ,  $l = [100, 200]$
- Classe 2:  $m = 8$ ,  $l = [100, 200]$
- Classe 3:  $m = 10$ ,  $l = [200, 500]$
- Classe 4:  $m = 20$ ,  $l = [200, 500]$
- Classe 5:  $m = 10$ ,  $l = [300, 800]$
- Classe 6:  $m = 20$ ,  $l = [300, 800]$

Cada classe contém 10 instâncias. As Classes 1-6 são usadas para comparar o tempo gasto por cada modelo para resolver os problemas e para comparar a qualidade da relaxação linear dos modelos.

Para realizar os testes computacionais foi utilizada a linguagem de programação Julia, Versão 1.1.1 (2019), que pode ser encontrada em <https://www.julialang.org/>. Esta linguagem é abordada em detalhes em [Bezanson et al. \(2017\)](#). Em conjunto com a linguagem

Julia, foi utilizado o *software* IBM CPLEX versão 12.9. O computador utilizado tem as seguintes configurações: 3.20 GHz, Intel(R)Core(TM) i7-8700 CPU e 16 GB de memória RAM.

## 6.2 Quantidade de variáveis

O grande número de variáveis inteiras e/ou binárias dificulta a resolução dos modelos, portanto, o tamanho do modelo é um aspecto importante para se determinar sua eficácia. Desta forma, comparamos a quantidade de variáveis presentes em cada formulação. Tome como  $N$  o número total de padrões de corte e  $M$  a quantidade de tipos distintos de itens, aqui analisamos a quantidade de variáveis presentes em cada formulação. Para facilitar a contagem de variáveis vamos estabelecer os seguintes conjuntos:

$$P = \{p_j, j \in J\},$$

$$V = \{v_j, j \in J\},$$

$$W = \{w_{jk}, j \in J, k \in K\},$$

$$Z = \{z_{kt}, k \in K \setminus \{1\}, t \in T\},$$

$$Y = \{y_{kt}, k \in K, t \in T\}.$$

Considerando a formulação YLI, temos que  $J$  se refere ao conjunto de todos os padrões de corte e  $K$  a ordem na qual estes padrões de corte são executados, logo, temos  $|J| = |K| = N$ , já  $T$  contém os tipos distintos de itens e, portanto,  $|T| = M$ . Desta forma, temos:

- $|P| = N$ ,
- $|V| = N$ ,
- $|W| = N^2$ ,
- $|Z| = (N - 1)M$ ,
- $|Y| = NM$ .

Apenas o conjunto de variáveis referentes a  $P$  contém variáveis inteiras, os demais conjuntos possuem variáveis binárias. Sendo assim, temos  $N$  variáveis inteiras e  $2MN + N^2 + N - M$  variáveis binárias implicando na existência de  $2MN + N^2 + 2N - M$  variáveis no total.

Para as formulações baseadas no MTSP de [Laporte, Salazar-González e Semet \(2004\)](#) os conjuntos definidos anteriormente podem ser reaproveitados. É importante frisar que, ao incluir o padrão de corte fictício, aumentamos em 1 a quantidade de padrões de corte. Além disso, dado que  $w_{jk}$  representa se a execução do padrão de corte  $j$  antecede a execução do padrão de corte  $k$  não é preciso considerar o caso em que  $j = k$ . Assim, reduzimos em  $N + 1$  o número de variáveis. Seja  $U$  o conjunto de variáveis usadas para impedir a formação de subciclos, logo  $U = \{u_i, i = 2, \dots, N + 1\}$ . Desta forma, temos:

- $|P| = N + 1,$
- $|V| = N + 1,$
- $|W| = (N + 1)^2 - (N + 1),$
- $|Z| = (N + 1)M,$
- $|Y| = (N + 1)M,$
- $|U| = N.$

Apenas as variáveis contidas em  $P$  são inteiras, sendo as demais binárias, portanto, temos  $N + 1$  variáveis inteiras e  $2(N + 1)M + (N + 1)^2 + N$  variáveis binárias, e um total de  $N(2M + N + 4) + 2M + 2$  variáveis.

A formulação [CI2](#) baseada em [Catanzaro, Golveia e Labbé \(2015\)](#) exige a modificação do conjunto  $Y$ . Para esta formulação, o conjunto  $Y$  é dado por  $Y = \{y_{jk}^t, j \in J', k \in J', j \neq k, (t \in T \setminus T_i) \cup (T \in T_i, t \notin T_j)\}$ . Além disso, as variáveis presentes no conjunto  $Z$  são eliminadas para essa formulação. Essencialmente, estas são as únicas diferenças entre esse modelo e os baseados em [Laporte, Salazar-González e Semet \(2004\)](#) no que se refere à quantidade de variáveis. Portanto, a cardinalidade dos conjuntos é dada por:

- $|P| = N + 1,$
- $|V| = N + 1,$
- $|W| = (N + 1)^2 - (N + 1),$
- $|Y| = \sum_{\substack{i, k \in J' \\ i \neq k}} |(T \setminus T_i) \cup (T_i \setminus T_k)|,$
- $|U| = N.$



Assim como nos casos anteriores, somente as variáveis presentes em  $P$  são inteiras, logo, há um total de  $N + 1$  variáveis inteiras e  $(N + 1)^2 + \sum_{\substack{i,k \in J' \\ i \neq j}} |(T \setminus T_i) \cup (T_i \setminus T_k)| + N$  variáveis binárias e um total de  $(N + 1)^2 + \sum_{\substack{i,k \in J' \\ i \neq j}} |(T \setminus T_i) \cup (T_i \setminus T_k)| + 2N + 1$  variáveis.

Na prática, o número de padrões de corte é muito maior que o de tipos de itens distintos. Em nossos testes o valor máximo de  $M$  é 20, enquanto o máximo de  $N$  foi 2174. Desta forma, quando o número de variáveis depende de uma potência de  $N$ , o tamanho do modelo pode ser tornar muito grande para ser tratado na prática. Para ilustrar, na [Tabela 6](#) apresentamos a média da quantidade de variáveis das 5 primeiras instâncias das classes 3, 5 e 6 apresentadas na [Seção 6.1](#) para os modelos [YLI](#), [LGSRV](#) e [CI2](#).

Classe	Nº de variáveis <a href="#">YLI</a>	Nº de variáveis <a href="#">LGSRV</a>	variáveis <a href="#">CI2</a>
3	$266,46 \times 10^2$	$269,55 \times 10^2$	$245,08 \times 10^3$
5	$147,82 \times 10$	$155,48 \times 10$	$967,56 \times 10$
6	$134,04 \times 10^2$	$136,33 \times 10^2$	$198,83 \times 10^3$

Tabela 6 – Comparação entre o número de variáveis presentes nos modelos [YLI](#), [LGSRV](#) e [CI2](#). A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 3, 5 e 6 propostas nesta dissertação.

Com base na [Tabela 6](#), concluímos que, para as classes testadas, o modelo [YLI](#) é o que apresenta o menor número de variáveis. O modelo [LGSRV](#) possui uma quantidade de variáveis semelhante à formulação [YLI](#) e é o segundo modelo mais compacto em termos de quantidade de variáveis. Já a formulação [CI2](#) possui um número significativamente maior de variáveis quando comparado aos outros dois modelos.

### 6.3 Quantidade de restrições

A qualidade de um modelo está relacionada não apenas com a sua quantidade de variáveis, mas também, com o número de restrições presentes no modelo. De forma geral é interessante que o modelo seja o mais compacto possível. O foco desta [Seção](#) está em comparar a quantidade de restrições presentes nas formulações [YLI](#), [LGSRV](#) e [CI2](#). Não foram consideradas as restrições referentes ao domínio das variáveis. Considere as restrições presentes no modelo [YLI](#) dadas por:

$$\sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i = 1, \dots, M \quad (4.7)$$

$$\sum_{i=2}^N \sum_{t \in T} z_{it} \leq M - C, \quad (4.8)$$

$$v_j \leq p_j, \quad j = 1, \dots, N \quad (4.9)$$

$$Kv_j \geq p_j, \quad j = 1, \dots, N \quad (4.10)$$

$$\sum_{i=1}^N w_{ij} = 1, \quad j = 1, \dots, N \quad (4.11)$$

$$\sum_{j=1}^N w_{ij} = 1, \quad i = 1, \dots, N \quad (4.12)$$

$$w_{ij} + (v_j - 1) \leq y_{jt}, \quad i \in J_t, j = 1, \dots, N, t \in T \quad (4.13)$$

$$\sum_{t \in T} y_{it} \leq C, \quad i = 1, \dots, N \quad (4.14)$$

$$y_{it} - y_{(i-1)t} \leq z_{it}, \quad i = 2, \dots, N, t \in T \quad (4.15)$$

$$w_{ij} \in \{0, 1\}, \quad i = 1, \dots, N, j = 1, \dots, N \quad (4.16)$$

$$y_{it} \in \{0, 1\}, \quad i = 1, \dots, N, t \in T \quad (4.17)$$

$$z_{it} \in \{0, 1\}, \quad i = 1, \dots, N, t \in T. \quad (4.18)$$

O conjunto de restrições (4.7) possui um total de  $M$  restrições, já cada um dos conjuntos (4.9), (4.10), (4.11), (4.12) e (4.14) possuem  $N$  restrições. As restrições (4.16), (4.17) e (4.18) são referentes ao domínio das variáveis e, portanto, não foram contabilizadas. Uma vez que  $|T| = M$ , o conjunto de restrições (4.15) tem cardinalidade igual a  $(N - 1)M$ . O conjunto de restrições (4.13) depende da cardinalidade dos conjuntos  $J_t \forall t \in T$ , portanto o número de restrições presentes no conjunto (4.13) depende do problema específico que está sendo tratado. A cardinalidade deste conjunto é dada por  $(\sum_{t \in T} |J_t|)N$ . Por último, deve-se levar em conta a restrição (4.8). Somando a cardinalidade de todos os conjuntos de restrições há um total de

$$M + 5N + (N - 1)M + (\sum_{t \in T} |J_t|)N + 1 =$$

$$N(5 + M + \sum_{t \in T} |J_t|) + 1$$

restrições.

O mesmo procedimento pode ser feito para encontrar o número de restrições presentes em **LGSRV**. Considere as restrições presentes no modelo **LGSRV** dadas por:

$$\sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i \in T \quad (5.5)$$

$$\sum_{k \in K \setminus \{1\}} \sum_{t \in T} z_{kt} \leq M - C, \quad (5.6)$$

$$v_j \leq p_j, \quad j \in J \quad (5.7)$$

$$Kv_j \geq p_j, \quad j \in J \quad (5.8)$$

$$v_1 = 1, \quad (5.9)$$

$$\sum_{j \in J' \setminus \{i\}} x_{ij} = v_i, \quad i \in J' \quad (5.10)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = v_j, \quad j \in J' \quad (5.11)$$

$$y_{it} \geq v_i, \quad i \in J, t \in T_i \quad (5.12)$$

$$\sum_{t \in T} y_{it} \leq C, \quad i \in J \quad (5.13)$$

$$x_{ij} + y_{jt} - y_{it} \leq z_{it} + 1, \quad i \in J', j \in J, t \in T \quad (5.14)$$

$$1 \leq u_i \leq N, \quad i \in J \quad (5.15)$$

$$u_i - u_j + Nx_{ij} \leq N - 1, \quad i, j \in J \quad (5.16)$$

$$z_{it} = 0, \quad i \in J, t \in T \setminus T_i \quad (5.17)$$

$$v_i \in \{0, 1\}, \quad i \in J' \quad (5.18)$$

$$x_{ij} \in \{0, 1\}, \quad i \in J', j \in J' \quad (5.19)$$

$$y_{it} \in \{0, 1\}, \quad i \in J, t \in T \quad (5.20)$$

$$z_{it} \in \{0, 1\}, \quad i \in J, t \in T \quad (5.21)$$

$$p_i \in Z^+, \quad i \in J' \quad (5.22)$$

$$u_i \in Z^+, \quad i \in J. \quad (5.23)$$

Uma vez que  $|T| = M$ , o conjunto de restrições (5.5) possui  $M$  restrições. Os conjuntos (5.7), (5.8), (5.13) e (5.15) possuem  $N$  restrições. Os conjuntos (5.10) e (5.11) possuem  $N + 1$  restrições. A quantidade de restrições presentes no conjunto (5.12) depende da cardinalidade de  $T_i \forall i \in J$ , sendo assim, o número de restrições contidas em (5.12) não depende apenas dos valores de  $M$  e  $N$ , mas também do problema específico que está sendo resolvido. Há um total de  $\sum_{i \in J} |T_i|$  restrições em (5.12) e o conjunto de restrições (5.17) possui  $\sum_{i \in J} |T \setminus T_i|$  restrições. A cardinalidade do conjunto de restrições (5.16) é igual a  $N^2$  e há um total de  $(N + 1)NM$  restrições em (5.14). As restrições (5.18)-(5.23) são

referentes ao domínio das variáveis  $e$ , portanto, não são contabilizadas. Por último, deve-se levar em conta as restrições (5.6) e (5.9). Somando a cardinalidade de todos os conjuntos de restrições há um total de

$$M + 4N + 2(N + 1) + \sum_{i \in J} |T_i| + \sum_{i \in J} |T \setminus T_i| + N^2 + (N + 1)NM + 2 =$$

$$N(6 + M + N + NM) + \sum_{i \in J} |T_i| + \sum_{i \in J} |T \setminus T_i| + M + 4$$

restrições

De forma análoga ao que foi feito para os modelos YLI e LGSRV, a quantidade de restrições presentes no modelo CI2 pode ser encontrada a partir da contagem de restrições presentes nos conjuntos a seguir:

$$\sum_{j \in J'} \alpha_i^{(j)} p_j \geq d_i, \quad i \in T \quad (5.35)$$

$$\sum_{\substack{i, j \in J \\ i \neq j}} \sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \leq M - C, \quad (5.36)$$

$$\sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \geq 0, \quad i, j \in J, i \neq j \quad (5.37)$$

$$v_j \leq p_j, \quad j \in J \quad (5.38)$$

$$K v_j \geq p_j, \quad j \in J \quad (5.39)$$

$$v_1 = 1, \quad (5.40)$$

$$\sum_{j \in J' \setminus \{i\}} x_{ij} = v_i, \quad i \in J' \quad (5.41)$$

$$\sum_{i \in J' \setminus \{j\}} x_{ij} = v_j, \quad j \in J' \quad (5.42)$$

$$1 \leq u_i \leq \sum_{j \in J} v_j, \quad i \in J \quad (5.43)$$

$$u_i - u_j + N x_{ij} \leq N - 1, \quad i, j \in J, i \neq j \quad (5.44)$$

$$\sum_{k \in J' \setminus \{i\}} y_{ki}^t - \sum_{j \in J' \setminus \{i\}} y_{ij}^t \geq 0, \quad i \in J', t \in T \setminus T_i \quad (5.45)$$

$$\sum_{\substack{t \in T: \\ t \in T_i \\ t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T: \\ t \notin T_i \\ t \notin T_j}} y_{ij}^t \leq (C - |T_j|)x_{ij}, \quad i, j \in J', i \neq j \quad (5.46)$$

$$y_{ij}^t \leq x_{ij}, \quad i, j \in J', i \neq j, t \in T \setminus T_j \quad (5.47)$$

$$x_{ij} \in \{0, 1\}, \quad i, j \in J', i \neq j \quad (5.48)$$

$$y_{ij}^t \in \{0, 1\}, \quad i, j \in J', i \neq j, \quad t \in (T \setminus T_i) \cup (T_i \setminus T_j) \quad (5.49)$$

$$v_j \in \{0, 1\}, \quad j \in J', \quad (5.50)$$

$$p_i \in Z^+, \quad i \in J' \quad (5.51)$$

$$u_i \in Z^+, \quad i \in J. \quad (5.52)$$

O conjunto de restrições (5.35) possui  $M$  restrições enquanto os conjuntos de restrições (5.38), (5.39) e (5.43) possuem  $N$  restrições cada um. Há um total de  $N + 1$  restrições em cada um dos conjuntos (5.41) e (5.42). Os conjuntos (5.37) e (5.44) possuem  $N(N - 1)$  cada, enquanto o conjunto de restrições (5.45) possui  $\sum_{i \in J'} |T \setminus T_i|$  restrições. O conjunto de restrições (5.46) possui  $(N + 1)^2 - (N + 1)$  restrições. A cardinalidade do conjunto (5.47) é igual a  $\sum_{\substack{i, j \in J' \\ i \neq j}} |T \setminus T_j|$ . Os conjuntos de restrições (5.48)-(5.52) são referentes ao domínio das variáveis e, portanto, não são considerados na contagem das restrições. Por último, devem ser consideradas as restrições (5.36) e (5.40), somando um total de:

$$M + 3N + 2(N + 1) + 2N(N - 1) + \sum_{i \in J'} |T \setminus T_i| + (N + 1)^2 - (N + 1) + \sum_{\substack{i, j \in J' \\ i \neq j}} |T \setminus T_j| + 2 =$$

$$M + N(4 + 3N) + \sum_{i \in J'} |T \setminus T_i| + \sum_{\substack{i, j \in J' \\ i \neq j}} |T \setminus T_j| + 4$$

restrições.

O número exato de restrições presentes nas formulações não depende apenas de  $N$  e  $M$ , mas também do problema específico que está sendo tratado. Para compararmos a quantidade de restrições presentes em cada modelo são utilizadas as 5 primeiras instâncias das Classes 3, 5 e 6 apresentadas na Seção 6.1. Note que esta comparação permite entender o comportamento das formulações, porém, não é possível garantir que os resultados aqui encontrados se mantêm para outros conjuntos de dados. A média da quantidade de restrições destas instâncias para os modelos YLI, LGSRV e CI2 é apresentada na Tabela 7.

Classe	Nº de restrições <b>YLI</b>	Nº de restrições <b>LGSRV</b>	Nº de restrições <b>CI2</b>
3	$601,91 \times 10^2$	$260,57 \times 10^3$	$230,89 \times 10^3$
4	$187,32 \times 10$	$999,24 \times 10$	$105,68 \times 10^2$
6	$184,82 \times 10^2$	$201,22 \times 10^3$	$204,59 \times 10^3$

Tabela 7 – Comparação entre o número de restrições presentes nos modelos **YLI**, **LGSRV** e **CI2**. A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 3, 4 e 6 propostas nesta dissertação.

A partir dos resultados apresentados na [Tabela 7](#) concluímos que o modelo **YLI** é o que contém o menor número de restrições para as instâncias e classes testadas. Os modelos **CI2** e **LGSRV** apresentam resultados similares para as classes testadas.

## 6.4 Análise da qualidade da relaxação linear

Para comparar a qualidade da relaxação linear dos modelos são consideradas as Classes 3, 4, 5 e 6 apresentadas na [Seção 6.1](#). A quantidade de padrões de corte de uma determinada classe depende do comprimento dos itens. Para itens de tamanho menor existe um número excessivamente grande de padrões de corte possíveis, dificultando a resolução do modelo de forma exata. Desta forma, não foi possível encontrar uma solução ótima, dentro do limite de tempo de 7200 segundos, para o problema relaxado para nenhuma das instâncias pertencentes à Classe 4 e, portanto, esta classe foi omitida da [Tabela 8](#).

Para as Classes 3, 5 e 6 propostas nesta dissertação, foram consideradas dez instâncias de cada classe para realizar os testes computacionais. Considerando 7200 segundos como o limite máximo de tempo para encontrar uma solução ótima de uma instância qualquer. Na [Tabela 8](#) apresentamos uma comparação da média do valor da função objetivo do modelo PLI com a média do valor da função objetivo da relaxação linear do modelo. Para fazer os testes são utilizados os modelos **YLI**, **LGSRV** e **CI2**. Para todos os testes, o número máximo permitido de pilhas abertas durante o sequenciamento foi fixado em 1.

Classe	Valor da função objetivo do modelo com variáveis inteiras (PLI)	Valor da função objetivo da relaxação linear parcial	Diferença
Yanasse Lamosa Integrado <b>YLI</b>			
1	976,00	856,62	13,92%
3	1999,60	1701,83	17,50%
4	3844,20	3215,50	19,56%
Laporte Gomes e Semet com remoção de vértices <b>LGSRV</b>			
1	976,00	972,76	0,33%
3	1999,60	1998,35	0,06%
4	3844,20	3841,92	0,06%
Catanzaro et al. Integrado 2 <b>CI2</b>			
1	976,00	972,76	0,33%
3	1999,60	1998,35	0,06%
4	3844,20	3841,92	0,06%

Tabela 8 – Qualidade da relaxação linear dos modelos **YLI**, **LGSRV** e **CI2**. Os dados contidos na tabela são referente às 10 instâncias das Classes 3, 5 e 6 apresentadas na Seção 6.1.

Como pode ser visto na Tabela 8, para os modelos **LGSRV** e **CI2** a diferença entre o valor da função objetivo para o problema original e o relaxado foi inferior a 0,5% para as três classes testadas. Já para o modelo **YLI** a diferença entre o valor da função objetivo, para o problema original e o relaxado, foi, em média, de aproximadamente 17%.

## 6.5 Resultados para as instâncias do CS-LOSP unidimensional

A quantidade de padrões de corte de uma determinada classe depende do comprimento dos itens e da quantidade de itens de tipos distintos. Para classes, tais que o tamanho dos itens é pequeno ou que há uma grande quantidade de itens de tipos distintos, o número de padrões de corte factíveis pode ser excessivamente grande, dificultando a resolução do modelo de forma exata.

Desta forma, nesta dissertação, não foi possível obter uma solução ótima para todas as classes de problemas quando consideramos todos os padrões de cortes possíveis, mesmo quando consideramos a relaxação linear parcial dos problemas. Felizmente, quando trabalhamos com o problema de corte de estoque com restrição no número máximo de pilhas abertas, uma parte dos padrões de corte pode ser eliminada do problema sem afetar a solução ótima.

Toda vez que um padrão de corte é executado, devem ser abertas as pilhas referentes aos itens pertencentes a este. Sendo assim, se a quantidade de itens de tipos

distintos, contidos em um padrão de corte, supera o limite de pilhas que podem estar abertas simultaneamente, este padrão de corte não pode ser utilizado, podendo então, ser removido do problema sem alterar a solução ótima do CS-LOSP.

O número de variáveis e restrições do problema está intimamente ligado à quantidade de padrões de corte considerados. Desta forma, adotar a estratégia de remover os padrões de corte ineficazes reduz consideravelmente a complexidade computacional do problema e, portanto, este procedimento é usado nos testes computacionais.

Na Seção 6.4 esta estratégia foi usada para obter o valor da função ótima do problema inteiro, já na relaxação linear foram considerados todos os padrões de corte possíveis. Note que um padrão de corte ineficaz para o problema inteiro não é necessariamente ineficaz para o problema relaxado. Outros padrões de corte podem ser eliminados do problema sem alterar o valor da função objetivo em uma solução ótima do mesmo. Considere a propriedade a seguir:

**Propriedade 1:** Dado dois padrões de corte:  $\alpha^{(i)} = (\alpha_1^{(i)}, \alpha_2^{(i)}, \dots, \alpha_M^{(i)})^t$  e  $\alpha^{(j)} = (\alpha_1^{(j)}, \alpha_2^{(j)}, \dots, \alpha_M^{(j)})^t$  tais que:

$$\alpha_k^{(i)} = 0 \iff \alpha_k^{(j)} = 0, \quad k = 1, \dots, M \quad (I)$$

$$\alpha_k^{(i)} \leq \alpha_k^{(j)}, \quad \forall \alpha_k^{(i)} > 0, \quad k = 1, \dots, M. \quad (II)$$

Então, considerando um CS-LOSP, no qual é permitido que a produção de itens supere a demanda preestabelecida e o objetivo do problema é de minimizar o número de objetos utilizados, o padrão de corte  $\alpha^{(i)}$  pode ser removido do problema sem alterar a solução ótima do CS-LOSP.

*Demonstração.* Considere uma solução ótima para o CS-LOSP na qual o padrão de corte  $\alpha^{(i)}$  é utilizado. A hipótese (II) garante que a quantidade de itens, obtida a partir da execução do padrão de corte  $j$ , é igual ou maior que a quantidade de itens obtida a partir da execução do padrão de corte  $i$ , para qualquer tipo de item. Portanto, ao substituir o padrão de corte  $i$  pelo padrão de corte  $j$ , a quantidade de itens produzidos se mantém ou aumenta, logo as restrições referentes à demanda são satisfeitas e a quantidade de objetos utilizados se mantém a mesma. Além disso, a hipótese (I) garante que o número de pilhas abertas é exatamente o mesmo, uma vez que ambos os padrões de corte produzem os mesmos tipos de itens quando executados. Portanto, a execução do padrão de corte  $i$  pode ser substituída pela execução do padrão de corte  $j$  sem alterar o valor da função objetivo. Desta forma, a remoção do padrão de corte  $i$  não altera em nada o valor da função objetivo em uma solução ótima. Neste caso, o padrão de corte  $i$  é dito dominado pelo padrão de corte  $j$ .  $\square$



A Propriedade I garante que todos os padrões de corte dominados por algum outro padrão de corte podem ser removidos sem alterar a solução ótima do problema. Assim, o número de padrões de corte factíveis é reduzido significativamente. Considere  $P_1$  o conjunto contendo todos os padrões de corte factíveis e  $P_2$  o conjunto contendo apenas os padrões de corte dominantes. Com o intuito de ilustrar a efetividade das estratégias abordadas anteriormente, o gráfico da Figura 11 apresenta a comparação entre a quantidade de padrões de corte factíveis em função da limitação na abertura de pilhas para  $P_1$  e  $P_2$ . Os dados utilizados são oriundos da primeira instância da Classe 3.

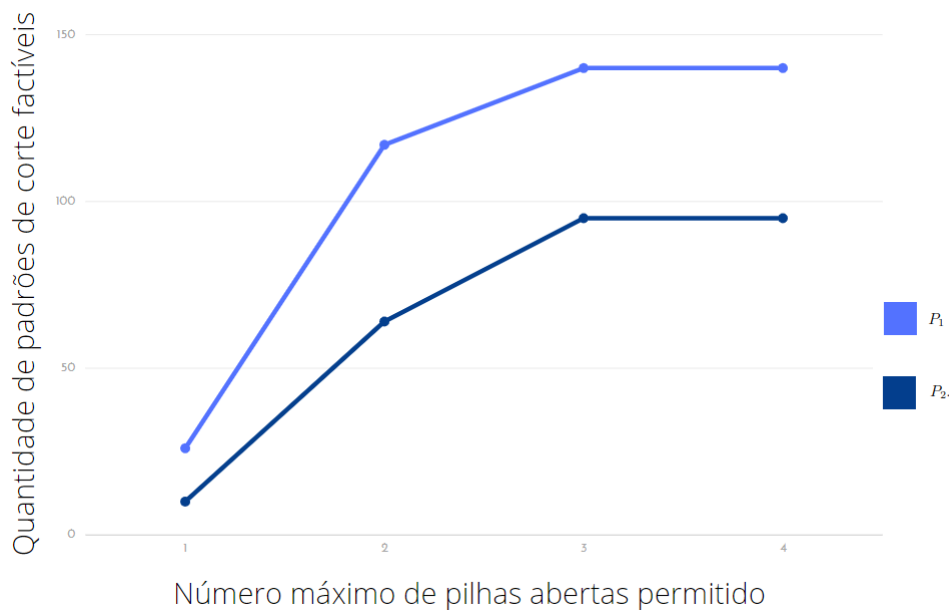


Figura 11 – Comparação entre a quantidade de padrões de corte factíveis total e a quantidade de padrões de corte dominantes em função da limitação no número máximo de pilhas abertas considerando os dados da primeira instância da Classe 3.

A limitação no número máximo de pilhas abertas durante o sequenciamento dos padrões de corte é um importante parâmetro na resolução de um CS-LOSP, impactando na solução ótima, nos padrões de corte utilizados e no tempo gasto para se resolver o problema, sendo assim, é interessante testar valores distintos para este parâmetro. Considere  $C$  como o número máximo de pilhas que podem ser abertas durante o sequenciamento dos padrões de corte. Nesta Seção, o conjunto de dados é testado utilizando três valores distintos de  $C$ , sendo estes  $C = 2$ ,  $C = 3$  e  $C = 4$ . Para cada classe há três possibilidades:

- É possível encontrar, dentro da limitação de tempo de 7200 segundos, uma solução ótima para cada uma das 10 instâncias pertencentes à classe analisada. Neste caso, as Tabelas 9, 12 e 15 indicam a média de tempo gasto para resolver 10 instâncias das 6 classes propostas nesta dissertação utilizando os modelos YLI, LGSRV e CI2,

além do valor médio da função objetivo avaliada em uma solução ótima quando consideramos  $C = 2$ ,  $C = 3$  e  $C = 4$ , respectivamente.

- Só é possível encontrar uma solução ótima dentro do limite de tempo para algumas das 10 instâncias pertencentes à classe. Neste caso, o caractere “\*” é utilizado para simbolizar que a classe foi parcialmente resolvida. As Tabelas 10, 13 e 16 contêm os resultados para as classes parcialmente resolvidas em detalhes quando consideramos  $C = 2$ ,  $C = 3$  e  $C = 4$ , respectivamente.
- Não foi possível resolver nenhuma das 10 instâncias. Neste caso, o caractere “–” é utilizado para simbolizar a impossibilidade de resolver qualquer uma das instâncias dentro do tempo limite.

Classe	YLI Tempo(s)	LGSRV Tempo(s)	CI2 Tempo(s)	Valor da Função Objetivo
1	704,70	5,87	2,48	191,50
2	*	*	1002,95	317,90
3	*	*	*	*
4	–	–	*	*
5	2,85	93,61	1,51	1702,10
6	*	*	307,37	3215,50

Tabela 9 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 6 classes propostas, nesta dissertação, para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 2. Nesta tabela, o caractere “–” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Com base nos resultados apresentados na Tabela 9 é possível afirmar que o modelo CI2 se sobressaiu aos demais. Para esta formulação, considerando o limite no número máximo de pilhas que podem ser abertas igual a 2, foi possível resolver quatro classes de forma completa e duas de forma parcial, totalizando 48 instâncias. Enquanto para os modelos LGSRV e YLI foi possível resolver duas classes de forma completa e três de forma parcial, totalizando 38 e 31 instâncias, respectivamente.

Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado (YLI)			
2	(1174,84; 5620,86)	(7; 9)	(416; 314)
3	(205,42; 419,93; 27,72; 246,36; 87,94; 18,61)	(1; 4; 7; 8; 9; 10)	(824; 978; 1033; 994; 989; 872)
6	(40,91; 19,20; 4,36)	(8; 9; 10)	(3579; 3280; 3470)
Laporte Gomes e Semet com remoção de vértices (LGSRV)			
2	(1411,73; 898,23; 2781,38; 4443,03; 1138,73; 4862,46; 208,16; 175,22; 3139,46)	(1; 2; 3 4; 5; 6; 7; 9; 10)	(275; 311; 231; 327; 339; 311 416; 314; 347)
3	(414,49; 1117,12; 4484,94; 98,16; 361,56; 17,81; 39,96)	(1; 4; 5; 7; 8; 9; 10)	(824; 978; 549; 1033; 994; 989; 872)
6	(220,73; 307,78)	( 9; 10)	(3280; 3470)
Catanzaro et al. Integrado 2 (CI2)			
3	(6,46; 16,64; 109,64; 5,77 63,89; 6,27; 15,43)	(1; 4; 5; 7; 8; 9; 10)	(824; 978; 549; 1033; 994; 989; 872)
4	4360.46	3	1807

Tabela 10 – Resultados em detalhes para as classes propostas para o CS-LOSP unidimensional parcialmente resolvidas utilizando os modelos YLI e LGSRV. A primeira coluna indica a classe, a segunda indica o tempo em segundos para se obter uma solução ótima, a terceira quais instâncias foram resolvidas dentro do tempo limite (7200s) e, a última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 2$ .

É possível observar, com base nos resultados apresentados na Tabela 9, que o comprimento dos itens é um fator de grande impacto no tempo computacional. Para as classes com itens de tamanho menor o tempo exigido para se determinar uma solução ótima foi maior, de forma que o número de instâncias resolvidas para estas classes foi menor. Este resultado é esperado uma vez que quanto menor o tamanho dos itens maior o número de padrões de corte possíveis. Outro parâmetro que influencia o tempo gasto para se determinar uma solução ótima é a quantidade de itens de tipos distintos. Classes contendo um maior número de itens exigem um tempo computacional maior. Na Tabela 11 é ilustrada a relação entre a quantidade de itens de tipos distintos e o tamanho dos itens, com o tempo gasto para se determinar uma solução ótima quando consideramos  $C = 2$ . Para esta tabela, são consideradas as instâncias pertencentes às Classes 3, 4, 5 e 6.

	YLI		LGSRV		CI2	
	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)
$l_i = [200, 500]$	6	166,63	7	933,43	8	573,07
$l_i = [300, 800]$	13	7,15	12	122,05	20	154,44
$M = 10$	16	64,66	17	439,42	17	14,07
$M = 20$	3	21,40	2	264,26	11	675,83

Tabela 11 – Comparação entre o tempo gasto e o número de instâncias resolvidas, em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das Classes 3, 4, 5 e 6, considerando  $C = 2$ .

A Tabela 12 é similar à Tabela 9, sendo a única diferença entre as duas o valor de  $C$ . Enquanto, na Tabela 9, os testes computacionais foram realizados considerando  $C = 2$ , na Tabela 12 o valor de  $C$  foi fixado como 3.

Classe	YLI Tempo(s)	LGSRV Tempo(s)	CI2 Tempo(s)	Valor da Função Objetivo
1	*	*	*	*
2	—	—	—	—
3	*	*	*	*
4	—	—	—	—
5	4,67	3,49	655,73	1702,10
6	*	*	—	*

Tabela 12 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 6 classes, propostas nesta dissertação, para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 3. Nesta tabela, o caractere “—” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Como pode ser visto na Figura 11, à medida que o valor de  $C$  aumenta, o número de padrões de corte factíveis cresce rapidamente, dificultando a resolução do modelo para o problema integrado. Por outro lado, ao aumentar o valor de  $C$ , a restrição imposta sobre o número máximo de pilhas abertas é enfraquecida e, conseqüentemente, uma quantidade maior de sequenciamentos se torna factível. Os modelos respondem diferentemente a este fator, enquanto para os modelos CI2 e LGSRV foi possível resolver mais instâncias para  $C = 2$ , para a formulação YLI mais instâncias foram resolvidas quando considerado  $C = 3$ .

Considerando  $C = 3$ , o modelo YLI foi o que apresentou os melhores resultados. Para este modelo foi possível resolver uma classe de forma completa e outras três classes de forma parcial, totalizando 32 instâncias.

Para as formulações **LGSRV** e **CI2** foi possível resolver uma instância de forma completa e outras duas de forma parcial. Utilizando os modelos **LGSRV** e **CI2** foi possível resolver 23 e 21 instâncias, respectivamente.

Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado ( <b>YLI</b> )			
1	(924,47; 942,81; 632,87; 34,61; 807,84; 360,71; 170,70; 2501,24)	(1; 2; 3; 4; 5; 6; 8; 10)	(181; 150; 182; 207; 206; 235; 160; 187)
3	(21,53; 14,18; 4142,17; 276,24; 112,09)	(4; 7; 8; 9; 10)	(965; 1009; 961; 962; 844)
6	(91,89; 3583,31; 200,58; 152,58; 3263,33; 231,53; 82,92; 11,51; 2,68)	(1; 3; 4; 5; 6; 7; 8; 9; 10)	(3151; 2778; 3043; 3429; 3350; 3356; 3579; 3280; 3470)
Laporte Gomes e Semet com remoção de vértices ( <b>LGSRV</b> )			
1	(189,75; 8,28; 18,99; 43,07; 33,14; 8,43; 213,34; 15,26)	(2; 3; 4; 5; 6; 8; 9; 10)	(150; 182; 207; 206; 235; 160; 193; 187)
3	(521,51; 58,87 ;1084,13; 812,09)	(4; 7; 9; 10)	(965; 1009; 962; 844)
6	1984,78	8	3579
Catanzaro et al. Integrado 2 ( <b>CI2</b> )			
1	(1184.19; 176.15; 9.79; 35.51; 83.18; 26.25; 12.09; 32.53; 127.70)	(1; 2; 3; 4; 5; 6; 8; 9; 10)	(181; 150; 182; 207; 206; 235; 160; 193; 187)
3	(2513,78; 839,22)	(4; 7)	(965; 1009)

Tabela 13 – Resultados em detalhes para as classes propostas para o CS-LOSP unidimensional parcialmente resolvidas utilizando os modelos **YLI**, **LGSRV** e **CI2**. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 3$ .

Os resultados obtidos considerando  $C = 3$  são semelhantes aos da [Tabela 11](#). Novamente, as classes com maior quantidade de tipos distintos de itens foram as que exigiram mais tempo para se determinar uma solução ótima. O aumento do valor de  $C$  amplificou a influência que o comprimento dos itens tem no tempo requerido para se encontrar uma solução ótima. Para  $C = 3$ , nota-se que o tamanho dos itens repercute grandemente no tempo gasto para se encontrar uma solução ótima. Para  $C = 3$ , para todos os modelos, a quantidade de instâncias resolvidas foi proporcional ao tamanho dos itens, conforme o esperado. Na [Tabela 14](#) apresentamos a relação entre quantidade de itens de tipos distintos e tamanho dos itens com o tempo gasto para se achar uma solução ótima quando consideramos as Classes 3, 4, 5 e 6 e  $C = 3$ .

	YLI		LGSRV		CI2	
	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)
$l_i = [200, 500]$	5	913,24	4	619,15	2	1676,50
$l_i = [300, 800]$	19	403,54	11	183,61	10	655,73
$M = 10$	15	307,54	14	179,39	12	825,86
$M = 20$	9	846,70	1	1984,78	–	–

Tabela 14 – Comparação entre o tempo gasto e o número de instâncias resolvidas em função da demanda requerida, do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das Classes 3, 4, 5 e 6 utilizando  $C = 3$ .

Por último, na Tabela 15 apresentamos os resultados referentes aos testes computacionais para o CS-LOSP unidimensional quando utilizamos  $C = 4$ .

Classe	YLI Tempo(s)	LGSRV Tempo(s)	CI2 Tempo(s)	Valor da Função Objetivo
1	683,98	38,12	877,38	190,20
2	–	–	–	–
3	*	*	*	*
4	–	–	–	–
5	0,65	4,14	*	1702,10
6	*	*	–	*

Tabela 15 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 6 classes, propostas nesta dissertação, para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 4. Nesta tabela, o caractere “–” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Para os modelos YLI e LGSRV foi possível resolver duas classes de forma completa e outras duas de forma parcial, totalizando 35 e 28 instâncias, respectivamente. Enquanto para a formulação CI2 foi possível resolver uma classe de forma completa e 2 de forma parcial, totalizando 19 instâncias.

Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado (YLI)			
3	(5435,38; 8,86; 9,76; 1522,82; 48,21; 81,83)	(1; 4; 7; 8; 9; 10)	(786; 965; 1009; 961; 962; 844)
6	(73,69; 66,81; 46,73; 47,83; 643,77; 260,56; 29,91; 12,42; 4,83)	(1; 3; 4; 5; 6; 7; 8; 9; 10)	(3151; 2778; 3043; 3429; 3350; 3356; 3579; 3280; 3470)
Laporte Gomes e Semet com remoção de vértices (LGSRV)			
3	(390,01; 5153,71; 169,43; 276,64; 5371,82; 447,09)	(4; 6; 7; 8; 9; 10)	(965; 878; 1009; 961; 962; 844)
6	(401,71; 2363,07)	(9; 10)	(3280; 3470)
Catanzaro et al. Integrado 2 (CI2)			
3	2221,14	4	965
5	(12,94; 6,67; 30,23; 1,22; 64,25; 9,25; 8,99; 29,60)	(3; 4; 5; 6; 7; 8; 9; 10)	(2349; 1699; 1689 2525; 1514; 1752; 1640; 1404)

Tabela 16 – Resultados em detalhes para as classes propostas para o CS-LOSP unidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 4$ .

A relação entre demanda, comprimento dos itens, quantidade de tipos distintos de itens e a quantidade de instâncias resolvidas é semelhante à relação apresentada anteriormente na Tabela 14.

	YLI		LGSRV		CI2	
	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)
$l_i = [200, 500]$	6	1184,48	6	1968,12	1	2221,14
$l_i = [300, 800]$	19	62,79	12	283,85	8	20,39
$M = 10$	16	444,58	16	740,63	9	264,91
$M = 20$	9	131,84	2	1382,39	–	–

Tabela 17 – Comparação entre o tempo gasto e número de instâncias resolvidas em função da demanda requerida, do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das Classes 3, 4, 5 e 6 utilizando  $C = 4$ .

De forma geral, o valor de  $C$  limita a quantidade de padrões de corte factíveis. Assim, quão menor for o valor da limitação no número máximo de pilhas que podem ser abertas, mais fácil será resolver o problema. Porém, caso o comprimento dos itens seja elevado, a quantidade de itens gerados por um padrão de corte qualquer já é naturalmente baixa. Neste caso, mesmo valores relativamente baixos de  $C$  não reduzem significativamente a cardinalidade do conjunto de padrões de corte factíveis do problema. Afim de ilustrar o

impacto do parâmetro  $C$  na quantidade de padrões de corte em função do comprimento dos itens a [Figura 12](#) utiliza a primeira instância das Classes 1, 3 e 5 para exemplificar a relação entre a porcentagem de padrões de corte factíveis e o valor de  $C$ .

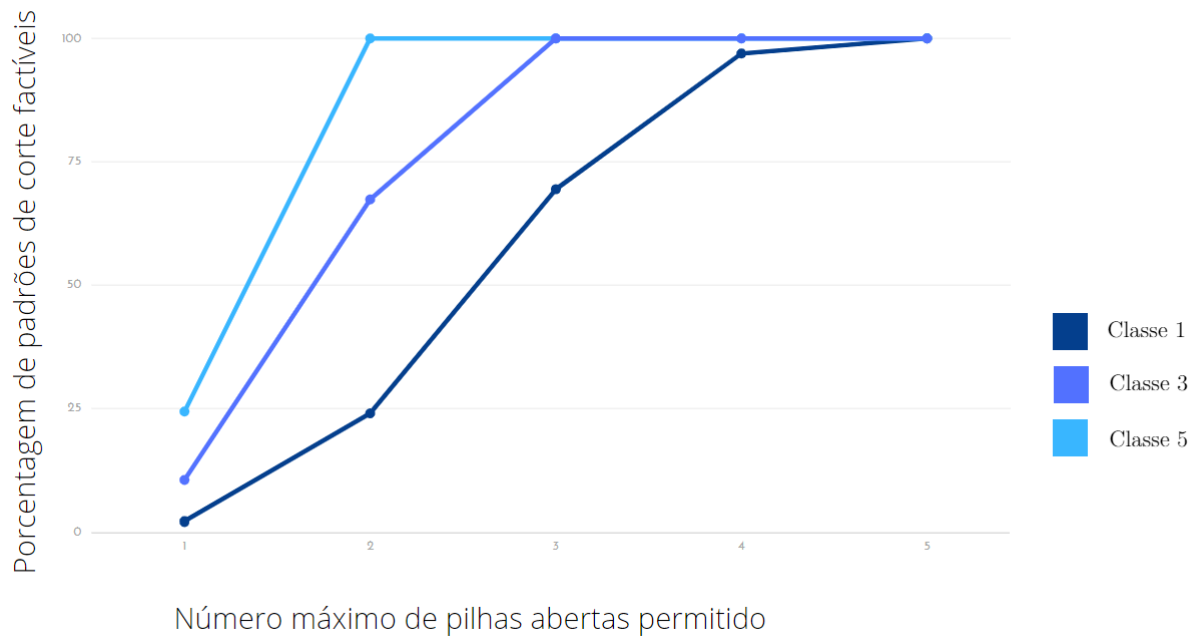


Figura 12 – Quantidade de padrões de corte factíveis em função da limitação no número máximo de pilhas abertas considerando os dados da primeira instância das Classes 1, 3 e 5.

Com base na [Figura 12](#) é possível perceber que, para a primeira instância das Classes 3 e 5, não há diferença na porcentagem de padrões de corte factíveis quando o número máximo de pilhas que podem ser abertas é superior a 3. Sendo assim, a redução no número de padrões de corte quando aumentamos o valor de  $C = 3$  para  $C = 4$  é nulo.

Os modelos reagem de forma distinta ao aumento do valor de  $C$  de três para quatro. Enquanto para as formulações [YLI](#) e [LGSRV](#) o número de instâncias resolvidas considerando  $C = 4$  é maior que o número de instâncias resolvidas considerando  $C = 3$ . Para a formulação [CI2](#), o número de instâncias resolvidas considerando  $C = 3$  é maior que o número de instâncias resolvidas considerando  $C = 4$ .

A quantidade de variáveis e restrições presentes nos três modelos estudados depende da quantidade de padrões de corte e do número de tipos distintos de itens. Para as instâncias maiores não foi possível sequer iniciar a execução do *solver* de otimização, uma vez que o tamanho dos modelos é demasiadamente grande. Para estas instâncias, os testes foram interrompidos por sobrecarga, culminando na reinicialização do sistema.

Com o intuito de evitar este problema, nos testes computacionais para o CS-LOSP unidimensional realizados neste capítulo, foi adotado um limite de tempo de 200 segundos, de forma que, nos casos nos quais o tempo demandado para executar o *software*



de otimização e iniciar o processo de busca por uma solução ótima excedeu este limite de tempo, os testes computacionais foram interrompidos. Uma vez iniciada a execução do *software* de otimização foi estabelecido um limitante de tempo de 7200 segundos para determinar uma solução ótima.

Para  $C = 2$ , todas as 6 classes propostas no trabalho foram executadas sem sobrecarregar a máquina utilizada. Para  $C = 3$  os três modelos apresentaram dificuldades para executar as instâncias pertencentes à Classe 4. Para esta classe, só foi possível executar as instâncias com menor número de padrões de corte, as demais excederam a capacidade da máquina resultando na interrupção dos testes. Para  $C = 4$ , não foi possível executar *software* de otimização, respeitando o limite de tempo de 200 segundos, para a maioria das instâncias das Classes 2 e 4. O modelo YLI é o mais compacto dentre os três modelos testados nesta seção, notamos que para esta formulação foi possível executar *software* de otimização para um maior número de instâncias para as Classes 2 e 4, embora, não tenha sido possível determinar uma solução ótima, respeitando a limitação de tempo de 7200 segundos, para nenhuma instâncias destas classes quando considerado  $C = 3$  ou  $C = 4$ .

Nas tabelas apresentadas, não é possível identificar as instâncias, que não foram resolvidas por não determinarem uma solução ótima dentro do limite de tempo de 7200 segundos ou por exceder a capacidade da máquina em lidar com o tamanho do modelo gerado.

Dentre os três modelos analisados, quando considerado que o número máximo de pilhas abertas deve ser igual ou inferior a 2, a formulação CI2 é a que apresenta os melhores resultados. Porém, para valores maiores de  $C$ , o modelo YLI se sobressai aos demais. Considerando todos os testes computacionais efetuados, foi possível resolver 98, 89 e 88 instâncias utilizando os modelos YLI, LGSRV e CI2, respectivamente. Como pode ser visto nas Tabelas 6 e 7, o modelo YLI é o mais compacto dos três estudados, tanto em termos de variáveis quanto de restrições. Por outro lado, a qualidade da relaxação linear da formulação YLI é inferior a dos modelos LGSRV e CI2.

Apesar de apresentar bons resultados, os modelos YLI, LGSRV e CI2 são insuficientes para determinar uma solução ótima em tempo razoável para algumas das instâncias pertencentes às classes propostas nesta dissertação. Com base nos resultados apresentados nesta seção, concluímos que há dois fatores com grande impacto no tempo requerido para resolver o CS-LOSP de forma exata, sendo esses: o comprimento dos itens e a quantidade de itens de tipos distintos. Na Tabela 18 é apresentada a quantidade de instâncias resolvidas para cada um dos modelos YLI, LGSRV e CI2 em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos.

Yanasse Lamosa Integrado <b>YLI</b>				
	M=5	M=8	M=10	M=20
$l_i = [100, 200]$	28	2	–	–
$l_i = [200, 500]$	–	–	17	0
$l_i = [300, 800]$	–	–	30	21
Laporte Gomes e Semet com remoção de vértices <b>LGSRV</b>				
	M=5	M=8	M=10	M=20
$l_i = [100, 200]$	28	9	–	–
$l_i = [200, 500]$	–	–	17	0
$l_i = [300, 800]$	–	–	30	5
Catanzaro et al. Integrado 2 <b>CI2</b>				
	M=5	M=8	M=10	M=20
$l_i = [100, 200]$	29	10	–	–
$l_i = [200, 500]$	–	–	10	1
$l_i = [300, 800]$	–	–	28	10

Tabela 18 – Número de instâncias resolvidas utilizando os modelos **YLI**, **LGSRV** e **CI2** em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos. Nesta tabela consideramos os três valores de  $C$  utilizados nos testes.

O modelo **CI2** foi o que apresentou os melhores resultados para as classes nas quais o comprimento dos itens está contido no intervalo  $[100, 200]$ . Considerando os três valores de  $C$  utilizados na dissertação, todas as classes são compostas de 30 instâncias no total. Para a Classe 1, contendo 5 tipos distintos de itens, todos os três modelos apresentaram bons resultados. Utilizando o modelo **CI2**, foi possível determinar uma solução ótima para 29 instâncias. Já para os modelos **YLI** e **LGSRV** foi possível determinar uma solução ótima para 28 instâncias. Para as instâncias da Classe 2, o tempo exigido para determinar uma solução ótima foi significativamente maior. Para o modelo **CI2** somente para 10 instâncias foi possível determinar uma solução ótima respeitando a limitação de tempo de 7200 segundos. Para as formulações **YLI** e **LGSRV**, só foi possível resolver 2 e 9 instâncias, respectivamente. O aumento no tempo computacional exigido se dá devido ao aumento no número de tipos distintos de itens, enquanto na Classe 1 as instâncias possuíam 5 tipos distintos de itens, na Classe 2, esse número cresce para 8. Comparando classes nas quais o comprimento dos itens está contido no mesmo intervalo, para todos os modelos, o aumento no número de tipos distintos de itens implicou no aumento do tempo exigido para se determinar uma solução ótima.

As Classes 3 e 4 contêm itens de tamanho intermediário, o comprimento dos itens pertencentes as instâncias destas classes está contido no intervalo  $[200, 500]$ . Por outro lado, a quantidade de tipos distintos de itens também é consideravelmente maior, a Classe 3 possui instâncias com 10 tipos distintos de itens e a Classe 4 contêm instâncias com 20 tipos

distintos de itens. Para a Classe 3, os modelos [YLI](#) e [LGSRV](#) apresentaram os melhores resultados, pois, para estes modelos, foi possível resolver 17 instâncias, ao passo que, para a formulação [CI2](#) foi possível resolver 11 instâncias. Para a Classe 4, os resultados obtidos foram insatisfatórios. De forma geral, os modelos encontraram dificuldades em resolver as Classes com 20 tipos distintos de itens (Classes 4 e 6), porém, como o comprimento dos itens na Classe 6 é relativamente grande, para esta Classe foi possível resolver um número razoável de instâncias, o mesmo não ocorre para a Classe 4 que contém itens de comprimento menor. Para esta classe, somente uma instância foi resolvida utilizando o modelo [CI2](#). Não foi possível resolver nenhuma instância utilizando as formulações [YLI](#) e [LGSRV](#).

As Classes 5 e 6 são as classes que contêm os itens de maior tamanho dentre as 6 classes utilizadas nos testes computacionais para o CS-LOSP unidimensional. O comprimento dos itens pertencentes as instâncias destas classes está contido no intervalo  $[300, 800]$ . Para a Classe 5, os modelos [YLI](#) e [LGSRV](#) obtiveram sucesso em resolver todas as instâncias desta classe para os três valores de  $C$  utilizados, enquanto para a formulação [CI2](#) foi possível resolver 28 instâncias. Para a classe 6, o modelo [YLI](#) se sobressaiu aos demais. Para esta formulação foi possível resolver 21 instâncias. Já para os modelos [CI2](#) e [LGSRV](#) foi possível resolver 10 e 5 instâncias, respectivamente. Note que, apesar das instâncias pertencentes às Classes 4 e 6 possuírem a mesma quantidade de tipos distintos, a diferença no comprimento dos itens impactou severamente no número de instâncias resolvidas.

Concluimos, então, que os modelos [YLI](#), [LGSRV](#) e [CI2](#) são ineficientes para resolver instâncias com uma grande quantidade de itens de tipos distintos ou com itens de tamanho menor. Sendo assim, a busca por formulações melhores ainda é necessária para garantir a solução de problemas de maior complexidade. Com o intuito de melhorar a eficiência dos modelos, no Capítulo 7 são propostas melhorias para as formulações [YLI](#) e [CI2](#).

Os modelos [Arbib, Marinelli e Ventura-1](#) e [Kantorovich Integrado](#) baseiam-se na formulação de [Kantorovich \(1960\)](#) para lidar com a parte do problema referente ao problema de corte de estoque. Conforme discutido na Seção 2.1, o número de variáveis e restrições presentes no modelo proposto em [Kantorovich \(1960\)](#) depende da demanda requerida de itens. Para todas as instâncias propostas nesta dissertação, a demanda requerida de itens foi gerada aleatoriamente no intervalo  $[100, 400]$ . Estes valores de demanda se mostraram demasiadamente grandes para a resolução das instâncias utilizando o modelo [Kantorovich Integrado](#) proposto nesta dissertação. Para permitir a avaliação do modelo, foram realizados testes preliminares com instâncias nas quais o valor da demanda está contido no intervalo  $[1, 5]$  e o comprimento dos itens requeridos está contido no intervalo  $[200, 500]$ . Para todos os testes foram considerados 5 tipos distintos de itens e

$C = 2$ . Para este conjunto de dados, foi possível determinar a solução ótima dos problemas respeitando a limitação de 7200 segundos, sendo assim, concluímos que a aplicação do modelo é válida para valores pequenos de demanda. Uma vantagem do modelo [Kantorovich Integrado](#) sobre as formulações [YLI](#), [LGSRV](#) e [CI2](#) é que este modelo não requer a geração do conjunto de padrões de corte antes da sua execução. Assim, não apresentamos resultados computacionais para as classes de instâncias propostas nesta dissertação para o modelo [Kantorovich Integrado](#) pois o modelo não foi capaz de resolver nem sequer as menores instâncias propostas na Seção 6.1.

## 6.6 Resultados para as instâncias do CS-LOSP bidimensional

As formulações [Kantorovich Integrado](#), [Arbib, Marinelli e Ventura-1](#) e [Arbib, Marinelli e Ventura-2](#) foram propostas especificamente para o CS-LOSP unidimensional. Desta forma, ainda que acreditemos existir a possibilidade de extensão dos modelos para o caso bidimensional, a utilização destes três modelos para resolver problemas de duas dimensões exige alterações nas formulações. Todos os demais modelos propostos para o CS-LOSP apresentados nesta dissertação são válidos para CS-LOSPs de quaisquer dimensão. Embora as formulações [YLI](#), [LGSRV](#) e [CI2](#) não necessitem de nenhuma modificação para lidar com problemas de dimensões distintas, na prática, à medida que a dimensão do problema aumenta, a complexidade dos problemas também cresce.

Na literatura os testes computacionais realizados para o CS-LOSP são frequentemente realizados para problemas com apenas uma dimensão. Nesta seção, consideramos um conjunto de padrões de corte bidimensionais e analisamos o efeito que a dimensão dos objetos e itens tem nos resultados obtidos.

Os testes são ancorados em uma base de dados proposta nesta dissertação. Nesta seção o conjunto de dados proposto é detalhado e, posteriormente, utilizado para comparar o tempo gasto por cada modelo para resolver as instâncias propostas.

## 6.7 Conjunto de dados para o CS-LOSP bidimensional

Para comparar os modelos [YLI](#), [LGSRV](#) e [CI2](#) para problemas bidimensionais um conjunto de dados é proposto. Assim, considere:

$L$  : comprimento do objeto a ser cortado;

$W$  : largura do objeto a ser cortado;

$m$  : número de tipos distintos de itens;

$l$  : intervalo no qual o comprimento dos itens está contido;

$w$  : intervalo no qual a largura dos itens está contida;

$d$  : intervalo no qual a demanda requerida para cada tipo de item está contida.

Para todas as instâncias geradas, a demanda requerida  $d$  foi gerada aleatoriamente no intervalo  $[100, 400]$ , o comprimento do objeto a ser cortado  $L$  é igual a 1000 e a largura do objeto a ser cortado  $W$  é igual a 500. Dois intervalos  $l$  foram utilizados para gerar as instâncias, sendo estes,  $l = [200, 500]$  e  $l = [300, 800]$ . O comprimento dos itens requeridos para cada instância foi gerado aleatoriamente no intervalo sem repetição, ou seja, itens de tipos distintos tem comprimentos distintos. A largura dos itens requeridos  $w$  foi gerada aleatoriamente como sendo um dos valores do conjunto  $\{100, 200, 300, 400\}$ , ou seja, todos os itens requeridos possuem largura igual a 100, 200, 300 ou 400. A quantidade de tipos distintos de itens presentes em cada instância é 5 ou 10. Foram considerados apenas padrões de corte guilhotinados 2-estágios exatos. O conjunto de dados pode ser dividido em 4 classes, sendo estas:

- Classe 1:  $m = 5$ ,  $l = [200, 500]$ ;
- Classe 2:  $m = 10$ ,  $l = [200, 500]$ ;
- Classe 3:  $m = 5$ ,  $l = [300, 800]$ ;
- Classe 4:  $m = 10$ ,  $l = [300, 800]$ .

Cada classe contém 10 instâncias. As Classes 1-4 são usadas para comparar o tempo gasto por cada modelo para resolver os problemas.

## 6.8 Resultados para as instâncias do CS-LOSP bidimensionais.

Nesta seção, o conjunto de dados bidimensional apresentado na Seção 6.7 é utilizado para a realização de testes computacionais semelhantes aos realizados na Seção 6.5. O intuito desta seção é de avaliar o impacto que a dimensão do problema tem no tempo computacional exigido para se determinar uma solução ótima além de comparar a qualidade dos modelos apresentados na dissertação para o caso bidimensional.

De forma análoga ao que foi feito para os testes computacionais para o CS-LOSP unidimensional, nesta seção, três valores distintos para o parâmetro  $C$  são utilizados nos

testes computacionais:  $C = 2$ ,  $C = 3$  e  $C = 4$ . Nas Tabelas 19, 22 e 25 são apresentadas as médias de tempo gasto para resolver 10 instâncias das quatro classes para o CS-LOSP bidimensional propostas nesta dissertação utilizando os modelos YLI, CI2 e CI2-Modificado além do valor médio da função objetivo avaliada em uma solução ótima quando consideramos  $C = 2$ ,  $C = 3$  e  $C = 4$ , respectivamente. As Tabelas 31, 33 e 35 são semelhantes às Tabelas 9, 12 e 15 apresentadas na seção referente aos testes para o CS-LOSP unidimensional.

De forma similar, as Tabelas 32, 34 e 36 são análogas às Tabelas 10, 13 e 16 apresentadas na seção referente aos testes para o CS-LOSP unidimensional, ou seja, contêm os resultados para as classes parcialmente resolvidas em detalhes quando consideramos  $C = 2$ ,  $C = 3$  e  $C = 4$ , respectivamente.

Classe	YLI Tempo(s)	LGSRV Tempo(s)	CI2 Tempo(s)	Valor da Função Objetivo
1	*	*	*	*
2	*	*	*	*
3	0,63	0,32	0,19	507,08
4	*	*	15,77	1083,50

Tabela 19 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 4 classes para o CS-LOSP bidimensional propostas nesta dissertação, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 2. Nesta tabela, o caractere “ – ” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “ \* ” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Para o conjunto de dados testado, o modelo CI2 foi o que conseguiu resolver o maior número de instâncias. Dentre as 40 instâncias testadas, para este modelo foi possível determinar uma solução ótima dentro do limite de tempo (7200s) para 37 delas, já para o modelo LGSRV 32 instâncias foram resolvidas, enquanto para o modelo YLI foi possível determinar uma solução ótima para 30 instâncias. Para a formulação CI2 foi possível resolver duas classes de forma completa e uma classe de forma parcial, ao passo que, para os modelos YLI e LGSRV foi possível resolver uma classe de forma completa e três classes de forma parcial.

Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado (YLI)			
1	(1,57; 10,36; 2,17; 74,37; 0,08; 2,27; 2,33; 0,74; 0,83)	(1; 2; 3; 4; 5; 6; 8; 9; 10)	(467; 334; 401; 187; 249; 243; 387; 395; 574)
2	(3598,46; 1777,11)	(2; 6)	(520; 537)
4	(18,65; 30,58; 4506,57; 96,67; 0,67; 7,91 26,31; 12,42; 8,88)	(1; 2; 3; 5; 6; 7; 8; 9; 10)	(914; 1090; 606; 1312; 1478; 1093 1388; 1168; 711)
Laporte Gomes e Semet com remoção de vértices (LGSRV)			
1	(4,03; 9,14; 6,55; 192,59; 3,61; 110,29; 11,58; 2,21; 2,83)	(1; 2; 3; 4; 5; 6; 8; 9; 10)	(467; 334; 401; 187; 249; 243; 387; 395; 574)
2	(54,03; 79,10; 2162,40; 832,80)	(2; 6; 7; 8)	( 520; 537; 391; 529)
4	(17,70; 44,09; 614,44; 87,74; 2,05; 25,48; 8,32; 5,99; 16,87)	(1; 2; 3; 5; 6; 7; 8; 9; 10)	(914; 1090; 606; 1312; 1478; 1093 1388; 1168; 711)
Catanzaro et al. Integrado 2 (CI2)			
1	(0,10; 0,08; 0,10; 0,44; 0,10; 0,87 0,13; 0,11; 0,07)	(1; 2; 3; 4; 5; 6; 8; 9; 10)	(467; 334; 401; 187; 249; 243; 387; 395; 574)
2	(40,88; 31,93; 768,87; 1599,43; 350,68; 16,85; 1189,92; 37,48)	(1; 2; 3; 4; 5; 6; 7; 8)	(426; 520; 448; 486; 523; 537; 391; 529)

Tabela 20 – Resultados em detalhes para as classes propostas para o CS-LOSP bidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicada a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 2$ .

Assim como para o caso unidimensional, o comprimento dos itens e a quantidade dos itens distintos influenciam no tempo gasto para determinar uma solução ótima. Novamente, as classes tais que o comprimento dos itens é menor e a quantidade de tipos distintos é maior são as mais difíceis de resolver. Na Tabela 21 apresentamos a relação entre a quantidade de itens de tipos distintos e o tamanho dos itens com o tempo gasto para se determinar uma solução ótima quando consideramos  $C = 2$ .

	YLI		LGSRV		CI2	
	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)
$l_i = [200, 500]$	11	497,30	13	267,01	18	224,38
$l_i = [300, 800]$	19	248,15	19	43,47	20	7,98
$M = 5$	19	5,34	19	18,21	20	0,24
$M = 10$	11	916,75	13	303,92	18	232,99

Tabela 21 – Comparação entre o tempo gasto e número de instâncias resolvidas em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das classes para o CS-LOSP bidimensional utilizando  $C = 2$ .

Na Tabela 22 apresentamos os resultados computacionais para os testes bidimensionais quando o valor de  $C$  é igual a três.

Classe	YLI Tempo(s)	LGSRV Tempo(s)	CI2 Tempo(s)	Valor da Função Objetivo
1	22,78	2,91	3,22	343,01
2	*	-	*	*
3	1,60	1,28	2,73	507,80
4	*	*	*	*

Tabela 22 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 4 classes, propostas nesta dissertação, para o CS-LOSP bidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos YLI, LGSRV e CI2. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 3. Nesta tabela, o caractere “-” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Considerando  $C = 3$ , assim como para os testes realizados para o CS-LOSP unidimensional, o modelo YLI foi o que obteve os melhores resultados, pois foi possível resolver 30 instâncias dentro do limite de tempo (7200s). Já para as formulações CI2 e LGSRV foi possível encontrar uma solução ótima em menos de 7200 segundos para 29 e 27 instâncias, respectivamente. Para os modelos YLI e CI2 duas classes foram resolvidas de forma completa e outras duas, de forma parcial. Enquanto, para a formulação LGSRV foi possível resolver 2 duas classes de forma completa e uma de forma parcial.

De forma análoga aos resultados apresentados para o CS-LOSP unidimensional, de forma geral, o tempo exigido para se determinar uma solução ótima foi aumentado quando o valor de  $C$  aumentou de 2 para 3.



Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado (YLI)			
2	7132,79	9	555
4	(1043,14; 955,90; 18,16; 53,90; 0,38; 18,47; 16,47; 2,73; 1,49)	(1; 2; 4; 5; 6; 7; 8; 9; 10)	(913; 1090; 1042 1312; 1478; 1093; 1388; 1168; 711)
Laporte Gomes e Semet com remoção de vértices (LGSRV)			
4	(2896,91; 216,45; 4,32; 756,47; 353,12; 104,90; 27,62)	(2; 4; 6; 7; 8; 9; 10)	(1090; 1042; 1478; 1093 1388; 1168; 711)
Catanzaro et al. Integrado 2 CI2			
2	3944.94	9	555
4	(583.80; 74,55; 1299,12; 5,22; 472,57; 2158,66; 297,72; 73,83)	(2; 4; 5; 6; 7; 8; 9; 10)	(1090; 1042; 1312; 1478; 1093; 1388; 1168; 711)

Tabela 23 – Resultados em detalhes para as classes propostas para o CS-LOSP bidimensional parcialmente resolvidas utilizando os modelos YLI, LGSRV e CI2. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 3$ .

Mais uma vez, o tempo necessário para se determinar uma solução ótima foi proporcional ao número total de padrões de corte, ou seja, quanto maior o número de itens de tipos distintos e quanto menor o comprimento dos itens requeridos maior o tempo exigido para determinar uma solução ótima. Na Tabela 24 apresentamos a relação entre a quantidade de itens de tipos distintos e o comprimento dos itens com o tempo gasto para se determinar uma solução ótima quando consideramos  $C = 3$ .

	YLI		LGSRV		CI2	
	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)
$l_i = [200, 500]$	11	669,14	10	2,91	11	361,56
$l_i = [300, 800]$	19	111,93	17	257,21	18	277,38
$M = 5$	20	12,19	20	2,10	20	2,98
$M = 10$	10	924,34	7	622,83	9	990,05

Tabela 24 – Comparação entre o tempo gasto e número de instâncias resolvidas em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das classes elaboradas para o CS-LOSP bidimensional utilizando  $C = 3$ .

Por último, a Tabela 25 refere-se aos resultados computacionais para o CS-LOSP bidimensional quando o valor de  $C$  é igual a quatro.

Classe	YLI Tempo(s)	LGSRV Tempo(s)	CI2 Tempo(s)	Valor da Função Objetivo
1	43,76	3,28	27,66	343,10
2	-	*	-	*
3	0,54	0,94	5,07	507,80
4	*	*	*	*

Tabela 25 – Média de tempo gasto para resolver as 10 instâncias de cada uma das 4 classes, propostas nesta dissertação, para o CS-LOSP bidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos **YLI**, **LGSRV** e **CI2**. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 4. Nesta tabela, o caractere “-” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Para o modelo **LGSRV** foi possível resolver duas classes de forma completa e duas classes de forma parcial totalizando 30 instâncias. Para as formulações **YLI** e **CI2** foram resolvidas duas classes de forma completa e uma classe de forma parcial. No total, foram resolvidas 29 e 23 instâncias para os modelos **YLI** e **CI2**, respectivamente.

Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado ( <b>YLI</b> )			
4	(236,31; 230,15; 37,52; 778,48; 0,29; 115,33; 19,90; 4,60; 8,24)	(1; 2; 4; 5; 6; 7; 8; 9; 10)	(913; 1090; 1042; 1312; 1478; 1093; 1388; 1168; 711)
Laporte Gomes e Semet com remoção de vértices ( <b>LGSRV</b> )			
2	536,11	8	475
4	(599,30; 4361,86; 541,71; 1914,53; 1,36; 2180,55, 415,03; 17,10; 19,32)	(1; 2; 4; 5; 6; 7; 8; 9; 10)	(913; 1090; 1042; 1312; 1478; 1093; 1388; 1168; 711)
Catanzaro et al. Integrado 2 ( <b>CI2</b> )			
4	(194,18; 9,89; 495,92)	(4; 6; 10)	(1042; 1478; 711)

Tabela 26 – Resultados em detalhes para as classes propostas para o CS-LOSP bidimensional parcialmente resolvidas utilizando os modelos **YLI**, **LGSRV** e **CI2**. Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima; na terceira coluna, quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 4$ .

Os resultados obtidos para  $C = 4$  seguem a mesma linha que aqueles encontrados para  $C = 2$  e  $C = 3$ , novamente as classes com maior número de itens de tipos distintos e tais que o comprimento dos itens é menor foram as que exigiram mais tempo para se determinar uma solução ótima. Na **Tabela 27** apresentamos a relação entre quantidade de itens de tipos distintos e o comprimento dos itens com o tempo gasto para se determinar

uma solução ótima quando consideramos  $C = 4$ .

	YLI		LGSRV		CI2	
	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)	nº instâncias	Tempo(s)
$l_i = [200, 500]$	10	43,76	11	51,70	10	26,67
$l_i = [300, 800]$	19	75,59	19	529,48	13	57,74
$M = 5$	20	22,15	20	2,10	20	16,37
$M = 10$	9	158,98	10	1058,69	3	233,33

Tabela 27 – Comparação entre o tempo gasto e número de instâncias resolvidas em função do tamanho dos itens e da quantidade de tipos distintos de itens. Os dados contidos nesta tabela são referentes à solução das classes propostas para o CS-LOSP bidimensional utilizando  $C = 4$ .

O valor do parâmetro  $C$  impacta os testes bidimensionais de forma similar aos testes para o CS-LOSP unidimensional. Utilizando a formulação YLI, foi possível resolver 30 instâncias para  $C = 2$  e  $C = 3$  e 29 instâncias para  $C = 4$ . Já para a formulação LGSRV, foram resolvidas 32, 27 e 30 instâncias considerando  $C = 2$ ,  $C = 3$  e  $C = 4$ , respectivamente. Assim como para os testes para o CS-LOSP unidimensional, a quantidade de instâncias resolvidas utilizando o modelo CI2 foi inversamente proporcional ao valor de  $C$ . Considerando  $C = 2$ ,  $C = 3$  e  $C = 4$ , foi possível resolver 37, 29 e 23 instâncias, respectivamente. Considerando todos os testes computacionais efetuados, foi possível resolver 89 instâncias para os três modelos. Note que, mais uma vez, os resultados do modelo CI2 são melhores que os demais modelos para  $C = 2$ , porém, para valores maiores de  $C$  o modelo YLI é capaz de resolver um número superior de instâncias quando comparado à formulação CI2. Considerando  $C = 4$ , o modelo LGSRV é o que apresenta os melhores resultados, pois utilizando este modelo foi possível resolver 30 instâncias. Na Tabela 28 é apresentada a quantidade de instâncias resolvidas para cada um dos modelos YLI, LGSRV e CI2 em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos, considerando os testes bidimensionais.

Yanasse Lamosa Integrado <b>YLI</b>		
	M=5	M=10
$l_i = [200, 500]$	29	3
$l_i = [300, 800]$	30	27
Laporte Gomes e Semet com remoção de vértices <b>LGSRV</b>		
	M=5	M=10
$l_i = [200, 500]$	29	5
$l_i = [300, 800]$	30	25
Catanzaro et al. Integrado 2 <b>CI2</b>		
	M=5	M=10
$l_i = [200, 500]$	29	9
$l_i = [300, 800]$	30	21

Tabela 28 – Número de instâncias resolvidas utilizando os modelos **YLI**, **LGSRV** e **CI2** em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos. Nesta tabela consideramos os três valores de  $C$  utilizados nos testes.

Novamente, a formulação **CI2** se sobressaiu às demais quando consideradas as classes contendo itens de menor comprimento, ou seja, as classes nas quais o comprimento dos itens está contido no intervalo  $[200, 500]$ . Todos os três modelos apresentaram bons resultados para a Classe 1. Esta classe possui 5 tipos distintos de itens. Para todos os três modelos foi possível resolver 29 instâncias. Para todos os modelos testados, o número de instâncias resolvidos para a Classe 2 é significativamente menor do que o número de instâncias resolvidas para a Classe 1. Isso ocorre devido ao aumento no número de tipos de itens distintos. Para os modelos **CI2**, **LGSRV** e **YLI**, foi possível resolver 9, 5 e 3 instâncias, respectivamente.

As Classes 3 e 4 contêm itens de maior comprimento, o comprimento dos itens nestas classes está contido no intervalo  $[300, 800]$ . Todos os três modelos conseguiram resolver todas as instâncias da Classe 3. Para a Classe 4, utilizando os modelos **CI2**, **LGSRV** e **YLI** foi possível resolver 21, 25 e 27 instâncias, respectivamente.

Nota-se que a dimensão do problema é um fator com grande influência no tempo exigido para se determinar uma solução ótima. Considerando os três valores de  $C$ , para os testes computacionais para o CS-LOSP unidimensional, foi possível resolver cerca de 74% das instâncias contendo até 10 tipos de itens distintos, enquanto, para os problemas bidimensionais, apenas 50% das instâncias com até 10 tipos distintos de itens foram resolvidas. Uma vez que, o aumento da dimensão do problema implicou em uma redução no número de instâncias resolvidas e a quantidade de instâncias resolvidas para as classes propostas para o CS-LOSP unidimensional com 20 tipos distintos de itens foi consideravelmente baixa, decidimos não efetuar testes computacionais para o CS-LOSP

bidimensional com 20 tipos distintos de itens. Por esta razão, os testes conduzidos nesta seção contêm 5 ou 10 tipos distintos de itens.

Mesmo considerando um número pequeno de itens de tipos distintos e itens com comprimento relativamente grande, não foi possível resolver todas as instâncias propostas. Desta forma, concluímos que a aplicação direta dos modelos [CI2](#), [LGSRV](#) e [YLI](#) para a resolução de problemas bidimensionais com dez ou mais tipos distintos de itens é ineficiente, especialmente quando o comprimento dos itens é pequeno. Uma vez que o CS-LOSP bidimensional é pouco abordado na literatura e é um problema com grande aplicação prática, julgamos necessário a construção de formulações mais eficientes para o problema. No capítulo 7 são propostas melhorias para os modelos [CI2](#) e [YLI](#). Estas melhorias são testadas utilizando as classes unidimensionais 1, 2, 3 e 4, propostas neste capítulo.

## 7 Melhorias

Um aspecto não desejável do CS-LOSP é que, uma vez determinada a frequência na qual cada padrão de corte deve ser executado, existem diversas sequências distintas que respeitam a limitação no número máximo de pilhas abertas, desta maneira, é possível obter um número grande de soluções distintas que apresentam o mesmo valor de função objetivo.

Em [Faggioli e Bentivoglio \(1998\)](#), para resolver o problema de sequenciamento dos padrões de corte para minimização do número máximo de pilhas abertas, os autores sugerem ignorar um padrão de corte  $i$ , caso exista algum padrão de corte  $j$ , tal que, todos os itens gerados ao cortar o padrão de corte  $i$  sejam também gerados ao cortar o padrão de corte  $j$ . Ou seja, considerando o conjunto  $P_i$  como o conjunto contendo todos os itens gerados ao executar o padrão de corte  $i$  e  $P_j$  o conjunto contendo todos os itens gerados ao executar o padrão de corte  $j$ , caso  $P_i \subseteq P_j$  o padrão de corte  $i$  é ignorado. Os autores demonstram que essa estratégia garante a otimalidade da solução. De forma similar, em [Becceneri, Yanasse e Soma \(2004\)](#) os padrões de corte que não interferem na obtenção de uma solução ótima são eliminados durante a fase de pré-processamento.

O procedimento utilizado em [Faggioli e Bentivoglio \(1998\)](#) e [Becceneri, Yanasse e Soma \(2004\)](#) não pode ser utilizado no contexto do CS-LOSP para os modelos propostos neste trabalho, pois o CS-LOSP é a composição do PCE e MOSP e, sendo assim, embora os padrões de corte redundantes possam ser removidos sem alterar o número máximo de pilhas abertas durante o sequenciamento, para garantir a solução ótima do PCE é preciso considerar todos os padrões de corte possíveis. Portanto, uma abordagem diferente se faz necessária. Ao invés de eliminar os padrões de corte dominados durante a fase do pré-processamento nossa abordagem será de agrupá-los em conjuntos e, então, buscar o sequenciamento ótimo destes conjuntos. O método proposto aqui é uma adaptação do proposto em [Faggioli e Bentivoglio \(1998\)](#) e [Becceneri, Yanasse e Soma \(2004\)](#), uma vez que só os conjuntos distintos são considerados.

**Definição:** Dois padrões de corte quaisquer são ditos do mesmo tipo se quando executados geram os mesmos tipos de itens, independente das quantidades de cada item. Do contrário, são padrões de corte de tipos distintos.

Seja  $P$  um conjunto de padrões de corte qualquer, com cardinalidade igual a  $N$ , contendo  $K$  tipos distintos de padrões de corte. Os padrões de corte contidos em  $P$  podem ser divididos em  $K$  subconjuntos, de forma que, padrões de corte do mesmo tipo pertençam ao mesmo subconjunto. Sejam  $C_i$ ,  $i = 1, \dots, K$  subconjuntos de  $P$ , tais que,

todos os padrões de corte pertencentes a  $P$ , que pertençam a um mesmo tipo, estejam contidos em  $C_i$  para algum  $i = 1, \dots, K$  e se  $i \neq j \iff C_i \neq C_j$ .

**(Exemplo III)**

Considere o conjunto de padrões de corte  $P$  a seguir:

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 2 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 2 & 1 & 0 & 1 & 0 & 0 & 2 & 1 & 2 \\ 0 & 1 & 0 & 2 & 0 & 1 & 1 & 0 & 0 & 3 & 2 & 0 & 2 & 1 & 1 & 0 \\ 9 & 7 & 6 & 5 & 5 & 4 & 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & 1 & 0 \end{pmatrix}$$

este conjunto possui 9 tipos distintos de padrões de corte, portanto, é possível separá-lo em 9 subconjuntos distintos tais que, os padrões de corte pertencentes a cada subconjunto gerem os mesmos tipos de itens quando executados. Sendo estes:

$$C_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 9 \end{pmatrix} \quad C_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 2 & 3 \\ 7 & 5 & 2 \end{pmatrix} \quad C_3 = \begin{pmatrix} 0 & 0 \\ 1 & 2 \\ 0 & 0 \\ 6 & 3 \end{pmatrix}$$

$$C_4 = \begin{pmatrix} 1 & 2 \\ 0 & 0 \\ 0 & 0 \\ 5 & 2 \end{pmatrix} \quad C_5 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & 2 & 1 \\ 4 & 2 & 1 \end{pmatrix} \quad C_6 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 2 \\ 3 & 1 \end{pmatrix}$$

$$C_7 = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 3 \end{pmatrix} \quad C_8 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad C_9 = \begin{pmatrix} 1 \\ 2 \\ 0 \\ 0 \end{pmatrix}$$

A divisão do conjunto de padrões de corte, em subconjuntos  $C_i$ , apresentam duas interessantes propriedades:

**Propriedade 1:** Dado um sequenciamento de padrões de corte, no qual  $n$  padrões de corte pertencentes a  $C_i$ , são executados um após o outro, a ordem na qual estes  $n$  padrões de corte são executados pode ser permutada sem alterar o número máximo de pilhas abertas por esta sequência.

*Demonstração.* Seja  $S$  um sequenciamento de padrões de corte arbitrário, no qual  $n$  padrões de corte pertencentes a  $C_i$  são executados um após o outro. A sequência  $S$  pode ser dividida em três subsequências, sendo estas,  $S_1$ ,  $S_2$  e  $S_3$ . Seja  $S_1$  a subsequência equivalente a parte da sequência  $S$  anterior à execução dos padrões de corte pertencentes a  $C_i$ , a subsequência  $S_2$  como a ordem na qual os padrões de corte pertencentes a  $C_i$  são executados e a subsequência  $S_3$  equivalente à parte da sequência  $S$  posterior à execução dos padrões de corte pertencentes a  $C_i$ .

Seja  $\overline{K}$  o número máximo de pilhas abertas ao executar a sequência  $S_1$  e  $K$  o número de pilhas abertas após a execução de  $S_1$ . Quando um padrão de corte é executado, para cada tipo distinto de item gerado pelo padrão de corte deve ser aberta uma nova pilha se e somente se a pilha referente ao item não tenha sido aberta anteriormente. Sendo assim, considere  $l$  como a quantidade de itens de tipos distintos gerados ao cortar um objeto utilizando um padrão de corte pertencente a  $C_i$  e  $m$ , como o número de pilhas referentes aos itens gerados ao cortar um objeto utilizando um padrão de corte pertencente a  $C_i$  que foram abertas anteriormente, ou seja, durante a execução da sequência  $S_1$ . Então o número de pilhas abertas a partir da execução do primeiro padrão de corte da sequência  $S_2$  é dado por  $K + l - m$ . Note que após a execução do 1º padrão de corte, todas as pilhas referentes aos itens gerados ao cortar um objeto utilizando um padrão de corte pertencente a  $C_i$  já estão abertas. Assim, ao executar o restante da sequência  $S_2$ , nenhuma nova pilha será aberta. A quantidade de pilhas abertas após a execução da sequência  $S_2$  é dada por  $K + l - m - f$ , sendo  $f$  a quantidade de pilhas fechadas após a execução de  $S_2$ . A execução de  $S_3$  permite a abertura e fechamento de novas pilhas. Seja  $\overline{K}_2$  o número máximo de pilhas abertas durante a execução da sequência  $S_3$ , então o número máximo de pilhas abertas durante a execução de  $S$  é dado por:

$$\max \{ \overline{K}, K + l - m, \overline{K}_2 \}.$$

Observe que, por hipótese,  $S_2$  é uma sequência arbitrária, logo, o número máximo de pilhas abertas durante a execução de  $S$  independe da sequência  $S_2$ . Portanto, a ordem na qual os elementos da sequência  $S_2$  são executados pode ser permutada sem alterar o número máximo de pilhas abertas.

□



**Propriedade 2:** Dado um conjunto de padrões de corte  $P$ , seja  $S$  um conjunto contendo todos os sequenciamentos possíveis para este conjunto  $P$ . Sendo  $\bar{S} \subset S$  o conjunto de todos os sequenciamentos possíveis dos padrões de corte contidos em  $P$ , nos quais todos os padrões de corte que geram os mesmos tipos de item são sempre executados em sequência e  $\bar{S}^c$  o complementar de  $\bar{S}$ . Então, para toda sequência  $S^* \in \bar{S}^c$  é possível encontrar uma sequência  $\bar{S}^* \in \bar{S}$  tal que o número máximo de pilhas abertas a partir da execução de  $\bar{S}^*$  seja igual ou inferior ao número máximo de pilhas abertas a partir da execução de  $S^*$ .

*Demonstração.* Seja  $S^* \in \bar{S}^c$  uma sequência arbitrária de padrões de corte:

$$S^* = p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_N.$$

Seja  $C_1$  o conjunto contendo todos os padrões de corte que geram os mesmos tipos de item do padrão de corte  $p_1$  e seja a cardinalidade de  $C_1$  igual a  $n_1$ . Logo, na sequência  $S^*$ , no máximo, os primeiros  $n_1$  termos são pertencentes a  $C_1$ . Seja  $n_1^*$  o maior valor tal que,  $p_i \in C_1, \forall 1 \leq i \leq n_1^*$ . Por construção,  $p_{n_1^*+1} \notin C_1$ . Seja  $C_2$  o conjunto contendo todos os padrões de corte que geram os mesmos tipos de item do padrão de corte  $p_{n_1^*+1}$  e seja a cardinalidade de  $C_2$  igual a  $n_2$ . Logo, na sequência  $S^*$ , no máximo, os primeiros  $n_1 + n_2$  termos são pertencentes a  $C_1 \cup C_2$ , seja  $n_2^*$  o maior valor tal que,  $p_i \in C_1 \cup C_2, \forall 1 \leq i \leq n_2^*$ . Por construção,  $p_{n_2^*+1} \notin C_1 \cup C_2$ . Suponha que existam  $K$  conjuntos distintos  $C_i$  tais que todos os padrões de corte pertencentes a  $C_i$  gerem os mesmos tipos de itens. Este processo pode ser repetido de forma a se obter todos conjuntos  $C_k, 1 \leq k \leq K$  e os valores  $n_k^*$ , de forma que,  $p_i \in \bigcup_{j=1}^k C_j, \forall 1 \leq i \leq n_k^*$ , e  $n_k^* \leq \sum_j n_j$ . Seja  $\bar{S}^* \in \bar{S}$  uma sequência na qual os padrões de corte pertencentes a  $C_1$  são executados, seguidos da execução de todos os padrões de corte pertencentes a  $C_2$  e, assim em diante, ou seja:

$$\bar{S}^* = C_1 \rightarrow C_2 \rightarrow \dots \rightarrow C_K.$$

Será demonstrado que o número máximo de pilhas abertas a partir da execução de  $\bar{S}^*$  é igual ou inferior ao número máximo de pilhas abertas a partir da execução de  $S^*$ . O número de pilhas abertas em um instante  $t$  é igual ao número total de pilhas abertas menos o número total de pilhas fechadas. Seja então  $\bar{A}_t$  o número de pilhas abertas após a execução do  $t$  – ésimo padrão de corte da sequência  $\bar{S}^*$ . Considerando  $l_i$  a quantidade de itens de tipos distintos que são gerados ao executar um padrão de corte do conjunto  $C_i$ ,  $l_1$  pilhas são abertas ao executar o primeiro padrão de corte da sequência  $\bar{S}^*$ . Ao se executar os demais  $n_1 - 1$  padrões de corte nenhuma nova pilha é aberta, pois todos os padrões de corte pertencentes a  $C_1$  geram os mesmos tipos de itens, logo, todas as pilhas

referentes a estes itens já estão abertas. Ao executar o padrão de corte  $n_1 + 1$  devem ser abertas as pilhas referentes aos padrões de corte pertencentes a  $C_2$  que não foram abertas anteriormente devido a execução dos padrões de corte pertencentes a  $C_1$ . Logo, o total de pilhas que foram abertas até o instante  $t = n_1 + 1$  é igual a  $l_1 + (l_2 - |J_1 \cap J_2|)$ , sendo  $J_i$  o conjunto contendo quais tipos de itens são gerados ao executar um padrão de corte pertencente a  $C_i$ . A execução dos próximos  $n_2 - 1$  padrões de corte não abre nenhuma nova pilha, pois todas as pilhas referentes a  $C_2$  já foram abertas anteriormente. Quando o padrão de corte  $n_2 + 1$  é executado acontece exatamente o mesmo processo que ocorreu anteriormente, logo o total de pilhas que foram abertas até o instante  $t = n_2 + 1$  é igual a  $l_1 + (l_2 - |J_1 \cap J_2|) + (l_3 - |J_1 \cup J_2 \cap J_3|)$ . De forma geral, a função  $\overline{A}_t$  é dada por:

$$\overline{A}_t = \begin{cases} l_1, & \text{para } t \leq n_1, \\ l_1 + (l_2 - |J_1 \cap J_2|), & \text{para } n_1 < t \leq n_2, \\ \vdots \\ \sum_j^K l_j - \sum_j^K |(\bigcup_{i=1}^{j-1} J_i) \cap J_j|, & \text{para } n_{K-1} < t \leq n_K. \end{cases}$$

Agora, considere  $s^*$  uma subsequência de  $S^*$  composta pela execução dos primeiros  $n_{K-1}^* + 1$  padrões de corte da sequência  $S^*$ , na mesma ordem de execução de  $S^*$ . Seja  $A_t^*$  o número total de pilhas abertas no instante  $t$ . É fácil mostrar que, ao executar o primeiro padrão de corte da sequência  $s^*$ , são abertas  $l_1$  pilhas e, ao executar os demais  $n_1^* - 1$  padrões de corte, nenhuma nova pilha é aberta, pois todos os padrões de corte pertencentes a  $C_1$  geram os mesmos tipos de itens. Portanto, todas as pilhas referentes a estes itens já estão abertas. Ao executar o padrão de corte  $n_1^* + 1$  devem ser abertas as pilhas referentes aos padrões de corte pertencentes a  $C_2$  que não foram abertas anteriormente devido a execução dos padrões de corte pertencentes a  $C_1$ . Logo, o total de pilhas que foram abertas até o instante  $t = n_1^* + 1$  é igual a  $l_1 + (l_2 - |J_1 \cap J_2|)$ . A execução dos próximos  $n_3^* - n_1^* - 1$  padrões de corte não abre nenhuma nova pilha, pois somente padrões de cortes pertencentes a  $C_1 \cup C_2$  serão executados e todas as pilhas referentes a  $C_1$  e  $C_2$  já foram abertas anteriormente. Quando o padrão de corte  $n_3^* + 1$  é executado acontece exatamente o mesmo processo que ocorreu anteriormente. Logo, o total de pilhas que foram abertas até o instante  $t = n_3^* + 1$  é igual a  $l_1 + (l_2 - |J_1 \cap J_2|) + (l_3 - |J_1 \cup J_2 \cap J_3|)$ . Portanto, a função  $A_t^*$  é muito parecida com a função  $\overline{A}_t$ , sendo a única diferença entre as duas os intervalos de  $t$ :

$$A_t^* = \begin{cases} l_1, & \text{para } t \leq n_1^*, \\ l_1 + (l_2 - |J_1 \cap J_2|), & \text{para } n_1^* < t \leq n_2^*, \\ \vdots \\ \sum_j^K l_j - \sum_j^K |(\bigcup_{i=1}^{j-1} J_i) \cap J_j|, & \text{para } n_{K-1}^* < t \leq n_K^*. \end{cases}$$

Seja  $\overline{F}_t$  o número de pilhas fechadas após a execução do  $t$ -ésimo padrão de corte da sequência  $\overline{S}^*$ . Se  $t \leq n_1$  então nenhuma pilha pode ser fechada, pois não há nenhum item tal que já foram executados todos os padrões de corte contendo este item. No instante  $t = n_1 + 1$ , todos os padrões de corte pertencentes a  $C_1$  foram executados, portanto, as pilhas referentes aos itens presentes em  $C_1$ , que não estão presentes em mais nenhum padrão de corte, podem ser fechadas. Portanto, há um total de  $|J_1 \setminus \{(\bigcup_{i=2}^K J_i)\}|$  pilhas fechadas neste instante. De forma análoga ao que foi feito anteriormente, nenhuma pilha pode ser fechada até o instante  $t = n_2 + 1$ . Neste instante, todos os padrões de corte referentes aos conjuntos  $C_1$  e  $C_2$  foram executados. Portanto, podem ser fechadas as pilhas que estão presentes em  $C_1 \cup C_2$  e não estão presentes em mais nenhum conjunto  $C_i$ , ou seja  $|J_1 \cup J_2 \setminus \{(\bigcup_{i=3}^K J_i)\}|$  pilhas estão fechadas, de forma geral:

$$\overline{F}_t = \begin{cases} 0, & \text{para } t \leq n_1, \\ |J_1 \setminus \{(\bigcup_{i=2}^K J_i)\}|, & \text{para } n_1 < t \leq n_2, \\ |J_1 \cup J_2 \setminus \{(\bigcup_{i=3}^K J_i)\}|, & \text{para } n_2 < t \leq n_3, \\ \vdots \\ |\bigcup_{i=1}^{K-2} J_i \setminus \{(\bigcup_{i=K-1}^K J_i)\}|, & \text{para } n_{K-2} < t \leq n_{K-1}, \\ |\bigcup_{i=1}^{K-1} J_i \setminus \{(\bigcup_{i=K}^K J_i)\}|, & \text{para } n_{K-1} < t \leq n_K. \end{cases}$$

Seja  $\overline{P}_t$  a quantidade de pilhas que estão abertas após a execução do  $t$ -ésimo padrão de corte da sequência  $\overline{S}^*$ , então:

$$\overline{P}_t = \overline{A}_t - \overline{F}_t.$$

O número máximo de pilhas que estão abertas simultaneamente durante a execução da sequência  $\overline{S}^*$  é dado por:

$$\max \{\overline{P}_1, \overline{P}_2, \dots, \overline{P}_N\} = P_{\bar{t}},$$

para algum  $\bar{t} \in \{1, 2, \dots, N\}$ . Note que  $\bar{t}$  está contido no intervalo  $\{n_j, n_{j+1}\}$  para algum  $j$  e, portanto:

$$\begin{aligned} \overline{P}_t &= \overline{A}_t - \overline{F}_t \\ &= \left[ \sum_k^{j+1} l_k - \sum_k^{j+1} \left| \left( \bigcup_{i=1}^{k-1} (J_i) \cap J_k \right) \right| \right] - \left[ \left| \bigcup_{i=1}^j (J_i) \setminus \left\{ \left( \bigcup_{i=j+1}^K (J_i) \right) \right\} \right| \right]. \end{aligned}$$

Seja  $P_t^*$  a quantidade de pilhas que estão abertas após a execução do  $t$  –ésimo padrão de corte da sequência  $\overline{s}^*$ , então:

$$P_t^* = A_t^* - F_t^*,$$

sendo  $F_t^*$  o número de pilhas fechadas após a execução do  $t$  –ésimo padrão de corte da sequência  $s^*$ . A menos que todos os padrões de corte de um conjunto  $C_i$  tenham sido executados, as pilhas referentes aos itens gerados por este conjunto não podem ser fechadas. No instante  $t^* \in (n_j^*, n_{j+1}^*]$  todos os padrões de corte executados, são pertencentes a um dos conjuntos  $C_1, C_2, \dots, C_{n_j^*}$  e  $C_{n_{j+1}^*}$ . Seja  $C$  o conjunto contendo todos os conjuntos  $C_i$ , tais que, todos os padrões de corte pertencentes a  $C_i$  foram executados até o instante  $t^*$  e  $C^C$  o complementar de  $C$ , logo,  $C \subset \{1, 2, \dots, j\}$  e, portanto:

$$F_{t^*}^* = \left( \left| \bigcup_{i \in C} (J_i) \setminus \left( \bigcup_{i \in C^c} (J_i) \right) \right| \right) \leq \left( \left| \bigcup_{i=1}^j (J_i) \setminus \left( \bigcup_{i=j+1}^K (J_i) \right) \right| \right) = \overline{F}_t.$$

Ao mesmo tempo temos:

$$A_{t^*}^* = \sum_k^{j+1} l_k - \sum_k^{j+1} \left( \bigcup_{i \leq (k-1)} (J_i) \cap J_k \right) = \overline{A}_t,$$

logo:

$$\overline{P}_t \leq P_t^* \leq \max \{P_1^*, P_2^*, \dots, P_{n_{k-1}^*+1}^*\} \leq P^*,$$

sendo  $P^*$  o número máximo de pilhas abertas ao executar a sequência  $S^*$ . Portanto, para toda sequência pertencente a  $\overline{S}^C$  é possível encontrar uma sequência  $\overline{S}^* \in \overline{S}$  tal que, o número máximo de pilhas abertas a partir da execução de  $\overline{S}^*$  seja igual ou inferior ao número máximo de pilhas abertas a partir da execução de  $S^*$ .

□

Originalmente, o CS-LOSP busca determinar a frequência na qual os padrões de corte devem ser executados e, ao mesmo tempo, determinar um sequenciamento de padrões de corte que, quando executado, respeite a restrição imposta sobre o número

máximo de pilhas abertas. A Propriedade 2 garante que, se existe um sequenciamento de padrões de corte que abra, no máximo,  $C$  pilhas, então, separando os padrões de corte em conjuntos  $C_i$ , de forma que padrões de corte que gerem os mesmos tipos de item pertençam ao mesmo conjunto, existe um sequenciamento de conjuntos  $C_i$  que, quando executado, abre, no máximo,  $C$  pilhas. Enquanto a Propriedade 1 garante que, quando  $n$  padrões de corte pertencentes a um mesmo conjunto  $C_i$  são executados em sequência, a ordem na qual estes padrões de corte são executados pode ser permutada sem alterar o número máximo de pilhas abertas obtido através do sequenciamento. Sendo assim, um CS-LOSP pode ser reformulado como um problema de determinar a frequência na qual os padrões de corte devem ser executados de forma a minimizar o material desperdiçado e, ao mesmo tempo, encontrar um sequenciamento dos conjuntos  $C_i$  de forma que sua execução abra, no máximo,  $C$  pilhas. De fato, a partir de uma solução para o CS-LOSP reformulado é possível determinar uma solução para o CS-LOSP usual e vice-versa, como garantem as Propriedades 3 e 4, definidas a seguir:

**Propriedade 3:** Seja uma solução para o CS-LOSP reformulado contendo a frequência na qual os padrões de corte devem ser executados e um sequenciamento de conjuntos  $C_i$ . A partir desta solução, é possível determinar um sequenciamento de padrões de corte, cuja execução abre o mesmo número de pilhas que o sequenciamento dos conjuntos  $C_i$ , ou equivalentemente, dada uma solução ótima do CS-LOSP reformulado é possível obter uma solução ótima para o CS-LOSP usual.

*Demonstração.* Considere um CS-LOSP no qual há um total de  $N$  padrões de corte que podem ser divididos em  $k$  subconjuntos de padrões de corte do mesmo tipo. Sem perda de generalidade, considere os conjuntos  $C_i$  dados por:

$$\begin{aligned} C_1 &= \{p_1, p_2, \dots, p_{n_1}\}, \\ C_2 &= \{p_{n_1+1}, p_{n_1+2}, \dots, p_{n_2}\}, \\ C_3 &= \{p_{n_2+1}, p_{n_2+2}, \dots, p_{n_3}\}, \\ &\vdots \\ C_k &= \{p_{(n_{k-1})+1}, p_{(n_{k-1})+2}, \dots, p_{n_k}\}, \end{aligned}$$

e considere uma solução ótima para o CS-LOSP reformulado contendo a frequência na qual os padrões de corte devem ser executados:

$$F = \{f_1, f_2, \dots, f_N\},$$

sendo  $f_i$  a quantidade de vezes que um objeto deve ser cortado utilizando o padrão de corte  $i$ . Considere os conjuntos  $C_i^* \subseteq C_i$ ,  $i = 1, 2, \dots, k$ , tais que:

$$\begin{aligned} C_1^* &= \{p_i, 1 \leq i \leq n_1 \mid f_i \neq 0\}, \\ C_2^* &= \{p_i, n_1 + 1 \leq i \leq n_2 \mid f_i \neq 0\}, \\ &\vdots \\ C_k^* &= \{p_i, n_{k-1} + 1 \leq i \leq n_k \mid f_i \neq 0\}. \end{aligned}$$

Em um CS-LOSP, não necessariamente todos os conjuntos  $C_i$  são usados. Sem perda de generalidade, suponha que em uma solução ótima para o CS-LOSP reformulado  $k_2$  conjuntos de padrões de corte do mesmo tipo sejam utilizados. A solução ótima de um CS-LOSP reformulado é dada pela frequência na qual os conjuntos de padrões de corte devem ser executados juntamente com uma sequência na qual os padrões de corte devem ser executados de forma que o número máximo de pilhas abertas seja menor que o limite preestabelecido. Seja essa sequência dada por:

$$C_{l_1} \rightarrow C_{l_2} \rightarrow C_{l_3} \rightarrow \dots C_{l_{k_2}},$$

os padrões de corte não utilizados podem ser retirados da sequência, resultando em:

$$C_{l_1}^* \rightarrow C_{l_2}^* \rightarrow C_{l_3}^* \rightarrow \dots C_{l_{k_2}}^*.$$

A Propriedade 1 garante que quando  $n$  padrões de corte pertencentes a um mesmo conjunto  $C_i$  são executados em sequência, a ordem na qual estes padrões de corte são executados pode ser permutada sem alterar o número máximo de pilhas abertas obtido através do sequenciamento. Basta então executar os padrões de corte pertencentes a  $C_{l_1}^*$  em qualquer ordem e, em seguida, executar aqueles pertencentes a  $C_{l_2}^*$  e assim em diante, até serem executados todos os padrões de corte contidos em  $C_{l_{k_2}}^*$ . Este sequenciamento de padrões de corte em conjunto com a frequência na qual os padrões de corte devem ser executados, dada por  $F$ , equivale a uma solução ótima para o CS-LOSP usual.

□

**Propriedade 4:** Seja uma solução ótima para o CS-LOSP usual contendo a frequência na qual os padrões de corte devem ser executados e um sequenciamento de padrões de corte factível. A partir desta solução é possível determinar um sequenciamento de conjuntos de padrões de corte, cuja execução abre um número de pilhas igual ou inferior que o sequenciamento dos padrões de corte, ou equivalentemente, a partir de uma solução ótima do CS-LOSP usual é possível obter uma solução ótima para o CS-LOSP reformulado.

*Demonstração.* Uma solução ótima para o CS-LOSP usual é composta da frequência na qual os padrões de corte devem ser executados e do sequenciamento destes padrões de corte de forma a abrir, no máximo,  $C$  pilhas, sendo  $C$  o limitante no número máximo de pilhas que podem ser abertas. Considere uma solução ótima do CS-LOSP usual. Para esta solução ótima, seja  $P$  o conjunto de padrões de corte utilizados na solução ótima, ou seja, os padrões de corte cuja frequência de execução é diferente de 0 e um sequenciamento de padrões de corte dado por  $S^*$  que, ao ser executado, abre, no máximo,  $C$  pilhas. Seja  $S$  um conjunto contendo todos os sequenciamentos possíveis para o conjunto  $P$ . Considere  $\bar{S} \subset S$  como o conjunto de todos os sequenciamentos possíveis dos padrões de corte contidos em  $P$ , nos quais todos os padrões de corte que geram os mesmos tipos de item são sempre executados em sequência. Caso  $S^* \subset \bar{S}^c$ ,  $S^*$  é dada por:

$$S^* = p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_N.$$

Seja  $C_1$  o conjunto contendo todos os padrões de corte que geram os mesmos tipos de item do padrão de corte  $p_1$ . Seja a cardinalidade de  $C_1$  igual a  $n_1$ . Logo na sequência  $S^*$ , no máximo, os primeiros  $n_1$  termos são pertencentes a  $C_1$ . Seja  $n_1^*$  o maior valor tal que,  $p_i \in C_1, \forall 1 \leq i \leq n_1^*$ . Por construção,  $p_{n_1^*+1} \notin C_1$ . Seja  $C_2$  o conjunto contendo todos os padrões de corte que geram os mesmos tipos de item do padrão de corte  $p_{n_1^*+1}$ . Seja a cardinalidade de  $C_2$  igual a  $n_2$ . Logo na sequência  $S^*$ , no máximo, os primeiros  $n_1 + n_2$  termos são pertencentes a  $C_1 \cup C_2$ . Seja o maior valor  $n_2^*$  tal que,  $p_i \in C_1 \cup C_2, \forall 1 \leq i \leq n_2^*$ . Por construção,  $p_{n_2^*+1} \notin C_1 \cup C_2$ . Suponha que existam  $K$  conjuntos distintos  $C_i$  tais que todos os padrões de corte pertencentes a  $C_i$  gerem os mesmos tipos de itens. Este processo pode ser repetido de forma a se obter todos conjuntos  $C_k, 1 \leq k \leq K$  e os valores  $n_k^*$ , de forma que,  $p_i \in \bigcup_{j \leq k} C_j, \forall 1 \leq i \leq n_k^*$ , e  $n_k^* \leq \sum_j^k n_j$ . Seja  $\bar{S}^* \in \bar{S}$  uma sequência, na qual todos os padrões de corte pertencentes a  $C_1$  são executados, seguidos da execução de todos os padrões de corte pertencentes a  $C_2$  e assim em diante, ou seja:

$$\bar{S}^* = C_1 \rightarrow C_2 \rightarrow \dots \rightarrow C_K.$$

Conforme foi demonstrado na Propriedade 2, o número máximo de pilhas abertas ao executar a sequência  $\bar{S}^*$  é igual ou menor que o número máximo de pilhas abertas ao executar a sequência  $S^*$ . Porém, como  $S^*$  é uma solução ótima do CS-LOSP então este sequenciamento é factível, portanto, no máximo,  $C$  pilhas são abertas ao executar esta sequência. Sendo assim, o número máximo de pilhas abertas ao executar  $\bar{S}^*$  é igual ou inferior a  $C$  e, portanto, este sequenciamento é também factível. Logo a frequência na qual os padrões de corte devem ser executados, juntamente com a sequência  $\bar{S}^*$ , representam uma solução ótima para o CS-LOSP reformulado. Caso  $S^* \subset \bar{S}$  então  $S^*$

representa uma das permutações possíveis da sequência  $\overline{S}^*$  e, como foi demonstrado na Propriedade 3, o número máximo de pilhas abertas por  $\overline{S}^*$  e  $S^*$  é o mesmo.  $\square$

As Propriedades 1, 2, 3 e 4 são suficientes para garantir a equivalência entre o CS-LOSP usual e reformulado. Sendo assim, de agora em diante, não será feita nenhuma distinção entre os dois problemas.

**Definição:** Seja  $S_1$  um sequenciamento de padrões de corte e  $S_2$  um sequenciamento de conjuntos de padrões de corte do mesmo tipo.  $S_1$  é equivalente a  $S_2$  se  $S_1$  é uma das permutações obtidas a partir de  $S_2$  utilizando a Propriedade 3 ou, se  $S_2$  é obtido a partir de  $S_1$ , quando a Propriedade 4 é aplicada.

**Observação:** Dado um sequenciamento de conjuntos de padrões de corte do mesmo tipo, existem diversas possibilidades para o sequenciamento de padrões de corte equivalente. Neste trabalho, sempre que foi necessário encontrar o sequenciamento de padrões de corte equivalente, foi considerada a permutação na qual padrões de corte pertencentes a um mesmo conjunto são executadas na ordem que os padrões de corte foram gerados inicialmente.

(Exemplo IV)

Considere o conjunto de padrões de corte  $P$  e os conjuntos  $C_i$  dados no **Exemplo III**, então as sequências de padrões de corte:

$$p_2 \rightarrow p_3 \rightarrow p_5 \rightarrow p_1 \rightarrow p_7$$

$$p_3 \rightarrow p_2 \rightarrow p_5 \rightarrow p_1 \rightarrow p_7$$

$$p_2 \rightarrow p_1 \rightarrow p_7$$

$$p_2 \rightarrow p_3 \rightarrow p_1 \rightarrow p_5 \rightarrow p_7$$

e a sequência de conjuntos de padrões de corte do mesmo tipo

$$C_2 \rightarrow C_1 \rightarrow C_3$$

são equivalentes.

Como veremos a seguir, sequenciar os conjuntos  $C_i$  é mais eficiente do que sequenciar os padrões de corte, tomando como base os modelos **YLI** e **CI2** e as Propriedades 1, 2, 3 e 4, dois novos modelos podem ser construídos. A construção das novas formulações



dependem da definição de novos índices e parâmetros. Seja  $P$  um conjunto de padrões de corte, com cardinalidade igual a  $N$ , contendo  $N_2$  tipos distintos de padrões de corte. Os padrões de corte contidos em  $P$  podem ser divididos em  $N_2$  subconjuntos de  $P$ , de forma que, padrões de corte do mesmo tipo pertençam ao mesmo subconjunto. Sejam  $C_i$ ,  $i = 1, \dots, N_2$  subconjuntos de  $P$ , tais que todos os padrões de corte pertencentes a  $P$  que pertençam a um mesmo tipo estejam contidos em  $C_i$ , para algum  $i = 1, \dots, N_2$  e se  $i \neq j \iff C_i \neq C_j$ . Considere:

### Índices:

$N$  : quantidade de padrões de corte distintos;

$M$  : quantidade de tipos distintos de itens;

$N_2$  : quantidade de conjuntos  $C_i$  distintos, incluindo o conjunto fictício  $C_1$ .

### Parâmetros:

$L_i$  : conjunto contendo os índices dos padrões de corte contidos no conjunto  $C_i$ ;

$J'_2$  : conjunto contendo todos os conjuntos  $C_i$ , inclusive o conjunto fictício;

$J_2^t$  : conjunto contendo os conjuntos de padrões de corte do mesmo tipo que geram o item  $t$ ;

$J_2$  : conjunto contendo os conjuntos  $C_i$ , excluindo o conjunto fictício;

$J'$  : conjunto contendo todos os padrões de corte, inclusive o padrão de corte fictício;

$J$  : conjunto contendo todos os padrões de corte, excluindo o padrão de corte fictício;

$T$  : conjunto contendo todos os tipos distintos de itens;

$T_i$  : conjunto de itens gerados ao executar um padrão de corte pertencente ao conjunto  $C_i$ ;

$K$  : constante dada por (4.4);

$d_i$  : demanda do item do tipo  $i$ .

Os índices e parâmetros definidos nesta seção são utilizados para a construção dos dois modelos propostos neste capítulo. Sendo um deles baseado no modelo YLI e o outro baseado no modelo CI2. A formulação resultante da aplicação das melhorias no modelo de YLI é denominado de Yanasse e Lamosa Integrado-Modificado (YLI-Modificado) e o modelo baseado em CI2 é denominado de Catanzaro Integrado II-Modificado (CI2-Modificado). Ambos os modelos são apresentados nas seções subsequentes.

## 7.1 Yanasse e Lamosa Integrado-Modificado

O primeiro passo para aplicar as melhorias propostas neste capítulo na formulação YLI consiste em alterar o significado das variáveis  $w_{ij}$ . Esta alteração se faz necessária uma vez que ao invés de sequenciar padrões de corte como foi feito em YLI, nosso objetivo nesta seção está no sequenciamento de conjuntos de padrões de corte do mesmo tipo, sendo assim, considere as variáveis:

$$w_{ij} = \begin{cases} 1, & \text{se o conjunto de padrões de corte do mesmo tipo } C_i \text{ é o } j\text{-ésimo} \\ & \text{a ser processado,} \\ 0, & \text{caso contrário.} \end{cases}$$

Note que, se ao menos um padrão de corte pertencente a  $C_i$ , para algum  $i$ , for utilizado, então o conjunto de padrões de corte do mesmo tipo  $C_i$  deve ser executado em algum momento no sequenciamento desta solução, sendo assim, as restrições (4.11) e (4.12) também devem ser modificadas. Assim, considere:

$$K \left( \sum_{j=1}^{N_2} w_{ij} \right) \geq \left( \sum_{k \in L_i} v_k \right), \quad i = 1, \dots, N_2 \quad (7.1)$$

$$\sum_{j=1}^{N_2} w_{ij} \leq \left( \sum_{k \in L_i} v_k \right), \quad i = 1, \dots, N_2 \quad (7.2)$$

$$\sum_{j=1}^{N_2} w_{ij} \leq 1, \quad i = 1, \dots, N_2 \quad (7.3)$$

$$K \left( \sum_{i=1}^{N_2} w_{ij} \right) \geq \left( \sum_{k \in L_j} v_k \right), \quad j = 1, \dots, N_2 \quad (7.4)$$

$$\sum_{i=1}^{N_2} w_{ij} \leq \left( \sum_{k \in L_j} v_k \right), \quad j = 1, \dots, N_2 \quad (7.5)$$

$$\sum_{i=1}^{N_2} w_{ij} \leq 1, \quad j = 1, \dots, N_2 \quad (7.6)$$

Os conjuntos de restrições (7.1) e (7.2) fazem com que  $\sum_{j=1}^{N_2} w_{ij} = 0$  se e somente se  $\sum_{k \in L_i} v_k = 0$ , desde que o valor da constante  $K$  seja suficientemente grande. Sendo assim, o conjunto de padrões de corte  $C_i$  é sequenciado se e somente se algum padrão de corte pertencente a  $C_i$  é utilizado. Dado que  $w_{ij}$  é uma variável binária para todo  $i, j \in \{1, \dots, N_2\}$ , o conjunto de restrições (7.3) garante que, caso algum padrão de corte pertencente ao conjunto de padrões de corte do mesmo tipo  $C_i$  seja utilizado, este conjunto de padrões de corte do mesmo tipo seja o  $j$ -ésimo a ser processado, para um único  $j \in \{1, \dots, N_2\}$ . De forma análoga, os conjuntos de restrições (7.4) e (7.5) garantem que  $\sum_{i=1}^{N_2} w_{ij} = 0$  se e somente se  $\sum_{k \in L_j} v_k = 0$ , desde que o valor da constante  $K$  seja suficientemente grande. Portanto, caso nenhum padrão de corte pertencente ao conjunto de padrões de corte do mesmo tipo  $C_j$  for utilizado, então nenhum conjunto de padrões de corte é executado no instante  $j$ . Desta forma, este instante pode ser ignorado no sequenciamento. Uma vez que  $w_{ij}$  é uma variável binária para todo  $i, j \in \{1, \dots, N_2\}$ , o conjunto de restrições (7.6) garante que, no máximo, um conjunto de padrões de corte do mesmo tipo seja executado no instante  $j$ , para todo  $j \in \{1, \dots, N_2\}$ . Note que, uma vez que existem padrões de corte que podem não ser utilizados na solução final do problema, é permitido que existam instantes  $j$  nos quais nenhum padrão de corte é executado, dessa maneira, no sequenciamento ótimo é necessário ignorar os instantes nos quais nenhum conjunto de padrões de corte é utilizado.

A última modificação necessária no modelo está no conjunto de restrições (4.13). Considerando os novos parâmetros estabelecidos nesta seção, as restrições (4.13) são dadas por:

$$w_{ij} + (v_j - 1) \leq y_{jt}, \quad i \in J_2^t, j = 1, \dots, N_2, t \in T.$$

Note que, caso um conjunto de padrões de corte do mesmo tipo  $i$  não seja utilizado então  $w_{ij} = 0$ ,  $j \in \{1, \dots, N_2\}$ . Sendo assim, o conjunto de restrições (4.13) se torna trivial independentemente do valor de  $v_j$ . Portanto, o termo entre parênteses pode ser eliminado, resultando em:

$$w_{ij} \leq y_{jt}, \quad i \in J_2^t, j = 1, \dots, N_2, t \in T.$$

Desta maneira, consideramos as variáveis de decisão dadas a seguir:

**Variáveis de decisão:**

$p_j$  : quantidade de vezes que o padrão de corte  $j$  é utilizado;

$z_{it}$  : variável binária utilizada para contabilizar o número de pilhas abertas. No contexto do MTSP,  $z_{it}$  é igual a 1 se e somente se a ferramenta  $t$  é inserida na máquina para o processamento da tarefa  $i$ ;

$$w_{ij} = \begin{cases} 1, & \text{se o conjunto de padrões de corte do mesmo tipo } C_i \text{ é o } j\text{-ésimo a} \\ & \text{ser processado,} \\ 0, & \text{caso contrário;} \end{cases}$$

$$y_{it} = \begin{cases} 1, & \text{se o item } t \text{ é obtido ao utilizar o algum padrão de corte pertencente} \\ & \text{ao conjunto de padrões de corte do mesmo tipo } C_i, \\ 0, & \text{caso contrário;} \end{cases}$$

$$v_j = \begin{cases} 1, & \text{se o padrão de corte } j \text{ é utilizado na solução do problema,} \\ 0, & \text{caso contrário.} \end{cases}$$

O modelo de Yanasse e Lamosa Integrado-Modificado (YLI-Modificado) é dado por:

(YLI-Modificado)

$$\min \sum_{j=1}^N c_j p_j \quad (7.7)$$

$$\text{sujeito a: } \sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i = 1, \dots, M \quad (7.8)$$

$$\sum_{i=2}^{N_2} \sum_{t \in T} z_{it} \leq M - C, \quad (7.9)$$

$$v_j \leq p_j, \quad j = 1, \dots, N \quad (7.10)$$

$$K v_j \geq p_j, \quad j = 1, \dots, N \quad (7.11)$$

$$K \left( \sum_{j=1}^{N_2} w_{ij} \right) \geq \left( \sum_{k \in L_i \setminus} v_k \right), \quad i = 1, \dots, N_2 \quad (7.12)$$

$$\sum_{j=1}^{N_2} w_{ij} \leq \left( \sum_{k \in L_i \setminus} v_k \right), \quad i = 1, \dots, N_2 \quad (7.13)$$

$$\sum_{j=1}^{N_2} w_{ij} \leq 1, \quad j = 1, \dots, N_2 \quad (7.14)$$

$$K \left( \sum_{i=1}^{N_2} w_{ij} \right) \geq \left( \sum_{k \in L_j} v_k \right), \quad j = 1, \dots, N_2 \quad (7.15)$$

$$\sum_{i=1}^{N_2} w_{ij} \leq \left( \sum_{k \in L_j} v_k \right), \quad j = 1, \dots, N_2 \quad (7.16)$$

$$\sum_{i=1}^{N_2} w_{ij} \leq 1, \quad j = 1, \dots, N_2 \quad (7.17)$$

$$w_{ij} \leq y_{jt}, \quad i \in J_2^t, j = 1, \dots, N_2, t \in T \quad (7.18)$$

$$\sum_{t \in T} y_{jt} \leq C, \quad j = 1, \dots, N_2 \quad (7.19)$$

$$y_{jt} - y_{(j-1)t} \leq z_{jt}, \quad i = 2, \dots, N_2, t \in T \quad (7.20)$$

$$p_j \in Z^+, \quad j = 1, \dots, N \quad (7.21)$$

$$v_j \in \{0, 1\}, \quad j = 1, \dots, N \quad (7.22)$$

$$w_{ij} \in \{0, 1\}, \quad i = 1, \dots, N_2, j = 1, \dots, N_2 \quad (7.23)$$

$$y_{jt} \in \{0, 1\}, \quad j = 1, \dots, N_2, t \in T \quad (7.24)$$

$$z_{jt} \in \{0, 1\}, \quad j = 1, \dots, N_2, t \in T. \quad (7.25)$$

A função objetivo (7.7) é similar à considerada em (2.12). Esta função objetivo minimiza o desperdício de material. De forma análoga, as restrições (7.8) equivalem ao conjunto de restrições (2.13), para garantir que a demanda requerida seja satisfeita. O conjunto de restrições (7.9) garante que o número máximo de pilhas abertas seja igual

a  $C$ , este conjunto de restrições é utilizado para transformar um MTSP em um MOSP conforme foi discutido na Seção 3.3. As restrições (7.10) e (7.11) garantem que  $v_j = 0$  se e somente se  $p_j = 0$ . As restrições (7.12)-(7.17) foram explicadas anteriormente nesta seção. O conjunto de restrições (7.18) garante que, caso um padrão de corte pertencente ao conjunto de padrões de corte  $C_i$  seja utilizado, então todas as pilhas abertas a partir da execução deste conjunto devem ser levadas em conta, do contrário a restrição se torna trivial. As restrições (7.19) impedem que o limite da capacidade de pilhas abertas seja ultrapassado. O conjunto de restrições (7.20) assegura a contabilização do número de pilhas abertas através das variáveis  $z_{kt}$  sempre que necessário. As restrições (7.21)-(7.23) definem o domínio das variáveis.

## 7.2 Catanzaro Integrado II-Modificado

Para aplicar as melhorias propostas neste capítulo no modelo CI2, o procedimento utilizado é o mesmo da seção anterior. Ou seja, primeiramente deve-se alterar a interpretação das variáveis de decisão, considere:

$$x_{ij} = \begin{cases} 1, & \text{se a execução do conjunto de padrões de corte do mesmo tipo } C_j \text{ é feita} \\ & \text{imediatamente após a execução do conjunto } C_i, \\ 0, & \text{caso contrário.} \end{cases}$$

Em seguida, os conjuntos de restrições (5.41) e (5.42) devem ser substituídos pelas restrições (7.1)-(7.6), adaptando os índices e parâmetros para o modelo de Catanzaro Integrado II. Além disso, é necessário adicionar as restrições a seguir:

$$\sum_{i \in J'_2} x_{i1} = 1, \quad (7.26)$$

$$\sum_{j \in J'_2} x_{1j} = 1, \quad (7.27)$$

$$v_1 = 1. \quad (7.28)$$

Os conjuntos de restrições (7.26)-(7.28) garantem que o conjunto de padrões de corte fictício seja utilizado. Considerando as variáveis de decisão dadas por:

**Variáveis de decisão:**

$p_j$  : quantidade de vezes que o padrão de corte  $j$  é utilizado;

$$x_{ij} = \begin{cases} 1, & \text{se a execução do conjunto de padrões de corte do mesmo tipo } C_j \\ & \text{é feita imediatamente após a execução do conjunto } C_i, \\ 0, & \text{caso contrário;} \end{cases}$$

$y_{ij}^t$  : variável binária utilizada na contabilização do número de pilhas abertas;

$$v_j = \begin{cases} 1, & \text{se o padrão de corte } j \text{ é utilizado na solução do problema,} \\ 0, & \text{caso contrário;} \end{cases}$$

$u_i$  : variável auxiliar usada para impedir a formação de subciclos, conforme proposto em [Miller, Tucker e Zemlin \(1960\)](#).

O modelo [CI2-Modificado](#) é dado então por:

(CI2-Modificado)

$$\min \sum_{j=1}^N p_j \tag{7.29}$$

$$\text{sujeito a: } \sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i \in T \tag{7.30}$$

$$\sum_{\substack{i,j \in J \\ i \neq j}} \sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \leq M - C, \tag{7.31}$$

$$\sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \geq 0, \quad i, j \in J_2, i \neq j \tag{7.32}$$

$$v_j \leq p_j, \quad j \in J \tag{7.33}$$

$$K v_j \geq p_j, \quad j \in J \tag{7.34}$$

$$\sum_{i \in J'_2 \setminus \{1\}} x_{i1} = 1, \tag{7.35}$$

$$\sum_{i \in J'_2 \setminus \{1\}} x_{1i} = 1, \quad (7.36)$$

$$K\left(\sum_{j \in J'_2 \setminus \{i\}} x_{ij}\right) \geq \left(\sum_{k \in L_i \setminus} v_k\right), \quad i \in J_2 \quad (7.37)$$

$$\sum_{j \in J'_2 \setminus \{i\}} x_{ij} \leq \left(\sum_{k \in L_i \setminus} v_k\right), \quad i \in J_2 \quad (7.38)$$

$$\sum_{j \in J'_2 \setminus \{i\}} x_{ij} \leq 1, \quad j \in J_2 \quad (7.39)$$

$$K\left(\sum_{i \in J'_2 \setminus \{j\}} x_{ij}\right) \geq \left(\sum_{k \in L_j} v_k\right), \quad j \in J_2 \quad (7.40)$$

$$\sum_{i \in J'_2 \setminus \{j\}} x_{ij} \leq \left(\sum_{k \in L_j} v_k\right), \quad j \in J_2 \quad (7.41)$$

$$\sum_{i \in J'_2 \setminus \{j\}} x_{ij} \leq 1, \quad j \in J_2 \quad (7.42)$$

$$v_1 = 1, \quad (7.43)$$

$$1 \leq u_i \leq N_2, \quad i \in J_2 \quad (7.44)$$

$$u_i - u_j + N_2 * x_{ij} \leq N_2 - 1, \quad i, j \in J_2, i \neq j \quad (7.45)$$

$$\sum_{k \in J'_2 \setminus \{i\}} y_{ki}^t - \sum_{j \in J'_2 \setminus \{i\}} y_{ij}^t \geq 0, \quad i \in J'_2, t \in T \setminus T_i \quad (7.46)$$

$$\sum_{\substack{t \in T: \\ t \in T_i \\ t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T: \\ t \notin T_i \\ t \notin T_j}} y_{ij}^t \leq (C - |T_j|)x_{ij}, \quad i, j \in J'_2, i \notin j \quad (7.47)$$

$$y_{ij}^t \leq x_{ij}, \quad i \in J_2, t \in T \setminus T_j \quad (7.48)$$

$$x_{ij} \in \{0, 1\}, \quad i \in J'_2, j \in J'_2, i \neq j \quad (7.49)$$

$$y_{ij}^t \in \{0, 1\}, \quad i, j \in J'_2, i \neq j, t \in (T \setminus T_i) \cup (T_i \setminus T_j) \quad (7.50)$$

$$v_j \in \{0, 1\}, \quad j \in J' \quad (7.51)$$

$$p_j \in Z^+, \quad j \in J' \quad (7.52)$$

$$u_i \in Z^+, \quad i \in J_2. \quad (7.53)$$

A função objetivo (7.29) é similar à considerada em (2.12). Esta função objetivo minimiza o desperdício de material. De forma análoga, as restrições (7.30) equivalem ao conjunto de restrições (2.13) que garantem o atendimento da demanda. O conjunto de restrições (7.31) garante que o número máximo de pilhas abertas seja igual a  $C$ , este conjunto de restrições é utilizado para transformar um MTSP em um MOSP conforme foi discutido na Seção 3.3. O conjunto de restrições (7.32) elimina a presença de sequenciamentos ineficazes. Este conjunto foi proposto em (5.33). As restrições (7.33) e (7.34) garantem que  $v_j = 0$  se e somente se  $p_j = 0$ . Note que, uma vez que as restrições (7.33) e (7.34) só estão definidas para  $j \in J$ , a variável  $p_1$  é livre para assumir qualquer valor pois,



como a função objetivo busca minimizar a soma das variáveis  $p_j$ , o valor de  $p_1$  sempre será igual a zero, ou seja o padrão de corte fictício não interfere na contagem de objetos necessários para satisfazer a demanda.

A restrição (7.43) garante que o padrão de corte fictício seja utilizado. Os conjuntos de restrições (7.35)-(7.42) foram explicados anteriormente. O conjunto de restrições (7.44) se baseia nas restrições propostas em Miller, Tucker e Zemlin (1960), este conjunto de restrições é apresentado em (5.24) e juntamente com as restrições (7.45) impedem a formação de subciclos. As restrições (7.45) foram também propostas em Miller, Tucker e Zemlin (1960) e são apresentadas em mais detalhes na Seção 3.6. O conjunto de restrições (7.47) garante que o limite máximo no número de pilhas abertas não seja ultrapassado.

As restrições (7.46) e (7.48) foram elaboradas em Catanzaro, Golveia e Labbé (2015) para o MTSP, neste contexto, as restrições (7.46) impõem que, se a ferramenta  $t$  é carregada na transição da tarefa  $i$  para alguma tarefa  $j$ , então esta ferramenta deve ter sido carregada ou inserida na transição de alguma tarefa  $k$  para a tarefa  $i$ ,  $i \neq k$ . O conjunto de restrições (7.48) permite que uma ferramenta  $t$  pertencente a  $T \setminus T_j$  seja carregada na transição das tarefas  $i$  e  $j$ , se o processamento da tarefa  $j$  ocorre logo após o processamento da tarefa  $i$  em uma solução ótima. Como foi demonstrado em Yanasse (1997), uma solução ótima do MTSP, que realiza um total de  $M - C$  trocas de ferramentas, representa um sequenciamento para o MOSP no qual, no máximo,  $C$  pilhas são abertas simultaneamente, quando consideramos as ferramentas como itens e as tarefas como padrões de corte. Embora os conjuntos de restrições (7.46) e (7.48) não possuam uma interpretação bem definida para o contexto do CS-LOSP, elas são essenciais para garantir o funcionamento do modelo, uma vez que, em conjunto com as restrições (7.31), asseguram que a limitação no número máximo de pilhas que podem ser abertas não seja ultrapassado. As restrições (7.49)-(7.53) são referentes ao domínio das variáveis.

### 7.3 Quantidade de variáveis

Como foi visto na Seção 6.2, a quantidade de variáveis e restrições presentes nos modelos YLI e CI2 dependem fortemente do parâmetro  $N$ , ou seja, a quantidade de padrões de corte presentes no problema. Ao sequenciar os conjuntos de padrões de corte invés dos padrões de corte em si, o tamanho dos modelos é reduzido significativamente, uma vez que a quantidade de conjuntos  $C_i$  distintos, dada por  $N_2$ , tende a ser consideravelmente menor que  $N$ . Nesta seção são estudados os impactos das modificações propostas neste capítulo em relação a quantidade de variáveis presentes na formulação. Para tal fim, são comparados os modelos CI2-Modificado e CI2 e os modelos YLI-Modificado e YLI. De forma similar ao que foi feito na Seção 6.2, considere os conjuntos:

$$P = \{p_j, j \in J\}$$

$$V = \{v_j, j \in J\}$$

$$W = \{w_{jk}, j, k \in J_2\}$$

$$Z = \{z_{kt}, k \in J_2 \setminus \{1\}, t \in T\}$$

$$Y = \{y_{kt}, k \in J_2, t \in T\}$$

Considerando o modelo [YLI-Modificado](#), temos que  $J$  se refere ao conjunto contendo todos os conjuntos  $C_i$ , ou seja, os conjuntos de padrões de corte do mesmo tipo  $i$  e  $K$  a ordem na qual estes conjuntos são executados, logo, temos  $|J| = |K| = N_2$ , já  $T$  contém os tipos distintos de itens e, portanto,  $|T| = M$ . Sendo assim, temos:

- $|P| = N$
- $|V| = N$
- $|W| = N_2^2$
- $|Z| = (N_2 - 1)M$
- $|Y| = N_2M$

Somente o conjunto de variáveis referentes a  $P$  contém variáveis inteiras, os demais conjuntos possuem variáveis binárias. Portanto, temos  $N$  variáveis inteiras e  $2MN_2 + N_2^2 + N - M$  variáveis binárias implicando na existência de  $2MN_2 + N_2^2 + 2N - M$  variáveis no total. A quantidade de variáveis presentes nos conjuntos  $V$  e  $P$  é a mesma para os dois modelos, a diferença entre eles está nos conjuntos  $W, Y$  e  $Z$ . É justamente nestes conjuntos que está presente a grande maioria das variáveis da formulação [YLI](#). É evidente que  $N \geq N_2$ , portanto, o número de variáveis presentes em [YLI-Modificado](#) é igual ou inferior a quantidade de variáveis presentes em [YLI](#). Mas, na prática, o valor de  $N_2$  pode ser muito menor que  $N$  resultando em uma redução significativa no número total de variáveis.

Para auxiliar na contagem de variáveis do modelo [CI2-Modificado](#), considere os seguintes conjuntos:

$$P = \{p_j, j \in J\}$$

$$V = \{v_j, j \in J\}$$

$$W = \{w_{jk}, j \in J'_2, k \in J'_2, k \neq j\}$$

$$Y = \{y_{jk}^t, j \in J'_2, k \in J'_2, j \neq k, (t \in T \setminus T_i) \cup (T \in T_i, t \notin T_j)\}$$

$$U = \{u_i, i = 2, \dots, N_2 + 1\}$$

a cardinalidade dos conjuntos é dada por:

- $|P| = N + 1$
- $|V| = N + 1$
- $|W| = (N_2 + 1)^2 - (N_2 + 1)$
- $|Y| = \sum_{\substack{i,k \in J'_2 \\ i \neq j}} |(T \setminus T_i) \cup (T_i \setminus T_k)|$
- $|U| = N_2$

Portanto, há um total de  $2(N + 1) + (N_2 + 1)^2 - (N_2 + 1) + \sum_{\substack{i,k \in J'_2 \\ i \neq j}} |(T \setminus T_i) \cup (T_i \setminus T_k)| + N_2$  variáveis. Enquanto para o modelo **CI2** a quantidade de variáveis é dada por:  $(N + 1) + (N + 1)^2 + \sum_{\substack{i,k \in J' \\ i \neq j}} |(T \setminus T_i) \cup (T_i \setminus T_k)| + N$ . As modificações propostas neste capítulo resultam em uma redução significativa no número de variáveis para os modelos **YLI-Modificado** e **CI2-Modificado** quando comparados às formulações **YLI** e **CI2**. Para ilustrar, apresentamos na **Tabela 29** a quantidade de variáveis nos modelos **YLI-Modificado**, **YLI**, **CI2-Modificado** e **CI2** para as primeiras cinco instâncias das Classes 1, 3 e 5.

Classe	Nº de variáveis YLI	Nº de variáveis YLI-Modificado	Nº de variáveis CI2	Nº variáveis CI2-Modificado
1	$705,88 \times 10^2$	$177,18 \times 10$	$275,17 \times 10^3$	$508,48 \times 10$
3	$266,46 \times 10^2$	$181,15 \times 10^2$	$245,08 \times 10^3$	$161,76 \times 10^3$
5	$147,82 \times 10$	$145,92 \times 10$	$967,56 \times 10$	$951,54 \times 10$

Tabela 29 – Comparação entre o número de variáveis presentes nos modelos **YLI**, **YLI-Modificado**, **CI2** e **CI2-Modificado**. A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 1, 3 e 5 propostas neste trabalho.

A partir dos resultados apresentados é possível perceber que o número de variáveis nos modelos **CI2-Modificado** e **YLI-Modificado** é consideravelmente menor que a quantidade de variáveis nos modelos **CI2** e **YLI**. O impacto das modificações é especialmente perceptível nas classes com itens de menor comprimento. Para a Classe 1, o número de variáveis foi drasticamente reduzido, enquanto para a Classe 5, com itens maiores, a redução é insignificante.

## 7.4 Quantidade de restrições

Como foi visto na Seção 7.3 sequenciar os conjuntos de padrões de corte invés dos padrões de corte em si reduz significativamente a quantidade de variáveis. Uma vez

que o número total de restrições dos modelos [CI2](#) e [YLI](#) também depende do parâmetro  $N$ , é esperado que a quantidade de restrições presentes nas formulações [CI2-Modificado](#) e [YLI-Modificado](#) também seja inferior à quantidade de restrições presentes nos modelos [CI2](#) e [YLI](#). Nesta seção é comparada a quantidade de restrições presentes nos modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#). Não são contabilizadas as restrições referentes ao domínio das variáveis. Considere as restrições presentes no modelo [YLI-Modificado](#) dadas por:

$$\sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i = 1, \dots, M \quad (7.8)$$

$$\sum_{i=2}^{N_2} \sum_{t \in T} z_{it} \leq M - C, \quad (7.9)$$

$$v_j \leq p_j, \quad j = 1, \dots, N \quad (7.10)$$

$$K v_j \geq p_j, \quad j = 1, \dots, N \quad (7.11)$$

$$K \left( \sum_{j=1}^{N_2} w_{ij} \right) \geq \left( \sum_{k \in L_i \setminus} v_k \right), \quad i = 1, \dots, N_2 \quad (7.12)$$

$$\sum_{j=1}^{N_2} w_{ij} \leq \left( \sum_{k \in L_i \setminus} v_k \right), \quad i = 1, \dots, N_2 \quad (7.13)$$

$$\sum_{j=1}^{N_2} w_{ij} \leq 1, \quad j = 1, \dots, N_2 \quad (7.14)$$

$$K \left( \sum_{i=1}^{N_2} w_{ij} \right) \geq \left( \sum_{k \in L_j} v_k \right), \quad j = 1, \dots, N_2 \quad (7.15)$$

$$\sum_{i=1}^{N_2} w_{ij} \leq \left( \sum_{k \in L_j} v_k \right), \quad j = 1, \dots, N_2 \quad (7.16)$$

$$\sum_{i=1}^{N_2} w_{ij} \leq 1, \quad j = 1, \dots, N_2 \quad (7.17)$$

$$w_{ij} \leq y_{jt}, \quad i \in J_2^t, j = 1, \dots, N_2, t \in T \quad (7.18)$$

$$\sum_{t \in T} y_{jt} \leq C, \quad j = 1, \dots, N_2 \quad (7.19)$$

$$y_{jt} - y_{(j-1)t} \leq z_{jt}, \quad i = 2, \dots, N_2, t \in T \quad (7.20)$$

$$p_j \in Z^+, \quad j = 1, \dots, N \quad (7.21)$$

$$v_j \in \{0, 1\}, \quad j = 1, \dots, N \quad (7.22)$$

$$w_{ij} \in \{0, 1\}, \quad i = 1, \dots, N_2, j = 1, \dots, N_2 \quad (7.23)$$

$$y_{jt} \in \{0, 1\}, \quad j = 1, \dots, N_2, t \in T \quad (7.24)$$

$$z_{jt} \in \{0, 1\}, \quad j = 1, \dots, N_2, t \in T. \quad (7.25)$$

O conjunto de restrições (7.8) tem  $M$  restrições. Os conjuntos de restrições (7.10) e (7.11) possuem  $N$  restrições cada. Os conjuntos de restrições (7.12)-(7.17) possuem  $N_2$  restrições cada, assim como o conjunto de restrições (7.19). O conjunto de restrições (7.20) possui cardinalidade igual a  $(N_2 - 1)M$ . Há um total de  $(\sum_{t \in T} |J_2^t|)N_2$  restrições no conjunto (7.18). Por último, devem ser levadas em conta a restrição (7.9). Somando a cardinalidade de todos os conjuntos de restrições há um total de:

$$2N + M + 7N_2 + (N_2 - 1)M + (\sum_{t \in T} |J_2^t|)N_2 + 1 =$$

$$2N + N_2(7 + M + \sum_{t \in T} |J_2^t|) + 1$$

De forma análoga, considere as restrições presentes no modelo [CI2-Modificado](#) dadas por:

$$\sum_{j=1}^N \alpha_i^{(j)} p_j \geq d_i, \quad i \in T \quad (7.30)$$

$$\sum_{\substack{i,j \in J \\ i \neq j}} \sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \leq M - C, \quad (7.31)$$

$$\sum_{\substack{t \in T: \\ t \notin T_i \\ t \in T_j}} (x_{ij} - y_{ij}^t) \geq 0, \quad i, j \in J_2, i \neq j \quad (7.32)$$

$$v_j \leq p_j, \quad j \in J \quad (7.33)$$

$$K v_j \geq p_j, \quad j \in J \quad (7.34)$$

$$\sum_{i \in J_2' \setminus \{i\}} x_{i1} = 1, \quad i \in J_2' \quad (7.35)$$

$$\sum_{i \in J_2' \setminus \{j\}} x_{1i} = 1, \quad i \in J_2' \quad (7.36)$$

$$K * (\sum_{j \in J_2' \setminus \{i\}} x_{ij}) \geq (\sum_{k \in L_i \setminus \{i\}} v_k), \quad i \in J_2 \quad (7.37)$$

$$\sum_{j \in J_2' \setminus \{i\}} x_{ij} \leq (\sum_{k \in L_i \setminus \{i\}} v_k), \quad i \in J_2 \quad (7.38)$$

$$\sum_{i \in J'_2 \setminus \{i\}} x_{ij} \leq 1, \quad j \in J_2 \quad (7.39)$$

$$K * \left( \sum_{i \in J'_2 \setminus \{i\}} x_{ij} \right) \geq \left( \sum_{k \in L_k \setminus \{i\}} v_k \right), \quad j \in J_2 \quad (7.40)$$

$$\sum_{i \in J'_2 \setminus \{i\}} x_{ij} \leq \left( \sum_{k \in L_k \setminus \{i\}} v_k \right), \quad j \in J_2 \quad (7.41)$$

$$\sum_{i \in J'_2 \setminus \{i\}} x_{ij} \leq 1, \quad j \in J_2 \quad (7.42)$$

$$v_1 = 1, \quad (7.43)$$

$$1 \leq u_i \leq N_2, \quad i \in J_2 \quad (7.44)$$

$$u_i - u_j + N_2 * x_{ij} \leq N_2 - 1, \quad i, j \in J_2, i \neq j \quad (7.45)$$

$$\sum_{k \in J'_2 \setminus \{i\}} y_{ki}^t - \sum_{j \in J'_2 \setminus \{i\}} y_{ij}^t \geq 0, \quad i \in J'_2, t \in T \setminus T \quad (7.46)$$

$$\sum_{\substack{t \in T: \\ t \in T_i \\ t \notin T_j}} y_{ij}^t + \sum_{\substack{t \in T: \\ t \notin T_i \\ t \notin T_j}} y_{ij}^t \leq (C - |T_j|)x_{ij}, \quad i, j \in J'_2, i \notin j \quad (7.47)$$

$$y_{ij}^t \leq x_{ij}, \quad i, j \in J'_2, i \neq j, t \in T \setminus T_j \quad (7.48)$$

$$x_{ij} \in \{0, 1\}, \quad i \in J'_2, j \in J'_2, i \neq j \quad (7.49)$$

$$y_{ij}^t \in \{0, 1\}, \quad i, j \in J_2, i \neq j, t \in (T \setminus T_i) \cup (T_i \setminus T_j) \quad (7.50)$$

$$v_j \in \{0, 1\}, \quad j \in J' \quad (7.51)$$

$$p_j \in Z^+, \quad j \in J' \quad (7.52)$$

$$u_i \in Z^+, \quad i \in J_2 \quad (7.53)$$

O conjunto de restrições (7.30) possui  $M$  restrições, já os conjuntos (7.33) e (7.34) possuem  $N$  restrições cada. Há um total de  $N_2$  restrições em cada um dos conjuntos (7.37), (7.38), (7.39), (7.40), (7.41), (7.42) e (7.44), enquanto os conjuntos (7.35) e (7.36) possuem  $(N_2 + 1)$  restrições cada um. Os conjuntos de restrições (7.32) e (7.45) possuem  $N_2(N_2 - 1)$  restrições cada, enquanto o conjunto de restrições (7.46) possui  $\sum_{i \in J'_2} |T \setminus T_i|$  restrições no total. O conjunto de restrições (7.47) possui  $(N_2 + 1)^2 - (N_2 + 1)$  restrições, a cardinalidade do conjunto (7.48) é igual a  $\sum_{\substack{i, j \in J'_2 \\ i \neq j}} |T \setminus T_j|$ , os conjuntos de restrições (7.49)-(7.53) são referentes ao domínio das variáveis e, portanto, não são consideradas na contagem das restrições. Por último devem ser consideradas as restrições (7.31) e (7.43). Logo, o total de restrições na formulação **CI2-Modificado** é dado por:

$$M + 2N + 7N_2 + 2 * (N_2 + 1) + 2N_2(N_2 - 1) + \sum_{i \in J'_2} |T \setminus T_i| + ((N_2 + 1)^2 - (N_2 + 1)) + \sum_{\substack{i, j \in J'_2 \\ i \neq j}} |T \setminus T_j| + 2$$

$$= M + 2N + 8N_2 + 3N_2^2 + \sum_{i \in J'_2} |T \setminus T_i| + \sum_{\substack{i, j \in J'_2 \\ i \neq j}} |T \setminus T_j| + 4.$$

O valor de  $N_2$  pode ser significativamente inferior ao valor de  $N$ . Por esta razão, o número de restrições presentes nos modelos [CI2-Modificado](#) e [YLI-Modificado](#) tende a ser menor do que o número de restrições presentes nas formulações [CI2](#) e [YLI](#). Apresentamos na [Tabela 30](#) a comparação entre a quantidade de restrições presentes nos modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#) para a primeira instância das Classes 1, 3 e 5.

Classe	Nº de restrições YLI	Nº de restrições YLI-Modificado	Nº de restrições CI2	Nº de restrições CI2-Modificado
1	$214,34 \times 10^3$	$335,38 \times 10$	$264,49 \times 10^3$	$626,38 \times 10$
3	$601,91 \times 10^2$	$412,29 \times 10^2$	$227,12 \times 10^3$	$165,57 \times 10^3$
5	$187,32 \times 10$	$195,30 \times 10$	$964,52 \times 10$	$105,07 \times 10^2$

Tabela 30 – Comparação entre o número de restrições presentes nos modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#). A comparação entre os modelos é feita utilizando as 5 primeiras instâncias das Classes 1, 3 e 5 propostas neste trabalho.

Com base nos resultados apresentados na [Tabela 30](#), nota-se que o número de restrições presentes nos modelos [CI2-Modificado](#) e [YLI-Modificado](#) é significativamente menor que o número de restrições presentes nas formulação [CI2](#) e [YLI](#), respectivamente. Esta diferença é intensificada quando consideradas classes com itens de menor comprimento. Para a Classe 3, os modelos [CI2](#) e [YLI](#) possuem um número um pouco menor de restrições quando comparados aos modelos [CI2-Modificado](#) e [YLI-Modificado](#), respectivamente. Concluimos com base nas Seções [7.3](#) e [7.4](#) que as modificações propostas neste capítulo são eficientes em reduzir o tamanho do modelo.

## 7.5 Resultados computacionais

A efetividade das melhorias propostas neste capítulo é avaliada a partir de testes computacionais semelhantes aos apresentados no Capítulo 6. O conjunto de dados utilizado para os testes computacionais são as 10 instâncias das Classes 1, 2, 3 e 4, propostas nesta dissertação, para o CS-LOSP unidimensional. Os testes feitos consideram três valores distintos para o parâmetro  $C$ , sendo  $C$  o número máximo de pilhas que podem ser abertas. Os valores de  $C$  utilizados nesta seção são  $C = 2$ ,  $C = 3$  e  $C = 4$ . As Tabelas [31](#), [33](#) e [35](#) indicam a média de tempo gasto para resolver as 10 instâncias das 4 classes propostas neste trabalho utilizando os modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#), além do valor médio da função objetivo avaliada em uma solução ótima quando consideramos  $C = 2$ ,  $C = 3$  e  $C = 4$ , respectivamente. As Tabelas [31](#), [33](#) e [35](#) são semelhantes às Tabelas [9](#), [12](#) e [15](#) apresentadas no Capítulo 6.

De forma análoga, as Tabelas 32, 34 e 36 são análogas às Tabelas 10, 13 e 16 apresentadas no Capítulo 6 e contêm os resultados para as classes parcialmente resolvidas em detalhes quando consideramos  $C = 2$ ,  $C = 3$  e  $C = 4$ , respectivamente.

Classe	YLI Tempo(s)	YLI-Modificado Tempo(s)	CI2 Tempo(s)	CI2-Modificado Tempo(s)	Valor da função objetivo
1	704,70	0,57	2,48	0,50	191,50
2	*	*	1002,95	9,65	317,90
3	*	*	*	*	*
4	—	—	*	*	*

Tabela 31 – Média de tempo gasto para resolver as 10 instâncias das Classes 1, 2, 3 e 4, propostas nesta dissertação para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos **YLI**, **YLI-Modificado**, **CI2** e **CI2-Modificado**. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 2. Nesta tabela, o caractere “—” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

A redução no tamanho dos modelos reflete diretamente no tempo necessário para se encontrar uma solução ótima. Com base nos resultados apresentados na Tabela 31 pode-se concluir que as modificações propostas neste capítulo aumentam a eficiência das formulações. O modelo **CI2-Modificado** apresentou os melhores resultados dentre todos os modelos testados. Para este modelo, assim como para a formulação **CI2**, foi possível resolver 2 classes de forma completa e 2 de forma parcial. No total, foram resolvidas 30 e 28 instâncias utilizando os modelos **CI2-Modificado** e **CI2**, respectivamente, considerando  $C = 2$ . Enquanto para os modelos **YLI** e **YLI-Modificado** foi possível resolver uma classe de forma completa e 2 de forma parcial, totalizando 18 e 25 instâncias, respectivamente. As melhorias propostas neste capítulo têm maior influência para as classes com itens de tamanho menor, para as Classes 1 e 2. O tempo exigido para se determinar uma solução ótima para os modelos com as modificações (**CI2-Modificado** e **YLI-Modificado**) foi significativamente menor do que para os modelos sem as melhorias (**CI2** e **YLI**). Este fato é justificado nas Seções 7.3 e 7.4. Na Tabela 32, apresentamos os resultados em detalhes para as classes parcialmente resolvidas, para  $C = 2$ .



Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado (YLI)			
2	(1174,84; 5620,86)	(7; 9)	(416; 314)
3	(205,42; 419,93; 27,72; 246,36; 87,94; 18,61)	(1; 4; 7; 8; 9; 10)	(824; 978; 1033; 994; 989; 872)
Yanasse Lamosa Integrado-Modificado (YLI-Modificado)			
2	(6,52; 104,21; 12,78; 34,48; 32,79; 2066,70; 1,28; 17,30; 561,46)	(1; 2; 3; 4; 5; 6; 7; 9; 10)	275; 311; 231; 327; 339; 311; 416; 314; 347)
3	(35,44; 640,18; 1883,47; 105,25; 14,14; 7,79)	(1; 4; 7 8; 9; 10)	(824; 978; 1033; 994; 989; 872)
Catanzaro et al. Integrado 2 (CI2)			
3	(6,46; 16,64; 109,64; 5,77 63,89; 6,27; 15,43)	(1; 4; 5; 7; 8; 9; 10)	(824; 978; 549; 1033; 994; 989; 872)
4	4360.46	3	1807
Catanzaro et al. Integrado 2-Modificado (CI2-Modificado)			
3	(14,85; 4042,87; 4,33; 223,81; 77,09; 6,31; 4,65; 1,75)	(1; 2; 4; 5; 7; 8; 9; 10)	(824; 830; 978; 549; 1033; 994; 989; 872)
4	(2372,10; 2673,14)	(1; 4)	(1496; 1757)

Tabela 32 – Resultados em detalhes para as classes parcialmente resolvidas utilizando os modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#). Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 2$ .

Conforme foi discutido no Capítulo 6, o tempo exigido para se determinar uma solução ótima depende do comprimento dos itens e da quantidade de itens distintos do problema. Este resultado é reiterado nesta seção. Novamente, as classes nas quais o comprimento dos itens é menor, assim como aquelas que possuem uma grande quantidade de itens de tipos distintos foram as mais difíceis de se resolver. Porém, é notável que o impacto que o comprimento dos itens tem no tempo exigido para se determinar uma solução ótima é menor para os modelos com as modificações.

A [Tabela 33](#) é similar à [Tabela 31](#) sendo a única diferença entre as duas o valor de  $C$ , enquanto na [Tabela 31](#) os testes computacionais foram realizados considerando  $C = 2$ , na [Tabela 33](#) o valor de  $C$  foi fixado em 3.

Classe	YLI Tempo(s)	YLI-Modificado Tempo(s)	CI2 Tempo(s)	CI2-Modificado Tempo(s)	Valor da função objetivo
1	*	101,46	*	8,31	190,30
2	—	*	—	*	*
3	*	*	*	*	*
4	—	—	—	—	—

Tabela 33 – Média de tempo gasto para resolver as 10 instâncias das Classes 1, 2, 3 e 4, propostas nesta dissertação para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#). Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 3. Nesta tabela, o caractere “—” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Para os modelos [YLI-Modificado](#) e [CI2-Modificado](#) foi possível resolver uma classe de forma completa e duas de forma parcial, no total foram resolvidas 22 e 17 instâncias para os modelos [YLI-Modificado](#) e [CI2-Modificado](#), respectivamente. Para os modelos [YLI](#) e [CI2](#) foi possível resolver duas classes de forma parcial, ou seja, para estes modelos, respeitando a limitação de tempo de 7200 segundos, não foi possível resolver nenhuma classe de forma completa. No total, foram resolvidas 13 e 11 instâncias para os modelos [YLI](#) e [CI2](#), respectivamente. Na [Tabela 34](#) apresentamos os resultados em detalhes para as classes parcialmente resolvidas, para  $C = 3$ .

Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado (YLI)			
1	(924,47; 942,81; 632,87; 34,61; 807,84; 360,71; 170,70; 2501,24)	(1; 2; 3; 4; 5; 6; 8; 10)	(181; 150; 182; 207; 206; 235; 160; 187)
3	(21,53; 14,18; 4142,17; 276,24; 112,09)	(4; 7; 8; 9; 10)	(965; 1009; 961; 962; 844)
Yanasse Lamosa Integrado-Modificado (YLI-Modificado)			
2	(1619,32; 301,27; 7064,03; 1312,07; 667,51; 242,64)	(1; 2; 4; 5; 7; 8)	(273; 310; 326; 336; 410; 308)
3	(13,96; 3051,68; 1,84; 89,89; 185,34; 51,57)	(4; 6; 7; 8; 9; 10)	(965; 878; 1009; 961; 962; 844)
Catanzaro et al. Integrado 2 (CI2)			
1	(1184,19; 176,15; 9,79; 35,51; 83,18; 26,25; 12,09; 32,53; 127,70)	(1; 2; 3; 4; 5; 6; 8; 9; 10)	(181; 150; 182; 207; 206; 235; 160; 193; 187)
3	(2513,78; 839,22)	(4; 7)	(965; 1009)
Catanzaro et al. Integrado 2-Modificado (CI2-Modificado)			
2	(923,32; 476,00; 1952,36)	(1; 4; 8)	(273; 326; 308)
3	(619,48; 801,15; 1703,81; 947,64)	(4; 7; 8; 10)	(965; 1009; 961; 844)

Tabela 34 – Resultados em detalhes para as classes parcialmente resolvidas utilizando os modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#). Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 3$ .

Os resultados apresentados na [Tabela 33](#) novamente confirmam a hipótese de que as classes contendo itens de tamanho menor ou grande quantidade de itens de tipos distintos exigem mais tempo para se determinar uma solução ótima. Por último, na [Tabela 35](#) apresentamos os resultados para  $C = 4$ .

Classe	YLI Tempo(s)	YLI-Modificado Tempo(s)	CI2 Tempo(s)	CI2-Modificado Tempo(s)	Valor da função objetivo
1	683,98	0,45	877,38	36,21	190,20
2	–	*	–	*	*
3	*	1139.69	*	–	857
4	–	–	–	–	–

Tabela 35 – Média de tempo gasto para resolver as 10 instâncias das Classes 1, 2, 3 e 4, propostas nesta dissertação para o CS-LOSP unidimensional, além do valor da função objetivo em uma solução ótima, considerando os modelos **YLI**, **YLI-Modificado**, **CI2** e **CI2-Modificado**. Foi considerada a limitação de que o número máximo de pilhas abertas deve ser igual ou inferior a 4. Nesta tabela, o caractere “–” simboliza que o tempo gasto para encontrar uma solução ótima, para cada uma das 10 instâncias, foi superior ao limite preestabelecido (7200 segundos) e o caractere “\*” significa que foi possível encontrar uma solução ótima, dentro do limite de tempo (7200 segundos), para algumas das 10 instâncias pertencentes à classe, mas não para todas.

Para o modelo **YLI-Modificado** foi possível resolver duas classes de forma completa e 1 de forma parcial. Para os demais modelos testados, foram resolvidas 1 classe de forma completa e 1 classe de forma parcial. No total, 28 instâncias foram resolvidas utilizando o modelo **YLI-Modificado**, enquanto para o modelo **YLI**, foram resolvidas 16 instâncias no total. Embora, mais uma vez, as modificações efetuadas no modelo **CI2-Modificado** obtiveram êxito em reduzir o tempo exigido para se determinar uma solução ótima para as classes com itens de menor comprimento, o número total de instâncias resolvidas pelo modelos **CI2-Modificado** e **CI2** foi o mesmo, ambos os modelos foram capazes de resolver 11 instâncias respeitando a limitação de tempo de 7200 segundos. Na [Tabela 36](#), apresentamos os resultados em detalhes para as classes parcialmente resolvidas, para  $C = 4$ .

Classe	Tempo(s)	Instâncias	Valor da função objetivo
Yanasse Lamosa Integrado (YLI)			
3	(5435,38; 8,86; 9,76; 1522,82; 48,21; 81,83)	(1; 4; 7; 8; 9; 10)	(786; 965; 1009; 961; 962; 844)
Yanasse Lamosa Integrado-Modificado (YLI-Modificado)			
2	(1434,01; 1241,45; 2248,10; 2834,07; 1570,94; 1080,90; 458,11; 3032,97)	(1; 2; 3; 4; 5; 6; 7; 10)	(273; 310; 229; 326; 336; 308; 409; 344)
Catanzaro et al. Integrado 2 (CI2)			
3	2221,14	4	965
Catanzaro et al. Integrado 2-Modificado (CI2-Modificado)			
2	2386,22	7	409

Tabela 36 – Resultados em detalhes para as classes parcialmente resolvidas utilizando os modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#). Na primeira coluna está indicado a classe, na segunda coluna o tempo em segundos para se obter uma solução ótima, na terceira coluna quais instâncias foram resolvidas dentro do tempo limite (7200s) e, na última, o valor da função objetivo para cada uma das instâncias analisadas. Os dados contidos nesta tabela são referentes à solução das classes utilizando  $C = 4$ .

O modelo [YLI-Modificado](#) contém uma quantidade consideravelmente menor de variáveis e restrições quando comparado aos demais modelos testados nesta dissertação. Para este modelo, foi possível executar o *software* de otimização e iniciar o processo de busca por uma solução ótima para todas as instâncias propostas no trabalho, respeitando o limite de tempo de 200 segundos. Apesar disso, não foi possível determinar uma solução ótima para nenhuma instância da Classe 4. Para o modelo [CI2-Modificado](#) foi possível executar o *software* de otimização e iniciar o processo de busca por uma solução ótima para todas as instâncias da Classe 2. Para a Classe 4, utilizando a formulação [CI2-Modificado](#), somente para a instância 6 foi possível executar o *software* de otimização quando considerado  $C = 3$  e para  $C = 4$ , novamente, não foi possível executar o *software* de otimização para nenhuma instância.

Com base nos resultados apresentados nesta seção concluímos que as melhorias propostas neste capítulo são eficientes na redução do tempo exigido para se determinar uma solução ótima para o CS-LOSP. Em relação ao número de instâncias resolvidas, é possível observar que o número decresce à medida que o valor de  $C$  aumenta, utilizando tanto o modelo [CI2](#) quanto o modelo [CI2-Modificado](#). Para  $C = 2$ , a formulação [CI2-Modificado](#) foi a que apresentou os melhores resultados e, para valores maiores de  $C$ , o modelo [YLI-Modificado](#) se sobressaiu aos demais.

Considerando todos os testes computacionais efetuados, foi possível resolver 47, 75, 50 e 58 instâncias utilizando os modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#), respectivamente. Nota-se que os resultados obtidos a partir da formulação [YLI-Modificado](#)

foram significativamente superiores aos demais modelos testados. Assim como no modelo [CI2-Modificado](#), a formulação [YLI-Modificado](#), visa determinar um sequenciamento factível de conjuntos de padrões de corte do mesmo tipo ao invés de buscar um sequenciamento factível de padrões de corte. Conforme pode ser visto nas Seções [7.3](#) e [7.4](#), esta modificação reduz significativamente o número de variáveis e restrições do modelo. Além desta modificação, a estratégia de remoção de pilhas foi substituída pela estratégia de remoção de vértices, como comentado no Capítulo 5. Entretanto, com base apenas nos testes realizados, não é possível garantir que o sucesso do modelo [YLI-Modificado](#) em resolver um grande número de instâncias se deve a esta modificação. Para se comparar as estratégia de remoção de vértices e de remoção de pilhas é necessário conduzir um maior número de testes comparando os modelos utilizando as duas estratégias.

No Capítulo 6, concluímos que há dois fatores que influenciam a dificuldade de resolução das instâncias a partir da aplicação dos modelos, sendo estes, o comprimento dos itens e o número de itens de tipos distintos. A aplicação das melhorias propostas neste capítulo reduzem o impacto que o comprimento dos itens tem no tempo exigido para se determinar uma solução ótima para a instância. Infelizmente, os modelos propostos neste capítulo ainda apresentam grande dificuldade em resolver instâncias com um grande número de tipos distintos de itens. A quantidade de instâncias resolvidas para cada um dos modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#) em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos é apresentada na [Tabela 37](#).

Yanasse Lamosa Integrado <a href="#">YLI</a>				
	M=5	M=8	M=10	M=20
$l_i = [100, 200]$	28	2	–	–
$l_i = [200, 500]$	–	–	17	0
Yanasse Lamosa Integrado-Modificado <a href="#">YLI-Modificado</a>				
	M=5	M=8	M=10	M=20
$l_i = [100, 200]$	30	23	–	–
$l_i = [200, 500]$	–	–	22	0
Catanzaro et al. Integrado 2 <a href="#">CI2</a>				
	M=5	M=8	M=10	M=20
$l_i = [100, 200]$	29	10	–	–
$l_i = [200, 500]$	–	–	10	1
Catanzaro et al. Integrado 2-Modificado <a href="#">CI2-Modificado</a>				
	M=5	M=8	M=10	M=20
$l_i = [100, 200]$	30	14	–	–
$l_i = [200, 500]$	–	–	12	2

Tabela 37 – Número de instâncias resolvidas utilizando os modelos [YLI](#), [YLI-Modificado](#), [CI2](#) e [CI2-Modificado](#) em função do intervalo no qual o comprimento dos itens está contido e da quantidade de itens distintos. Nesta tabela consideramos os três valores de  $C$  utilizados nos testes.

Para todas as Classes testadas, com exceção da Classe 4, o modelo [YLI-Modificado](#) conseguiu resolver um maior número de instâncias, quando comparado aos demais modelos testados. O impacto das melhorias propostas neste capítulo é especialmente notável para as classes nas quais o comprimento dos itens está contido no intervalo  $[100, 200]$ . Para a Classe 1, contendo 5 tipos distintos de itens, os modelos [YLI-Modificado](#) e [CI2-Modificado](#) obtiveram êxito em resolver todas as instâncias. Enquanto, para os modelos [YLI](#) e [CI2](#) foi possível resolver 28 e 29 instâncias, respectivamente. Já para a Classe 2, contendo 8 tipos distintos de itens, utilizando os modelos [YLI-Modificado](#) e [CI2-Modificado](#), foi possível resolver 23 e 14 instâncias, respectivamente, um número significativamente maior do que para os modelos sem as melhorias, uma vez que, utilizando as formulações [YLI](#) e [CI2](#), foi possível resolver 2 e 10 instâncias, respectivamente.

Embora as modificações propostas neste capítulo tenham sido eficientes em reduzir o tempo exigido para se determinar uma solução ótima para as Classes 3 e 4, nota-se que, para classes contendo um grande número de tipos distintos de itens, os modelos propostos neste capítulo ainda apresentam resultados insatisfatórios. O comprimento dos itens pertencentes às instâncias das Classes 3 e 4 está contido no intervalo  $[200, 500]$ . Para a Classe 3, os modelos baseados na formulação de [Tang e Denardo](#), ou seja, os modelos [YLI](#) e [YLI-Modificado](#) se sobressaem aos modelos baseados na formulação de

Catanzaro et al. II, ou seja, os modelos [CI2](#) e [CI2-Modificado](#). Para os modelos [YLI](#) e [YLI-Modificado](#) foi possível resolver 17 e 22 instâncias, respectivamente. Enquanto, para as formulações [CI2](#) e [CI2-Modificado](#) foi possível resolver 11 e 12 instâncias, respectivamente. Para a Classe 4, não foi possível resolver nenhuma instância utilizando os modelos [YLI](#) e [YLI-Modificado](#). Utilizando os modelos [CI2](#) e [CI2-Modificado](#) foi possível resolver 1 e 2 instâncias, respectivamente.

Sendo assim, as melhorias propostas neste capítulo demonstram potencial para lidar com as classes com itens de tamanho menor. Por outro lado, os modelos propostos neste trabalho são ineficazes para lidar com instâncias contendo uma grande quantidade de itens de tipos distintos. Desta maneira, se faz necessário a busca por métodos e modelos ainda mais eficientes para a resolução do CS-LOSP.



## 8 Considerações finais

O foco deste trabalho está no problema de corte de estoque com limitação no número máximo de pilhas abertas, em inglês, *Cutting Stock with Limited Open Stacks Problem* (CS-LOSP). Inicialmente, é feita uma revisão sobre o PCE o MOSP e o CS-LOSP. A revisão se inicia no Capítulo 2 e tem como foco o problema de corte de estoque. Neste capítulo, os modelos de [Gilmore e Gomory \(1961, 1963, 1965\)](#) e [Kantorovich \(1960\)](#) são apresentados em detalhes e é feita uma breve revisão sobre as formulações propostas em [Wolsey \(1977\)](#), [Carvalho \(1999\)](#) e [Macedo, Alves e Carvalho \(2010\)](#). Além disso, o problema de geração de padrões de corte e o método de geração de colunas também são abordados neste capítulo.

No Capítulo 3, o problema de sequenciamento dos padrões de corte para minimização de pilhas abertas (MOSP) é revisado em conjunto com o problema de sequenciamento de tarefas para minimização de trocas de ferramentas (MTSP). Os modelos de [Tang e Denardo \(1988\)](#), [Laporte, Salazar-González e Semet \(2004\)](#) e [Catanzaro, Golveia e Labbé \(2015\)](#) para o MTSP são também revisados neste capítulo. Já para o MOSP são revisadas as formulações propostas em [Yanasse \(1997\)](#) e [Pinto \(2004\)](#). Além de revisar estes modelos, também é abordado o método de solução proposto em [Yanasse \(1997\)](#), que permite transformar um modelo proposto para o MTSP em uma formulação para o MOSP.

A revisão dos problemas é concluída no Capítulo 4, no qual são revisadas as formulações para o CS-LOSP propostas em [Yanasse e Lamosa \(2007\)](#) e [Arbib, Marinelli e Ventura \(2016\)](#). Em conjunto com a revisão dos modelos, também são revisadas as heurísticas propostas em [Pileggi \(2003\)](#) para a resolução do CS-LOSP.

No Capítulo 5, propomos novas formulações para o CS-LOSP. Os modelos são desenvolvidos a partir da junção entre uma formulação elaborada para o PCE e outra para o MOSP, este procedimento é similar ao utilizado em [Pinto \(2004\)](#). As formulações propostas para o CS-LOSP se baseiam nos modelos propostos em [Gilmore e Gomory \(1961, 1963, 1965\)](#) e [Kantorovich \(1960\)](#) para o PCE e nas formulações [Yanasse \(1997\)](#), [Pinto \(2004\)](#), baseada no modelo elaborado em [Laporte, Salazar-González e Semet \(2004\)](#) para o MTSP e, em uma nova formulação para o MOSP, construída a partir de uma das formulações propostas em [Catanzaro, Golveia e Labbé \(2015\)](#) para o MTSP.

No Capítulo 6, os modelos propostos no Capítulo 5, juntamente com o proposto em [Yanasse e Lamosa \(2007\)](#), são comparados em termos de quantidade de variáveis e restrições, qualidade da relaxação linear e o tempo exigido para determinar uma solução ótima para um conjunto de problemas. São utilizadas duas bases de dados, propostas nesta dissertação, para realizar os testes. A primeira destas é composta de seis classes, na

qual cada classe contém dez instâncias referentes a CS-LOSPs unidimensionais. A segunda base de dados é referente a CS-LOSPs bidimensionais, esta base é composta de quatro classes, na qual cada classe contém 10 instâncias. Considerando  $C$  como o número máximo de pilhas abertas no CS-LOSP, três valores de  $C$  foram empregados nos testes, sendo estes  $C = 2$ ,  $C = 3$  e  $C = 4$ . A influência do valor de  $C$  no tempo exigido para se determinar uma solução ótima também foi analisado durante a execução dos testes computacionais.

Durante a execução dos testes computacionais, notamos que o comprimento dos itens e a quantidade de itens distintos influenciam no tempo exigido para encontrar uma solução ótima para o CS-LOSP. Para todos os modelos testados, o tempo necessário para encontrar uma solução ótima foi maior para as classes nas quais o comprimento dos itens é menor e o número de itens de tipos distintos é maior.

Percebemos também que a dimensão do problema altera o tempo exigido para se determinar uma solução ótima. Os três modelos testados encontraram maior dificuldade para resolver os CS-LOSPs bidimensionais.

Dentre as formulações propostas neste trabalho a formulação **CI2**, baseada em uma das formulações propostas em [Catanzaro, Golveia e Labbé \(2015\)](#) e no modelo proposto em [Gilmore e Gomory\(1961, 1963, 1965\)](#) para o PCE, foi a que apresentou melhores resultados para  $C = 2$  e para as classes nas quais o comprimento dos itens é menor. Para valores maiores de  $C$  e nas classes nas quais o comprimento dos itens é maior, o modelo proposto em [Yanasse e Lamosa \(2007\)](#) foi superior à formulação **CI2**.

No Capítulo 7, nós propomos algumas melhorias para os modelos **CI2** e **YLI**. As melhorias tem o intuito de reduzir a simetria do problema e reduzir o número de variáveis e restrições presentes no modelo. Testes computacionais realizados neste capítulo comprovam a validade das melhorias. Utilizando o modelo com as melhorias foi possível resolver um número maior de instâncias respeitando o tempo limite de 7200 segundos. Os resultados são particularmente interessantes para as classes com itens de tamanho menor, pois foi obtida uma redução significativa do tempo computacional exigido para se determinar uma solução ótima. Dentre todas as formulações testadas neste trabalho, a formulação **YLI-Modificado**, resultante da aplicação das melhorias propostas no Capítulo 7 no modelo proposto em [Yanasse e Lamosa \(2007\)](#), foi a que apresentou os melhores resultados.

Concluimos que os modelos propostos nesta dissertação representam um avanço na resolução do CS-LOSP, especialmente para o caso bidimensional, no qual, do nosso conhecimento, existem, na literatura, poucas formulações aplicáveis.

## 8.1 Perspectivas de trabalho

Os modelos propostos neste trabalho, assim como a formulação proposta em [Yanasse e Lamosa \(2007\)](#), apresentaram resultados limitados para o problema de corte de estoque com limitação no número máximo de pilhas abertas bidimensional. A aplicação direta dos modelos se mostrou insuficiente para determinar uma solução ótima para os problemas mesmo considerando instâncias com uma pequena quantidade de itens de tipos distintos e tais que o comprimento dos itens requeridos é grande em relação ao objeto. Em [Arbib, Marinelli e Ventura \(2016\)](#) os autores apresentam um método eficiente para a resolução do CS-LOSP. Infelizmente, o método proposto é construído apenas para o problema unidimensional. Tendo em vista a necessidade de resolver o CS-LOSP bidimensional de forma eficiente, assim como a ausência de métodos competitivos para a resolução deste problema, pretendemos estender o modelo e o método de geração de colunas propostos em [Arbib, Marinelli e Ventura \(2016\)](#) para o CS-LOSP bidimensional. Além disso, pretendemos construir uma nova formulação para o CS-LOSP utilizando o modelo [Catanzaro et al. III](#) em conjunto com as melhorias propostas no capítulo 7.

Por último, outra proposta para investigação futura é executar um maior número de testes computacionais para o modelo [Kantorovich Integrado](#) e escrever uma análise detalhada dos resultados obtidos, avaliando a performance do modelo para diversos parâmetros.

## Referências

- ARBIB, C.; MARINELLI, F.; VENTURA, P. One-dimensional cutting stock with a limited number of open stacks: bounds and solutions from a new integer linear programming model. *International Transactions in Operational Research*, v. 23, n. 1-2, p. 47–63, 2016. Citado 15 vezes nas páginas 19, 20, 62, 63, 64, 65, 66, 67, 68, 69, 70, 84, 85, 153 e 155.
- BAZARAA MOKHTAR S., J. J. J.; SHERALI, H. D. *Linear programming and network flows*. USA: John Wiley Sons, Inc., 2011. ISBN 978-0-470-46272-0. Citado na página 34.
- BECCENERI, J. C.; YANASSE, H. H.; SOMA, N. Y. A method for solving the minimization of the maximum number of open stacks problem within a cutting process. *Computers & Operations Research*, v. 31, n. 14, p. 2315–2332, 2004. Citado na página 118.
- BEEZAO, A. C. *O problema de minimização de trocas de ferramentas*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016. Citado na página 39.
- BEEZAO, A. C.; CORDEAU, J.-F.; LAPORTE, G.; YANASSE, H. H. Scheduling identical parallel machines with tooling constraints. *European Journal of Operational Research*, v. 257, n. 3, p. 834–844, 2017. Citado na página 39.
- BEZANSON, J.; EDELMAN, A.; KARPINSKI, S.; SHAH, V. B. Julia: A fresh approach to numerical computing. *SIAM review*, SIAM, v. 59, n. 1, p. 65–98, 2017. Disponível em: <<https://doi.org/10.1137/141000671>>. Citado na página 86.
- BROWNE, J.; DUBOIS, D.; RATHMILL, K.; SETHI, S. P.; STECKE, K. E. Classification of flexible manufacturing systems. *FMS Magazine*, v. 2, n. 2, p. 114–117, 1984. Citado na página 19.
- CARVALHO, J. M. V. V. de. Exact solution of bin-packing problems using column generation and branch-and-bound. *Annals of Operations Research*, n. 86, p. 629–659, 1999. Citado 2 vezes nas páginas 22 e 153.
- CATANZARO, D.; GOLVEIA, L.; LABBÉ, M. Improved integer linear programming formulations for the job sequencing and tool switching problem. *European Journal of Operational Research*, v. 244, n. 3, p. 766–777, 2015. Citado 15 vezes nas páginas 21, 46, 48, 53, 54, 55, 70, 76, 77, 80, 81, 88, 137, 153 e 154.
- CRAMA, Y.; OERLEMANS, A. G.; SPIEKSMAN, F. C. R. Minimizing the number of tool switches on a flexible machine. *Production Planning in Automated Manufacturing*, p. 165–195, 1994. Citado na página 53.
- DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, v. 2, n. 4, p. 393–410, 1954. Citado 5 vezes nas páginas 10, 45, 54, 55 e 56.

FAGGIOLI, E.; BENTIVOGLIO, C. A. Heuristic and exact methods for the cutting sequencing problem. *European Journal of Operational Research*, v. 110, n. 3, p. 564–575, 1998. Citado na página 118.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. *Operations Research*, v. 9, n. 6, p. 849–859, 1961. Citado 16 vezes nas páginas 18, 20, 22, 24, 25, 28, 30, 31, 35, 57, 59, 66, 70, 81, 153 e 154.

\_\_\_\_\_. A linear programming approach to the cutting-stock problem—part II. *Operations Research*, v. 11, n. 6, p. 863–888, 1963. Citado 16 vezes nas páginas 18, 20, 22, 24, 25, 28, 30, 31, 35, 57, 59, 66, 70, 81, 153 e 154.

\_\_\_\_\_. Multi-stage cutting stock problems of two and more dimensions. *Operations Research*, v. 13, n. 1, p. 94–120, 1965. Citado 16 vezes nas páginas 18, 20, 22, 24, 25, 28, 30, 31, 35, 57, 59, 66, 70, 81, 153 e 154.

KANTOROVICH, L. V. Mathematical methods of organizing and planning production. *Management Science*, v. 6, n. 4, p. 366–422, 1960. Citado 12 vezes nas páginas 18, 20, 22, 23, 24, 35, 62, 70, 81, 84, 107 e 153.

LAPORTE, G.; SALAZAR-GONZÁLEZ, J. J.; SEMET, F. Exact algorithms for the job sequencing and tool switching problem. *IEEE transactions*, v. 36, n. 1, p. 37–45, 2004. Citado 20 vezes nas páginas 8, 16, 19, 21, 36, 43, 44, 45, 46, 47, 48, 55, 70, 74, 75, 76, 80, 81, 88 e 153.

MACEDO, R.; ALVES, C.; CARVALHO, J. M. V. V. de. Arc-flow model for the two-dimensional guillotine cutting stock problem. *Computers Operations Research*, v. 37, n. 6, p. 991–1001, 2010. Citado 2 vezes nas páginas 22 e 153.

MARTELLO, S.; TOTH, P. *Knapsack Problems: Algorithms and Computer Implementations*. USA: John Wiley Sons, Inc., 1990. ISBN 0471924202. Citado na página 27.

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, v. 7, n. 4, p. 326–329, 1960. Citado 9 vezes nas páginas 10, 55, 56, 72, 74, 78, 80, 135 e 137.

MOHAMED, Z. M.; YOUSSEF, M. A.; HUQ, F. The impact of machine flexibility on the performance of flexible manufacturing systems. *International Journal of Operations & Production Management*, v. 21, n. 5/6, p. 707–727, 2001. Citado na página 19.

PILEGGI, G. C. F. *Abordagens para otimização integrada dos problemas de geração e sequenciamento de padrões de corte*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2003. Citado 2 vezes nas páginas 68 e 153.

PINTO, M. J. *Algumas contribuições à resolução do problema de corte integrado ao problema de sequenciamento dos padrões*. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais, 2004. Citado 8 vezes nas páginas 19, 20, 21, 36, 43, 45, 68 e 153.

TANG, C. S.; DENARDO, E. V. Models arising from a flexible manufacturing machine, part I: minimization of the number of the tool switches. *Operations Research*, v. 36, n. 5, p. 767–777, 1988. Citado 11 vezes nas páginas 19, 21, 36, 41, 42, 48, 70, 76, 80, 81 e 153.

- WIDMER, M. Job shop scheduling with tooling constraints: a tabu search approach. *The Journal of the Operational Research Society*, v. 42, n. 1, p. 75–82, 1991. Citado na página 39.
- WOLSEY, L. A. Valid inequalities, covering problems and discrete dynamic programs. *Annals of Discrete Mathematics*, v. 1, p. 527–538, 1977. Citado 2 vezes nas páginas 22 e 153.
- YANASSE, H. H. On a pattern sequencing problem to minimize the maximum number of open stacks. *European Journal of Operational Research*, v. 100, n. 3, p. 454–463, 1997. Citado 13 vezes nas páginas 19, 20, 21, 36, 41, 42, 45, 57, 60, 80, 81, 137 e 153.
- YANASSE, H. H.; LAMOSA, M. J. P. An integrated cutting stock and sequencing problem. *European Journal of Operational Research*, v. 183, n. 3, p. 1353–1370, 2007. Citado 9 vezes nas páginas 20, 21, 57, 59, 70, 81, 153, 154 e 155.
- YUEN, B. J. Improved heuristics for sequencing cutting patterns. *European Journal of Operational Research*, v. 87, n. 1, p. 57–64, 1995. Citado na página 68.
- ZEBALLOS, L. J. A constraint programming approach to tool allocation and production scheduling in flexible manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, v. 26, n. 6, p. 725–743, 2010. Citado na página 39.