



LOCALIZAÇÃO DE ARMAZÉNS COM EFEITO DE CONSOLIDAÇÃO DE
ESTOQUES: UMA ANÁLISE COMPARATIVA DO DESEMPENHO DE
DIFERENTES FORMULAÇÕES

Gustavo Cunha de Bittencourt

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Produção, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Produção.

Orientadores: Virgílio José Martins Ferreira Filho
Laura Silvia Bahiense da S. Leite

Rio de Janeiro

Março de 2014

LOCALIZAÇÃO DE ARMAZÉNS COM EFEITO DE CONSOLIDAÇÃO DE
ESTOQUES: UMA ANÁLISE COMPARATIVA DO DESEMPENHO DE
DIFERENTES FORMULAÇÕES

Gustavo Cunha de Bittencourt

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA
(COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE
DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE
EM CIÊNCIAS EM ENGENHARIA DE PRODUÇÃO.

Examinada por:

Prof^a. Laura Silvia Bahiense da Silva Leite, D.Sc.

Prof. Virgílio José Martins Ferreira Filho, D.Sc.

Prof. Nelson Maculan Filho, D.Sc.

Prof. Peter Fernandes Wanke, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2014

Bittencourt, Gustavo Cunha de

Localização de Armazéns com Efeito de Consolidação de Estoques / Gustavo Cunha de Bittencourt – Rio de Janeiro: UFRJ/COPPE, 2014.

XII, 78 p.: il.; 29,7 cm.

Orientadores: Virgílio José Martins Ferreira Filho

Laura Silvia Bahiense da Silva Leite

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia de Produção, 2014.

Referências Bibliográficas: p. 53-56.

1. Pesquisa Operacional. 2. Localização de Armazéns. 3. Programação Inteira Mista. I. Ferreira Filho, Virgílio José Martins *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Produção. III. Título.

*A Zilda e Bentos, com todo o
amor que um filho pode ter.*

AGRADECIMENTOS

À medida que o conhecimento humano avança, alguns fenômenos – antes tidos como místicos – tornam-se explicáveis, modeláveis e até reproduzíveis. Outros, entretanto, permanecem um mistério para nós. Agradeço àquela singularidade que se expandiu, à Deus, à essa energia que nos une e nos move ou simplesmente à sequência de eventos aleatórios (ou não) que me permitiram estar exatamente onde estou, bem como à qualquer combinação destes fatores.

Agradeço aos meus pais, Zilda e Bentos, por todo o amor e carinho que me foi dedicado ao longo de toda minha vida, e em especial nesta etapa árdua que se concluiu. Também aos meus irmãos, Rafael e Thiago, às irmãs Ana, Anny e Alejandra, e à minha namorada Isis Lee, fonte de muita força e inspiração. Estes agradecimentos são extensivos à toda a minha família: aos tios Robert, Alonso, Celso, Fernando, João, José Rodrigues e Rubens; às tias Joaquina, Neuza, Vera; bem como aos primos, sobrinhos e amigos queridos que não posso citar nominalmente por limitações no número de caracteres, mas que moram no meu coração. Dedico um “muito obrigado” especial à minha madrinha, Betinha, por todo o apoio e amor quase maternal.

Gostaria de agradecer ao Prof. Virgílio e à Prof^a. Laura todo o carinho, apoio, compreensão e confiança. O que vocês fizeram por mim não tem preço, e o resto da vida será pouco para agradecer tudo isso.

Agradeço ao Prof. Nelson Maculan e ao Prof. Peter Wanke por aceitarem o meu convite para participar da banca que examinou este trabalho.

Sou grato também dos colegas, professores e funcionários do PEP (Programa de Engenharia de Produção) e do LORDE (Laboratório de Otimização de Recursos, de Simulação Operacional e de Apoio a Decisões na Indústria do Petróleo) que tanto contribuíram para a realização deste trabalho.

Por último, agradeço à Petrobras pelo suporte financeiro que me foi concedido, e à UFRJ por me acolher e expandir meus horizontes.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

LOCALIZAÇÃO DE ARMAZÉNS COM EFEITO DE CONSOLIDAÇÃO DE
ESTOQUES: UMA ANÁLISE COMPARATIVA DO DESEMPENHO DE
DIFERENTES FORMULAÇÕES

Gustavo Cunha de Bittencourt

Março/2014

Orientadores: Virgílio José Martins Ferreira Filho
Laura Silvia Bahiense da Silva Leite

Programa: Engenharia de Produção

Este trabalho apresenta duas formulações matemáticas para o Problema de Localização de Armazéns. Nestas modelagens é considerado o efeito de consolidação de estoques, que prevê reduções nos estoques de segurança dos clientes de uma cadeia de suprimentos com a centralização destes em armazéns. Uma das formulações analisa o problema sob a óptica de dois elos da cadeia, sendo baseado em variáveis de índice duplo; enquanto a outra engloba uma variável de fluxo de produtos de índice triplo. Dada a natureza não linear de uma das equações componentes do sistema, os modelos são linearizados utilizando uma aproximação por linearização por partes. Dois métodos são propostos e analisados para a divisão dos segmentos nesta aproximação; e os dois modelos são comparados em diversos aspectos e cenários.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

INVENTORY LOCATION PROBLEM WITH RISK-POOLING EFFECT: A
COMPARATIVE ANALYSIS BETWEEN DIFFERENT FORMULATIONS
PERFORMANCES

Gustavo Cunha de Bittencourt

March/2014

Advisors: Virgílio José Martins Ferreira Filho

Laura Silvia Bahiense da Silva Leite

Department: Production Engineering

This work presents two mathematical formulations for the Inventory Location Problem. These models consider the risk-pooling effect, which provides reductions in customers' safety stocks of a supply chain by centralizing these in central depots. One of the formulations analyzes the problem from the perspective of a two echelon supply chain, being based on double index variables, while the other includes a triple index flow variable. Given the nonlinear nature of the components of the system equations, the models are linearized using a piecewise linearization approximation. To divide the segments, two methods are proposed and analyzed in this approach, and the two models are compared in several aspects and scenarios.

SUMÁRIO

1. INTRODUÇÃO	1
1.1 JUSTIFICATIVA E MOTIVAÇÃO DO TEMA	3
1.2 OBJETIVOS	3
1.3 ESTRUTURA	4
2. REVISÃO BIBLIOGRÁFICA	6
2.1 MODELOS DE GESTÃO DE ESTOQUE	6
2.2 CONSOLIDAÇÃO DE ESTOQUES	8
2.3 PROBLEMA DE LOCALIZAÇÃO DE FACILIDADES	9
A. MODELOS QUE DESCONSIDERAM OS CUSTOS DE MANUTENÇÃO DE ESTOQUES	10
B. MODELOS QUE CONSIDERAM CUSTOS DE MANUTENÇÃO DE ESTOQUES SIMPLIFICADOS	11
C. MODELOS QUE CONSIDERAM OS EFEITOS DE CENTRALIZAÇÃO DE ESTOQUES	12
D. MODELOS QUE CONSIDERAM OS EFEITOS DE CENTRALIZAÇÃO DE ESTOQUES APLICADOS À INDÚSTRIA PETROLÍFERA	12
3. DEFINIÇÃO DO PROBLEMA	14
3.1 SUPOSIÇÕES E HIPÓTESES	15
4. FORMULAÇÕES DO PROBLEMA	17
4.1 FORMULAÇÃO DE ÍNDICE DUPLO	19
4.2 FORMULAÇÃO DE ÍNDICE TRIPLO	23
5. METODOLOGIA	26
5.1 SEGMENTAÇÃO DO EIXO X	26
5.2 GERAÇÃO DE INSTÂNCIAS DE TESTE (PROJETO DE EXPERIMENTOS)	28
6. EXPERIMENTAÇÃO E ANÁLISE	35
6.1 EXPERIMENTOS ALEAT (INSTÂNCIAS ALEATÓRIAS)	35
6.2 EXPERIMENTOS FUF (FATOR DE UTILIZAÇÃO DOS FORNECEDORES)	40
6.3 EXPERIMENTOS FUA (FATOR DE UTILIZAÇÃO DOS ARMAZÉNS)	42
6.4 EXPERIMENTOS NSEG (NÚMERO DE SEGMENTOS)	43
6.5 EXPERIMENTOS GAP (DIFERENÇA ENTRE LIMITANTES)	46
6.6 EXPERIMENTOS ALEAT-RL (CROSS-FILLING)	48
7. CONCLUSÕES	50
7.1 TRABALHOS FUTUROS	51

8. REFERÊNCIAS BIBLIOGRÁFICAS	53
APÊNDICE I	57
APÊNDICE II	58
APÊNDICE III	61
APÊNDICE IV	69

LISTA DE FIGURAS

FIGURA 1.1 – REPRESENTAÇÃO DO PROBLEMA DE LOCALIZAÇÃO DE ARMAZÉNS	2
FIGURA 2.1 – GRÁFICO DENTE-DE-SERRA – MODELO LEC.....	7
FIGURA 3.1 – REPRESENTAÇÃO ESQUEMÁTICA DO PROBLEMA DE LOCALIZAÇÃO DE ARMAZÉNS	14
FIGURA 4.1 – REPRESENTAÇÃO DO MODELO DE ÍNDICE DUPLO	17
FIGURA 4.2 – REPRESENTAÇÃO DO MODELO DE ÍNDICE TRIPLO	18
FIGURA 4.3 – APROXIMAÇÃO POR LINEARIZAÇÃO POR PARTES.....	21
FIGURA 5.1 – SEGMENTAÇÃO PELA DIVISÃO EM INTERVALOS IGUALMENTE ESPAÇADOS (SL)	27
FIGURA 5.2 – SEGMENTAÇÃO PELA DIVISÃO QUADRÁTICA (SQ).....	28
FIGURA 5.3 – CIDADES MAIS POPULOSAS DOS ESTADOS UNIDOS.....	31
FIGURA 6.1 – INSTÂNCIAS ALEATÓRIAS X TEMPO	37
FIGURA 6.2 – TEMPO MÉDIO POR MODELO (INSTÂNCIAS ALEAT)	38
FIGURA 6.3 – ERRO DE LINEARIZAÇÃO.....	39
FIGURA 6.4 – ERRO MÉDIO DE LINEARIZAÇÃO SL x SQ (INSTÂNCIAS ALEAT)	39
FIGURA 6.5 – TEMPO MÉDIO E ERRO MÉDIO POR MODELO (INSTÂNCIAS ALEAT)	40
FIGURA 6.6 – FATOR DE UTILIZAÇÃO DOS FORNECEDORES X TEMPO	41
FIGURA 6.7 – FATOR DE UTILIZAÇÃO DOS ARMAZÉNS X TEMPO	43
FIGURA 6.8 – TEMPO MÉDIO E ERRO MÉDIO POR NÚMERO DE SEGMENTOS (ID-SL)	45
FIGURA 6.9 – TEMPO MÉDIO E ERRO MÉDIO POR NÚMERO DE SEGMENTOS (ID-SQ)	45
FIGURA 6.10 – TEMPO MÉDIO E ERRO MÉDIO POR NÚMERO DE SEGMENTOS (IT-SL).....	45
FIGURA 6.11 – TEMPO MÉDIO E ERRO MÉDIO POR NÚMERO DE SEGMENTOS (IT-SQ).....	46
FIGURA 6.12 – GAP MÉDIO APÓS 2H (INSTÂNCIAS GRANDES)	47
FIGURA 6.13 – GAP MÉDIO X ERRO MÉDIO APÓS 2H (INSTÂNCIAS GRANDES)	47
FIGURA 6.14 – TEMPO MÉDIO DE PROCESSAMENTO (ALEAT X ALEAT-RL).....	49
FIGURA 6.15 – VALOR DA FUNÇÃO OBJETIVO (ALEAT X ALEAT-RL).....	49

LISTA DE SÍMBOLOS

Índices e Conjuntos

$j \in J$	Índice de potenciais fornecedores a serem abertos
$k \in K$	Índice de potenciais armazéns a serem abertos
$l \in L$	Índice de clientes
$r \in R$	Índice de segmentos na aproximação por linearização por partes

Parâmetros

f_j	Custo anual de abertura de um fornecedor na localidade j
P_j	Capacidade de produção do fornecedor j
c_{jk}	Custo unitário de produção e envio de um produto do fornecedor j para o armazém k
g_k	Custo anual de abertura de um armazém na localidade k
h_k	Custo anual unitário de manutenção de estoques no armazém k
V_k	Capacidade de armazenagem do armazém k
Γ	Horizonte de planejamento: número de dias de operação em um ano.
t_{kl}	Custo unitário de transporte de produtos entre o armazém k e o cliente l
φ_{kl}	<i>Lead time</i> médio de transporte do depósito k para o cliente l
ξ_{kl}	Cobertura de clientes: 1 se o cliente l pode ser atendido pelo depósito k , 0 caso contrário.
μ_l	Média da demanda diária de produtos do cliente l
σ_l	Desvio padrão da demanda diária de produtos do cliente l
σ_l^2	Variância da demanda diária de produtos do cliente l
α	Nível de serviço desejado para os clientes
Z_α	Fator de segurança correspondente ao nível de serviço α desejado pelos clientes
F_k^r	Coefficiente linear do segmento r na aproximação por linearização por partes
C_k^r	Coefficiente angular do segmento r na aproximação por linearização

por partes

Variáveis

SS_k	Estoque de segurança consolidado no armazém k
w_j	Abertura de fornecedores: 1 se o fornecedor j é aberto, 0 caso contrário
x_{jk}	Número de unidades produzidas no fornecedor j e enviadas para o armazém k (modelo de índice duplo)
x_{jkl}	Número de unidades produzidas no fornecedor j e enviadas para o armazém k a fim de atender à demanda do cliente l (modelo de índice triplo)
xS_{jk}	Número de unidades produzidas no fornecedor j e enviadas para o armazém k para suprir o seu estoque de segurança (modelo de índice triplo)
z_k	Abertura de armazéns: 1 se o armazém k é aberto, 0 caso contrário
y_{kl}	Alocação de clientes: 1 se o cliente l é alocado ao armazém k , 0 caso contrário
$v_k^r, v_k, v_{kl}, v_{kl}^r$	Variáveis auxiliares contínuas da aproximação por linearização por partes
u_k^r	Variáveis auxiliares binárias da aproximação por linearização por partes

1. Introdução

*“Those who dream by day are cognizant of many things
which escape those who dream only by night.”*

— Edgar Allan Poe

O projeto do sistema logístico pertence às decisões estratégicas de uma organização, isto é, aquelas com efeitos de longo prazo e que influenciam de forma marcante o desempenho e a imagem da companhia. Estas decisões envolvem a determinação do número e localização de terminais, o tipo e a quantidade de equipamentos a serem utilizados, os tipos de linhas e capacidades a serem adicionadas, entre outras. Embora Bektas e Crainic (2007) estivessem tratando especificamente de redes de transporte intermodais, estes conceitos também se aplicam ao estudo desenvolvido neste trabalho.

As decisões tomadas no nível estratégico serão as premissas utilizadas para o projeto da rede de serviços, que é o plano de transporte para atender à demanda enquanto opera o sistema de maneira eficiente (nível tático); e para o planejamento operacional.

O Council of Supply Chain Management define logística como "a parte do Gerenciamento da Cadeia de Abastecimento que planeja, implementa e controla o fluxo e armazenamento eficiente e econômico de matérias-primas, materiais semi-acabados e produtos acabados, bem como as informações a eles relativas, desde o ponto de origem até o ponto de consumo, com o propósito de atender às exigências dos clientes" (COUNCIL OF SUPPLY CHAIN MANAGEMENT, 2012). O projeto e gestão de cadeias de suprimento é um dos problemas mais críticos enfrentados pelas organizações modernas. As empresas e profissionais de logística devem estar aptos a projetar e avaliar diversas configurações e cenários com o objetivo gerenciar as incertezas, nível de serviço e custos de logísticos.

Dentre os problemas de logística comumente abordados em pesquisa operacional, destacam-se os de transporte e armazenagem de produtos. Na indústria do petróleo, a modelagem e a solução eficiente destes problemas mostram-se ainda mais

relevantes. Isso se deve, principalmente, à ordem de grandeza dos valores envolvidos, tanto nos investimentos de capital quanto nas operações de exploração, refino e distribuição do óleo.

O problema de localização de armazéns pode ser descrito como o projeto de um sistema de produção-estocagem-distribuição em dois elos, onde um determinado conjunto de clientes deve ser abastecido por um conjunto de fornecedores. No nível intermediário devem ser alocados os armazéns, depósitos ou centros de distribuição (CDs) com o objetivo de consolidar os estoques de ciclo e os estoques de segurança dos clientes. Portanto, o primeiro elo da cadeia engloba o fluxo de produtos entre fornecedores e os armazéns, enquanto o segundo engloba o fluxo de produtos entre os armazéns e os clientes, conforme exemplificado na figura abaixo.

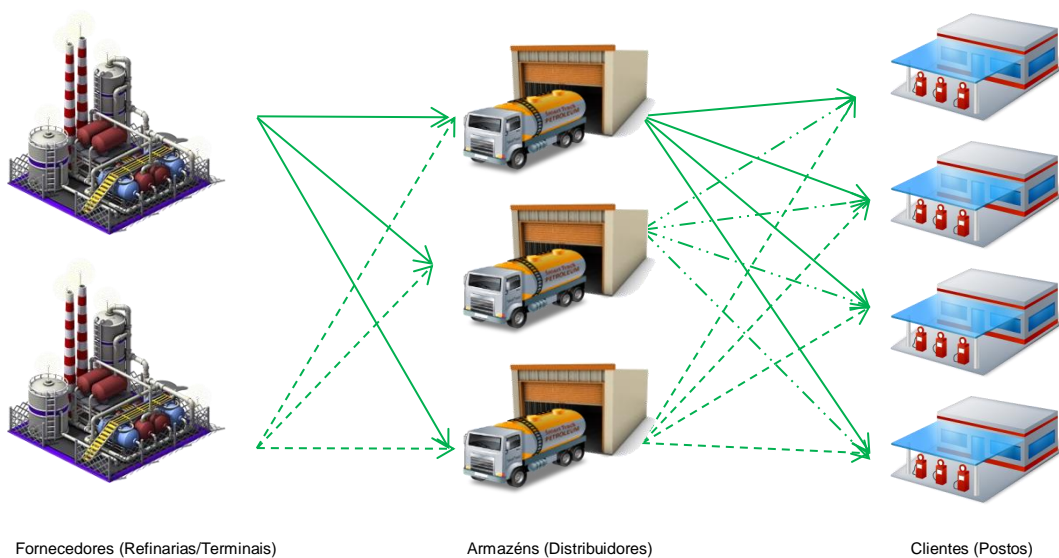


Figura 1.1 – Representação do Problema de Localização de Armazéns

Das diversas aplicações possíveis deste problema na indústria do petróleo, é possível destacar a alocação das bases de distribuição dos distribuidores, que recebem os derivados de petróleo das refinarias através dos terminais e podem consolidar os estoques de segurança dos postos de combustível. Tsuchida (2008) detalha o sistema de distribuição da Petróleo Brasileiro S/A (Petrobras). Embora os distribuidores e postos de combustível não sejam empresas pertencentes à Petrobras, as recentes tendências de integração das cadeias de suprimentos vistas na literatura técnica e estudos de caso de logística ensejam a possibilidade de acordos entre estas empresas para que políticas de localização e estoques sejam planejadas conjuntamente para todos os elos da cadeia.

1.1 Justificativa e motivação do tema

A modelagem e solução de problemas logísticos possuem grande relevância tanto do ponto de vista acadêmico quanto econômico. Em relação às formulações para problema de localização de armazéns especificamente, pode-se destacar o fato de representar uma ferramenta de apoio à tomada de decisões que engloba àquelas que concernem tanto ao nível estratégico quanto tático da organização. Isto se traduz em modelos mais abrangentes e com maiores perspectivas de soluções que se adequem à realidade dos negócios a que se destinam. Na indústria do petróleo, onde gigantescas quantias são despendidas em investimentos e nas operações, isso representa um potencial incrível para economias e ganhos de eficiência na cadeia de suprimentos como um todo. Além disso, também são tratadas as incertezas relativas à demanda, o que torna os modelos ainda mais aderentes à realidade.

Do ponto de vista acadêmico é possível destacar a complexidade do problema abordado, dada a sua natureza inteira (programação inteira mista) e a não linearidade da função que determina o estoque de segurança em ambas as formulações propostas. Isto faz com que seja necessária a aplicação de técnicas para a sua linearização que conciliem um baixo erro de aproximação com a possibilidade de se encontrar soluções viáveis com custo computacional razoável.

Por ser um campo de estudos relativamente novo, existem muitas oportunidades de melhorias nos métodos existentes e de descobertas relevantes.

1.2 Objetivos

O objetivo geral deste trabalho é realizar um estudo da modelagem do Problema de Localização de Armazéns, através da comparação entre duas formulações matemáticas propostas para a sua solução.

Como objetivos específicos, podemos elencar análises mais detalhadas das especificidades de cada um dos modelos, comparando o tempo de processamento para configurações de instâncias geradas aleatoriamente; a influência do fator de utilização das capacidades dos fornecedores e dos armazéns no custo computacional das soluções; e o *tradeoff* entre números de segmentos utilizados na linearização por partes da função do estoque de segurança (e a conseqüente qualidade das soluções obtidas) e o custo

computacional. Também serão comparadas duas formas distintas de segmentação da função do estoque de segurança pelo processo de linearização por partes.

1.3 Estrutura

Esta dissertação se divide em oito capítulos. Inicialmente, teve-se a introdução, que explicitou o problema abordado e o contexto no qual se insere. Em seguida, foram apresentadas as justificativas e motivações para a realização deste trabalho, bem como os objetivos gerais e específicos propostos.

No segundo capítulo, será apresentada uma breve revisão bibliográfica, com a evolução temporal dos modelos de gestão de estoques e de localização de facilidades, bem como a sua posterior integração. São apresentados modelos de localização de facilidades que desconsideram os custos de estoques, modelos que simplificam estes custos e modelos que consideram o também o efeito de consolidação de estoques. Por último, são mostrados exemplos de aplicação de problemas de localização de armazéns na indústria petrolífera.

O capítulo três apresenta a definição formal do Problema Localização de Armazéns. Também são apresentadas as suposições e hipóteses assumidas para a sua modelagem.

O quarto capítulo mostra as duas formulações propostas, tanto no seu formato mais básico quanto na versão implementada, que é a formulação fortalecida e linearizada.

No capítulo cinco é apresentada a metodologia de execução do trabalho e o processo de geração das instâncias de teste.

O sexto capítulo expõe os experimentos realizados e resultados obtidos com os modelos. Os resultados destes experimentos também são analisados neste capítulo.

A conclusão é apresentada no capítulo sete, com a síntese do que foi desenvolvido e alcançado, além de propostas futuras para a continuação deste trabalho.

Por fim, têm-se o conjunto de referências bibliográficas, seguido pelo Apêndice I - Diagrama de Classes do Gerador de Instâncias; Apêndice II - Capturas de Tela do

Gerador de Instâncias; Apêndice III - Código em OPL do Modelo de Índice Duplo e Código em OPL do Modelo de Índice Triplo; e Apêndice IV - Código em C++ da Interface de Integração.

2. Revisão Bibliográfica

“Os verdadeiros analfabetos são os que aprenderam a ler e não lêem.”

— Mário Quintana

Diversas abordagens e modelos já foram propostos na literatura para lidar com os problemas de estoques e localização, bem como suas combinações e desdobramentos. Este capítulo tem por objetivo realizar uma revisão bibliográfica dos trabalhos já realizados nesta área. Na primeira subseção são mostrados os modelos de gestão de estoque; seguidos pelas técnicas de *risk pooling* e, mais especificamente, os trabalhos relacionados à consolidação de estoques. Por último, são apresentados os modelos de localização e a progressiva incorporação do tratamento de estoques nestes estudos.

2.1 Modelos de Gestão de Estoque

O modelo do Lote Econômico de Compra (LEC ou EOQ - *Economic Order Quantity*) foi um dos primeiros a ser desenvolvido, proposto por Harris (1913) para otimizar os níveis de estoque no caso em que a demanda é contínua e perpétua. Posteriormente, o LEC se tornou parte integrante do modelo de Ponto de Pedido (PP), sendo que as demandas e os *lead times* de ressuprimento passaram a ser probabilísticos (BALLOU, 2005). Estes modelos serviram como base para a grande maioria dos modelos surgidos a partir daí, incluindo os que serão abordados neste trabalho.

O LEC determina a quantidade a ser comprada que minimiza os custos totais, sendo este o somatório dos custos de colocação de pedido (de compra ou distribuição) e dos custos de manutenção do estoque. No LEC a demanda e o *lead time* de entrega são considerados determinísticos e constantes, e por isso o modelo leva em conta apenas os estoques de ciclo, ou seja, aqueles que são utilizados para se atender a demanda propriamente dita, sem levar em conta as variações que a mesma pode apresentar durante o *lead time*. Também são premissas do modelo:

- Faltas de produtos e *backorders* (entregas com atraso) não são permitidas

- Custos de pedido e de estoque são independentes do tamanho da ordem (não existem descontos por quantidade) e não variam no tempo;
- O pedido chega completo em um único instante de tempo;
- Itens diversos são pedidos de forma independente, ou seja, não são consideradas possibilidades de um pedido com vários itens;
- Não existem restrições, como espaço de armazenagem e capacidade de transporte.

O ponto de pedido (PP), por sua vez, é o momento no qual a ordem de compra deve ser emitida; determinado pelo *lead time* de entrega e pela taxa de consumo média dos produtos (demanda).

Dada a natureza determinística do modelo, os níveis de estoque ao longo do tempo podem ser representados por um gráfico dente-de-serra, conforme pode ser visto na figura 2.1.

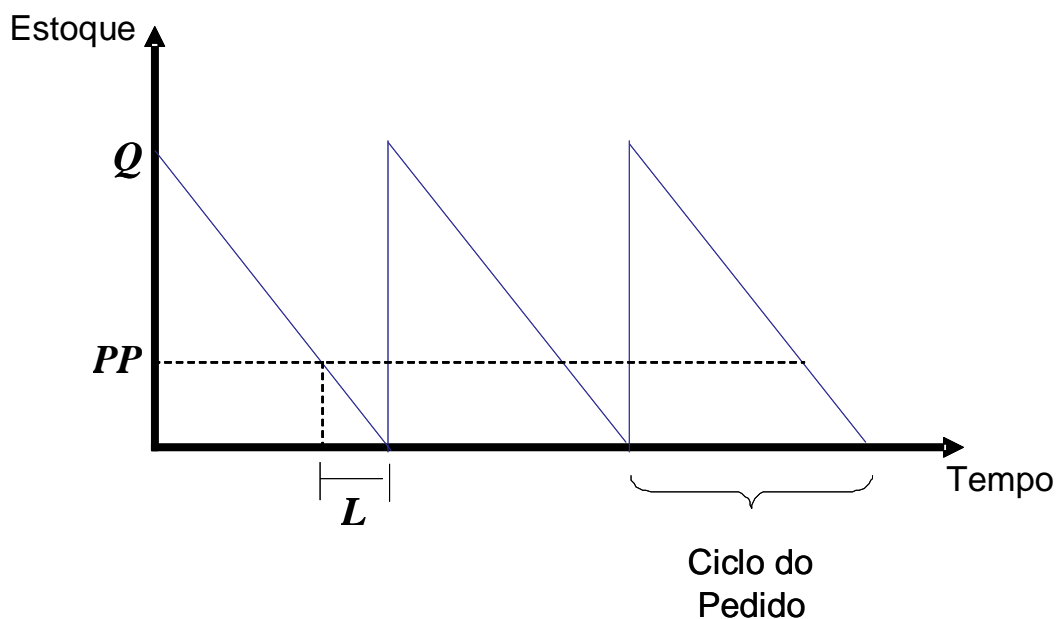


Figura 2.1 – Gráfico dente-de-serra – Modelo LEC
(Extraído de FERREIRA FILHO et al., 2006)

No LEC, um pedido com a quantidade Q de produtos é realizado sempre que o estoque chega ao nível de PP unidades, que é o ponto de pedido.

As incertezas da demanda dos consumidores e do suprimento por parte de fornecedores requerem que seja mantido um estoque adicional para os produtos armazenados, de modo a se obter uma proteção quanto à probabilidade de um esgotamento de estoque (*stock-out*); sendo este estoque adicional chamado de estoque de segurança (ES). Tallon (1993) e Bowersox e Closs (2001) calculam o estoque de segurança a partir da determinação da probabilidade de não faltar produto, ou seja, a probabilidade de não ocorrência de um *stock-out*, considerando a variabilidade (desvio-padrão) da demanda durante o *lead time* e a distribuição de probabilidade da demanda durante o *lead time*. Bowersox e Closs (2001) afirmam que a demanda durante o *lead time* é frequentemente considerada aderente à distribuição normal, sendo esta elencada como a mais simples dentre as utilizadas com a finalidade de controlar estoques.

2.2 Consolidação de Estoques

Jorge (2008) elenca em seu trabalho quatro subdivisões das técnicas de *risk pooling*: centralização ou consolidação de estoques, análise de múltiplas fontes de fornecimento (*order splitting*), múltiplo abastecimento de mercados consumidores (*cross-filling* ou *transshipment*) e a padronização de componentes ou peças para produção ou reposição. Destas, apenas o efeito de consolidação de estoques será abordado neste trabalho, tendo suas aplicações detalhadas abaixo.

A centralização ou consolidação dos estoques (também conhecida como *inventory pooling*, *demand pooling*, *portfolio effect*, *consolidation effect* e *virtual inventory management*) tem a finalidade de mitigar o risco decorrente das flutuações de demanda de consumidores separados geográfica ou mercadologicamente a partir do compartilhamento de unidades de distribuição na rede (BALLOU e BURNETAS, 2003).

As economias atribuídas a esta consolidação começaram a ser estudadas por Maister (1976) a partir da regra da raiz quadrada. Respeitando certas premissas a respeito da distribuição de probabilidade das demandas e do *lead time*, e da ausência de correlação entre as demandas, esta regra determina que a centralização de n depósitos em apenas um reduz os estoques de segurança proporcionalmente à \sqrt{n} .

Eppen (1979) aprofunda este estudo, derivando uma expressão do custo esperado de esgotamento e manutenção de estoques como uma função dos parâmetros de demanda de cada armazém, sendo considerado também o coeficiente de correlação entre as demandas.

Zinn, Levy e Bowersox (1989) derivaram uma expressão, baseada também na regra da raiz quadrada, para mensurar a redução percentual dos estoques de segurança a partir da centralização. Com a criação do conceito de magnitude, onde uma variável M_{ij} representa a proporção entre os desvios padrão de dois depósitos i e j quaisquer, os autores concluíram que os ganhos resultantes do efeito portfólio se devem à relação entre os valores dos desvios padrão, e não aos valores absolutos destes; bem como tornou mais clara a influência da correlação entre as demandas na redução final (demandas correlacionadas positivamente apresentam menores ganhos com o efeito portfólio, enquanto demandas correlacionadas negativamente maximizam este efeito).

Mahmoud (1992) expande os estudos anteriores, estudando o caso de sub-centralizações em múltiplos armazéns. A partir disso, apresenta um exemplo onde a sub-centralização em dois armazéns resulta em um estoque de segurança menor do que resultaria a centralização total, dada as características da demanda de cada mercado consumidor e as correlações existentes entre estas demandas.

2.3 Problema de Localização de Facilidades

Segundo Ambrosino e Scutellà (2005), o problema de localização de facilidades (*facility location problem*) consiste na determinação da melhor maneira de se transferirem bens para os pontos de demanda, escolhendo-se a estrutura da rede (os elos da rede, os diferentes tipos de facilidades operando nos diferentes elos, o número dessas facilidades e sua localização) ao mesmo tempo em que os custos totais da rede são minimizados. Os primeiros trabalhos envolvendo este tipo de decisão foram realizados, segundo Ballou (2004), por Thünen e Weber no final do século XIX e início do século XX. Entretanto, o crescimento destes estudos se deu por volta da década de 60 (OWEN; DASKIN, 1998).

a. Modelos que desconsideram os Custos de Manutenção de Estoques

O trabalho de Owen e Daskin (1998) revisa os problemas clássicos de localização de facilidades, onde os custos de manutenção de estoques não são considerados. Além disso, em sua grande maioria, apenas os custos de transporte são considerados.

O primeiro problema abordado é o de P-Medianas, proposto por Hakimi (1964). Basicamente, o problema consiste em localizar p facilidades (medianas) em uma rede, com o objetivo de minimizar a soma de todas as distâncias de cada ponto de demanda à sua mediana mais próxima. A natureza do problema é discreta, sendo considerado um conjunto finito de potenciais localidades para a instalação das facilidades.

Os problemas de centralização de facilidades se assemelham ao de P-Mediana. Porém, busca a minimização da máxima distância entre os pontos de demanda e as instalações mais próximas dos mesmos, sendo conhecido como problemas de MiniMax. Quando as potenciais localidades das facilidades são pré-determinadas em alguns nós da rede, o problema é conhecido como *vertex center problem*, tendo sido abordado por Halfin (1974), Hedetniemi et al. (1981) e Farley (1982). Quando as instalações podem ser localizadas em qualquer lugar da rede o problema é conhecido como *Absolute Center Problem*, tendo sido estudado por Hakimi (1964), Garfinkel et al. (1977) e Kariv e Hakimi (1979) (BRANDEAU e CHIU, 1989).

No problema de cobertura de um conjunto de locais, abordado por Minieka (1970) e Elzinga e Hearn (1972); é determinado qual o número mínimo de instalações necessário para que todas as demandas sejam atendidas, dado uma distância ou tempo máximos de atendimento. O objetivo deste problema é minimizar o custo fixo das instalações.

Uma variação dele foi proposta por Church e ReVelle (1974), conhecido por problema de máxima cobertura. Neste problema, busca-se a maximização da demanda coberta para uma distância ou tempo máximos de atendimento pré-determinados. Utiliza-se uma quantidade fixa de facilidades determinada a priori, sendo também pré-determinado o conjunto finito de potenciais localidades destas.

Os problemas de localização de facilidades com custo fixo e sem capacidade (*Fixed Charge Facility Location Problems*) são variações do problema de P-Medianas, onde o custo fixo das instalações é computado na função objetivo e há a relaxação da restrição relativa ao número máximo de facilidades. O objetivo é minimizar o custo total da rede, composto pelos custos fixos de construção e operação das facilidades e pelos custos de transporte. Em uma outra variação, os problemas de localização com custo fixo e capacidade (*Capacitated Facility Location Problems*), as capacidades das facilidades são limitadas, conforme abordado por Nozick e Turnquist (2001).

b. Modelos que consideram Custos de Manutenção de Estoques Simplificados

Os modelos de otimização apresentados na seção anterior consideravam apenas os custos de transporte e, em alguns casos, os custos fixos de instalação das facilidades. Os trabalhos revisados nessa seção já incorporam alguns custos de estoque, embora de maneira simplificada.

Jayaraman (1998) apresenta uma adaptação do modelo clássico de localização de facilidades com custo fixo e capacidade, incorporando múltiplos produtos e modais de transporte. Os custos dos estoques de ciclo em cada fábrica e armazém, bem como os custos do estoque em trânsito são computados na função objetivo, junto ao custo de abertura e operação de fábricas e armazéns e os custos de transporte ao longo da rede; determinando qual modal de transporte a ser utilizado para cada produto, o número e localização das facilidades a serem alocadas e os parâmetros do estoque de ciclo de cada instalação. Este modelo foi utilizado por Davariz (2006) para análise da rede de produção e distribuição do Biodiesel no Brasil.

Uma extensão do modelo de Jayaraman (1998) é abordado por Jayaraman e Pirkul (2001) com a modelagem de um sistema de três elos (fornecedores-fábricas, fábricas-armazéns, armazéns-clientes). O modelo determina a localização das fábricas e dos armazéns, bem como o fluxo de cada um dos múltiplos produtos através da rede, também com o objetivo de minimizar o custo total. Foram utilizados relaxação lagrangeana e métodos heurísticos para a solução do problema.

Ambrosino e Scutellà (2004) incluíram armazéns intermediários na sua formulação, além de também considerar a determinação de rotas de distribuição que podem passar por vários mercados consumidores a cada envio de produtos. Desta

forma, a distribuição não é necessariamente feita diretamente dos armazéns a cada um dos mercados.

c. Modelos que consideram os Efeitos de Centralização de Estoques

A simplificação da consideração dos custos de estoque, tratando apenas os estoques de ciclo e estoques em trânsito, resulta em equações de natureza linear. Com isso, os modelos apresentados anteriormente podem ser resolvidos através das técnicas clássicas utilizadas para a solução de problemas de programação inteira mista (ver Wolsey, 1998). Já a consolidação dos estoques de segurança implica no cômputo de uma função da raiz quadrada do somatório das variâncias dos estoques centralizados em um dado armazém, o que resulta na não-linearidade dos modelos.

O trabalho de Das e Tyagi (1997) apresenta uma das primeiras análises formais da decisão centralização de estoque em uma perspectiva mais ampla, com um modelo conjunto de localização de facilidades e tratamento de estoques de ciclo e de segurança não simplificado. São representadas expressões para vários elementos do custo total e seus efeitos individuais e combinados sobre a centralização dos estoques é examinada em cinco cenários: apenas o estoque de ciclo; estoque de ciclo e estoque de segurança; apenas custo de transportes; estoque de ciclo e custos de transporte; e estoque de ciclo, estoque de segurança e custo de transporte. Para cada cenário são analisados os *tradeoffs* entre centralização e descentralização, com a exemplificação de uma aplicação prática.

d. Modelos que consideram os Efeitos de Centralização de Estoques aplicados à Indústria Petrolífera

Das e Tyagi (1997) exemplificam o seu modelo que inclui o tratamento do estoque de ciclo, estoque de segurança e custo de transporte através de uma aplicação no transporte de óleo dos fornecedores para os clientes. O modelo utilizado no artigo considera apenas um fornecedor, e o enfoque se dá sobre a análise de grau de centralização para cada cenário, e não nos métodos empregados para a solução.

Montebeller Júnior (2009) desenvolveu um modelo que permite a consideração das demandas dos mercados com correlações e desvios padrão diferentes de zero, a consideração dos *lead times* de ressuprimento dos armazéns com desvios padrão

também diferentes de zero e a adoção da premissa do *cross-filling* (ou transferência regular). Foi conduzida uma análise de sensibilidade no intuito de se obterem insights gerenciais, onde foram gerados e otimizados diferentes cenários aleatórios, com o posterior tratamento estatístico dos resultados das otimizações.

Através das análises realizadas, Montebeller Júnior (2009) concluiu que para equipamentos e peças de reposição para petróleo e gás a política mais adequada seria a centralização total. Isso se deve ao fato destes produtos apresentarem valores mais altos, e conseqüentemente, maiores custos de manutenção de estoques. Além disso, a concentração da produção de tais produtos em determinadas áreas geográficas de um país ou do mundo acarreta naturalmente uma ampla faixa de *lead times* de ressuprimento entre fábricas e possíveis armazéns, e provavelmente altos *lead times*. Isso dependeria, em primeiro lugar, da distância entre a fábrica e a região, ou país, que albergaria o possível armazém, considerando-se que não há restrições de ordem política para a abertura de armazéns em diferentes regiões do mundo. Em segundo lugar, o nível de serviço do transporte disponível nas diversas regiões onde um armazém pode ser localizado pode variar bastante, contribuindo ainda mais para aumentar a amplitude dos *lead times* de ressuprimento.

Em sua dissertação, Silva (2012) realiza uma avaliação do gerenciamento dinâmico de estoques para o abastecimento de materiais e equipamentos para as unidades da cadeia produtora de petróleo a partir das plataformas produtoras de petróleo extraído da camada pré-sal. Devido à natureza dos campos de extração em ambiente marítimo, foram analisadas as unidades mais comumente utilizadas nas operações de abastecimento denominadas Unidades de Produção (UPs); sendo que a demanda varia ao longo do tempo (modelo dinâmico). São resolvidos dois modelos propostos, sendo um o original não linear e o outro uma adaptação deste, linearizado por partes.

fornecedores e dos depósitos abertos, os custos de transporte entre os fornecedores e depósitos, bem como entre os depósitos e clientes; e o custo de manutenção de estoques em cada depósito. Como restrições têm-se a capacidade dos fornecedores e dos armazéns; a conservação de fluxo nos armazéns, onde o fluxo dos fornecedores aos armazéns deve ser igual ao fluxo dos armazéns aos clientes (a soma das suas demandas no *lead time*) mais o estoque de segurança; a demanda dos clientes, que deve ser atendida na sua totalidade; e a cobertura dos clientes, sendo que cada cliente deve ser abastecido por exatamente um depósito (não é permitido *cross-filling*).

Para Ferreira Filho e Gendron (2012), a decisão estratégica envolve duas componentes: A primeira componente é o problema de localização/alocação, que determina o número e localização dos fornecedores e armazéns e atribui os clientes aos armazéns; enquanto a segunda componente lida com a determinação do nível de estoque de ciclo e de segurança nos armazéns, visando prover certo nível de serviço no atendimento aos clientes.

3.1 Suposições e Hipóteses

Algumas hipóteses devem ser assumidas e pré-requisitos cumpridos para que para que o problema possa ser modelado conforme as formulações que serão apresentadas, segundo Ferreira Filho e Gendron (2012):

- a. Cada fornecedor tem sua localização fixa e pré-determinada;
- b. Cada cliente tem sua localização fixa e pré-determinada;
- c. As possíveis alocações para os armazéns são pré-determinadas e finitas;
- d. O número de armazéns é finito e no máximo igual ao número de possíveis localizações;
- e. As demandas de cada cliente são independentes e seguem uma distribuição normal com desvio padrão diferente de zero e constante ao longo do tempo;
- f. O nível de serviço requerido por cada cliente é considerado na forma de um fator de segurança que garante certa probabilidade de não ocorrência de faltas de estoque (*stock-out*);
- g. Os clientes devem ser servidos por apenas um armazém, isto é, não é permitido *cross-filling*;
- h. Cada armazém pode servir a mais de um cliente;

- i. O estoque de segurança de todos os clientes servidos por um determinado armazém é mantido neste armazém;
- j. Os estoques de ciclo e estoques de segurança são considerados apenas nos armazéns;
- k. Não são considerados estoques nos fornecedores e nem nos clientes;
- l. O modelo de gerenciamento de estoque nos armazéns é assumido como sendo o do LEC (Lote Econômico de Compra) e PP (Ponto de Pedido);
- m. Os custos unitários de estocagem são funções do *lead time* de ressuprimento dos depósitos para cada fornecedor;
- n. Os custos unitários de transporte entre fornecedores e depósitos e entre depósitos e clientes depende da distância entre eles.

4. Formulações do Problema

“Essentially, all models are wrong, but some are useful.”

— George Edward Pelham Box

Podemos encarar o Problema de Localização de Armazéns a partir de duas perspectivas. A primeira, tratada na formulação contendo variáveis de índice duplo, secciona o problema entre os dois elos. No primeiro elo o fluxo de produtos entre os fornecedores e cada depósito (x_{jk}) é determinado, visando suprir os estoques de ciclo (representado pelo somatório da média das demandas alocadas a cada armazém $\sum_l \mu_l y_{kl}$) e de segurança (SS_k) centralizados nestes. No segundo elo é fixada a alocação dos clientes aos depósitos, sendo estes fluxos equilibrados através das restrições de conservação de fluxo, conforme figura abaixo:



Figura 4.1 – Representação do modelo de índice duplo

A segunda perspectiva a partir da qual o problema pode ser visto é enxergando o fluxo total de produtos através da cadeia que visa atender ao estoque de ciclo de cada cliente (x_{jkl}), somando isso à parcela referente ao estoque de segurança dos clientes alocados a cada depósito (xs_{jk}). Esta abordagem é tratada na formulação de índice triplo, representada na figura 4.2:



Figura 4.2 – Representação do modelo de índice tripla

A lista dos símbolos, conjuntos, parâmetros e variáveis utilizadas segue no quadro abaixo:

Índices e Conjuntos

$j \in J$	Índice de potenciais fornecedores a serem abertos
$k \in K$	Índice de potenciais armazéns a serem abertos
$l \in L$	Índice de clientes
$r \in R$	Índice de segmentos na aproximação por linearização por partes

Parâmetros

f_j	Custo anual de abertura de um fornecedor na localidade j
P_j	Capacidade de produção do fornecedor j
c_{jk}	Custo unitário de produção e envio de um produto do fornecedor j para o armazém k
g_k	Custo anual de abertura de um armazém na localidade k
h_k	Custo anual unitário de manutenção de estoques no armazém k
V_k	Capacidade de armazenagem do armazém k
Γ	Horizonte de planejamento: número de dias de operação em um ano.
t_{kl}	Custo unitário de transporte de produtos entre o armazém k e o cliente l
φ_{kl}	Lead time médio de transporte do depósito k para o cliente l
ξ_{kl}	Cobertura de clientes: 1 se o cliente l pode ser atendido pelo depósito k , 0 caso contrário.
μ_l	Média da demanda diária de produtos do cliente l
σ_l	Desvio padrão da demanda diária de produtos do cliente l

σ_l^2	Variância da demanda diária de produtos do cliente l
α	Nível de serviço desejado para os clientes
Z_α	Fator de segurança correspondente ao nível de serviço α desejado pelos clientes
F_k^r	Coefficiente linear do segmento r na aproximação por linearização por partes
C_k^r	Coefficiente angular do segmento r na aproximação por linearização por partes
Variáveis	
SS_k	Estoque de segurança consolidado no armazém k
w_j	Abertura de fornecedores: 1 se o fornecedor j é aberto, 0 caso contrário
x_{jk}	Número de unidades produzidas no fornecedor j e enviadas para o armazém k (modelo de índice duplo)
x_{jkl}	Número de unidades produzidas no fornecedor j e enviadas para o armazém k a fim de atender à demanda do cliente l (modelo de índice triplo)
xs_{jk}	Número de unidades produzidas no fornecedor j e enviadas para o armazém k para suprir o seu estoque de segurança (modelo de índice triplo)
z_k	Abertura de armazéns: 1 se o armazém k é aberto, 0 caso contrário
y_{kl}	Alocação de clientes: 1 se o cliente l é alocado ao armazém k , 0 caso contrário
$v_k^r, v_k, v_{kl}, v_{kl}^r$	Variáveis auxiliares contínuas da aproximação por linearização por partes
u_k^r	Variáveis auxiliares binárias da aproximação por linearização por partes

Quadro 1 – Lista de Símbolos, Conjuntos, Parâmetros e Variáveis

4.1 Formulação de Índice Duplo

A formulação básica não-linear do modelo a partir da primeira perspectiva mencionada é dada abaixo:

$$\text{Minimizar} \quad \sum_{j \in J} f_j w_j + \sum_{j \in J} \sum_{k \in K} (c_{jk} + h_k) x_{jk} + \sum_{k \in K} g_k z_k + \Gamma \sum_{k \in K} \sum_{l \in L} t_{jk} \mu_l y_{kl} \quad (1)$$

$$\text{Sujeito a} \quad SS_k = Z_\alpha \sqrt{\sum_{l \in L} \varphi_{kl} \sigma_l^2 y_{kl}} \quad \forall k \in K \quad (2)$$

$$\sum_{j \in J} x_{jk} = \Gamma \sum_{l \in L} \mu_l y_{kl} + SS_k \quad \forall k \in K \quad (3)$$

$$\sum_{k \in K} x_{jk} \leq P_j w_j \quad \forall j \in J \quad (4)$$

$$\Gamma \sum_{l \in L} \mu_l y_{kl} + SS_k \leq V_k z_k \quad \forall k \in K \quad (5)$$

$$\sum_{k \in K} y_{kl} = 1 \quad \forall l \in L \quad (6)$$

$$y_{kl} \leq \xi_{kl} z_k \quad \forall l \in L, \forall k \in K \quad (7)$$

$$x_{jk} \geq 0 \quad \forall j \in J, \forall k \in K \quad (8)$$

$$w_j, z_k \in \{0,1\} \quad \forall j \in J, \forall k \in K \quad (9)$$

$$y_{kl} \in \{0,1\} \quad \forall l \in L, \forall k \in K \quad (10)$$

A função objetivo (1) minimiza a soma dos custos de abertura e operação de fornecedores e armazéns (amortizados anualmente), os custos de transporte de produtos dos fornecedores para os depósitos, os custos de produção e transporte de produtos dos fornecedores para os depósitos, o custo de manutenção de estoques de ciclo e de segurança nos armazéns, e o custo de transporte dos produtos dos armazéns para os clientes. A restrição (2) contém o cálculo do estoque de segurança para cada armazém aberto. A restrição (3) determina a conservação de fluxo nos armazéns, que assegura que de todos os produtos que passam dos fornecedores para o armazém, uma parte atende à demanda dos clientes alocados àquele depósito e o restante é armazenado como estoque de segurança. As restrições (4) e (5) limitam a capacidade dos fornecedores e armazéns, e permitem a utilização apenas daqueles que estejam abertos. A restrição (6) impede o *cross-filling*, isto é, determina que cada cliente deve ser atendido por apenas um armazém; e a restrição (7) limita a cobertura de clientes, impedindo a alocação de clientes a depósitos que não possam atendê-los (por questões burocráticas ou limites de

distância, por exemplo). As restrições (8), (9) e (10) satisfazem aos requisitos de integralidade e não negatividade das variáveis.

Dada a não linearidade do modelo original devido à natureza da fórmula que determina o estoque de segurança na restrição (2), é necessário utilizar um procedimento para aproximar esta função por uma função linear, e com isso permitir que o modelo seja resolvido utilizando os métodos tradicionais empregados para a solução de problemas de programação linear inteira mista. Utilizando o procedimento de linearização por partes proposto por Croxton e Gendron (2003); Croxton, Gendron e Magnanti (2007) e Frangioni e Gendron (2009), a função $SS_k = Z_\alpha \sqrt{\sum_{l \in L} \varphi_{kl} \sigma_l^2 y_{kl}}$ é substituída por $Z_\alpha \sum_{r \in R} (F_k^r u_k^r + C_k^r v_{kl}^r)$, segmentada em r -segmentos. O coeficiente linear de cada um dos r -segmentos é dado por F_k^r , e o coeficiente angular é dado por C_k^r . A variável binária u_k^r determina qual segmento deverá ser utilizado na aproximação, e v_{kl}^r realiza a varredura no segmento selecionado. Uma representação gráfica deste procedimento pode ser visualizada abaixo:

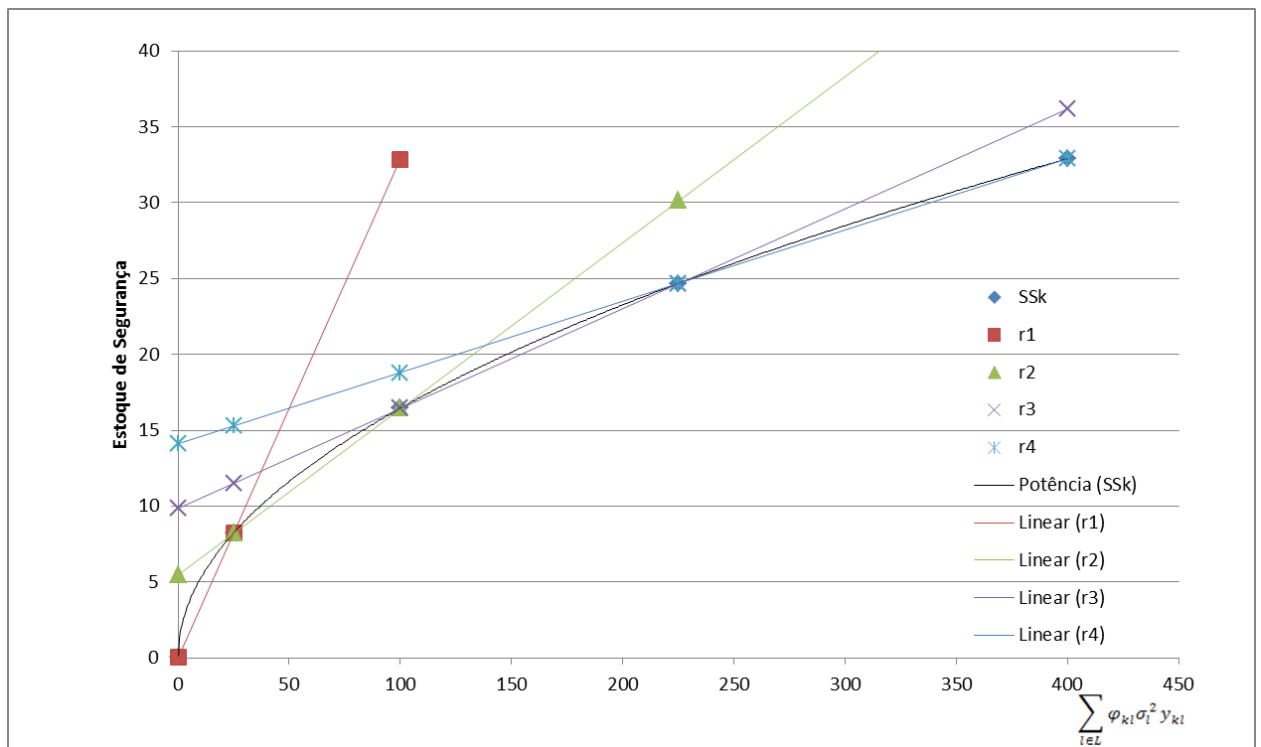


Figura 4.3 - Aproximação por Linearização por Partes

Frangioni e Gendron (2009) e Ferreira Filho e Gendron (2012) utilizam uma série de restrições redundantes com o objetivo de fortalecer a formulação do problema,

limitando melhor a envoltória convexa do modelo relaxado. O modelo de índice duplo linearizado por partes e contando com a formulação reforçada pode ser visto abaixo.

$$\text{Minimizar} \quad \sum_{j \in J} f_j w_j + \sum_{j \in J} \sum_{k \in K} (c_{jk} + h_k) x_{jk} + \sum_{k \in K} g_k z_k + \Gamma \sum_{k \in K} \sum_{l \in L} t_{jk} \mu_l y_{kl} \quad (1)$$

$$\text{Sujeito a} \quad SS_k = Z_\alpha \sum_{r \in R} \left(F_k^r u_k^r + C_k^r \sum_{l \in L} v_{kl}^r \right) \quad \forall k \in K \quad (2a)$$

$$\sum_{j \in J} x_{jk} = \Gamma \sum_{l \in L} \mu_l y_{kl} + SS_k \quad \forall k \in K \quad (3)$$

$$\sum_{k \in K} x_{jk} \leq P_j w_j \quad \forall j \in J \quad (4)$$

$$x_{jk} \leq P_j w_j \quad \forall j \in J, \forall k \in K \quad (14)$$

$$\sum_{j \in J} x_{jk} \leq V_k z_k \quad \forall k \in K \quad (15)$$

$$\sum_{j \in J} \sum_{k \in K} x_{jk} \geq \Gamma \sum_{l \in L} \mu_l \quad (16)$$

$$\sum_{j \in J} \sum_{k \in K} x_{jk} \leq \Gamma \sum_{l \in L} \mu_l + Z_\alpha \left\{ \sum_{k \in K} \left(\max_r F_k^r \right) + \left(\max_{k,r} C_k^r \right) \sum_{l \in L} \varphi_{kl} \sigma_l^2 \right\} \quad (17)$$

$$\Gamma \sum_{l \in L} \mu_l y_{kl} + SS_k \leq V_k z_k \quad \forall k \in K \quad (5)$$

$$\sum_{k \in K} y_{kl} = 1 \quad \forall l \in L \quad (6)$$

$$y_{kl} \leq \xi_{kl} z_k \quad \forall l \in L, \forall k \in K \quad (7)$$

$$\sum_{l \in L} \varphi_{kl} \sigma_l^2 y_{kl} = \sum_{l \in L} \sum_{r \in R} v_{kl}^r \quad \forall k \in K \quad (11)$$

$$L_k^{r-1} u_k^r \leq \sum_{l \in L} v_{kl}^r \leq L_k^r u_k^r \quad \forall k \in K, \forall r \in R \quad (12)$$

$$\sum_{r \in R} u_k^r \leq 1 \quad \forall k \in K \quad (13)$$

$$v_{kl}^r \leq \varphi_{kl} \sigma_l^2 u_k^r \quad \forall k \in K, \forall l \in L, \forall r \in R \quad (18)$$

$$x_{jk}, v_{kl}^r \geq 0 \quad \forall j \in J, \forall k \in K, \forall l \in L, \forall r \in R \quad (8)$$

$$w_j, z_k, u_k^r \in \{0,1\} \quad \forall j \in J, \forall k \in K, \forall r \in R \quad (9)$$

$$y_{kl} \in \{0,1\} \quad \forall l \in L, \forall k \in K \quad (10)$$

As restrições (2a), (11), (12) e (13) são responsáveis pela aproximação por linearização por partes, sendo válido notar que o valor de $L_k^0 = 0, \forall k \in K$. As restrições redundantes (14), (15), (16), (17) e (18) são as que fortalecem a formulação.

4.2 Formulação de Índice Triplo

A formulação básica não-linear do modelo a partir da segunda perspectiva mencionada é dada abaixo:

$$\begin{aligned} \text{Minimizar} \quad & \sum_{j \in J} f_j w_j + \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} (c_{jk} + t_{kl}) x_{jkl} + \sum_{k \in K} g_k z_k + \sum_{j \in J} \sum_{k \in K} c_{jk} x_{Sjk} \\ & + \sum_{k \in K} h_k \left(\sum_{j \in J} \sum_{l \in L} x_{jkl} + \sum_{j \in J} x_{Sjk} \right) \end{aligned} \quad (19)$$

$$\text{Sujeito a} \quad SS_k = Z_\alpha \sqrt{\sum_{l \in L} \varphi_{kl} \sigma_l^2 y_{kl}} \quad \forall k \in K \quad (2)$$

$$\sum_{j \in J} x_{jkl} = \Gamma \mu_l y_{kl} \quad \forall k \in K, \forall l \in L \quad (20)$$

$$\sum_{k \in K} \sum_{l \in L} x_{jkl} + \sum_{k \in K} x_{Sjk} \leq P_j w_j \quad \forall j \in J \quad (21)$$

$$\sum_{k \in K} \sum_{l \in L} x_{jkl} + \sum_{k \in K} x_{Sjk} \leq V_k z_k \quad \forall k \in K \quad (22)$$

$$\sum_{j \in J} x_{Sjk} = SS_k \quad \forall k \in K \quad (5)$$

$$\sum_{k \in K} y_{kl} = 1 \quad \forall l \in L \quad (6)$$

$$y_{kl} \leq \xi_{kl} z_k \quad \forall l \in L, \forall k \in K \quad (7)$$

$$x_{jkl} \geq 0, x_{Sjk} \geq 0 \quad \forall j \in J, \forall k \in K, \forall l \in L \quad (23)$$

$$w_j, z_k \in \{0,1\} \quad \forall j \in J, \forall k \in K \quad (9)$$

$$y_{kl} \in \{0,1\} \quad \forall l \in L, \forall k \in K \quad (10)$$

Assim como no modelo de índice duplo, a função objetivo (19) busca minimizar o custo total de instalação, logística e operação da cadeia; e as restrições seguem de maneira análoga. O modelo de índice triplo linearizado por partes e contando com a formulação reforçada segue abaixo.

$$\begin{aligned} \text{Minimizar} \quad & \sum_{j \in J} f_j w_j + \sum_{j \in J} \sum_{k \in K} \sum_{l \in L} (c_{jk} + t_{kl}) x_{jkl} + \sum_{k \in K} g_k z_k + \sum_{j \in J} \sum_{k \in K} c_{jk} x_{sjk} \\ & + \sum_{k \in K} h_k \left(\sum_{j \in J} \sum_{l \in L} x_{jkl} + \sum_{j \in J} x_{sjk} \right) \end{aligned} \quad (19)$$

$$\text{Sujeito a} \quad SS_k = Z_\alpha \sum_{r \in R} \left(F_k^r u_k^r + C_k^r \sum_{l \in L} v_{kl}^r \right) \quad \forall k \in K \quad (2a)$$

$$\sum_{j \in J} x_{jkl} = \Gamma \mu_l y_{kl} \quad \forall k \in K, \forall l \in L \quad (20)$$

$$\sum_{k \in K} \sum_{l \in L} x_{jkl} + \sum_{k \in K} x_{sjk} \leq P_j w_j \quad \forall j \in J \quad (21)$$

$$\sum_{j \in J} \sum_{l \in L} x_{jkl} + \sum_{j \in J} x_{sjk} \leq V_k z_k \quad \forall k \in K \quad (22)$$

$$\sum_{j \in J} x_{sjk} = SS_k \quad \forall k \in K \quad (5)$$

$$\Gamma \sum_{l \in L} \mu_l y_{kl} + SS_k \leq V_k z_k \quad \forall k \in K \quad (24)$$

$$\sum_{k \in K} y_{kl} = 1 \quad \forall l \in L \quad (25)$$

$$y_{kl} \leq \xi_{kl} z_k \quad \forall l \in L, \forall k \in K \quad (26)$$

$$\sum_{l \in L} \varphi_{kl} \sigma_l^2 y_{kl} = \sum_{l \in L} \sum_{r \in R} v_{kl}^r \quad \forall k \in K \quad (11)$$

$$L_k^{r-1} u_k^r \leq \sum_{l \in L} v_{kl}^r \leq L_k^r u_k^r \quad \forall k \in K, \forall r \in R \quad (12)$$

$$\sum_{r \in R} u_k^r \leq 1 \quad \forall k \in K \quad (13)$$

$$v_{kl}^r \leq \varphi_{kl} \sigma_l^2 u_k^r \quad \forall k \in K, \forall l \in L, \forall r \in R \quad (18)$$

$$x_{jkl}, x_{sjk}, v_{kl}^r \geq 0 \quad \forall j \in J, \forall k \in K, \forall l \in L, \forall r \in R \quad (23)$$

$$w_j, z_k, u_k^r \in \{0,1\} \quad \forall j \in J, \forall k \in K, \forall r \in R \quad (9)$$

$$y_{kl} \in \{0,1\} \qquad \forall l \in L, \forall k \in K \qquad (10)$$

As restrições (2a), (11), (12) e (13) são responsáveis pela aproximação por linearização por partes, tendo novamente o valor de $L_k^0 = 0, \forall k \in K$. As restrições redundantes (18) e (24) fortalecem a formulação.

5. Metodologia

“I am the one who knocks!”

— Walter White

O trabalho realizado pode ser dividido em quatro grandes blocos: o desenvolvimento dos modelos, testes iniciais e estudos de linearização; o desenvolvimento do método gerador das instâncias de teste; o projeto e implementação das interfaces de comunicação entre os modelos, o *solver* e os conjuntos de dados; e a realização dos experimentos propostos.

O *solver* escolhido para o processamento dos modelos foi o *IBM® ILOG CPLEX Optimizer*, reconhecido tanto no meio científico quanto industrial pela sua eficácia em lidar com problemas de grande porte. Optou-se pela simplicidade da linguagem OPL (*Optimization Programming Language*) para a modelagem, e as interfaces de comunicação foram criadas utilizando a camada *Concert* em integração com o C++. O código em OPL dos modelos está disponível no Anexo Apêndice III, e o código em C++ das interfaces de comunicação no Anexo Apêndice IV.

5.1 Segmentação do Eixo X

A equação (2) dos modelos, responsável pela determinação dos estoques de segurança em cada um dos armazéns, foi aproximada pela equação (2a) utilizando o método de linearização por partes. Foram testadas cinco possíveis configurações para a sua segmentação, isto é, a determinação dos valores de L_k^r em que cada segmento de reta utilizado na linearização estará ativo quando a variável u_k^r for igual a 1. Das cinco configurações, duas foram escolhidas para a realização dos experimentos.

A primeira, mais intuitiva e amplamente adotada na literatura (CROXTON; GENDRON; MAGNANTI, 2007; FERREIRA FILHO; GENDRON, 2012; FRANGIONI; GENDRON, 2009), divide o eixo das abscissas (que representa os $\sum_{l \in L} v_{kl}^r$) em intervalos igualmente espaçados de L_k^r , conforme mostrado na figura 5.1.

Os limitantes L_k^r são diretamente proporcionais ao índice do segmento r (sendo $L_k^0 = 0$), determinados pela fórmula abaixo:

$$L_k^r = \frac{r}{|R|} \cdot \text{Max} \sum_{l \in L} v_{kl}^r \quad r = 1, 2, \dots, |R|$$

$$\forall k \in K$$

Pela natureza linear da fórmula utilizada para a divisão, esta será referenciada de agora em diante como Segmentação Linear, ou SL. O limite superior $\text{Max} \sum_{l \in L} v_{kl}^r$ foi determinado de maneira direta, calculado como:

$$\text{Max} \sum_{l \in L} v_{kl}^r = \sum_{l \in L} (\text{Max} \varphi_{kl}) \sigma_l^2 \quad \forall k \in K$$

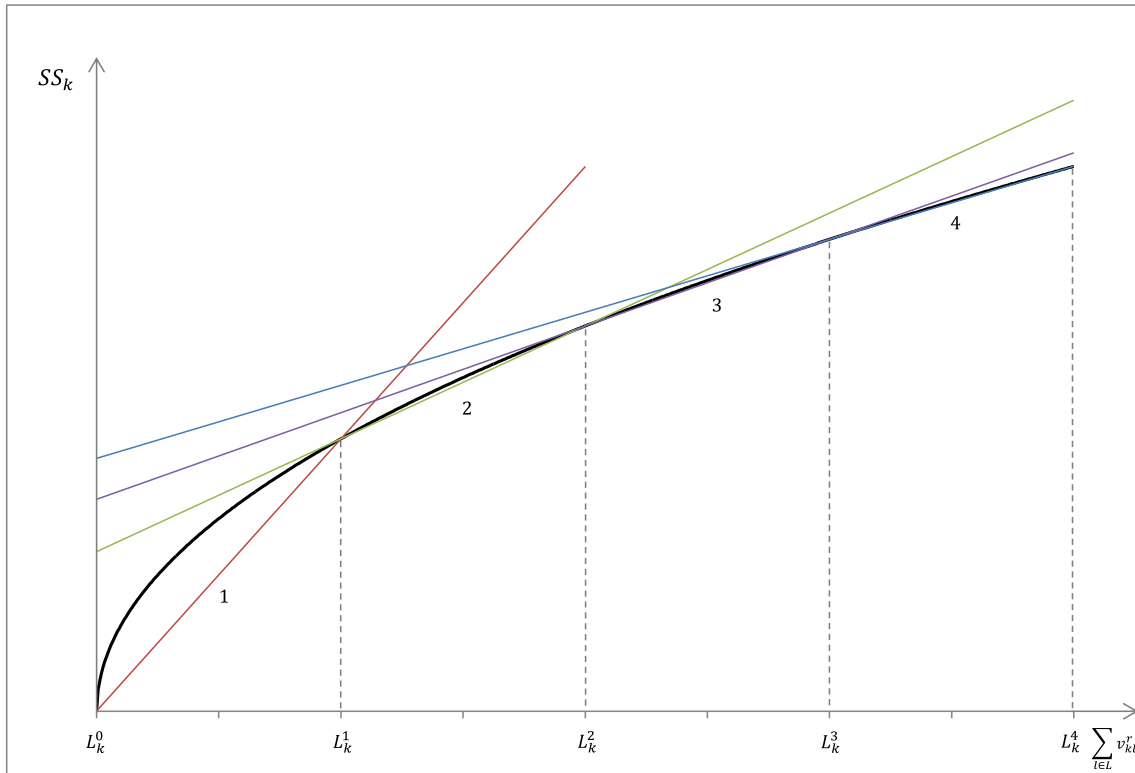


Figura 5.1 – Segmentação pela divisão em intervalos igualmente espaçados (SL)

A segunda forma de segmentação escolhida utilizou uma função inversa à que determina o estoque de segurança, isto é, uma função quadrática. Ela será referida de agora em diante como Segmentação Quadrática (SQ), e pode ser vista na figura 5.2. Os limitantes L_k^r são proporcionais ao quadrado do índice do segmento r , conforme abaixo, sendo o $\text{Max} \sum_{l \in L} v_{kl}^r$ adotado idêntico ao de SL:

$$L_k^r = \frac{r^2}{|R|^2} \cdot \text{Max} \sum_{l \in L} v_{kl}^r \quad r = 1, 2, \dots, |R|$$

$$\forall k \in K$$

A configuração quadrática foi escolhida por ser a que minimiza a diferença entre a área sob a curva da função do estoque de segurança e a área delimitada pelos segmentos nos intervalos de L_k^r , determinadas pelas integrais definidas. Com isso, espera-se que seja a configuração com o menor erro médio de linearização. As outras configurações testadas foram uma função cúbica, uma função do logaritmo natural e a própria função raiz quadrada.

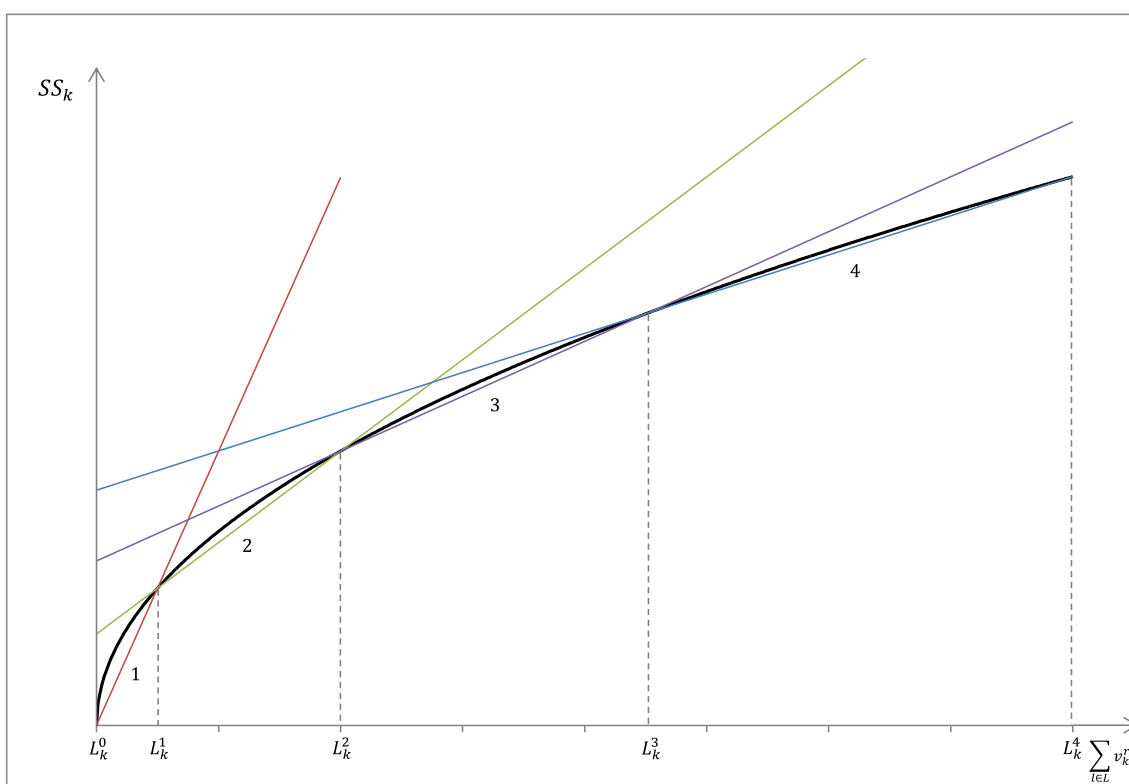


Figura 5.2 – Segmentação pela divisão quadrática (SQ)

5.2 Geração de Instâncias de Teste (Projeto de Experimentos)

Os conjuntos de dados de teste foram gerados utilizando um sistema implementado em VBA (*Visual Basic for Applications*) no Microsoft® Excel 2010, cujo diagrama de classes do projeto pode ser visto no Anexo Apêndice I e algumas capturas de tela podem ser vistas no Anexo Apêndice II. As instâncias especificadas são geradas automaticamente, junto com os respectivos arquivos de extensão *.dat*, que efetuam o

mapeamento entre as variáveis do modelo do OPL e a planilha que contém os dados de entrada.

Cada instância a ser gerada recebe como *inputs* o número de fornecedores, armazéns e clientes que deve conter; bem como o Coeficiente de Utilização dos Fornecedores (CUF), o Coeficiente de Utilização dos Armazéns (CUA) e a listagem dos números de segmentos em que a função linearizada do estoque de segurança deve ser repartida.

O CUF é um coeficiente utilizado no cálculo das capacidades de cada um dos fornecedores, podendo ser visto como o valor esperado do FUF (Fator de Utilização dos Fornecedores). Este fator, por sua vez, é dado pelo somatório das demandas μ_l e do máximo estoque de segurança possível de todos os clientes (ou $Z_\alpha \sqrt{\text{Max} \sum_{l \in L} v_{kl}^r}$), dividido pelo somatório das capacidades P_j de todos os fornecedores. O CUA se assemelha ao CUF e é utilizado da mesma forma, podendo ser visto como o valor esperado do FUA (Fator de Utilização dos Armazéns).

A nomenclatura das instâncias segue o padrão **IL-EX-|J|-|K|-|L| Cenário (N) |R|-FUF-FUA**, onde:

- |J|: Número de Fornecedores;
- |K|: Número de Armazéns;
- |L|: Número de Clientes;
- *Cenário*: Aleat, Gap, nSeg, etc. referente à rodada de experimentos realizados;
- *N*: Índice dentro do conjunto gerado de instâncias de mesma dimensão, isto é, de mesmo número de fornecedores, armazéns, clientes, cenário, CUF e CUA;
- |R|: Número de Segmentos;
- *FUF*: Fator de Utilização da Capacidade dos Fornecedores;
- *FUA*: Fator de Utilização da Capacidade dos Armazéns.

Um exemplo de instância gerada é **IL-EX-1-5-10 Gap (01) 05-0,40-0,19**, que tem um fornecedor disponível para alocação, cinco armazéns disponíveis, dez clientes a serem servidos, pertence a um conjunto de instâncias geradas para o experimento de mensuração do Gap, tem a aproximação do estoque de segurança por linearização por partes dividida em cinco segmentos, o fator de utilização da capacidade dos fornecedores é de 40% e dos armazéns é de 19%. Junto com arquivo de dados da instância no formato do Excel 2010 (*IL-EX-1-5-10 Gap (01) 05-0,40-0,19.xlsx*), são criados quatro arquivos com a extensão .dat: *IL-EX-1-5-10 Gap (01) 05-0,40-0,19 - ID-SL.dat*, *IL-EX-1-5-10 Gap (01) 05-0,40-0,19 - ID-SQ.dat*, *IL-EX-1-5-10 Gap (01) 05-0,40-0,19 - IT-SL.dat* e *IL-EX-1-5-10 Gap (01) 05-0,40-0,19 - IT-SQ.dat*.

O sufixo *ID-SL* indica que este é o mapeamento utilizado pelo modelo de índice duplo com a divisão linearmente segmentada em intervalos equivalentes dos limitantes que particionam a função do estoque de segurança na aproximação por linearização por partes. O sufixo *ID-SQ* indica o arquivo .dat utilizado pelo modelo de índice duplo com a divisão quadrática dos segmentos na aproximação por linearização por partes. Os sufixos *IT-SL* e *IT-SQ* são análogos aos anteriores no que concerne à divisão dos segmentos, mas indicam o mapeamento de variáveis para o modelo de índice triplo.

O método de construção dos conjuntos de dados de teste foi inspirado no descrito por Ferreira Filho e Gendron (2012), que por sua vez, se baseia na proposta de Vidyarthi, Elhedhli e Jewkes (2009) e Vidyarthi *et al.* (2007). Entretanto, a fim de prover maior flexibilidade e atender aos requisitos dos experimentos que foram planejados, algumas alterações foram realizadas.

Os exemplos foram gerados utilizando dados das 150 maiores cidades da parte continental dos Estados Unidos, obtidos do censo de 2000 (figura 5.3).



Figura 5.3 – Cidades mais populosas dos Estados Unidos
(Extraído de FERREIRA FILHO; GENDRON, 2012)

Os dados de entrada foram especificados de acordo com os experimentos realizados (exceto para o conjunto **Aleat**, onde foram determinados aleatoriamente), sendo as potenciais localizações de fornecedores (J) e de armazéns (K) limitadas às 20 maiores cidades, os clientes (L) limitados às 150 maiores cidades, e o número $|R|$ de segmentos gerados entre 4 e 20.

No processo de geração das instâncias, o horizonte de planejamento Γ (número de dias produtivos em um ano) foi igual a 1, com os custos de abertura dos fornecedores e armazéns ajustados proporcionalmente. O *lead time* médio de transporte do armazém k para o cliente l (em dias) foi calculado em função do tempo de colocação do pedido e despacho (1 dia) e da distância d_{kl} em milhas entre as localidades, considerando um trajeto percorrido de 300 *mpd* (milhas por dia), conforme abaixo:

$$\varphi_{kl} = 1 + \frac{d_{kl}}{300 \text{ (mpd)}} \quad \begin{array}{l} \forall k \in K \\ \forall l \in L \end{array}$$

O parâmetro de cobertura de clientes ξ_{kl} foi considerado igual a 1 para todos os armazéns e clientes. A demanda média diária μ_l de um dado cliente foi obtida pela divisão da população da cidade l por 1.000. O desvio padrão da demanda foi calculado como:

$$\sigma_l = \mu_l \cdot CV \cdot U[0, 1] \quad \forall l \in L$$

O coeficiente de variação (*CV*) adotado para os experimentos foi de 0,05; e a variância σ_l^2 é o quadrado do desvio padrão σ_l . O nível de serviço dos clientes, α , foi de 95%, o que corresponde a um fator de segurança $Z_\alpha = 1,64$.

O custo unitário de produção e envio de um produto do fornecedor j para o armazém k foi determinado como:

$$c_{jk} = \frac{d_{jk}}{100} \quad \begin{array}{l} \forall j \in J \\ \forall k \in K \end{array}$$

e o custo de envio de um produto do armazém k para o cliente l como:

$$t_{kl} = \frac{d_{kl}}{100} \quad \begin{array}{l} \forall k \in K \\ \forall l \in L \end{array}$$

A capacidade de produção dos fornecedores foi calculada em duas etapas. Na primeira, um valor aleatório chamado de capacidade original foi gerado para cada armazém, utilizando como parâmetros a necessidade total máxima de produtos (somatório das demandas e estoques de segurança máximos para todos os clientes) e o número de fornecedores disponíveis, bem como um número aleatório de distribuição uniforme entre 0 e 2 (buscando, desta forma, que a média seja 1); conforme a fórmula abaixo:

$$P_j^{(Orig)} = U[0, 2] \frac{\sum_{l \in L} \mu_l + \sum_{k \in K} Max SS_k}{|J|} \quad \forall j \in J$$

onde

$$\sum_{k \in K} Max SS_k = \sum_{k \in K} \sum_{l \in L} (Max \varphi_{kl}) \sigma_l^2$$

Também foi calculado Fator de Utilização da Capacidade dos Fornecedores (FUF) desta primeira etapa, chamado de $FUF^{(Orig)}$:

$$FUF^{(Orig)} = \frac{\sum_{l \in L} \mu_l + \sum_{k \in K} Max SS_k}{\sum_{j \in J} P_j^{(Orig)}}$$

A partir disso, pode ser calculada a capacidade de produção a ser efetivamente utilizada nos testes, baseada na $P_j^{(Orig)}$ mas levando também em consideração o Coeficiente de Utilização da Capacidade dos Fornecedores (CUF) esperado para aquele conjunto de dados:

$$P_j = \text{INT} \left(\frac{P_j^{(Orig)} \cdot CUF}{FUF^{(Orig)}} \right) \quad \forall j \in J$$

$$FUF = \frac{\sum_{l \in L} \mu_l + \sum_{k \in K} \text{Max } SS_k}{\sum_{j \in J} P_j}$$

O arredondamento de P_j visa facilitar o processamento posterior da instância pelo *solver*, e pode resultar em uma ligeira diferença entre o CUF e o FUF , mas nada muito expressivo.

Para evitar um grande número de instâncias inviáveis, a fórmula de cálculo de P_1 foi alterada para:

$$P_1 = \text{Max} \left[\text{INT} \left(\frac{P_j^{(Orig)} \cdot CUF}{FUF^{(Orig)}} \right), \text{INT} \left(\sum_{l \in L} \mu_l + \sum_{k \in K} \text{Max } SS_k - \sum_{j \in J - \{1\}} P_j \right) + 1 \right]$$

Os custos de abertura e operação possuem uma parte fixa e uma parte variável (que depende da capacidade P_j), além de um fator de aleatoriedade:

$$f_j = U[500, 1000] + U[1000, 2000] \sqrt{P_j} \quad \forall j \in J$$

A capacidade dos armazéns é determinada de maneira análoga à dos fornecedores:

$$V_k^{(Orig)} = U[0, 2] \frac{\sum_{l \in L} \mu_l + \sum_{k \in K} \text{Max } SS_k}{|K|} \quad \forall k \in K$$

$$FUA^{(Orig)} = \frac{\sum_{l \in L} \mu_l + \sum_{k \in K} \text{Max } SS_k}{\sum_{k \in K} V_k^{(Orig)}}$$

$$V_k = \text{INT} \left(\frac{V_k^{(Orig)} \cdot CUA}{FUA^{(Orig)}} \right) \quad \forall k \in K$$

$$FUA = \frac{\sum_{l \in L} \mu_l + \sum_{k \in K} Max SS_k}{\sum_{k \in K} V_k}$$

V_k foi arredondada para facilitar o processamento posterior da instância pelo *solver*, o que pode resultar em uma pequena diferença entre o CUA e o FUA. V_1 também foi alterada, com o objetivo de evitar muitas instâncias inviáveis:

$$V_1 = \text{Max} \left[\text{INT} \left(\frac{V_1^{(Orig)} \cdot CUA}{FUA^{(Orig)}} \right), \text{INT} \left(\sum_{l \in L} \mu_l + \sum_{k \in K} Max SS_k - \sum_{k \in K - \{1\}} V_k \right) + 1 \right]$$

Análogo a f_j , os custos de abertura e operação dos armazéns também possuem uma parte fixa e uma parte variável, mas esta é proporcional à raiz quarta de V_k :

$$g_k = U[100, 500] + U[500, 1000]^4 \sqrt[4]{V_k} \quad \forall k \in K$$

O custo anual unitário de armazenagem no armazém k foi determinado como:

$$h_k = U[5, 10] \quad \forall k \in K$$

6. Experimentação e Análise

“There are three kinds of lies: lies, damned lies, and statistics.”

— Benjamin Disraeli

Os modelos de índice duplo e triplo, tanto com a segmentação SL quanto SQ, tiveram o seu desempenho comparado para cinco conjuntos de dados gerados para análises mais específicas planejadas para este trabalho. Todos os dados, códigos fonte, e uma versão atualizada deste trabalho (bem como os artigos derivados dele) podem ser acessados através de um repositório hospedado em <http://bit.ly/InventoryLocation> (https://www.dropbox.com/sh/bm27w84qrgb3qqn/AACyC1V7Hfy_VGP-_KGPPV0Na/Pesquisa%20Operacional/Inventory%20Location)

Os experimentos foram realizados em 22 computadores pertencentes à *Amazon Elastic Compute Cloud* (Amazon EC2). Foram utilizadas instâncias (aqui no sentido de unidade de processamento) do tipo *c3.large*, otimizadas para aplicativos que necessitam de grande poder computacional. Cada uma possui 2 vCPUs de 64 bits, com 3,75 GB RAM e sistema operacional Microsoft® Windows Server 2012. Cada vCPU é um *hyperthread* de hardware dos processadores Intel® Xeon® E5-2680v2 de 2,8 GHz Ivy Bridge (AMAZON WEB SERVICES, 2014).

O *solver* escolhido, como dito anteriormente, foi o IBM® ILOG CPLEX Optimizer na sua versão 12.4. É válido salientar que os testes foram realizados sem a utilização do pré-processamento e da inserção de cortes padrão do CPLEX, bem como sem nenhum tipo de relaxação ou método avançado de solução *ad hoc* (geração de linhas e colunas, por exemplo).

6.1 Experimentos ALEAT (Instâncias Aleatórias)

O primeiro conjunto de dados testado foi o **Aleat**, composto por 10 instâncias cujas entradas (número de fornecedores, armazéns, clientes, segmentos, CUF e CUA) foram geradas aleatoriamente; totalizando 40 experimentos (10 instâncias x 4 modelos). Foram excluídas da análise as instâncias cujas soluções se mostraram inviáveis e

aquelas cujos modelos não chegaram à solução ótima dentro do intervalo de 2h (que serão analisadas no experimento **Gap**). Os tempos de processamento para cada um dos modelos se encontra na tabela 6.1:

Tabela 6.1 – Tempo de Processamento para as instâncias Aleat

Instâncias	Tempo (Segundos)			
	ID-SL	ID-SQ	IT-SL	IT-SQ
IL-EX-1-1-33 Aleat (01) 09-0,90-0,88	0,058	0,064	0,089	0,087
IL-EX-9-1-82 Aleat (01) 08-0,69-0,10	0,405	0,162	0,206	0,195
IL-EX-20-3-77 Aleat (01) 17-0,36-0,88	3,152	5,46	23,135	8,175
IL-EX-1-4-136 Aleat (01) 13-0,92-0,79	4,98	11,801	9,692	11,552
IL-EX-9-4-90 Aleat (01) 15-0,84-0,49	5,179	2,402	7,628	3,438
IL-EX-12-15-144 Aleat (01) 08-0,52-0,06	16,209	18,412	54,195	56,447
IL-EX-12-10-114 Aleat (01) 16-0,42-0,02	17,537	17,939	28,265	87,796
IL-EX-17-14-20 Aleat (01) 07-0,91-0,93	18,611	121,026	56,863	73,369
IL-EX-18-17-117 Aleat (01) 10-0,40-0,27	39,047	165,346	92,088	338,662
IL-EX-7-13-67 Aleat (01) 05-0,34-0,67	170,623	110,583	75,24	560,936

A figura 6.1 apresenta uma plotagem dos dados expostos na tabela 6.1, e a figura 6.2 sintetiza o tempo médio para todas as instâncias **Aleat**.

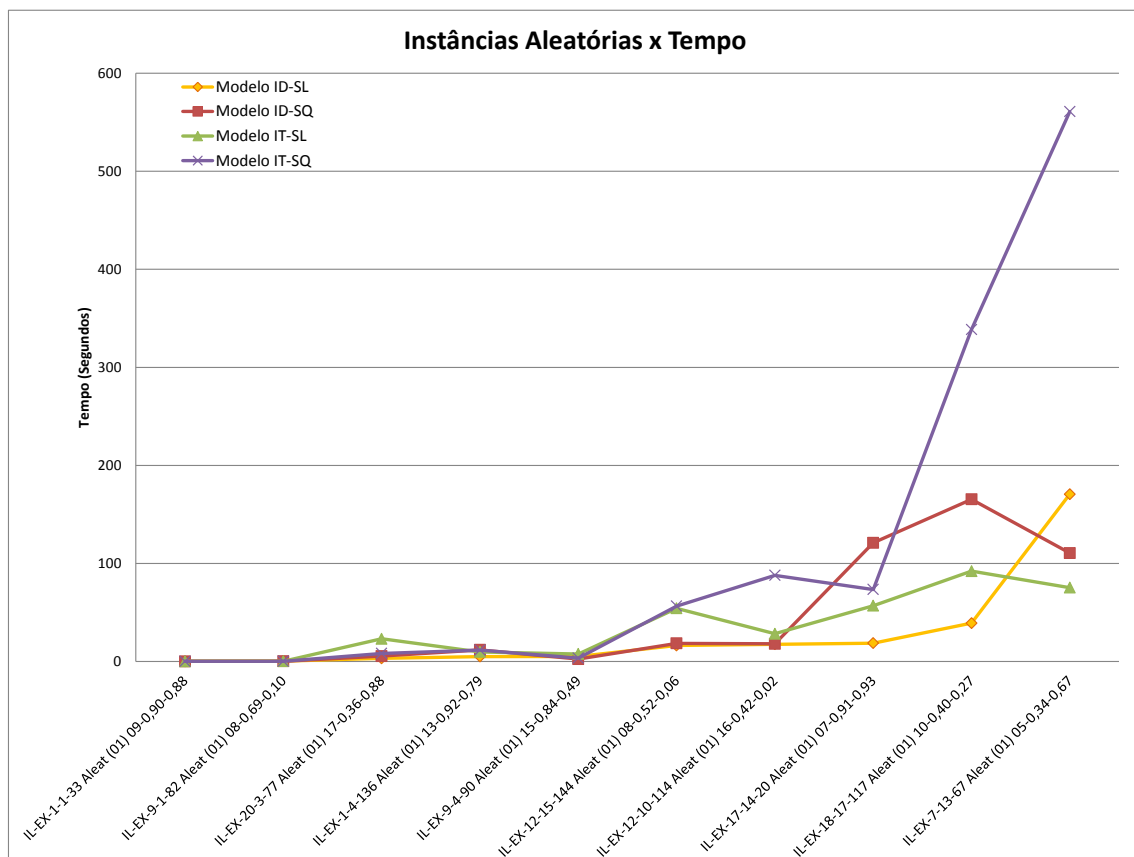


Figura 6.1 – Instâncias Aleatórias x Tempo

O modelo ID-SL apresentou o menor tempo médio de processamento (27,6 segundos), seguido pelo modelo IT-SL (34,7 segundos). Embora seja uma diferença pequena em valores absolutos, o modelo de índice triplo demandou um tempo médio 26% maior que o de índice duplo para chegar à solução ótima das instâncias testadas na segmentação SL. Já na segmentação quadrática (SQ), o tempo médio demandado pelo modelo de índice triplo (114,1 segundos) foi 2,5 vezes maior que o tempo do modelo de índice duplo (45,3 segundos).

É válido notar que o tempo de processamento do modelo IT-SQ cresce em um ritmo muito maior que o dos demais modelos à medida que a dimensão das instâncias aumenta. Outra observação importante a ser feita é a de, salvo raríssimas exceções, o tempo demandado pelo modelo de índice triplo é superior ao de índice duplo em ambas as linearizações. Isto se deve ao porte dos modelos gerados, conforme pode ser visto na tabela 6.2.

Tabela 6.2 – Porte dos Modelos (Instâncias Aleat)

Parâmetro	Número Médio	
	Índice Duplo	Índice Triplo
Variáveis	8.894	18.430
Variáveis Binárias	867	867
Restrições	9.095	9.734
Coef. Não Zerados	54.036	107.075

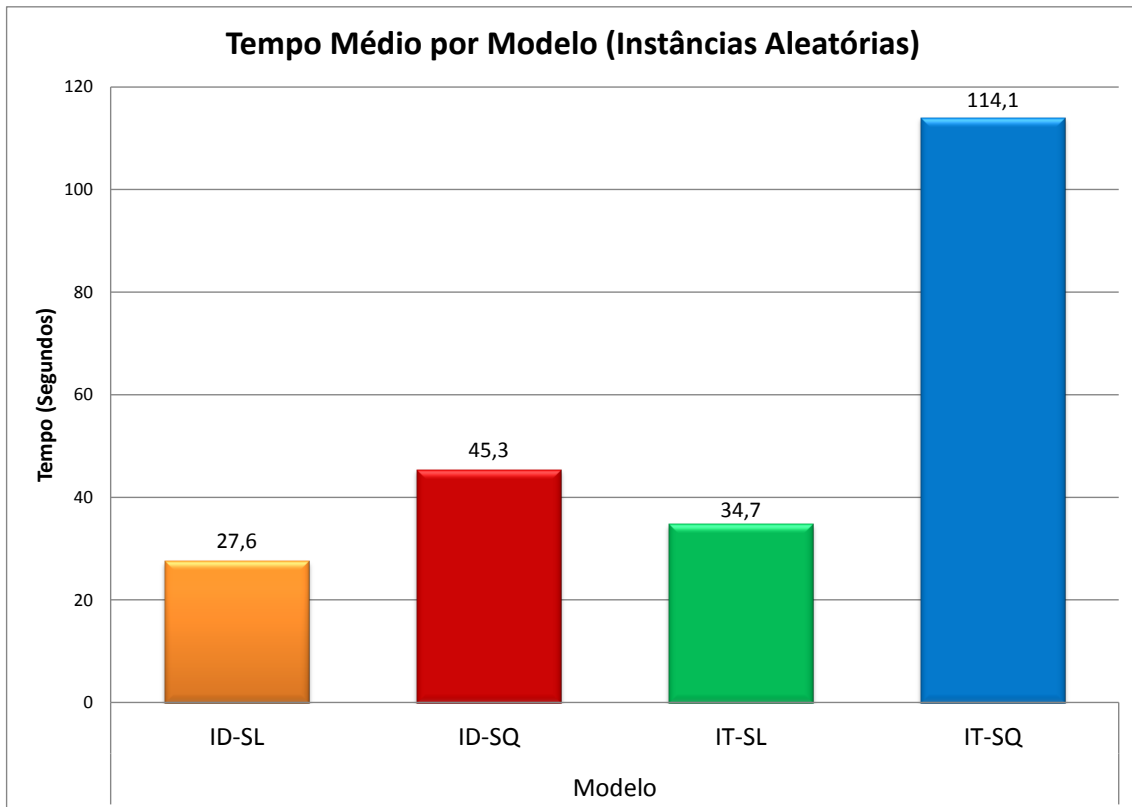


Figura 6.2 – Tempo Médio por Modelo (Instâncias Aleat)

Os erros médios de linearização tanto para a Segmentação Linear quanto para a Segmentação Quadrática são apresentados no gráfico da figura 6.4, e a figura 6.5 mostra a relação entre o tempo médio de processamento e o erro médio de linearização para cada um dos modelos.

O erro médio de linearização é computado através da fórmula abaixo, onde SS_{real} é o estoque de segurança real e SS_{lin} é o estoque de segurança linearizado:

$$Erro = \left(1 - \frac{SS_{real}}{SS_{lin}}\right) \cdot 100\%$$

O estoque de segurança real $SS_{real} = Z_{\alpha} \sqrt{\sum_{l \in L} \varphi_{kl} \sigma_l^2 y_{kl}}$, representado pela equação (2) dos modelos; e o linearizado $SS_{lin} = Z_{\alpha} \sum_{r \in R} (F_k^r u_k^r + C_k^r \sum_{l \in L} v_{kl}^r)$, representado pela equação (2a). A figura 6.3 representa o erro graficamente.

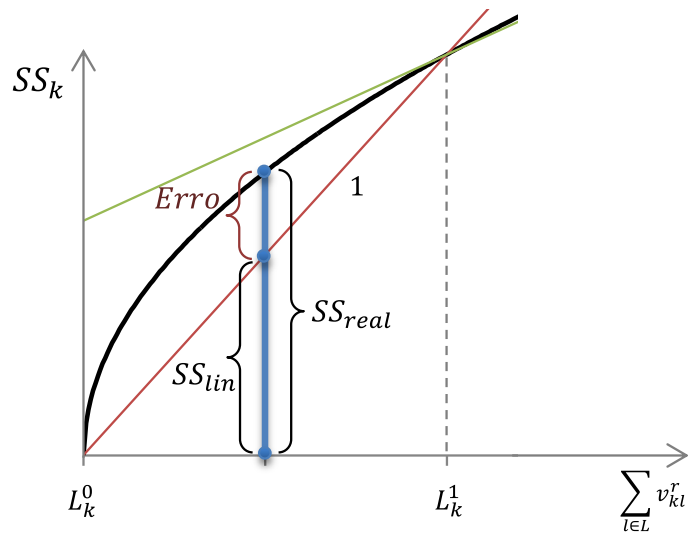


Figura 6.3 – Erro de Linearização

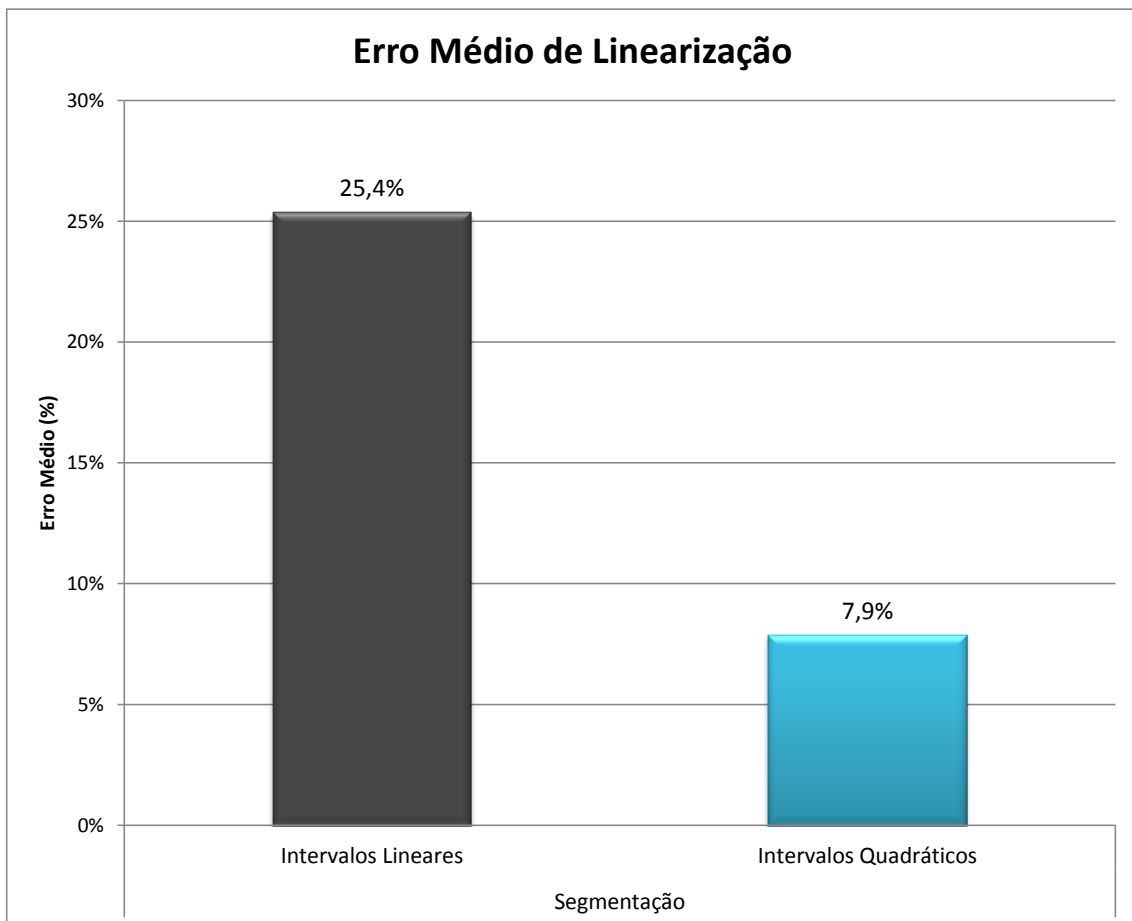


Figura 6.4 – Erro Médio de Linearização SL x SQ (Instâncias Aleat)

Embora o tempo médio de processamento tenha se mostrado significativamente superior para os modelos que utilizaram a segmentação quadrática (SQ), o erro médio de linearização para estes modelos foi muito menor do que o obtido pela segmentação linear (SL). Uma boa relação de equilíbrio é encontrada no modelo ID-SQ, que mesmo

com o tempo médio de processamento um pouco superior ao do ID-SL e IT-SL, apresentou erro de 7,9% (contra 25,4% nos modelos SL).

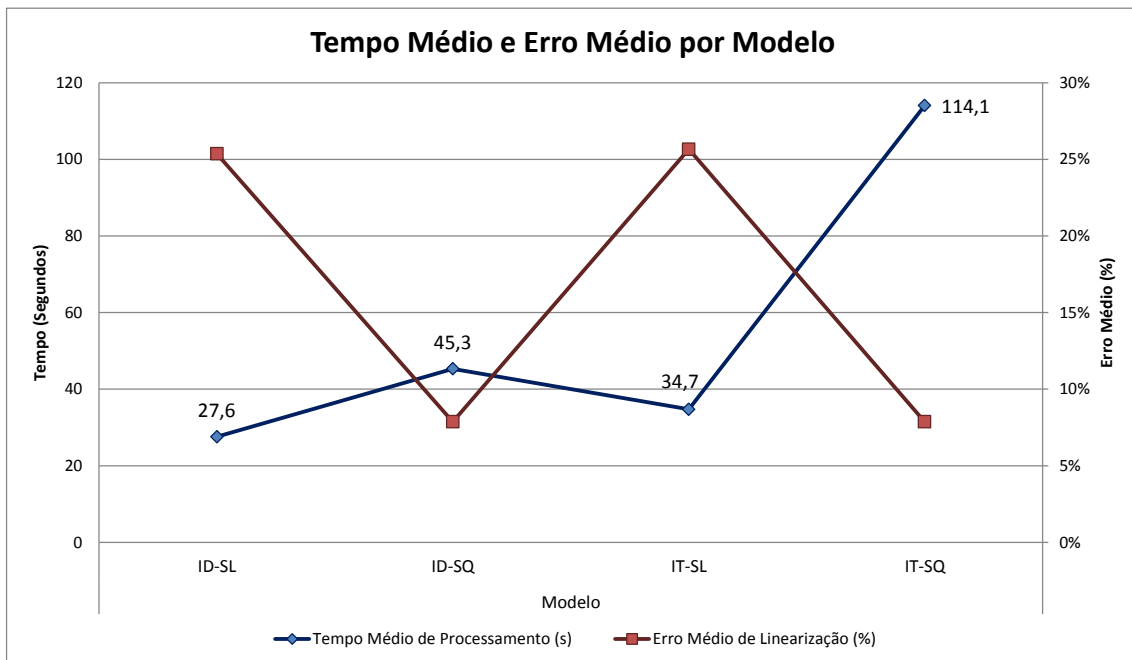


Figura 6.5 – Tempo Médio e Erro Médio por Modelo (Instâncias Aleat)

6.2 Experimentos FUF (Fator de Utilização dos Fornecedores)

O segundo conjunto de dados gerados foi o **FUF**, com o objetivo de analisar o impacto que o Fator de Utilização dos Fornecedores exerce sobre o tempo de processamento em cada um dos modelos. Foram criadas 10 instâncias ($N = 1, \dots, 10$) de tamanho médio com 10 fornecedores, 15 armazéns, 70 clientes, 5 segmentos e CUA de 50% (*IL-EX-10-15-70 FUF (N) 05-CUF-0,50*); sendo que para cada uma delas o CUF foi variado de 10% a 100%, com incrementos de 10%, totalizando 100 dados (ou 400 experimentos). O tempo médio de processamento para cada variação de CUF pode ser visto na tabela 6.3, a plotagem dos dados está no gráfico da figura 6.6.

Tabela 6.3 – Fator de Utilização dos Fornecedores x Tempo

FUF	Tempo			
	ID-SL	ID-SQ	IT-SL	IT-SQ
10%	101,99	40,36	585,69	124,53
20%	150,94	192,35	423,33	889,34
30%	86,57	39,21	541,94	110,01
40%	82,88	21,00	99,98	63,21

50%	60,11	33,96	74,31	90,06
60%	15,09	19,27	37,97	39,23
70%	18,39	19,87	78,07	71,32
80%	10,23	17,12	31,98	52,72
90%	20,23	19,90	66,01	64,68
100%	7,22	24,01	104,44	158,61

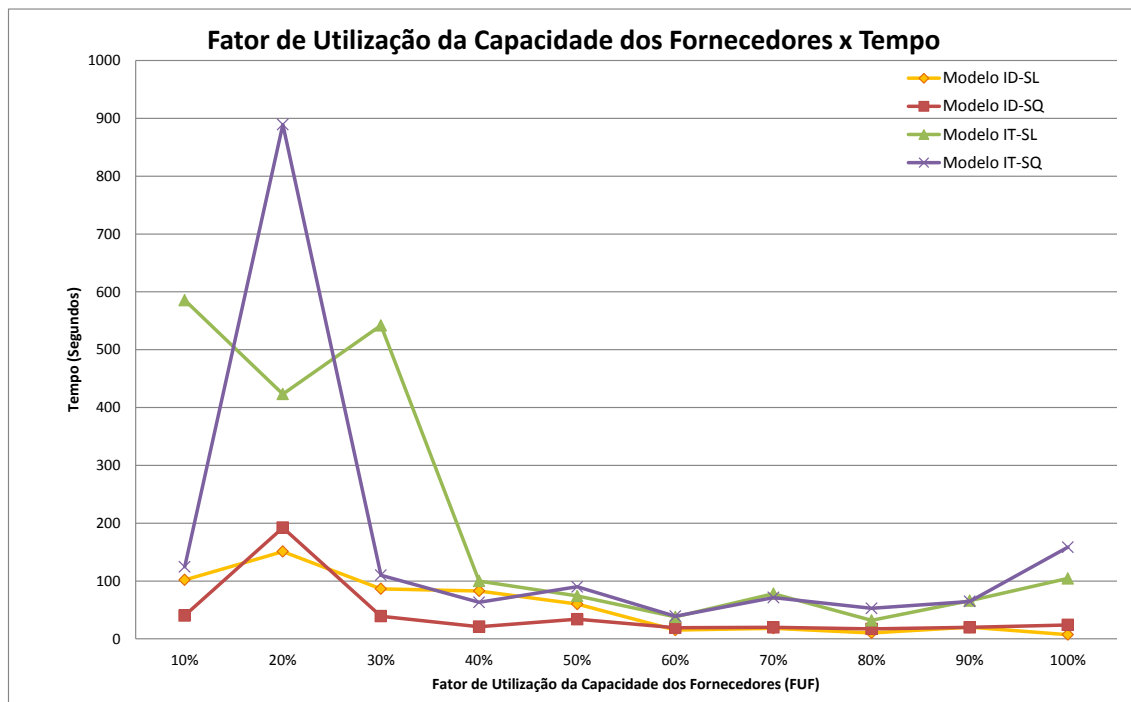


Figura 6.6 – Fator de Utilização dos Fornecedores x Tempo

A perspectiva inicial era a de que o aumento do FUF resultasse em uma diminuição do tempo de processamento, dado que a alocação dos armazéns aos fornecedores não ocorre de maneira rígida como a alocação dos clientes aos armazéns (determinada pela variável binária y_{kl} , que impede o *cross-filling*). Com isso, o número de alternativas viáveis a ser analisada diminui, podando a árvore de enumeração mais rapidamente. Esta perspectiva se confirmou parcialmente, tendo o tempo aumentado subitamente próximo à faixa dos 20%, seguido por uma queda gradual. Novos experimentos e análises mais profundas precisam ser conduzidos para que se possa determinar com maior exatidão o que ocorre quando o FUF se encontra próximo a 10%.

6.3 Experimentos FUA (Fator de Utilização dos Armazéns)

O terceiro conjunto de experimentos realizados teve o objetivo de analisar o impacto do FUA (Fator de Utilização dos Armazéns) sobre o custo computacional de cada modelo. De maneira análoga aos experimentos do FUF, foram criadas 10 instâncias ($N = 1, \dots, 10$) de tamanho médio com 10 fornecedores, 15 armazéns, 70 clientes, 5 segmentos e CUF de 50% (*IL-EX-10-15-70 FUA (N) 05-0,50-CUA*); sendo que para cada uma delas o CUA foi variado de 10% a 100%, com incrementos de 10%, também totalizando 100 dados (ou 400 experimentos). A tabela 6.4 e o gráfico da figura 6.7 mostram os resultados dos experimentos.

Tabela 6.4 – Fator de Utilização dos Armazéns x Tempo

FUA	Modelo			
	ID-SL	ID-SQ	IT-SL	IT-SQ
10%	2,5	3,9	6,2	7,5
20%	23,6	12,6	10,5	64,6
30%	12,5	7,1	40,6	20,7
40%	7,8	31,2	17,5	174,5
50%	1013,5	109,7	1491,2	274,2
60%	834,1	959,3	994,0	1316,1
70%	1355,6	1277,6	2574,5	1501,9
80%	1709,7	169,4	7200,3	743,6
90%	7200,1	7200,1	7200,2	7200,5
100%	7200,6	-	-	-

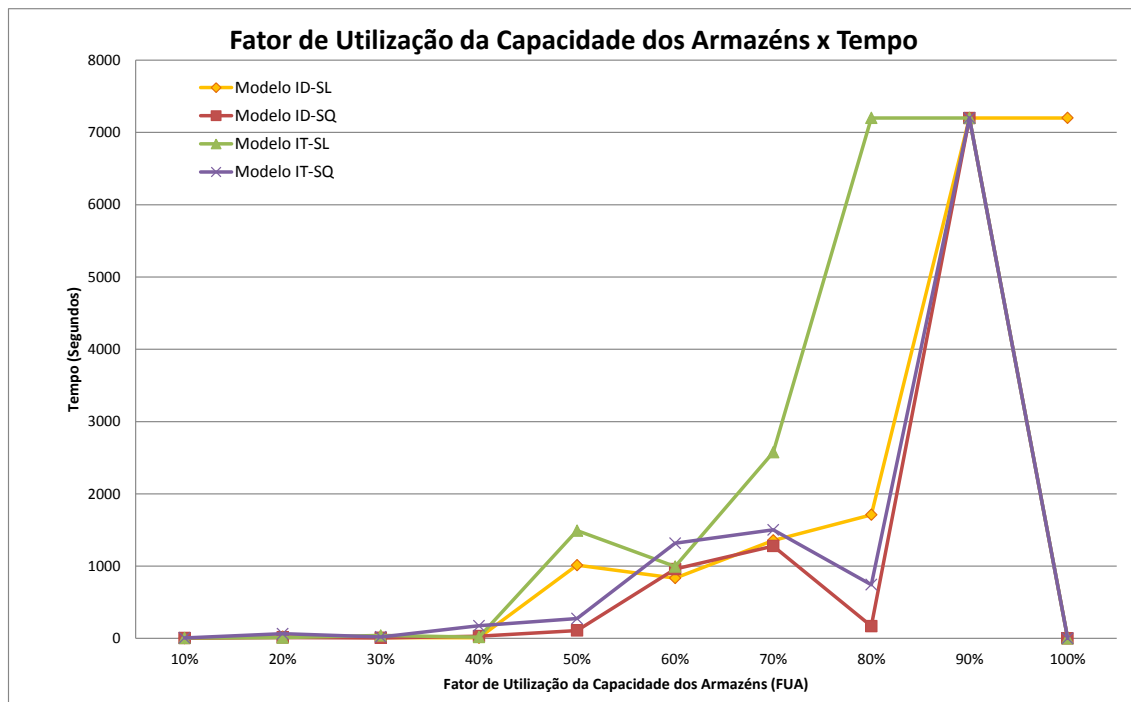


Figura 6.7 – Fator de Utilização dos Armazéns x Tempo

A partir do FUA de 80%, apenas uma das instâncias gerou soluções viáveis; e para o FUA de 100% apenas o modelo ID-SL encontrou uma solução viável dentro do limite de processamento imposto de 7.200 segundos (2 horas). Este tipo de resultado já era esperado, dado que a rigidez da alocação dos clientes aos armazéns faz com que a probabilidade de que determinadas configurações se mostrem inviáveis aumente com a diminuição das folgas de capacidade. Devido também a esta rigidez, o aumento gradual do tempo de processamento das instâncias com o aumento do FUA, previsto à priori, se confirmou.

6.4 Experimentos nSeg (Número de Segmentos)

O quarto experimento (**nSeg**) foi projetado com o intuito de verificar a influência do número de segmentos utilizados na linearização por partes sobre o tempo de processamento dos modelos, bem como sobre o erro resultante desta aproximação. Foram geradas 10 instâncias ($N = 1, \dots, 10$) de tamanho médio com 10 fornecedores, 15 armazéns, 50 clientes, CUF de 50% e CUA de 50% (*IL-EX-10-15-50 nSeg (N) nSeg-0,50-0,50*); sendo experimentados 5, 7, 10, 13, 16 e 20 segmentos. Com isso, 240 dados foram testados neste experimento, cujos resultados estão condensados na tabela 6.5 e nos gráficos da figura 6.8, figura 6.9, figura 6.10 e figura 6.11.

Tabela 6.5 – Tempo Médio e Erro Médio x Número de Segmentos

Modelo	Métrica	Segmentos					
		5	7	10	13	16	20
ID-SL	Tempo Médio de Processamento (s)	19,6	36,6	45,3	47,8	70,9	222,0
	Erro Médio de Linearização (%)	59,2%	52,7%	44,4%	39,7%	33,5%	28,7%
ID-SQ	Tempo Médio de Processamento (s)	27,4	33,0	184,6	279,4	1791,5	2433,6
	Erro Médio de Linearização (%)	26,4%	16,3%	9,0%	5,4%	3,4%	2,2%
IT-SL	Tempo Médio de Processamento (s)	71,5	124,7	129,5	93,8	120,2	513,6
	Erro Médio de Linearização (%)	59,2%	52,6%	45,0%	39,7%	33,5%	28,6%
IT-SQ	Tempo Médio de Processamento (s)	44,8	62,9	699,7	804,1	944,9	1402,6
	Erro Médio de Linearização (%)	26,5%	16,1%	8,9%	5,4%	3,4%	2,2%

Embora os tempos observados para os modelos SQ tenham sido significativamente superiores aos SL para quase todos os experimentos realizados com o mesmo número de segmentos, a diferença na qualidade das soluções obtidas também foi expressiva.

O erro médio resultante da linearização por partes da função do estoque de segurança dos modelos SL em 20 segmentos foi superior (embora próximo) ao erro encontrado com a segmentação em 5 partes nos modelos SQ nas instâncias testadas. Mesmo assim, o tempo médio de processamento dos modelos SL segmentados em 20 partes foi oito vezes maior que o dos modelos SQ segmentados em 5 para a formulação de índice duplo (ID) e mais de onze vezes maior para a formulação de índice triplo (IT).

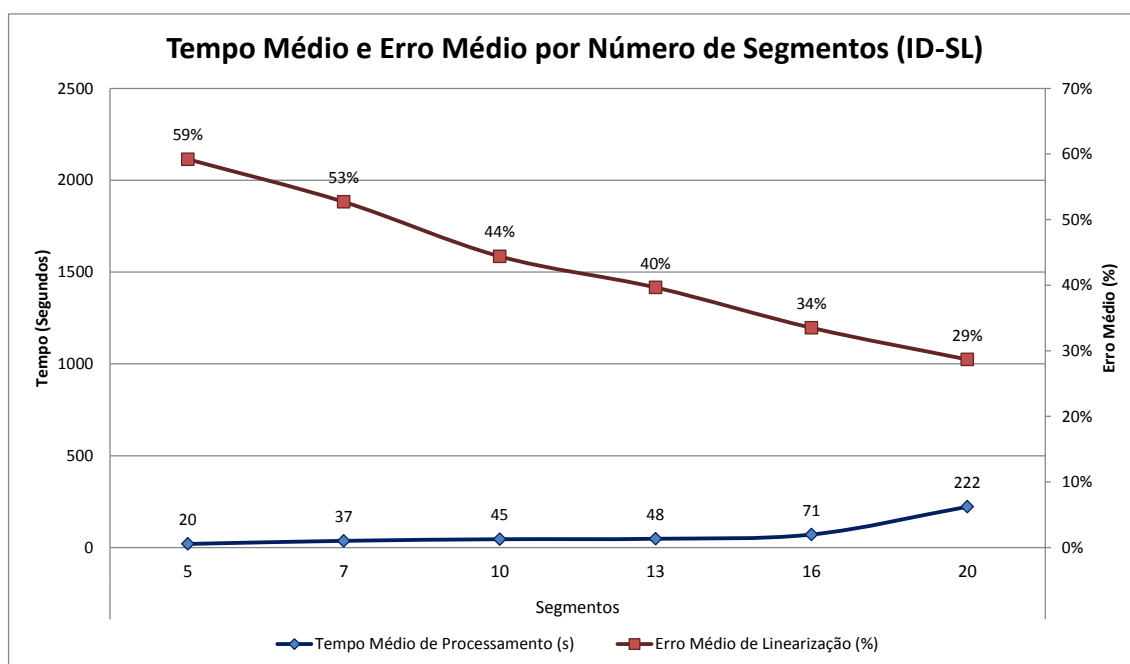


Figura 6.8 – Tempo Médio e Erro Médio por Número de Segmentos (ID-SL)

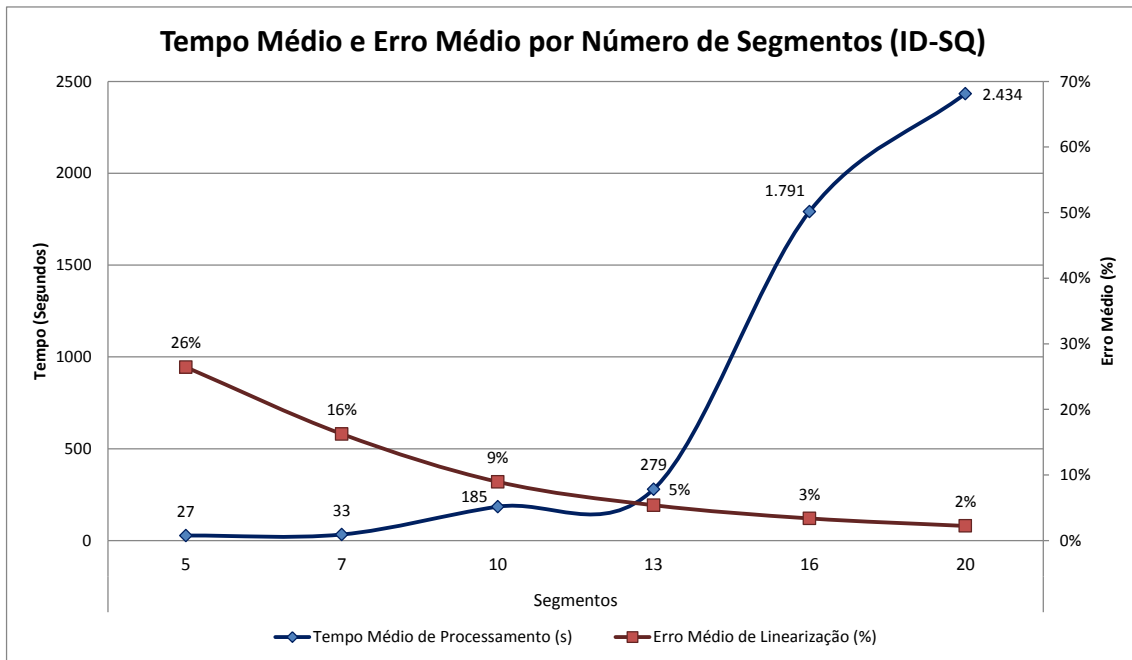


Figura 6.9 – Tempo Médio e Erro Médio por Número de Segmentos (ID-SQ)

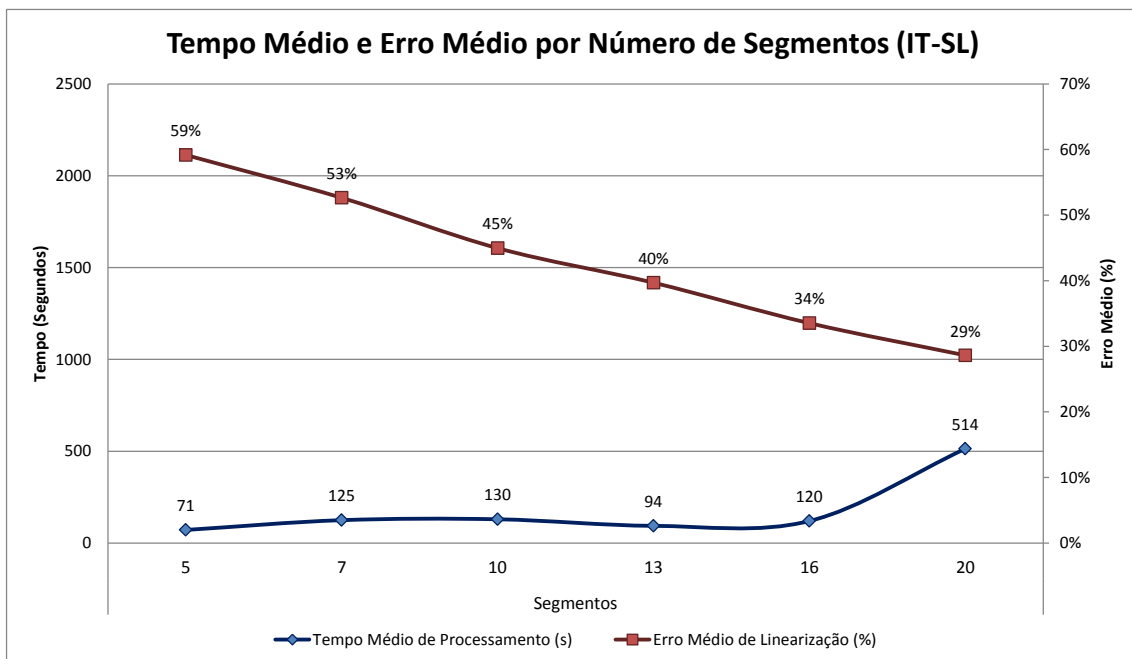


Figura 6.10 – Tempo Médio e Erro Médio por Número de Segmentos (IT-SL)

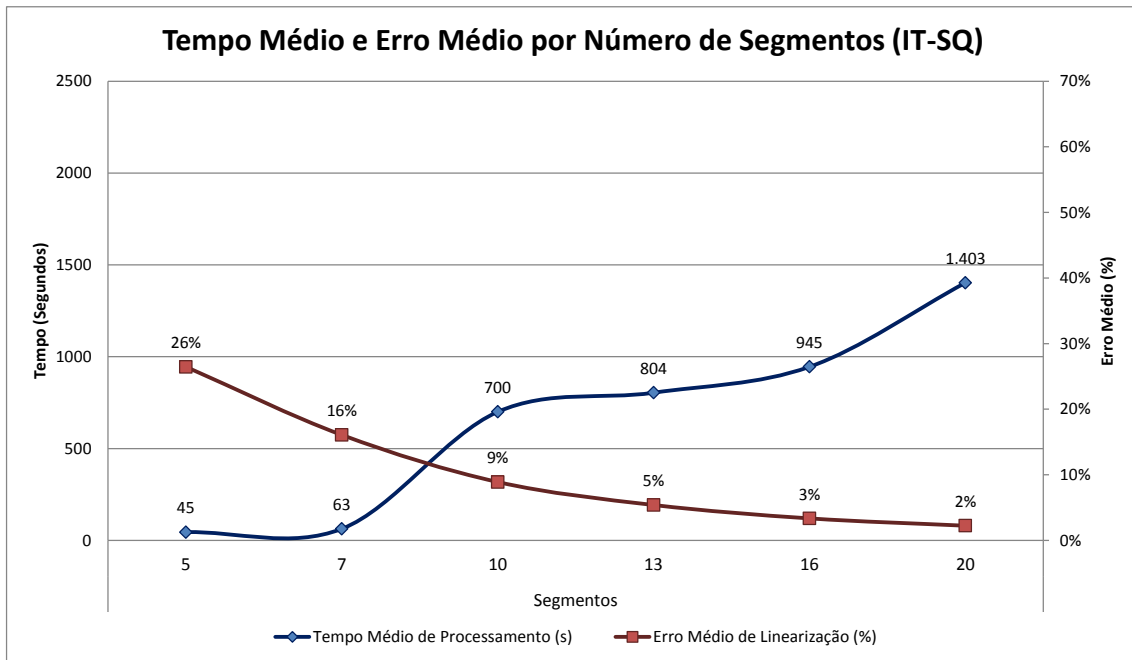


Figura 6.11 – Tempo Médio e Erro Médio por Número de Segmentos (IT-SQ)

6.5 Experimentos GAP (Diferença entre limitantes)

O quinto conjunto de experimentos realizados (**Gap**) buscou verificar o erro de linearização da função do estoque de segurança - de (2) para (2a) - e o *gap* existente entre os limitantes primais e duais da função objetivo após duas horas de processamento. Para tal, foram geradas 5 instâncias ($N = 1, \dots, 5$) de tamanho grande com 20 fornecedores, 20 armazéns, 150 clientes, CUF de 50%, CUA de 50% e divididas em 20 segmentos (*IL-EX-20-20-150 Gap (N) 20-0,50-0,50*); sendo analisado o erro e o *gap* após 7.200 segundos. Os resultados dos 20 dados testados seguem nos gráficos da figura 6.12 e da figura 6.13.

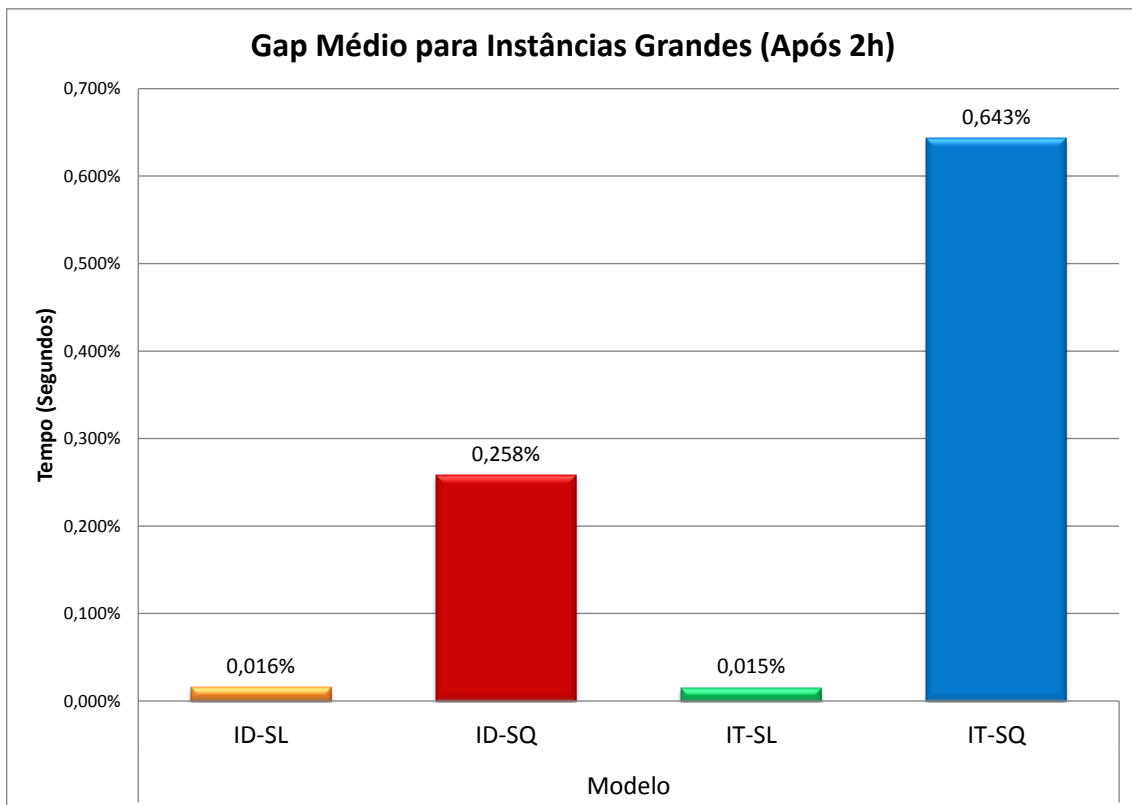


Figura 6.12 – Gap Médio após 2h (Instâncias Grandes)

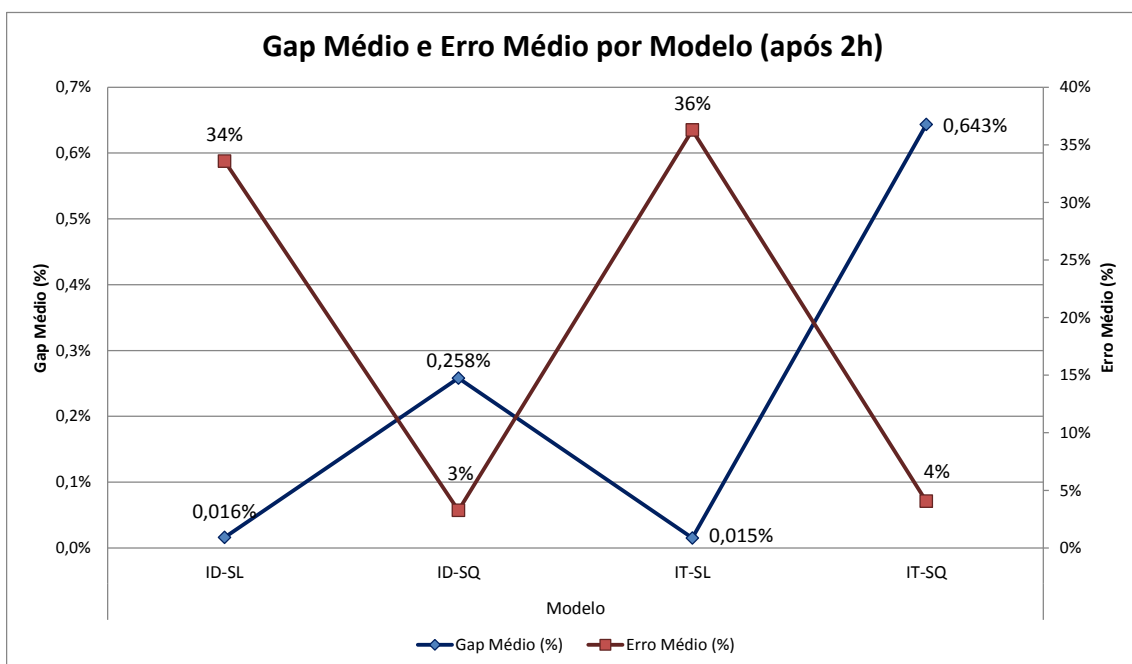


Figura 6.13 – Gap Médio x Erro Médio após 2h (Instâncias Grandes)

O *gap* encontrado entre os limitantes primais e duais estabelecidos durante a solução das instâncias foi menor nos modelos SL. Entretanto, esta diferença entre os *gaps* se mostrou irrelevante entre os conjuntos de dados testados, dada a ordem de grandeza dos erros resultantes do procedimento de linearização por partes.

Isso pode ser verificado ao analisarmos a instância IL-EX-20-20-150 Gap (02) 20-0,50-0,50, por exemplo. A execução do modelo ID-SL finalizou, após duas horas de processamento, com um *gap* de 0,02% e uma solução viável cujo erro foi de 43,24%. O estoque de segurança linearizado foi de aproximadamente 920 unidades, mas o estoque de segurança real para a solução encontrada seria de 1.622 unidades. Já o modelo ID-SQ finalizou a sua execução com um *gap* de 0,57%; mas como o erro de linearização foi consideravelmente menor (2,46%), a qualidade da solução encontrada foi melhor (1.512,4 unidades para a função linearizada e 1550,5 unidades para o estoque de segurança real).

6.6 Experimentos ALEAT-RL (Cross-Filling)

O último conjunto de experimentos realizados teve o objetivo de analisar o impacto que a política de proibição ao *cross-filling* exerce sobre o modelo, tanto em relação ao tempo computacional quanto ao custo da solução objetivo. A limitação ao *cross-filling* ocorre por meio da imposição de que a variável y_{kl} seja binária e que o seu somatório seja igual a 1. Com a relaxação desta condição, é permitido que um cliente seja atendido por múltiplos depósitos, o que diminui o número de variáveis inteiras no problema (resultando em queda do custo computacional) e a alocação se torna mais flexível, abrindo espaço para uma eventual queda nos custos de abertura, operação e manutenção da rede logística. Os experimentos foram testados com as 10 instâncias **Aleat** já utilizadas no primeiro experimento, com alterações apenas da condição de integralidade da variável y_{kl} nos modelos, gerando os experimentos **Aleat-RL**. Os gráficos da Figura 6.14 e da Figura 6.15 apresentam os resultados alcançados.

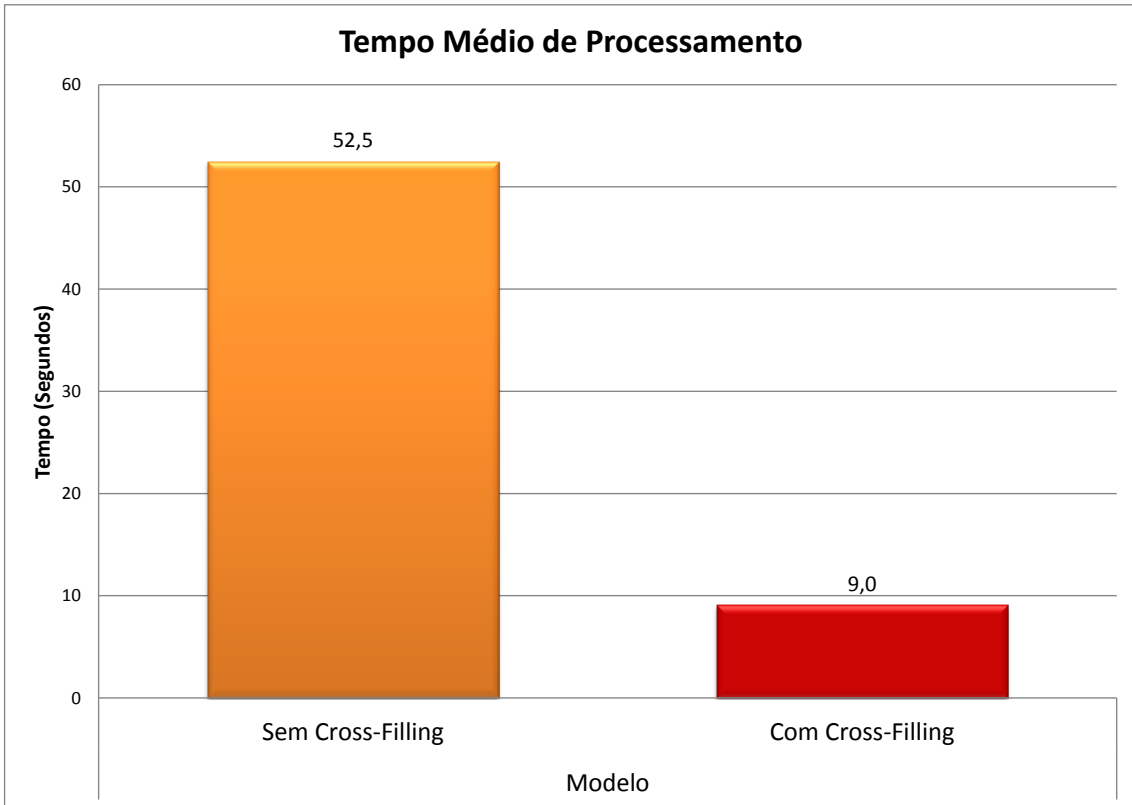


Figura 6.14 – Tempo Médio de Processamento (Aleat x Aleat-RL)

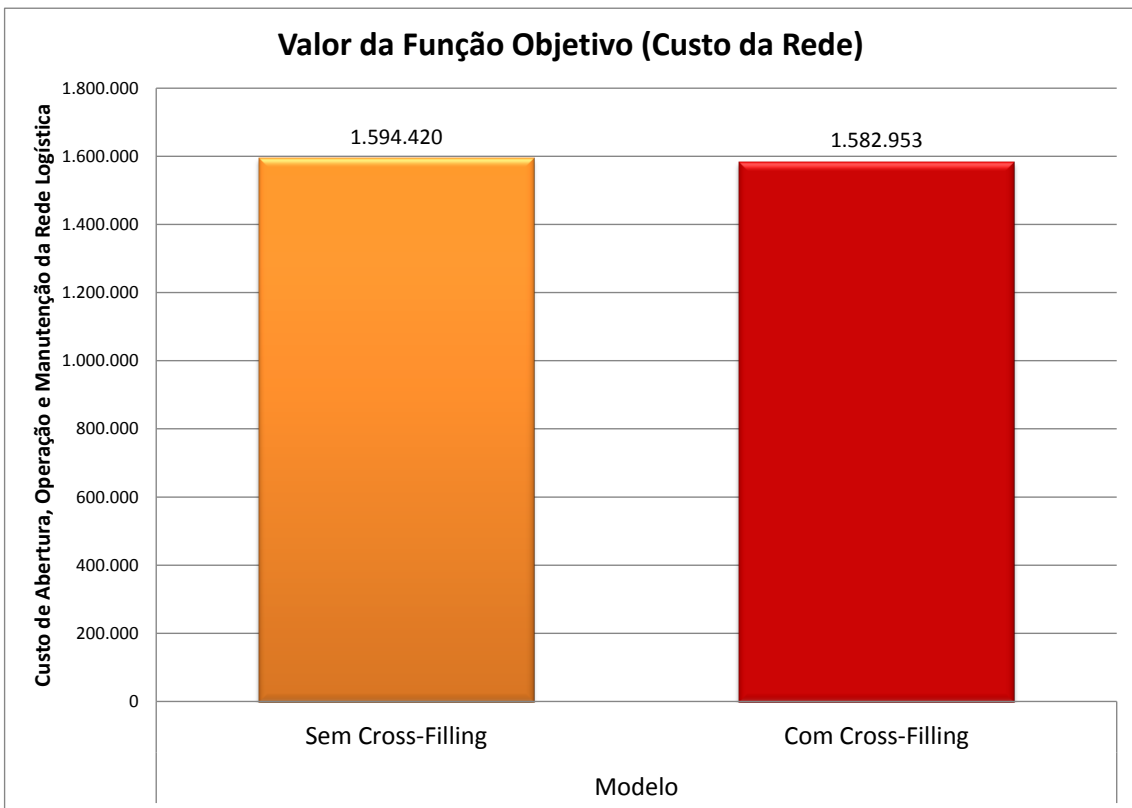


Figura 6.15 – Valor da Função Objetivo (Aleat x Aleat-RL)

7. Conclusões

“That's all Folks!”

— Bugs Bunny

Neste trabalho, foram apresentadas duas formulações para o Problema de Localização de Armazéns com a consideração do efeito de consolidação de estoques, que possibilita a redução dos níveis de estoque de segurança através da centralização destes em armazéns. Os modelos, que enxergam o problema a partir de ópticas distintas, buscam a minimização do custo total de projeto e operação da rede de suprimentos; e retornam os fornecedores a serem utilizados, os armazéns a serem abertos, a alocação de clientes a armazéns e os níveis de estoques em cada um dos armazéns, bem como o fluxo de produtos através da rede.

Com a realização do trabalho, ficou evidente que a inclusão de decisões relativas aos estoques de segurança no modelo aumentou significativamente a complexidade do problema; que foi suavizada com o procedimento de linearização por partes adotado.

Os experimentos mostraram que o limitante superior utilizado para a segmentação da função aproximativa do estoque de segurança foi superestimado, o que resultou em erros de linearização significativos. Entretanto, a linearização SQ mostrou bons resultados em tempos computacionais viáveis, em especial quando utilizada junto ao modelo de índice duplo (ID-SQ).

O aumento do número de segmentos na linearização por partes se mostrou uma ferramenta eficiente para a diminuição do erro de linearização, mas que deve ser utilizada com ponderação, dado o impacto que exerce sobre o tempo total de processamento. Na segmentação quadrática, foi possível diminuir o erro inicial de quase 26,4% (5 segmentos) para 2,2% (20 segmentos), mas com um acréscimo substancial do tempo.

Ainda sobre a segmentação quadrática, pode-se verificar através dos experimentos realizados que o modelo de índice duplo (ID) apresentou melhor desempenho quando o número de segmentos era menor (até 13 segmentos), enquanto o

modelo de índice triplo (IT) foi mais eficiente ao abordar instâncias cuja função do estoque de segurança foi dividida em um número maior de segmentos (16 e 20 segmentos).

Pode-se constatar através de testes específicos que o Fator de Utilização da Capacidade dos Fornecedores (FUF) e o Fator de Utilização da Capacidade dos Armazéns (FUA) exercem impactos extremamente relevantes no tempo de processamento dos modelos. Enquanto o aumento do FUF resulta em uma diminuição do custo computacional, o aumento do FUA tem como consequência o aumento do custo computacional ao lidar com dados cujas restrições de capacidade dos armazéns estão com poucas folgas. Isso é um desdobramento da dificuldade de alocação dos clientes aos armazéns com folgas de capacidade apertadas, dado o impedimento do *cross-filling*.

Por último, cabe destacar que o número de aplicações do Problema de Localização de Armazéns no setor petrolífero ainda é escasso, sendo este trabalho um passo inicial para desenvolvimento de novas soluções e modelos adaptados.

7.1 Trabalhos Futuros

Os modelos apresentados neste trabalho, embora complexos e de consistente valor acadêmico, ainda apresentam certas lacunas em relação às necessidades práticas das organizações. Como trabalhos futuros, vislumbra-se o enriquecimento dos modelos através de uma perspectiva dinâmica (cenários que variam ao longo do tempo) e da inclusão de novas características – como a possibilidade de fornecimento de múltiplas *commodities*.

Propõe-se também a utilização de métodos avançados de solução, como a utilização da relaxação lagrangeana em conjunto com o método de geração de linhas e colunas. Dada a importância da forma como os multiplicadores de Lagrange são atualizados durante a execução do algoritmo, buscar-se-á uma aplicação da versão revisada do algoritmo do volume, utilizado por Bahiense (2000) na atualização dos multiplicadores em um problema de steiner em grafos; em contraposição ao método tradicional do subgradiente. Uma das expectativas iniciais é a de que o modelo de índice

triplo se comporte melhor com a utilização do método de geração de linhas e colunas, dada a facilidade que este método tem de lidar com problemas de grande porte.

Outro ponto a ser estudado com maior profundidade são os métodos de linearização da função do estoque de segurança. Heurísticas simples para a determinação de limitantes superiores do máximo estoque de segurança começarão a ser testadas e comparadas tanto para o modelo SL quanto para o modelo SQ. O método iterativo convergente proposto por Gebennini, Gamberini e Manzini (2009) também pode ensejar diversos estudos sobre o tema e sua aplicação às formulações aqui apresentadas. Por fim, tem-se a pretensão de revisar o método de linearização por partes apresentado em Croxton e Gendron (2003) e Croxton, Gendron e Magnanti (2007), com determinação dos coeficientes e limitantes de cada segmento em tempo de execução.

8. Referências Bibliográficas

- AMAZON WEB SERVICES. **Instâncias do Amazon EC2**. Disponível em: <<http://aws.amazon.com/pt/ec2/instance-types/>>. Acesso em: 15 fev. 2014.
- AMBROSINO, D.; GRAZIA SCUTELLÀ, M. Distribution network design: New problems and related models. **European Journal of Operational Research**, v. 165, n. 3, p. 610–624, set. 2005.
- AMBROSINO, D.; SCUTELLÀ, M. G. Distribution Network Desing: New Problems and Related Models. **European Journal of Operational Research**, v. 165, p. 610–624, 2005.
- BACCHINI, L. **O Algoritmo do Volume: Convergência e Resolução do Problema de Steiner em Grafos**. [s.l.] Universidade Federal do Rio de Janeiro, 2000.
- BALLOU, R. H. **Business Logistics / Supply Chain Management: Planning, Organizing and Controlling the supply Chain**. 5. ed. [s.l.] Pearsin Prentice Hall, 2004.
- BALLOU, R. H. Expressing Inventory Control Policy in the Turnover Curve. **Jornal of Business Logistics**, v. 26, p. 143–164, 2005.
- BALLOU, R. H.; BURNETAS, A. Planning Multiple Location Inventories. **Journal of Business Logistics**, v. 24, p. 65–89, 2003.
- BOWERSOX, D. J.; CLOSS, D. J. **Logística Empresarial: O Processo de Integração da Cadeia de Suprimento**. [s.l.] São Paulo: Atlas, 2001.
- BRANDEAU, M. L.; CHIU, S. S. An Overview of Representative Problems in Location Research. **Management Science**, v. 35, p. 645–674, 1989.
- CHURCH, R. L.; REVELLE, C. S. The Maximal Covering Location Problem. **Papers of the Regional Science Association**, v. 32, p. 101–118, 1974.
- COUNCIL OF SUPPLY CHAIN MANAGEMENT. **Supply Chain Management Definitions**. Disponível em: <<http://cscmp.org/aboutcscmp/definitions.asp>>. Acesso em: 12 nov. 2012.
- CROXTON, K. L.; GENDRON, B. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. **Management Science**, v. 49, n. 9, p. 1268–1273, 2003.
- CROXTON, K. L.; GENDRON, B.; MAGNANTI, T. L. Variable Disaggregation in Network Flow Problems with Piecewise Linear Costs. **Operations Research**, v. 55, n. 1, p. 146–157, 1 jan. 2007.

- DAS, C.; TYAGI, R. Role of inventory and transportation costs in determining the optimal degree of centralization. **Transportation Research Part E: Logistics and Transportation Review**, v. 33, n. 3, p. 171–179, set. 1997.
- DAVARIZ, R. C. **Procedimento para Análise de Rede Logística**. [s.l.] Rio de Janeiro IME, 2006.
- ELZINGA, J.; HEARN, D. W. The Minimum Covering Sphere Problem. **Management Science**, v. 19, p. 96–104, 1972.
- EPPEN, G. D. Effects of Centralization on Expected Costs in a Multi-Location Newsboy Problem. **Management Science**, v. 25, n. 5, p. 498–501, 1979.
- FARLEY, A. M. Vertex Centers of Trees. **Transportation Science**, v. 16, p. 265–280, 1982.
- FERREIRA FILHO, V. J. M. et al. **Gestão de Estoques: Otimizando a logística e a cadeia de suprimentos**. 1^a. ed. Rio de Janeiro, Brasil: Editora e-papers, 2006. p. 144
- FERREIRA FILHO, V. J. M.; GENDRON, B. A Piecewise Linearization with Column-and-Row Generation Approach for a Two-Echelon Inventory Location Problem with Risk Pooling. p. 28, 2012.
- FRANGIONI, A.; GENDRON, B. 0–1 Reformulations of the Multicommodity Capacitated Network Design Problem. **Discrete Applied Mathematics**, v. 157, n. 6, p. 1229–1241, 28 mar. 2009.
- GARFINKEL, R. S.; NEEBE, A. W.; RAO, M. R. The m-Center Problem: Minimax Facility Location. **Management Science**, v. 23, p. 1133–1142, 1977.
- GEBENNINI, E.; GAMBERINI, R.; MANZINI, R. An integrated production–distribution model for the dynamic location and allocation problem with safety stock optimization. **International Journal of Production Economics**, v. 122, n. 1, p. 286–304, nov. 2009.
- HAKIMI, S. L. Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph. **Operations Research**, v. 10, p. 450–459, 1964.
- HALFIN, S. On Finding the Absolute and Vertex Centers of a Tree with Distances. **Transportation Science**, v. 8, p. 75–77, 1974.
- HARRIS, F. W. How Many Parts to Make at Once. **Factory, the Magazine of Management**, v. 10, p. 135–136, 1913.
- HEDETNIEMI, S. M.; COCKAYNE, E. J.; HEDETNIEMI, S. T. Linear Algorithms for Finding the Jordan Center and Path Center of a Tree. **Transportation Science**, v. 15, n. 2, p. 98–114, maio 1981.

JAYARAMAN, V. Transportation, facility location and inventory issues in distribution network design: An investigation. **International Journal of Operations & Production Management**, v. 18, n. 5, p. 471–494, 1998.

JAYARAMAN, V.; PIRKUL, H. Planning and coordination of production and distribution facilities for multiple commodities. **European Journal of Operational Research**, v. 133, n. 2, p. 394–408, jan. 2001.

JORGE, C. M. DE M. **Planejamento de redes logísticas via simulação: analisando fatores determinantes das decisões de localização e centralização da distribuição**. [s.l.] Universidade Federal do Rio de Janeiro, 2008.

KARIV, O.; HAKIMI, S. L. An Algorithmic Approach to Network Location Problems. II: The p -Medians. **SIAM Journal on Applied Mathematics**, v. 37, n. 3, p. 539–560, dez. 1979.

MAHMOUD, M. Optimal inventory consolidation schemes: a portfolio effect analysis. **Journal of Business Logistics**, 1992.

MAISTER, D. H. Centralisation of Inventories and the “Square Root Law”. **International Journal of Physical Distribution & Logistics Management**, v. 6, n. 3, p. 124–134, 1976.

MINIEKA, E. The m -Center Problem. **SIAM Review**, v. 12, n. 1, p. 138–139, jan. 1970.

MONTEBELLER JUNIOR, E. J. **Incorporando o Efeito de Consolidação dos Estoques no Planejamento de Redes Logísticas**. [s.l.] Universidade Federal do Rio de Janeiro, 2009.

NOZICK, L. K.; TURNQUIST, M. A. A two-echelon inventory allocation and distribution center location analysis. **Transportation Research Part E: Logistics and Transportation Review**, v. 37, n. 6, p. 425–441, dez. 2001.

OWEN, S. H.; DASKIN, M. S. Strategic facility location: A review. **European Journal of Operational Research**, v. 111, n. 3, p. 423–447, dez. 1998.

SILVA, T. C. M. **Gerenciamento Dinâmico da Localização de Armazéns e Alocação de Estoques Aplicado ao Abastecimento de Unidades de Produção Offshore**. [s.l.] Universidade Federal do Rio de Janeiro, 2012.

TALLON, W. J. The Impact of Inventory Centralization on Aggregate Safety Stock: The Variable Supply Lead Time Case. **Journal of Business Logistics**, 1993.

TSUCHIDA, T. DE C. **Modelagem da localização de pólos de venda de derivados de petróleo**. [s.l.] Pontifícia Universidade Católica do Rio de Janeiro, 2008.

VIDYARTHI, N. et al. Integrated Production-Inventory-Distribution System Design with Risk Pooling: Model Formulation and Heuristic Solution. **Transportation Science**, v. 41, n. 3, p. 392–408, 1 ago. 2007.

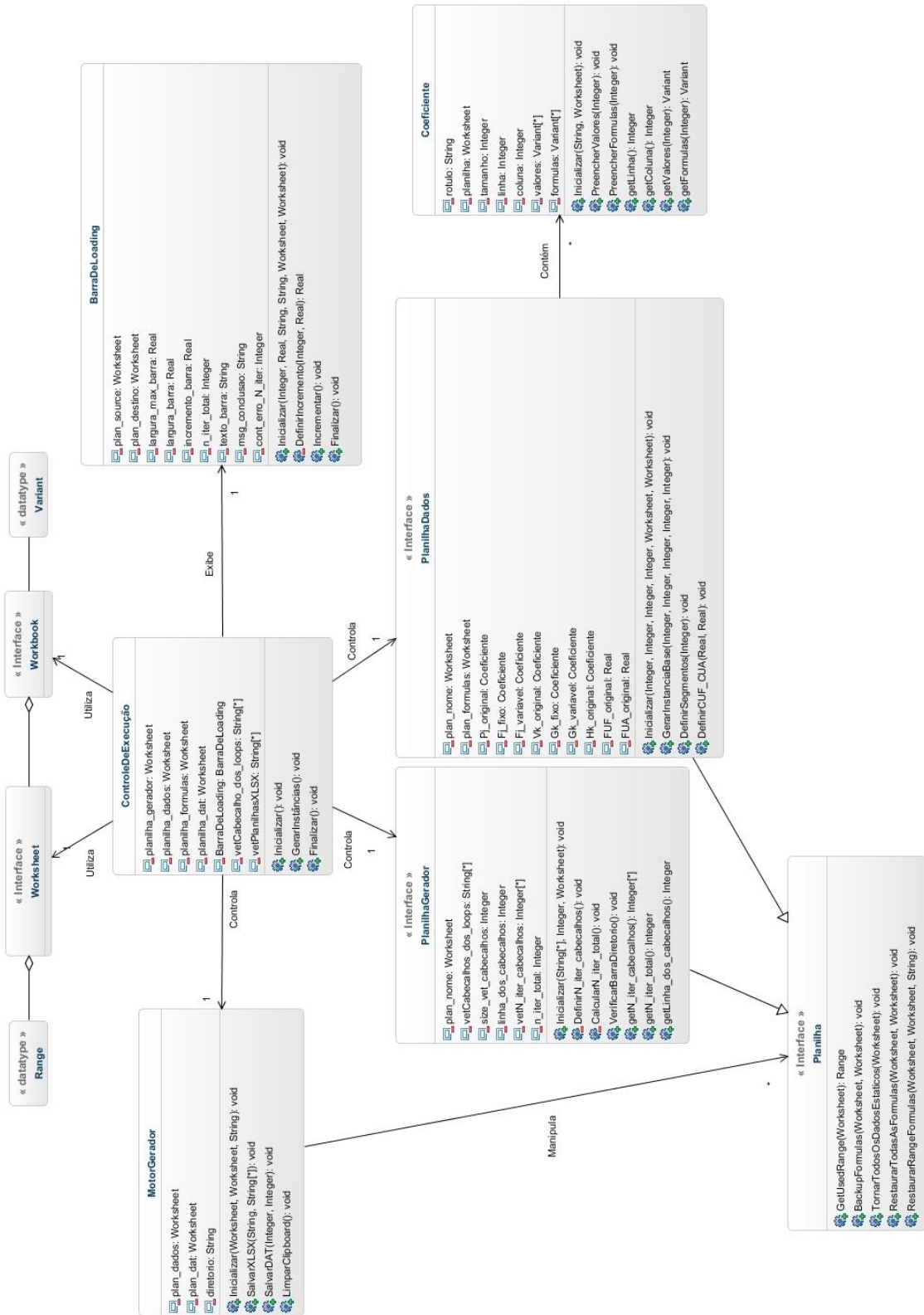
VIDYARTHI, N.; ELHEDHLI, S.; JEWKES, E. Response time reduction in make-to-order and assemble-to-order supply chain design. **IIE Transactions**, v. 41, n. 5, p. 448–466, 9 mar. 2009.

WOLSEY, L. A. **Integer Programming**. 1. ed. [s.l.] Wiley-Interscience, 1988. p. 288

ZINN, W.; LEVY, M.; BOWERSOX, D. J. Measuring the effect of inventory centralization/decentralization on aggregate safety stock: the “square root law” revisited. **Journal of Business Logistics**, v. 10, n. 1, p. 1–14, 1989.

Apêndice I

Diagrama de Classes do Gerador de Instâncias



Apêndice II

Capturas de Tela do Gerador de Instâncias

Gerador de Instâncias - FUA - Microsoft Excel


Gustavo Bitencourt (22/01/2014)

Gerador de Instâncias - Inventory Location

Caminho: D:\Instâncias 2\

Cenário: FUA Fator de Escala: 1

Service Level α : 0.95 Coeficiente de Variação: 0.05



Segmentos	CUF	CUA	Fornecedores	Armazéns	Clientes	N
5	50%	10%	10	15	70	01
6	50%	20%	10	15	70	02
7	50%	30%	10	15	70	03
8	50%	40%	10	15	70	04
9	50%	50%	10	15	70	05
10	50%	60%	10	15	70	06
	50%	70%	10	15	70	07
	50%	80%	10	15	70	08
	50%	90%	10	15	70	09
	50%	100%	10	15	70	10
			20	20	150	01
			20	20	150	02
			20	20	150	03
			20	20	150	04
			20	20	150	05

Gerador Dados ArqDat Resultados - ID-SL Resultados - ID-SQ Resultados - IT-SL Resultados - IT-SQ

Pronto

Basic data for Inventory Location Problem

IL-EX-5-10-15 FUA (01) 05-0,56-1,00

Tel	1
Service level α	0,95
Z_α	1,64
Demanda Total	28.438
Max C_{br}	401,7
Max ΣV_{br}	0,0026
Max ΣV_{br}	723.762
Max ΣSS_k	851

Scenario Multiplier	β
β_F	1
β_R	1
β_{V_r}	1
β_{G_h}	1
β_{H_h}	1

RESUMO DA INSTÂNCIA	
Fornecedores	5
Armazéns	10
Cientes	15
Segmentos	5
Service Level α	0,95
Fator de Escala	1
Coef. de Var.	0,05

nr (# segmentos) 5 Função Estoque de Segurança = Raiz $(Z_\alpha \cdot \phi_w \cdot \sigma \cdot r)$:: Divisão Linear dos Segmentos ::

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
r	0	144.752	289.504	434.256	579.008	723.762								
L_r	0	380,5	538,1	659,0	760,9	850,7								
SS_{br}/Z_α	0,0	0,0	222,9	296,2	353,1	401,7								
F_{br}	0,0	0,0	0,002628	0,001089	0,000835	0,000704	0,000620							
C_{br}	-													

nr (# segmentos) 5 Função Estoque de Segurança = Raiz $(Z_\alpha \cdot \phi_w \cdot \sigma \cdot r)$:: Divisão Quadrática dos Segmentos ::

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
r	0	28.950	115.602	260.554	463.208	723.762								
L_r	0	170,1	340,3	510,4	680,6	850,7								
SS_{br}/Z_α	0,0	0,0	113,4	204,2	291,7	378,1								
F_{br}	0,0	0,0	0,005877	0,001959	0,001175	0,000840	0,000653							
C_{br}	-													

Number of Suppliers: {1, 5, 10, 15} Total Demand 28.438

#j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
jSuppliers	New York NY	Los Angeles CA	Chicago IL	Houston TX	Philadelphia PA									
Custo Fixo de F_j	771	551	955	655	617									
Custo Var. de F_j	1.259	1.928	1.889	1.638	1.103									
F_j	74.966	73.738	199.991	138.704	189.384									

Gerador

Pronto

Gerador de Instâncias - FUA - Microsoft Excel

Arquivo | Página Inicial | Inserir | Layout da Página | Fórmulas | Dados | Revisão | Exibição | Suplementos | Team

Gerador de Arquivos DAT para o CPLEX

IL-EX-5-10-15 FUA (01) 05-0-47-1,00 - ID-SL.dat	
Fornecedores	5
Armazéns	10
Cientes	15
Segmentos	5

Índice Duplo - Segmentos Lineares
IL-EX-5-10-15 FUA (01) 05-0-47-1,00 - ID-SL.dat

SheetConnection sheet ("IL-EX-5-10-15 FUA (01) 05-0-47-1,00.xlsx");
 outfile from SheetRead(sheet,"Dados"E2");
 Tal from SheetRead(sheet,"Dados"C4");
 Zalfa from SheetRead(sheet,"Dados"C6");
 TotalDem from SheetRead(sheet,"Dados"C7");
 MaxFkr from SheetRead(sheet,"Dados"C8");
 MaxCkr from SheetRead(sheet,"Dados"C9");
 jSuppliers from SheetRead(sheet,"Dados"C29:G29");
 kDepots from SheetRead(sheet,"Dados"C37:L37");
 iClients from SheetRead(sheet,"Dados"C46:Q46");
 nr from SheetRead(sheet,"Dados"J8);
 Lr from SheetRead(sheet,"Dados"C15:H15");
 Fr from SheetRead(sheet,"Dados"D17:H17");
 Cr from SheetRead(sheet,"Dados"D18:H18");
 Fj from SheetRead(sheet,"Dados"C32:G32");
 Pj from SheetRead(sheet,"Dados"C33:G33");
 Cjk from SheetRead(sheet,"Dados"C54:L58");
 Gk from SheetRead(sheet,"Dados"C40:L40");
 Hk from SheetRead(sheet,"Dados"C42:L42");
 Vk from SheetRead(sheet,"Dados"C41:L41");
 Tkl from SheetRead(sheet,"Dados"C77:Q86");
 mil from SheetRead(sheet,"Dados"C47:Q47");
 sigl2 from SheetRead(sheet,"Dados"C49:Q49");
 fikl from SheetRead(sheet,"Dados"C100:Q109");
 qskl from SheetRead(sheet,"Dados"C123:Q132");

Índice Duplo - Segmentos Quadráticos
IL-EX-5-10-15 FUA (01) 05-0-47-1,00 - ID-SQ.dat

SheetConnection sheet ("IL-EX-5-10-15 FUA (01) 05-0-47-1,00.xlsx");
 outfile from SheetRead(sheet,"Dados"E2");
 Tal from SheetRead(sheet,"Dados"C4");
 Zalfa from SheetRead(sheet,"Dados"C6");
 TotalDem from SheetRead(sheet,"Dados"C7");
 MaxFkr from SheetRead(sheet,"Dados"C8");
 MaxCkr from SheetRead(sheet,"Dados"C9");
 jSuppliers from SheetRead(sheet,"Dados"C29:G29");
 kDepots from SheetRead(sheet,"Dados"C37:L37");
 iClients from SheetRead(sheet,"Dados"C46:Q46");
 nr from SheetRead(sheet,"Dados"J8);
 Lr from SheetRead(sheet,"Dados"C15:H15");
 Fr from SheetRead(sheet,"Dados"D17:H17");
 Cr from SheetRead(sheet,"Dados"D18:H18");
 Fj from SheetRead(sheet,"Dados"C32:G32");
 Pj from SheetRead(sheet,"Dados"C33:G33");
 Cjk from SheetRead(sheet,"Dados"C54:L58");
 Gk from SheetRead(sheet,"Dados"C40:L40");
 Hk from SheetRead(sheet,"Dados"C42:L42");
 Vk from SheetRead(sheet,"Dados"C41:L41");
 Tkl from SheetRead(sheet,"Dados"C77:Q86");
 mil from SheetRead(sheet,"Dados"C47:Q47");
 sigl2 from SheetRead(sheet,"Dados"C49:Q49");
 fikl from SheetRead(sheet,"Dados"C100:Q109");
 qskl from SheetRead(sheet,"Dados"C123:Q132");

Apêndice III

Código em OPL do Modelo de Índice Duplo

```

// ----- DADOS ----- //

// Outputfile
string outfile = ...;

// Parameters
int Tal = ...;
float Zalfa = ...;
float TotalDem = ...;
float MaxFkr = ...;
float MaxCkr = ...;

// Index
{string} jSuppliers = ...;
{string} kDepots = ...;
{string} lClients = ...;
int nr = ...;
range rSegments = 1..nr;
range rLSegments = 0..nr;

// Piecewise Linear Coeficients
float Lr [rLSegments]= ...;
float Fr [rSegments]= ...;
float Cr [rSegments]= ...;

float Fkr [k in kDepots][r in rSegments] = Fr [r] ;
float Ckr [k in kDepots][r in rSegments] = Cr [r] ;

// Costs and Capacities
int Fj[jSuppliers] = ...; //Custo Fixo de abertura dos Fornecedores
int Pj[jSuppliers] = ...; //Capacidade dos Fornecedores
float Cjk[jSuppliers][kDepots]= ...; //Custo de produzir e enviar uma
unidade do fornecedor J para o depósito K

int Gk[kDepots]= ...; //Custo anual de abertura de um
CD na localidade k
float Hk[kDepots]= ...; //Custo de manutenção de estoques
para uma unidade de produto por um ano no depósito k
float Vk[kDepots]= ...; //Capacidade do depósito k
float Tkl[kDepots][lClients]= ...; //Custo de envio de uma unidade
do CD k para o cliente l

// Demand (mean e deviation) leadtime and coverture parameter
float mil[lClients]= ...;
float sigl2[lClients]= ...;
float fikl[kDepots][lClients]= ...; //Lead time de envio
do depósito K para o cliente L
int qsikl[kDepots][lClients] = ...; //Cobertura dos depósitos - 1 if
DC k cover (is in an appropriate distance) client l; 0 otherwise.

// ----- VARIÁVEIS DE DECISÃO ----- //

// MIP

```

```

dvar boolean wj[jSuppliers]; //Abertura ou não de uma fábrica
(Supplier)
dvar boolean zk[kDepots]; //Abre ou não um depósito em k

dvar float+ xjk[jSuppliers][kDepots]; //Número de unidades de produto
enviada de j para k visando suprir a demanda de l
dvar boolean ykl[kDepots][lClients]; //Alocação ou não do cliente l ao
depósito k (por ser binária impede o Cross-filling)

dvar boolean ukr [kDepots][rSegments]; //Variável auxiliar binária da
linearização por partes
dvar float+ vklr [kDepots][lClients][rSegments]; //Variável auxiliar
contínua da linearização por partes
dvar float+ SSk [kDepots];

// Decision variables to track the Obj components

dvar float+ TC; /* Total Cost */
dvar float+ SFC; /* Suppliers Fixed Cost */
dvar float+ DFC; /* Depot Fixed Cost */
dvar float+ PTHC; /* Production, Transport e Holding Cost from
Suppliers to Depot*/
dvar float+ TSC; /* Transportation Cost from Depots to Customers
*/

// ----- MODELO ----- //

// Objective
minimize
    SFC + DFC + PTHC + TSC;

// Constraints
subject to {

    SFC == sum( j in jSuppliers) Fj[j]*wj[j];
    DFC == sum( k in kDepots) Gk[k]*zk[k];
    PTHC == sum( j in jSuppliers, k in kDepots) ((Cjk[j][k] +
Hk[k])*xjk[j][k]);
    TSC == Tal*sum( k in kDepots, l in lClients)
Tk1[k][l]*mil[l]*ykl[k][l];
    TC == SFC + DFC + PTHC + TSC;

    forall( k in kDepots )
        EstoqueSeguranca:
            SSk[k] == Zalfa*(sum( r in rSegments) (Fkr[k][r]*ukr[k][r] +
Ckr[k][r]*sum( l in lClients)vklr[k][l][r]));

    // (2a) e (3)
    forall( k in kDepots )
        FlowEquilibrium:
            sum( j in jSuppliers)xjk[j][k] == (sum( l in lClients)
Tal*mil[l]*ykl[k][l]) + SSk[k];

    // (4)
    forall (j in jSuppliers)
        SupplierCapacity:
            sum ( k in kDepots )xjk[j][k] <= Pj[j]*wj[j];

```

```

// (14)
forall (j in jSuppliers, k in kDepots )
    LinkFlow1EchSupplier:
        xjk[j][k] <= Pj[j]*wj[j];

// (15)
forall( k in kDepots )
    DepotCapacity:
        sum(j in jSuppliers ) xjk[j][k] <= Vk[k]*zk[k];

// (16)
SupplierXClientDemand:
    sum(j in jSuppliers, k in kDepots) xjk[j][k] >= Tal*sum(l in
lClients)mil[l];

// (17)
SupplierXClientDemand2:
    sum(j in jSuppliers, k in kDepots) xjk[j][k] <= Tal*sum(l in
lClients)mil[l] + Zalfa*(sum( k in kDepots ) (Fkr[k][nr]
+ Ckr[k][1]*sum( l in
lClients)fikl[k][l]*(sigl2[l])));
// (5)
forall( k in kDepots )
    DepotCapacity2:
        Tal*sum(l in lClients ) (mil[l]*ykl[k][l]) + SSk[k] <=
Vk[k]*zk[k];

// (6)
forall (l in lClients )
    Demandfulfillment2:
        sum( k in kDepots) ykl[k][l] == 1;

// (7)
forall( k in kDepots, l in lClients )
    CoverturesDepotClient:
        ykl[k][l] <= qsikl[k][l]*zk[k];

// (11)
forall( k in kDepots )
    PiecewiseLinear2:
        sum( l in lClients ) fikl[k][l]*(sigl2[l])*ykl[k][l] == sum(
r in rSegments, l in lClients ) vklr[k][l][r];

// (12) - Primeira Metade da Inequação
forall( k in kDepots, r in rSegments) //r>=2//
    PiecewiseLinear3a:
        Lr[r-1]*ukr[k][r] <= sum( l in lClients ) vklr[k][l][r];

// (12) - Segunda Metade da Inequação
forall( k in kDepots, r in rSegments )
    PiecewiseLinear3b:
        sum( l in lClients ) vklr[k][l][r] <= Lr[r]*ukr[k][r] ;

// (13)
forall( k in kDepots )
    PiecewiseLinear3:
        sum( r in rSegments) ukr[k][r] <= 1;

// (18)
forall( k in kDepots, l in lClients, r in rSegments )

```

```

    PiecewiseLinear4:
        vklr[k][l][r] <= fikl[k][l]*sigl2[l]*ukr[k][r];
}

// ----- PÓS-PROCESSAMENTO ----- //

int NSuppliers = sum(j in jSuppliers) wj[j];
int NDepots = sum(k in kDepots) zk[k];
float CUF_Aloc = TotalDem / (sum(j in jSuppliers) Pj[j]*wj[j]);
//Utilização dos Fornecedores Alocados (Diferente do CUF da instância,
qué dos fornecedores disponíveis)
float CUA_Aloc = TotalDem / (sum(k in kDepots) Vk[k]*zk[k]); //
Utilização dos Armazéns Alocados
float SSR = sum(k in kDepots) Zalfa * (sqrt ( sum( l in lClients )
fikl[k][l]*sigl2[l]*ykl[k][l])); // Estoque de Segurança Real
float SSL = sum(k in kDepots) Zalfa * (sum( r in rSegments)
(Fkr[k][r]*ukr[k][r] + Ckr[k][r]*sum( l in lClients ) vklr[k][l][r]));
// Estoque de Segurança Linearizado
float SSE = 100*(1 - SSL / SSR); // Erro Percentual

```

Código em OPL do Modelo de Índice Triplo

```
// ----- DADOS ----- //

// Outputfile
string outfile = ...;

// Parameters
int Tal = ...;
float Zalfa = ...;
float TotalDem = ...;
float MaxFkr = ...;
float MaxCkr = ...;

// Index
{string} jSuppliers = ...;
{string} kDepots = ...;
{string} lClients = ...;
int nr = ...;
range rSegments = 1..nr;
range rLSegments = 0..nr;

// Piecewise Linear Coeficients
float Lr [rLSegments]= ...;
float Fr [rSegments]= ...;
float Cr [rSegments]= ...;

float Fkr [k in kDepots][r in rSegments] = Fr [r] ;
float Ckr [k in kDepots][r in rSegments] = Cr [r] ;

// Costs and Capacities
int Fj[jSuppliers] = ...;
int Pj[jSuppliers] = ...;
float Cjk[jSuppliers][kDepots]= ...;

int Gk[kDepots]= ...; //Custo anual de abertura de um
CD na localidade k
float Hk[kDepots]= ...; //Custo de manutenção de estoques
para uma unidade de produto por um ano no depósito k
float Vk[kDepots]= ...; //Capacidade do depósito k
float Tkl[kDepots][lClients]= ...; //Custo de envio de uma unidade
do CD k para o cliente l

// Demand (mean e deviation) leadtime and coerture parameter
float mil[lClients]= ...;
float sigl2[lClients]= ...;
float fikl[kDepots][lClients]= ...; //Lead time de envio
do depósito K para o cliente L
int qsikl[kDepots][lClients] = ...; //Cobertura dos depósitos - 1 if
DC k cover (is in an appropriate distance) client l; 0 otherwise.

// ----- VARIÁVEIS DE DECISÃO ----- //

// MIP
dvar boolean wj[jSuppliers]; //Abertura ou não de uma fábrica
(Supplier)
dvar boolean zk[kDepots]; //Abre ou não um depósito em k
```

```

dvar float+ xjkl[jSuppliers][kDepots][lClients]; //Número de unidades
de produto enviada de j para k visando suprir a demanda de l
dvar float+ xsjk[jSuppliers][kDepots]; //Número de unidades de
produto enviada de j para k visando suprir o estoque de segurança de k
dvar boolean ykl[kDepots][lClients]; //Alocação ou não do cliente l ao
depósito k (por ser binária impede o Cross-filling)

dvar boolean ukr [kDepots][rSegments]; //Variável auxiliar binária da
linearização por partes
dvar float+ vklr [kDepots][lClients][rSegments]; //Variável auxiliar
contínua da linearização por partes

// Decision variables to track the Obj components

dvar float+ TC; /* Total Cost */
dvar float+ SFC; /* Suppliers Fixed Cost */
dvar float+ DFC; /* Depot Fixed Cost */
dvar float+ PTC; /* Production and Transport Cost through both
Echelons*/
dvar float+ PCSS; /* Production Cost of Safety Stock units */
dvar float+ TSC; /* ToTal Stock Cost */

// ----- MODELO ----- //

//Objective
minimize
    SFC + DFC + PTC + PCSS + TSC;

//Constraints
subject to {

    SFC == sum( j in jSuppliers) Fj[j]*wj[j];
    DFC == sum( k in kDepots) Gk[k]*zk[k];
    PTC == sum( j in jSuppliers, k in kDepots, l in lClients)
((Cjk[j][k] + Tkl[k][l])*xjkl[j][k][l]);
    PCSS == sum( j in jSuppliers, k in kDepots) Cjk[j][k]*xsjk[j][k];
    TSC == sum( k in kDepots) Hk[k]*(( sum( j in jSuppliers, l in
lClients) xjkl[j][k][l]) + (sum( j in jSuppliers) xsjk[j][k]));
    TC == SFC + DFC + PTC + PCSS + TSC;

    // (20)
    forall( k in kDepots, l in lClients )
        Demandfulfillment:
            sum( j in jSuppliers) xjkl[j][k][l] == Tal*mil[l]*ykl[k][l];
//O horizonte de tempo é igual a l

    // (21)
    forall (j in jSuppliers)
        SupplierCapacity:
            sum ( k in kDepots, l in lClients )xjkl[j][k][l] + (sum(k in
kDepots) xsjk[j][k]) <= Pj[j]*wj[j];

    // (22)
    forall( k in kDepots )
        DepotCapacity:
            (sum(j in jSuppliers, l in lClients ) xjkl[j][k][l]) +
(sum(j in jSuppliers) xsjk[j][k]) <= Vk[k]*zk[k];

```

```

// (23)
forall( k in kDepots )
    PiecewiseLinear1:
        sum( j in jSuppliers ) xsjk[j][k] == Zalfa * (sum( r in
rSegments) (Fkr[k][r]*ukr[k][r] + Ckr[k][r]*sum( l in lClients )
vklr[k][l][r]));

// (5)
forall( k in kDepots )
    DepotCapacity2:
        Tal * sum( l in lClients ) (mil[l]*ykl[k][l])
        + Zalfa * (sum( r in rSegments) (Fkr[k][r]*ukr[k][r] +
Ckr[k][r]*sum( l in lClients ) vklr[k][l][r])) <= Vk[k]*zk[k];

// (6)
forall ( l in lClients )
    Demandfulfillment2:
        sum( k in kDepots) ykl[k][l] == 1;

// (7)
forall( k in kDepots, l in lClients )
    CoverturesDepotClient:
        ykl[k][l] <= qsikl[k][l]*zk[k];

// (11)
forall( k in kDepots )
    PiecewiseLinear2:
        sum( l in lClients ) fikl[k][l]*sigl2[l]*ykl[k][l] == sum( r
in rSegments, l in lClients ) vklr[k][l][r];

// (12) - Primeira Metade da Inequação
forall( k in kDepots, r in rSegments : r !=1) //r>=2//
    PiecewiseLinear3a:
        Lr[r-1]*ukr[k][r] <= sum( l in lClients ) vklr[k][l][r];

// (12) - Segunda Metade da Inequação
forall( k in kDepots, r in rSegments : r !=1 )
    PiecewiseLinear3b:
        sum( l in lClients ) vklr[k][l][r] <= Lr[r]*ukr[k][r] ;

// (13)
forall( k in kDepots )
    PiecewiseLinear3:
        sum( r in rSegments) ukr[k][r] <= 1;

// (18)
forall( k in kDepots, l in lClients, r in rSegments )
    PiecewiseLinear4:
        vklr[k][l][r] <= fikl[k][l]*sigl2[l]*ukr[k][r];
}

// ----- PÓS-PROCESSAMENTO ----- //

int NSuppliers = sum(j in jSuppliers) wj[j];
int NDepots = sum(k in kDepots) zk[k];
float CUF_Aloc = TotalDem / sum(j in jSuppliers) Pj[j]*wj[j];
//Utilização dos Fornecedores Alocados (Diferente do CUF da instância,
qué dos fornecedores disponíveis)
float CUA_Aloc = TotalDem / sum(k in kDepots) Vk[k]*zk[k]; //
Utilização dos Armazéns Alocados

```

```

float SSR = sum(k in kDepots) Zalfa * (sqrt ( sum( l in lClients )
fikl[k][l]*sigl2[l]*ykl[k][l])); // Estoque de Segurança Real
float SSL = sum(k in kDepots) Zalfa * (sum( r in rSegments)
(Fkr[k][r]*ukr[k][r] + Ckr[k][r]*sum( l in lClients ) vklr[k][l][r]));
// Estoque de Segurança Linearizado
float SSE = 100*(1 - SSL / SSR); // Erro Percentual

```


Apêndice IV

Código em C++ das Interfaces de Comunicação

```
// --- Bibliotecas e Includes --- //
#include <ilopl/iloopl.h>
#include <time.h> //Biblioteca para cronometrar o tempo de execução
//Bibliotecas para manipulação de arquivos
#include <windows.h>
#include <iostream>
#include <fstream>
#include <string>
#include <cstdlib>
using namespace std;
#include <sstream>

// --- Declaração de Variáveis Globais --- //
struct Instancia
{
    string arq_dat; // Nome de um arquivo .dat de
uma instância (IL-EX-10-20-150 G1 (01) 05-0,49-0,40 - ID-SL.dat)
    string arq_dat2; // Nome da instancia com o 2 no final
para que ela não seja executada por nenhum outro processo (IL-EX-10-20-150 G1
(01) 05-0,49-0,40 - ID-SL.dat2)
    string arq_xlsx; // Nome de um arquivo .xlsx de uma
instância (IL-EX-10-20-150 G1 (01) 05-0,49-0,40.xlsx)
    string prefixo; // Prefixo de uma instância
(IL-EX-10-20-150 G1 (01) 05-0,49-0,40)
    string sufixo_modelo; // Sufixo de um arquivo .dat relativo ao
modelo a que se refere (ID-SL)
    string path; // Aponta o path completo da
instância a ser processada pela função ProcessarModelo
    string path2; // Aponta o path completo do
arquivo arq_dat2
    string path_exec; // Aponta o path completo para o qual a
instância deve ser movida após o seu processamento
};
struct Diretorios
{
    string arquivo_resultados;
    string raiz;
    string modelo;
    string instancias;
    string instancias_exec;
};
int status; //Status retornado à função main (0 caso o programa tenha sido
executado com sucesso)

// --- Protótipos das Funções --- //
string SelecionarModelo (void);
void ProcessarModelo (const Diretorios& diretorio, const Instancia& instancia);
void LoopModelos (const Diretorios& diretorio, const string& sufixo_modelo);
void ProcessarTodosOsModelos (Diretorios& diretorio);

int main(int argc, char* argv[]) {

    // Definição de compatibilidade com PT-BR
    setlocale(LC_ALL, "Portuguese");
    locale locBr("Portuguese-Brazilian");
    cout.imbue(locBr);
```

```

// --- Declaração de Variáveis --- //

//Diretórios do Problema
Diretorios diretorio;
//diretorio.raiz =
"C:\\Users\\Gustavo\\Desktop\\Gustavo\\Projetos\\Mestrado\\Dissertação\\Modelos e
Implementação\\";
diretorio.raiz = "C:\\Users\\Administrator\\Desktop\\Gustavo\\Modelos e
Implementação\\";
diretorio.instancias = diretorio.raiz + "Instâncias\\";
diretorio.instancias_exec = diretorio.instancias + "Executadas\\";
diretorio.arquivo_resultados = diretorio.raiz +
"Experimentos\\Resultados - Instâncias.csv"; //Endereço para gravação do
arquivo .CSV com os resultados gerados

//Variáveis Diversas
string sufixo_modelo; //Especifica o sufixo_modelo que será processado

// --- Seleção do sufixo_modelo sobre o qual os testes serão executados
--- //

sufixo_modelo = SelecionarModelo();

if (sufixo_modelo == "ID-SL" || sufixo_modelo == "ID-SQ")
{
    diretorio.modelo = diretorio.raiz + "Modelos\\Indice-Duplo-
MIP.mod";
    LoopModelos (diretorio, sufixo_modelo);
}
else if (sufixo_modelo == "IT-SL" || sufixo_modelo == "IT-SQ")
{
    diretorio.modelo = diretorio.raiz + "Modelos\\Indice-Triplo-
MIP.mod";
    LoopModelos (diretorio, sufixo_modelo);
}
else
{
    // -----> Processar todos os modelos <----- //
    ProcessarTodosOsModelos(diretorio);
}

//cout << endl << "--Pressione <Enter> para finalizar--" << endl;
//getchar();
cout << endl;
system("pause");

return status;
}

string SelecionarModelo (void)
{
    string sufixo_modelo = ""; //string que recebe tanto a opção a ser
escolhida quanto o sufixo_modelo que será trabalhado
    char op_modelo;

    cout << "Defina o modelo a ser executado:" << endl << endl
        << "1 - Modelo de Índice Duplo / Segmentos
Lineares (ID-SL)" << endl

```

```

Quadráticos (ID-SQ)" << endl          << "2 - Modelo de Índice Duplo / Segmentos
Lineares (IT-SL)" << endl              << "3 - Modelo de Índice Triplo / Segmentos
Quadráticos (IT-SQ)" << endl          << "4 - Modelo de Índice Triplo / Segmentos
                                          << "5 - Processa todas as Instâncias" << endl
                                          << "0 - Finaliza a execução" << endl << endl
                                          << "Opção: ";

cin >> op_modelo; //Armazena o número da opção como uma string

switch (op_modelo)
{
    case '1':
        sufixo_modelo = "ID-SL";
        cout << endl << endl << "--- Modelo de Índice Duplo /
Segmentos Lineares (ID-SL) ---"
        << endl << endl <<
"===== " << endl;
        return sufixo_modelo;

    case '2':
        sufixo_modelo = "ID-SQ";
        cout << endl << endl << "--- Modelo de Índice Duplo /
Segmentos Quadráticos (ID-SQ) ---"
        << endl << endl <<
"===== " << endl;
        return sufixo_modelo;

    case '3':
        sufixo_modelo = "IT-SL";
        cout << endl << endl << "--- Modelo de Índice Triplo /
Segmentos Lineares (IT-SL) ---"
        << endl << endl <<
"===== " << endl;
        return sufixo_modelo;

    case '4':
        sufixo_modelo = "IT-SQ";
        cout << endl << endl << "--- Modelo de Índice Triplo /
Segmentos Quadráticos (IT-SQ) ---"
        << endl << endl <<
"===== " << endl;
        return sufixo_modelo;

    case '5':
        sufixo_modelo = "Todos";
        cout << endl << endl << "--- Processando Todas as Instâncias
e Modelos ---"
        << endl << endl <<
"===== " << endl;
        return sufixo_modelo;

    case '0':
        exit(EXIT_SUCCESS);
        return "0";

    default:
        system ("cls");
        cout << "\a\nOpcao inválida!\n\n";
}

```

```

        SelecionarModelo();
        return "0";
    }
}

void ProcessarModelo (const Diretorios& directorio, const Instancia& instancia)
{
    // --- Variáveis Locais --- //

    IloEnv env;
    IloNum gap;
    fstream resultados;
    locale locBr("Portuguese-Brazilian");
    string dados_inst = instancia.arq_dat;

    status = 127;

    // --- Cabeçalho indicando a instância que está sendo
processada --- //
    cout << endl << endl << endl
        <<
"===== " << endl
<< "          |" << endl
        <<
"===== " <<
endl;

    // --- Processando o sufixo_modelo --- //

    try {
        IloOplErrorHandler handler(env,cout);
        IloOplModelSource modelSource(env,
diretorio.modelo.c_str()); //Arquivo do sufixo_modelo do OPL
        IloOplSettings settings(env,handler);
        IloOplModelDefinition def(modelSource,settings);
        IloCplex cplex(env);
        IloOplModel opl(def,cplex);
        IloOplDataSource dataSource(env,
instancia.path2.c_str()); //Arquivo .dat da instância
        opl.addDataSource(dataSource);
        opl.generate();

        // ##### Já posso mudar o arquivo da instância de
nome!!! #####

        //Desligar o Presolver do CPLEX
        cplex.setParam(IloCplex::PreInd,0);
        cplex.setParam(IloCplex::RelaxPreInd,0);
        cplex.setParam(IloCplex::PreslvNd,-1);
        cplex.setParam(IloCplex::Reduce, 0);
        cplex.setParam(IloCplex::TiLim, 7200); //Tempo
limite do CPLEX (60 segundos)
        cplex.setWarning(env.getNullStream()); // Desliga
os Warnings

        //Desligar a inserção de cortes
        cplex.setParam(IloCplex::CutsFactor,0);

        // --- Tempo de Processamento - Início --- //
        clock_t tempo; //Cria uma variável do tipo Clock
        tempo = clock(); //Inicia a contagem
    }
}

```

```

if ( cplex.solve() ) {

// --- Tempo de Processamento - Fim ---
//
// contagem
tempo = clock() - tempo; //Finaliza a

// --- Gap --- //
gap = 100 * cplex.getMIPRelativeGap();

// --- Impressão dos Resultados --- //
cout << endl
<< "-----" << endl
<< "Instância: " << instancia.arq_dat <<
endl
<< "Tempo de processamento: " <<
((float)tempo)/CLOCKS_PER_SEC << " segundos." << endl
<< "Valor da Função Objetivo: " << fixed
<< setprecision(2) << opl.getCplex().getObjValue() << endl
<< "Número de Variáveis: " << fixed <<
opl.getCplex().getNcols() << endl
<< "Numero de Variáveis Binárias: " <<
fixed << opl.getCplex().getNbinVars() << endl
<< "Numero de Coeficientes Não-Zerados:
" << fixed << opl.getCplex().getNNZs() << endl
<< "Numero de Restrições: " << fixed <<
opl.getCplex().getNrows() << endl
<< "NSuppliers = " <<
opl.getElement("NSuppliers").asInt() << endl
<< "NDepots = " <<
opl.getElement("NDepots").asInt() << endl
<< "CUF_Aloc = " <<
opl.getElement("CUF_Aloc").asNum() << endl
<< "CUA_Aloc = " <<
opl.getElement("CUA_Aloc").asNum() << endl
<< "SSR = " <<
opl.getElement("SSR").asNum() << endl
<< "SSL = " <<
opl.getElement("SSL").asNum() << endl
<< "SSE = " <<
opl.getElement("SSE").asNum() << "%" << endl
<< "Gap (%) = " << gap << "%" << endl
<< "Status = " << cplex.getStatus() <<
endl
<< "Cplex Status = " <<
cplex.getCplexStatus() << endl;

cout << endl << "-----***-----" <<

// --- Gravação do Arquivo --- //
resultados.imbue(locBr);
resultados.open
(diretorio.arquivo_resultados.c_str(), ios::out | ios::app);

// Separar campos do nome da instância
para o título

```

```

dados_inst.size()-4);
EX-"), 6, "");
1, ";");
1, ";");
1, ";");
("), 2, ";");
"), 2, ";");
1, ";");
1, ";");
"), 3, ";");
1, ";");

<< ";"
";"
((float)tempo)/CLOCKS_PER_SEC << ";"
setprecision(2) << opl.getCplex().getObjValue() << ";"
opl.getCplex().getNcols() << ";"
opl.getCplex().getNbinVars() << ";"
opl.getCplex().getNNZs() << ";"
opl.getCplex().getNrows() << ";"
opl.getElement("NSuppliers").asInt() << ";"
opl.getElement("NDepots").asInt() << ";"
opl.getElement("CUF_Aloc").asNum() << ";"
opl.getElement("CUA_Aloc").asNum() << ";"
opl.getElement("SSR").asNum() << ";"
opl.getElement("SSL").asNum() << ";"
opl.getElement("SSE").asNum() << "%" << ";"
<< "%" << ";"
cplex.getStatus() << ";"
cplex.getCplexStatus();

dados_inst.assign(dados_inst, 0,
dados_inst.replace(dados_inst.find("IL-
dados_inst.replace(dados_inst.find("-"),
dados_inst.replace(dados_inst.find("-"),
dados_inst.replace(dados_inst.find(" "),
dados_inst.replace(dados_inst.find("
dados_inst.replace(dados_inst.find("-"),
dados_inst.replace(dados_inst.find("-"),
dados_inst.replace(dados_inst.find(" -
dados_inst.replace(dados_inst.find("-"),

// Gravar resultados no arquivo .csv
resultados << endl << instancia.arq_dat

<< dados_inst <<
<<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed <<
<< fixed << gap
<< fixed <<
<< fixed <<

resultados.close();

```

```

// --- Pós-processamento --- //

opl.postProcess();
//opl.printSolution(cout);
status = 0;
    } else {
        cout << "\nProblema sem solução!\n" <<
            status = 1;
    }
} catch (IloOplException & e) {
    cout << "### OPL exception: " << e.getMessage() <<
endl;

} catch( IloException & e ) {
    cout << "### CONCERT exception: ";
    e.print(cout);
    status = 2;
} catch (...) {
    cout << "### UNEXPECTED ERROR ..." << endl;
    status = 3;
}

env.end();

// Move o arquivo .dat da instância após a sua execução
if (MoveFile(instancia.path2.c_str(),
instancia.path_exec.c_str()))
{
    cout << endl << "- O arquivo " <<
instancia.arq_dat << " foi movido com sucesso!" << endl;
}
else
{
    cout << endl << "- Erro ao mover o arquivo " <<
instancia.arq_dat << ": " << GetLastError() << endl;
}

}

void LoopModelos (const Diretorios& diretorio, const string& sufixo_modelo)
{
    // Declaração de Variáveis Locais
    string dat = "*" + sufixo_modelo + ".dat"; // Extensão dos arquivos de
instância de teste
    Instancia instancia;
    // Estrutura da instancia

    // --- Listagem de Instâncias no Diretório de Modelos --- //

    ifstream inputFile;
    WIN32_FIND_DATA FindData;
    HANDLE hFind;

    hFind = FindFirstFile((diretorio.instancias + dat).c_str(), &FindData);
    // Seleciona a primeira instância a ser processada

    // --- Laço que percorre todos os arquivos de instância .dat --- //
    do
    {

```

```

// Path da instância a ser processada e para onde ela deve
ser movida
instancia.path = diretorio.instancias + FindData.cFileName;
// Fornece o path completo da instância a ser processada na sequência
instancia.path_exec = diretorio.instancias_exec +
FindData.cFileName; // Fornece o path completo para o qual a instância deve ser
movida
instancia.arq_dat = FindData.cFileName;

// Cria arq_dat2 para processamento paralelo na Amazon AWS
instancia.arq_dat2 = instancia.arq_dat + "2";
instancia.path2 = diretorio.instancias + instancia.arq_dat2;
if (MoveFile(instancia.path.c_str(),
instancia.path2.c_str()))
{
    cout << endl << "=> Processando " <<
instancia.arq_dat << " ..." << endl;
}
else
{
    cout << endl << "- Erro ao criar o arquivo DAT2 da
instancia " << instancia.arq_dat << ": " << GetLastError() << endl;
}

// Processa a Instância
ProcessarModelo(diretorio, instancia);
}
while (FindNextFile(hFind, &FindData) > 0); // Seleciona a próxima
instância a ser processada, utilizando o critério fornecido no handler hFind

inputFile.close();
}

void ProcessarTodosOsModelos (Diretorios& diretorio)
{
    // --- Declaração de Variáveis --- //
    string dat = "*.dat";
    // Extensão dos arquivos de instância de teste
    Instancia instancia; // Cria
uma instância do objeto "Instância"

    // --- Listagem de Instâncias no Diretório de Modelos --- //

    ifstream inputFile;
    WIN32_FIND_DATA FindData;
    HANDLE hFind;

    hFind = FindFirstFile((diretorio.instancias + dat).c_str(), &FindData);
    // Seleciona a instância a ser processada

    while (hFind != INVALID_HANDLE_VALUE)
    {
        do
        {
            // Preenche os dados da struct Instância relativos
            à primeira instância escolhida
            instancia.arq_dat = FindData.cFileName;

```



```

instancia.arq_dat.size()-12);
instancia.arq_dat.size()-9, 5);
instancia.arq_dat;
instancia.arq_dat;

instancia.prefixo.assign(instancia.arq_dat, 0,
instancia.arq_xlsx = instancia.prefixo + ".xlsx";
instancia.sufixo_modelo.assign(instancia.arq_dat,
instancia.path = diretorio.instancias +
instancia.path_exec = diretorio.instancias_exec +

// Cria arq_dat2 para processamento paralelo na
Amazon AWS
instancia.arq_dat2 = instancia.arq_dat + "2";
instancia.path2 = diretorio.instancias +
instancia.arq_dat2;
instancia.path2.c_str()))
if (MoveFile(instancia.path.c_str(),
{
    cout << endl << "=> Processando " <<
instancia.arq_dat << " ..." << endl;
}
else
{
    cout << endl << "- Erro ao criar o
arquivo DAT2 da instancia " << instancia.arq_dat << ": " << GetLastError() <<
endl;
}

dat = instancia.prefixo + "*.dat"; // Altera a
string "dat" para que todos os arquivos .dat atrelados a uma mesma instância
(.xlsx) sejam executados antes de passar para o próximo

if (instancia.sufixo_modelo == "ID-SL" ||
instancia.sufixo_modelo == "ID-SQ")
{
    diretorio.modelo = diretorio.raiz +
"Modelos\\Indice-Duplo-MIP.mod";
}
else if (instancia.sufixo_modelo == "IT-SL" ||
instancia.sufixo_modelo == "IT-SQ")
{
    diretorio.modelo = diretorio.raiz +
"Modelos\\Indice-Triplo-MIP.mod";
}
else
{
    cout << "Erro no sufixo de determinação
do modelo da instancia " << instancia.arq_dat << endl
        << "Sufixo: " <<
instancia.sufixo_modelo;
    system("pause");
}

// Processar o modelo selecionado pelo handler
hFind
ProcessarModelo(diretorio, instancia);
FindClose(hFind);
hFind = FindFirstFile((diretorio.instancias +
dat).c_str(), &FindData); // Seleciona a instância a ser
processada

```

```

        //FindNextFile(hFind, &FindData);
    } while (hFind != INVALID_HANDLE_VALUE);

    // Mover o arquivo .xlsx após o processamento dos arquivos
    .dat associados a ele
    if (MoveFile((diretorio.instancias +
instancia.arq_xlsx).c_str(), (diretorio.instancias_exec +
instancia.arq_xlsx).c_str()))
    {
        cout << endl << "- O arquivo " <<
instancia.arq_xlsx << " foi movido com sucesso!" << endl << endl
        << endl << "---- FIM DO PROCESSAMENTO
DA INSTÂNCIA " << instancia.prefixo << " ----" << endl << endl << endl;
    }
    else
    {
        cout << endl << " Erro ao mover o arquivo " <<
instancia.arq_xlsx << ": " << GetLastError() << endl << endl
        << endl << "---- FIM DO PROCESSAMENTO
DA INSTÂNCIA " << instancia.prefixo << " ----" << endl << endl << endl;
    }

    // Encontrar o novo arquivo depois de mover o arquivo .xlsx
    e trocar o prefixo da instância
    dat = "*.dat";
    FindClose(hFind);
    hFind = FindFirstFile((diretorio.instancias + dat).c_str(),
&FindData);
        // Seleciona a instância a ser processada

    }

    cout << endl
    << "
----" << endl << endl;
}
    << "
---->> Experimentos Finalizados! <<-
}

```