

UNIVERSIDADE DE SÃO PAULO

Escola de Engenharia de São Carlos

MODELOS MATEMÁTICOS PARA
PROJETO DE REDES COM RESTRIÇÕES
DE CONFIABILIDADE

Luis Henrique Silvério Tirelli



São Carlos - SP

LUIS HENRIQUE SILVÉRIO TIRELLI

**MODELOS MATEMÁTICOS PARA
PROJETO DE REDES COM
RESTRIÇÕES DE CONFIABILIDADE**

Trabalho de Conclusão de Curso
Escola de Engenharia de São Carlos
Universidade de São Paulo
Engenharia Elétrica com ênfase em Eletrônica

ORIENTADOR: Prof. Dr. Alysson Machado Costa

São Carlos
2012

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Atendimentos ao Usuário do Serviço de
Biblioteca – EESC/USP.

T596m Tirelli, Luis Henrique Silvério
Modelos matemáticos para projeto de redes com
restrições de confiabilidade. / Luis Henrique Silvério
Tirelli; orientador Alysson Machado Costa. São Carlos,
2012.

Monografia (Graduação em Engenharia Elétrica com
ênfase em Eletrônica) -- Escola de Engenharia de São
Carlos da Universidade de São Paulo, 2012.

1. Pesquisa operacional. 2. Modelos lineares. 3.
Otimização. 4. Projeto de redes. 5. Confiabilidade. I.
Título.

FOLHA DE APROVAÇÃO

Nome: Luís Henrique Silvério Tirelli


Título: "Modelos Matemáticos para Projeto de Redes com Restrições de Confiabilidade"

Trabalho de Conclusão de Curso defendido e aprovado
em 20 / 6 / 2012,

com NOTA 9,0 (Nota, zero), pela comissão julgadora:


Prof. Dr. Alysson Machado Costa (Orientador) - ICMC/USP


Prof. Dr. José Carlos de Melo Vieira Júnior - EESC/USP


Prof. Associado Ivan Nunes da Silva - EESC/USP


Prof. Associado Homero Schiabel
Coordenador da CoC-Engenharia Elétrica
EESC/USP

DEDICATÓRIA

Por nunca medirem qualquer tipo de esforço para me apoiar, dedico esse trabalho aos meus pais, Claudiney e Irani. Todas as conquistas e realizações que um dia alcançarei em minha vida só serão possíveis porque Deus me proporcionou pais tão sábios e dispostos a me educar.

AGRADECIMENTOS

A Deus, por ser o meu porto seguro e por iluminar sempre minha trajetória na vida.

Aos meus pais, irmãos e demais familiares, por me apoiarem em todos os momentos, mesmo algumas vezes à distância. A eles devo tudo o que fiz e o que sou hoje.

Ao professor Alysson Machado Costa, pela paciência e dedicação com que me orientou durante todo o desenvolvimento deste trabalho. Por todo o conhecimento e conselhos passados a mim, muito obrigado.

Aos amigos de Lorena, Pedro, Mateus, Carlos, Beбето, Pâmela, Gustavo, Raul, Daniel e Diogo, por me fazerem entender o real significado da palavra amizade e o fato de que sem ela não somos nada.

Ao companheiro de república e quase um irmão Adriano, por tudo que compartilhamos juntos e pela amizade que sei que levaremos pra sempre.

Aos colegas de classe, que com o passar dos anos se tornaram mais do que amigos, compartilhando viagens, festas, churrascos e vários outros momentos de alegria.

SUMÁRIO

LISTA DE FIGURAS	i
LISTA DE TABELAS.....	ii
RESUMO.....	iii
ABSTRACT	iv
1. Introdução	1
2. A Pesquisa Operacional.....	3
2.1. Introdução	3
2.2. Importância	5
3. Estudo de Modelos Lineares.....	6
3.1. Exemplos de aplicações de Modelos Lineares	6
3.1.1. Problemas de Transporte	7
3.1.2. Problemas de mistura.....	9
3.2. Aplicações de Modelos Matemáticos em Projeto de Redes	11
4. Modelos Estudados.....	12
4.1. Introdução	12
4.2. Análise de Modelos Matemáticos em Projeto de Redes.....	12
4.2.1. Problema de Projeto de Redes Não-Capacitadas.....	12
4.2.2. Problema de Projeto de Redes Capacitadas.....	14
4.2.3. Um novo modelo de projeto de redes com restrições de HOP	14
4.2.4. Modelo de Projeto de Redes com Restrições “soft” de HOP	16
5. Implementações Básicas.....	18
5.1. Usando o <i>solver</i> do Microsoft Excel	18
5.1.1. Inserção dos Dados no Excel.....	18
5.1.2. Inserção da Função Objetivo e do Conjunto de Restrições	19
5.1.3. Utilização do <i>Solver</i>	21
5.1.4. Resultado	22
5.2. Gerando um arquivo .lp através de um programa em C	23
5.2.1. Arquivos de Dados	24
5.2.2. Arquivo Gerado	24
5.3. Usando a tecnologia CONCERT do IBM ILOG CPLEX	26
5.3.1. Declaração de Variáveis	26
5.3.2. Função objetivo	27
5.3.3. Restrições.....	27
5.3.4. Extração e Resolução do Modelo	27
5.3.5. Obtenção do Valor da Função Objetivo	27

6. Resultados e Análise	29
6.1. Redes Capacitadas e Modelos com HOP.....	29
6.2 - Modelos com Restrições “soft” de HOP	31
7. Considerações Finais.....	34
REFERÊNCIAS BIBLIOGRÁFICAS	35
APÊNDICE	37

LISTA DE FIGURAS

Figura 2.1: Esquema geral de trabalho em Pesquisa Operacional.....	4
Figura 3.1: Esquematização Básica de um Problema de Transporte	8
Figura 5.1: Inserção dos dados na planilha para o exemplo implementado no Excel.....	19
Figura 5.2: Inserção do conjunto de restrições na planilha para o exemplo implementado no Excel.....	20
Figura 5.3: Planilha completa e pronta para a utilização do <i>so/ver</i> no exemplo implementado no Excel.....	21
Figura 5.4: Tela Principal do <i>So/ver</i> do Excel.....	21
Figura 5.5: Adição dos Parâmetros no <i>So/ver</i> para o exemplo implementado no Excel	22
Figura 5.6: Modelo Completo Resolvido relacionado ao exemplo implementado no Excel ..	23
Figura 5.7: Fluxo correspondente à solução ótima do exemplo implementado	23
Figura 5.8: Dados utilizados como exemplo para a geração de um arquivo .lp seguindo o padrão das instâncias descritas por Ghamlouche et al. (2004).	24

LISTA DE TABELAS

Tabela 3.1: Dados do exemplo do Problema de Mistura	9
Tabela 5.1: Dados do Grafo para o exemplo implementado no Excel	18
Tabela 5.2: Dados dos produtos para o exemplo implementado no Excel	19
Tabela 6.1: Resultados para Redes Capacitadas e Modelos com HOP para as 5 instâncias de dados.....	29
Tabela 6.2 - Resultados para Redes Capacitadas e Modelos com HOP para 5 outras instâncias de dados	30
Tabela 6.3: Resultados para o Modelo com Restrições “soft” de HOP para as 5 primeiras instâncias.....	31
Tabela 6.4: Resultados para o Modelo com Restrições “soft” de HOP para 5 outras instâncias.....	32

RESUMO

Problemas de projeto de redes constituem uma classe importante dentre os problemas de otimização, aparecendo em inúmeras situações e desafios práticos. A resolução de tais problemas se baseia fortemente em métodos e modelos matemáticos, objetos de estudo deste trabalho de conclusão de curso. Sendo assim, um estudo profundo referente a modelos matemáticos para problemas de projeto de redes foi realizado, com um particular interesse pela questão da confiabilidade das redes, i.e., o comportamento das soluções obtidas quando da ocorrência de falhas. O trabalho apresenta, dessa forma, toda a abordagem feita aos modelos citados anteriormente bem como a implementação dos mesmos com a utilização de softwares, analisando os resultados para diversas instâncias de dados.

Palavras chave: Pesquisa operacional, modelos lineares, otimização, projeto de redes, confiabilidade.

ABSTRACT

Network design problems are an important class of optimization problems, appearing in numerous and practical situations. Many approaches for the solution of these problems rely on mathematical models and methods. In this study, we present an analysis of a class of mathematical models for network design problems with a particular interest in the reliability of these networks. Several computational tests were effected in order to verify the efficiency of the models and the effect of considering the obtained network reliability.

Keywords: Operational research, linear models, optimization, network design, reliability.

1. Introdução

Problemas de projeto de redes compõem uma importante parcela dentre os problemas de otimização, estando presente em diversos contextos práticos e em uma grande variedade de áreas, tais como logística, engenharias e telecomunicações.

Nesse contexto, uma rede pode ser matematicamente modelada através da ajuda de grafos. De fato, seja um grafo $G = (V, A, K)$, onde V é um conjunto de vértices, A um conjunto de arestas e K um conjunto de produtos a serem transportados. Diversos problemas de projeto e operação de redes podem ser definidos em G : os nós em V , as arestas em A e os elementos de K podem representar locais ou equipamentos (cidades, centros de consumo, centrais telefônicas, etc.), as conexões possíveis entre estes locais ou equipamentos (estradas, linhas de transmissão, trechos de fibras óticas, etc.) e os produtos a serem transportados (pessoas, eletricidade, pacotes de dados, etc.), respectivamente.

Este trabalho trata de problemas de projeto de redes em um âmbito geral, mas que podem simular um determinado tipo de rede, como de telecomunicações, por exemplo. Neste caso, o universo de estudo é extremamente vasto e as aplicações envolvem desde redes de acesso local (Corte-Real & Gouveia 2007) a redes de largo alcance (Cabral et al. 2008), passando por redes de comunicações ópticas (Tornatore et al. 2007) e redes especializadas na troca de bases de dados (Ruffolo et al. 2007), para citar apenas alguns exemplos. Em todos estes casos, o problema básico gira em torno da escolha de arestas em A conectando elementos de V e permitindo o fluxo dos produtos K .

Nos problemas citados anteriormente, a questão da confiabilidade é particularmente importante, em que o funcionamento contínuo das redes é uma característica fundamental. Isso faz com que bons modelos sejam obrigados a levar em consideração a existência de falhas e defeitos nos componentes das redes, levando a soluções robustas. De fato, as soluções obtidas devem ser de boa qualidade não apenas para situações de operação normal, mas também para situações de contingência.

Este trabalho de conclusão de curso apresenta, portanto, uma ampla análise de modelos e métodos matemáticos aplicados a redes em geral, com um particular interesse pela questão da confiabilidade dessas redes.

Primeiramente, é apresentada uma fundamentação teórica relacionada ao estudo feito sobre Pesquisa Operacional (PO), que teve por objetivo a criação de uma base sólida na nomenclatura e no conhecimento da área, sobretudo no que diz respeito a modelos lineares. Desse modo, é possível, em seguida, o desenvolvimento de novas formulações que representassem problemas de interesse.

Após isso, os problemas ligados a projeto de redes passaram a ser o principal enfoque. Modelos clássicos foram estudados, baseados em um artigo de revisão do estado da arte do tema (Costa, 2005). O estudo iniciou-se com um modelo de redes não-capacitadas e capacitadas, sendo que a diferença entre elas é que no segundo a quantidade de fluxo que circula em cada arco é limitada a uma certa capacidade. Posteriormente, foram analisados modelos que lidam com a questão da confiabilidade através de restrições de saltos ou hops (Costa et al. 2009), que limitam o número de conexões ou links entre uma central e cada nó ou consumidor servido. Esta questão também foi amplamente estudada na literatura (Gouveia, 1999). Para este caso, dois modelos foram estudados, sendo que o último deles apresenta uma formulação inédita.

Finalmente, após a compreensão dos modelos citados acima, estes foram implementados e resolvidos usando-se os pacotes comerciais disponíveis no Microsoft Excel e no software IBM ILOG CPLEX, a fim de se obter resultados para análise.

2. A Pesquisa Operacional

2.1. Introdução

O termo Pesquisa Operacional (PO) surgiu durante a segunda guerra mundial por volta de 1939 onde se buscava resolver problemas de operações militares. Ao final da guerra, tanto na Inglaterra como nos EUA, a pesquisa operacional evoluiu consideravelmente.

Na década de 50 foram fundadas várias sociedades relacionadas à pesquisa operacional. Nesta mesma década foi realizada a primeira conferência internacional de pesquisa operacional, realizada em Oxford. Esta conferência foi a primeira oportunidade de troca de informações e constatações de trabalhos diferentes em diversas áreas apresentados pelos cientistas ingleses, que apresentaram problemas específicos, enquanto os americanos apresentaram modelos e métodos matemáticos em temas diversos.

Em 1967, o periódico inglês *Operational Research* propôs uma definição para pesquisa operacional que diz, segundo a tradução de Arenales et al. (2007), “Pesquisa Operacional é a aplicação de métodos científicos a problemas complexos para auxiliar no processo de tomadas de decisão, tais como projetar, planejar e operar sistemas em situações que requerem alocações eficientes de recursos escassos”.

Na década de 60, a pesquisa operacional conquistou os pesquisadores brasileiros e em 1968, em São José dos Campos-SP, no ITA (Instituto Tecnológico de Aeronáutica), foi realizado o primeiro simpósio brasileiro de pesquisa operacional, tendo sido fundada, em seguida, a Sociedade Brasileira de Pesquisa Operacional (SOBRAPO).

“De forma sucinta, podemos dizer que a pesquisa operacional é um enfoque científico sobre a tomada de decisões. A denominação pesquisa operacional é comumente motivo de críticas e reflexões, pois não revela a abrangência da área e pode dar a falsa impressão de estar limitada a análise de operações... O componente científico está relacionado a ideias e processos para articular e modelar problemas de decisão, determinando os objetivos do tomador de decisão e as restrições sob as quais se devem operar. Também está relacionado a métodos matemáticos para otimizar sistemas numéricos que resultam quando se usam dados nos modelos” (Arenales et al., 2007).

Assim, resumidamente, a pesquisa operacional estuda a aplicação do método científico, por equipes multidisciplinares, a problemas envolvendo o controle de sistemas organizados de forma a fornecer soluções que otimizem um ou mais objetivos de um ou mais afetados pelo processo decisório (*stakeholders*). Para tal, são utilizados um conjunto de técnicas e métodos matemáticos para auxiliar na tomada de decisões.

A PO pode ser aplicada a problemas onde é necessário especificar, de forma quantitativa, a condução e a coordenação das operações ou atividades dentro de uma organização. A natureza desta organização pode ser financeira, industrial, governamental, militar, entre outras.

Um diagrama esquemático que ilustra as etapas principais de um projeto de PO pode ser visto na Figura 2.1.

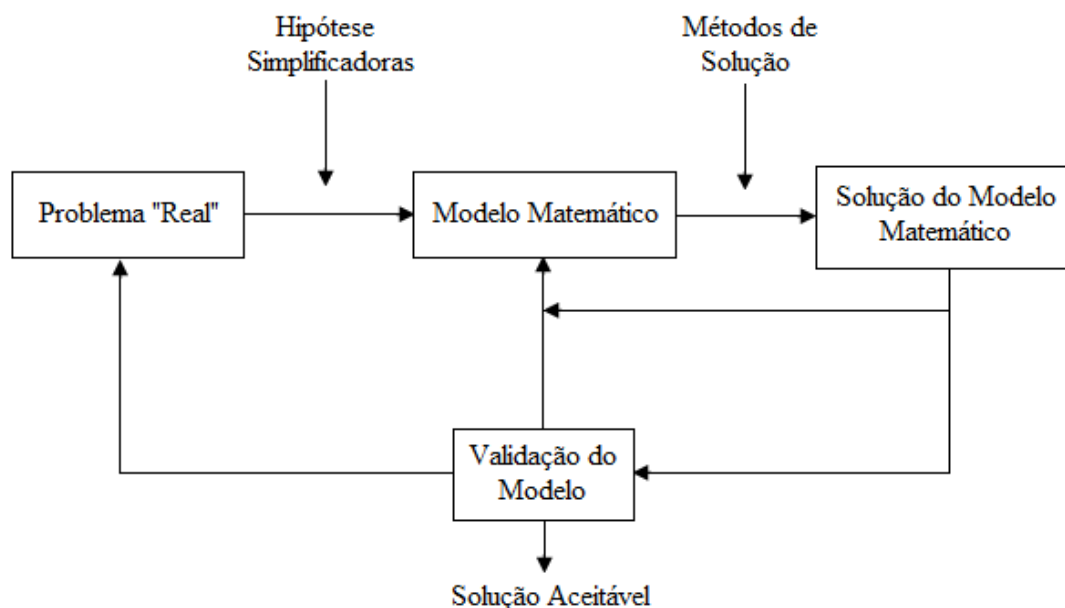


Figura 2.1: Esquema geral de trabalho em Pesquisa Operacional (Arenales et al, 2007)

Na construção de um modelo matemático devemos seguir uma série de etapas. A principal é ouvir aquele que lida com o problema real. Depois, deve-se descobrir o que será determinado (variáveis do problema), em função do que está disponível (dados do problema) e, por fim, reproduzir os caminhos que levam a uma solução (através da resolução de equações matemáticas ou implementação de algoritmos).

A busca de uma solução mais adequada entre diversas soluções alternativas traz consigo os elementos de um *problema de otimização*: um critério de avaliação das soluções alternativas, que nos permite dizer que uma solução é “melhor” que outra (de maneira objetiva ou subjetiva). A este critério de avaliação chamamos de função objetivo, que é justamente o que buscamos otimizar, ou seja, maximizar ou minimizar.

Por outro lado, as soluções obtidas devem ser passíveis de execução, ou seja, devem respeitar as restrições do problema.

2.2. Importância

De acordo com Mirshawka (1981), as características da PO são: pesquisa sobre as operações de toda a organização, a otimização das operações, aplicação dos mais recentes métodos e técnicas científicas, desenvolvimento e utilização dos modelos analíticos, projeto e utilização de operações experimentais, e, emprego de equipes mistas de pesquisa

A PO como ciência aplica-se a pessoas (organização e gerência, relações de trabalho, economia, decisões individuais, pesquisa do mercado, etc), a pessoas e máquinas (eficiência e produtividade, organização de fluxos em fábricas, métodos de controle de qualidade, organização de mudanças tecnológicas, etc) e ao movimento (transporte, estoque, distribuição, manipulação, comunicação, localização, etc).

Além disso, no mundo hoje em dia, a sobrevivência de uma organização impõe o planejamento. Com base nessas ideias, pode-se ratificar a importância da PO em qualquer tomada de decisão numa organização. Estas decisões devem ser efetuadas de forma concisa, tentando atingir um determinado objetivo e é neste ponto que entram os problemas de otimização.

3. Estudo de Modelos Lineares

Para Goldbarg & Luna (2000), um modelo tenta a representação substitutiva da realidade. Intuitivamente a maioria das pessoas já utilizou algum modelo para explicar algo a alguém. Por exemplo, quando um gráfico é feito, são utilizadas equações que representam sólidos (cone, cubo, pirâmide, etc), entre outras situações, ou seja, tenta-se transmitir e fazer interpretações da realidade através de metáforas/modelos.

“Um modelo não é igual à realidade, mas suficientemente similar para que as conclusões obtidas através de sua análise e/ou operação, possam ser estendidas à realidade” (Goldbarg & Luna, 2000).

A PO reúne as mais diversas técnicas e algoritmos que tentam estruturar e solucionar modelos quantitativos expressos matematicamente. Os principais modelos de pesquisa operacional são denominados de programação (no sentido de planejamento) matemática.

A Programação Matemática destaca-se principalmente devido a sua grande aplicabilidade na solução de problemas de otimização. Problemas de programação matemática podem ser classificados de acordo com as técnicas utilizadas para a resolução dos modelos matemáticos: problemas lineares (variáveis são contínuas e apresentam comportamento linear), problemas inteiros (se alguma variável está condicionada a assumir valores discretos), problemas não-lineares (quando exibe qualquer tipo de não-linearidade).

A dificuldade em modelar matematicamente um problema está em representar de forma adequada a realidade. Assim, um modelo que conseguir representar mais precisamente a realidade será de melhor qualidade.

Dentre os diversos modelos de programação citados anteriormente, o foco deste trabalho concentrou-se em problemas de programação linear.

3.1. Exemplos de aplicações de Modelos Lineares

Conforme já citado anteriormente, a PO se aplica a uma grande variedade de problemas. A maioria consiste em problemas de natureza tática e não estratégica. A distinção entre problemas táticos e problemas estratégicos se baseia em três aspectos:

1. Alcance do problema: Um problema é mais tático que outro se sua solução produzir efeito de duração mais curta ou se a solução pode ser modificada ou abandonada com facilidade.

2. Extensão do problema: Um problema é tanto mais estratégico quanto maior for a parte da organização diretamente afetada pela solução.

3. Orientação do problema: Um problema é tanto mais estratégico quanto mais envolver a determinação de finalidades, metas ou objetivos.

A aplicação da PO a grande variedade de problema táticos pode ser representada por um pequeno número de problemas típicos. Desenvolveram-se técnicas para modelá-los e obter soluções a partir dos modelos. Os problemas típicos são:

- Alocação;
- Estoque;
- Substituição ou reposição;
- Filas de espera;
- Seqüência e coordenação;
- Determinação de rotas;
- Situações de competição;
- Busca de informação.

Estas informações e o primeiro dos exemplos apresentados a seguir são baseados na apostila “Pesquisa Operacional I” de Baldo, T.A. Estes exemplos simplificados são alguns dos que foram estudados durante o trabalho e que fornecem a principal ideia dos problemas tratados pela otimização linear.

3.1.1. Problemas de Transporte

O Problema de Transporte pode ser considerado, talvez, o mais representativo dos Problemas de Programação Linear. Há uma vasta aplicação prática, tendo sido estudado por vários pesquisadores, embora tenha sido Dantzig o primeiro a estabelecer a sua formulação em programação linear (PL) e a propor um método sistemático de resolução.

O problema (geral) de transporte consiste em determinar a forma mais econômica (ou mais lucrativa) de enviar um bem disponível em quantidades limitadas de centros de produção aos mercados consumidores. Os produtos podem ser os mais variados possíveis: petróleo, equipamentos, máquinas, produção agrícola, energia elétrica, entre outros. Admite-se, geralmente, que as quantidades produzidas ou ofertadas em cada centro e as quantidades demandadas em cada mercado consumidor sejam conhecidas. Os transportes devem ser efetuados respeitando-se as limitações de oferta e atendendo à demanda.

Assim, o modelo dos transportes visa minimizar o custo total do transporte necessário para abastecer n centros consumidores (destinos), a partir de m centros fornecedores (origens), podendo ser esquematizado de acordo com a Figura 3.1.

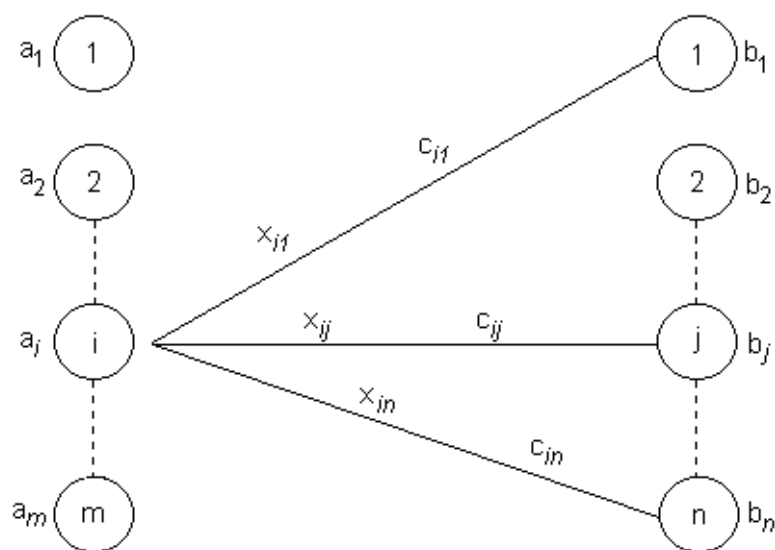


Figura 3.1: Esquematização Básica de um Problema de Transporte

O objetivo consiste em encontrar os valores de x_{ij} ($i = 1; \dots; m$ e $j = 1; \dots; n$) que minimize o custo total do transporte. Portanto, o modelo matemático será:

$$\text{Minimizar } \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (3.1)$$

sujeito a:

$$\sum_{j=1}^n x_{ij} \leq a_i \quad (i = 1 \dots m) \quad (3.2)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad (j = 1 \dots n) \quad (3.3)$$

$$x_{ij} \geq 0 \quad (3.4)$$

Em que,

c_{ij} : custo unitário de transporte da origem i para o destino j ;

a_i : quantidade disponível na origem i ;

b_j : quantidade requerida no destino j ;

x_{ij} : quantidade a ser transportada da origem i para o destino j . São as incógnitas do problema.

3.1.2. Problemas de mistura

Problemas deste tipo constituem em combinar materiais obtidos na natureza (ou resto de outros já combinados anteriormente) para gerar novos materiais ou produtos com características convenientes. Estão entre os primeiros problemas de otimização linear implementados com sucesso na prática. A seguir, buscou-se apresentar um exemplo numérico deste tipo de problema, que ilustra bem uma aplicação de modelos lineares.

Exemplo:

Uma agroindústria deve produzir um tipo de ração para determinado animal. Essa ração é produzida pela mistura de farinhas de três ingredientes básicos: osso, soja e resto de peixe. Cada um desses três ingredientes contém diferentes quantidades de dois nutrientes necessários a uma dieta nutricional balanceada: proteína e cálcio. O nutricionista especifica as necessidades mínimas desses nutrientes em 1kg de ração. Cada ingrediente é adquirido no mercado com um certo custo unitário (\$/kg). Na Tabela 3.1, os dados do problema são apresentados. Por exemplo, a farinha de osso é constituída de 20% de proteína e 60% de cálcio; a ração deve ser composta de pelo menos 30% de proteína e 50% de cálcio; 1kg de farinha de osso custa \$0,56 (os ingredientes podem ser constituídos por outros elementos, mas que não são importantes para o problema em questão).

Tabela 3.1: Dados do exemplo do Problema de Mistura (Arenales et al, 2007)

Nutrientes	Ingredientes			Ração
	Osso	Soja	Peixe	
Proteína	0,2	0,5	0,4	0,3
Cálcio	0,6	0,4	0,4	0,5
Custos (\$/kg)	0,56	0,81	0,46	

Deve-se determinar em que quantidades os ingredientes devem ser misturados de modo a produzir uma ração que satisfaça às restrições nutricionais com o mínimo custo. Defina a variável de decisão x_j como a quantidade (em kg) do ingrediente j que deve ser utilizada em uma unidade (1kg) da ração, j = osso, soja, peixe. Com isso, o custo da mistura é dado por:

$$f(x_{\text{osso}}, x_{\text{soja}}, x_{\text{peixe}}) = 0,56x_{\text{osso}} + 0,81x_{\text{soja}} + 0,46x_{\text{peixe}} \quad (3.5)$$

e as restrições de composição são dadas por:

$$0,2x_{\text{osso}} + 0,5x_{\text{soja}} + 0,4x_{\text{peixe}} \geq 0,3 \quad (3.6)$$

$$0,6x_{\text{osso}} + 0,4x_{\text{soja}} + 0,4x_{\text{peixe}} \geq 0,5. \quad (3.7)$$

Deve-se observar a pequena diferença no enunciado, que resulta em restrições de desigualdade (em vez de igualdade), pois foram estabelecidos percentuais mínimos dos componentes na mistura.

Temos também que a soma dos ingredientes resulta em uma unidade da mistura, ou seja,

$$x_{osso} + x_{soja} + x_{peixe} = 1 \quad (3.8)$$

e que esses ingredientes podem ser utilizados ou não, isto é,

$$x_{osso} \geq 0, x_{soja} \geq 0, x_{peixe} \geq 0. \quad (3.9)$$

O modelo matemático completo fica, então:

$$\text{Minimizar } f(x_{osso}, x_{soja}, x_{peixe}) = 0,56x_{osso} + 0,81x_{soja} + 0,46 x_{peixe} \quad (3.10)$$

$$0,2x_{osso} + 0,5x_{soja} + 0,4x_{peixe} \geq 0,3 \quad (3.11)$$

$$0,6x_{osso} + 0,4x_{soja} + 0,4x_{peixe} \geq 0,5 \quad (3.12)$$

$$x_{osso} + x_{soja} + x_{peixe} = 1 \quad (3.13)$$

$$x_{osso} \geq 0, x_{soja} \geq 0, x_{peixe} \geq 0. \quad (3.14)$$

A melhor solução, chamada solução ótima deste modelo, é dada por: $x_{osso}^* = 0,5$, $x_{soja}^* = 0$ e $x_{peixe}^* = 0,5$, o que significa que a ração deve ser constituída de 50% de farinha de osso e 50% de farinha de peixe. Pode-se determinar o custo dessa solução e procurar por soluções alternativas.

O exemplo apresentado teve como objetivo apenas ilustrar uma situação. Os dados utilizados não correspondem a valores reais, que podem ser obtidos em tabelas nutricionais ou por análise laboratorial. Este exemplo, ainda, se baseia na referência usada como base para este estudo (Arenales et al, 2007). Além do problema de mistura, diversos outros exemplos numéricos foram estudados, tais como problemas de planejamento da produção, de programação de projetos, de gestão financeira, de meio ambiente, entre outros.

3.2. Aplicações de Modelos Matemáticos em Projeto de Redes

Conforme já citado anteriormente, problemas de projeto de redes constituem uma classe importante dentre os problemas de otimização, aparecendo em inúmeras situações e desafios práticos. Desenvolver modelos e algoritmos para estes problemas é, portanto, de grande importância. Como exemplo de aplicações para estes projetos, podemos citar a engenharia elétrica, planejamentos de transporte, telecomunicações e ciência da computação.

Em engenharia elétrica, as aplicações de projetos de redes aparecem em diferentes situações desde projetos de componentes microeletrônicos até projeto de sistemas de transmissão e distribuição de energia elétrica.

Com relação ao planejamento de transporte, pode-se dizer que nas últimas décadas o transporte de pessoas e mercadorias tem crescido bastante e é esperado que esse crescimento seja mantido num mundo cada vez mais interconectado. Um bom projeto de rede pode ser usado para melhorar a eficiência do transporte, a satisfação do usuário e o lucro das empresas, enquanto atende outras metas, tais como o uso racional de combustível e outros recursos naturais.

Na área de ciência da computação, as aplicações de projeto de rede são bastante relacionadas com as citadas anteriormente. Além disso, podem-se citar aplicações que lidam com roteamento de internet e comparação aproximada de strings para unificar dados textuais vindos de diferentes fontes de dados.

Finalmente, podemos citar a área de telecomunicações, na qual a modernização da mesma é uma marca diferenciada em nossos tempos. Transmissões de TV ao vivo ao longo de todo o mundo, chamadas de telefone intercontinentais ou comunicações através da internet estão tão difundidos e com acesso imediato que é fácil esquecer que estes são mantidos por uma vasta rede de cabos, fibras ópticas e satélites. O projeto e a operação dessas redes dependem fortemente, também, dos métodos de projeto de redes.

A seguir são apresentados os modelos relacionados a projeto de redes explorados, sendo um deles inédito. Os resultados e eficiência computacional dos mesmos são comparados através de testes com pacotes comerciais e também mostrados na sequência.

4. Modelos Estudados

4.1. Introdução

Os modelos estudados têm como ideia básica estabelecer uma rede de links (estradas, fibras ópticas, linhas elétricas, entre outros) que permita o fluxo de produtos (pessoas, pacotes de dados, e eletricidade, por exemplo) com o intuito de satisfazer algumas das características de procura. Este trabalho teve particular interesse em problemas de projeto de redes com custo fixo, onde, para utilizar uma ligação, deverá ser pago um custo fixo representando, por exemplo, o custo de construção de uma estrada, ou a instalação de uma linha elétrica, etc.

Um grande número de aplicações práticas pode ser representado por modelos de projeto de redes com custo fixo. Uma área importante é o *problema de projeto de redes de serviço*, que aparece, por exemplo, em companhias aéreas e empresas de caminhões. A ideia é maximizar o lucro definindo rotas e horários, levando em consideração algumas limitações de recursos. Por exemplo, as companhias aéreas devem determinar as vias abrangidas e a frequência de vôos considerando, ainda, a disponibilidade de aeronaves e tripulação. Da mesma forma, empresas de entrega rápida (por exemplo, UPS, Fedex) devem estabelecer as rotas e aeronaves para atribuí-los e decidir sobre o fluxo dos pacotes.

Em sistemas de energia, projetos de redes com custo fixo são usados para planejar a transmissão de energia de centrais de produção para os centros de consumo e, ainda, a distribuição de energia no interior desses centros. Neste último caso, os modelos de rede também podem ser usados em um contexto operacional para obter a configuração que minimiza os custos de energia diária.

Finalmente, várias aplicações podem ser encontradas também em telecomunicações. Como exemplo, temos o projeto de uma rede de acesso local com uma ou duas tecnologias, o projeto de layout de um terminal e a interligação das redes existentes.

4.2. Análise de Modelos Matemáticos em Projeto de Redes

4.2.1. Problema de Projeto de Redes Não-Capacitadas

Primeiramente, foram analisados problemas de projeto de redes não-capacitadas. Nestes problemas, não há limite para o fluxo que pode circular através dos links selecionados. Muitas aplicações na vida real podem ser bem representadas por esses modelos.

O início do problema de projeto de redes não-capacitadas se fez com pares origem-destino. Neste caso, a cada mercadoria $k = 1, \dots, |K|$ está associada uma demanda d_k , um nó de origem $O(k)$ e um nó de destino $D(k)$. Estes problemas são definidos em uma rede subjacente representada por um grafo $G=(N,A)$, onde N é o conjunto de nós (cidades a serem conectadas, computadores a serem interligados, etc) e A é o conjunto de possíveis ligações entre os nós.

Uma formulação para esta classe de problemas pode ser verificada a seguir.

$$\text{Minimizar } \sum_{(i,j) \in A} \left(\sum_{k \in K} c_{ij}^k x_{ij}^k + f_{ij} y_{ij} \right) \quad (4.1)$$

sujeito a

$$\sum_{j|(i,j) \in A} x_{ij}^k - \sum_{j|(j,i) \in A} x_{ji}^k = \begin{cases} d_k, i = O(k), \forall k \in K, \\ 0, i \notin \{O(k), D(k)\}, \forall k \in K, \\ -d_k, i = D(k), \forall k \in K, \end{cases} \quad (4.2)$$

$$x_{ij}^k \leq d_k y_{ij}, \quad \forall (i,j) \in A, \quad \forall k \in K, \quad (4.3)$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, \quad \forall k \in K, \quad (4.4)$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A. \quad (4.5)$$

Variáveis x_{ij} representam o fluxo real do produto k indo do nó i ao nó j . Variáveis y_{ij} são variáveis binárias que estão associadas à construção da ligação $(i,j) \in A$: $y_{ij} = 1$ se (i,j) pertence à solução final, e $y_{ij} = 0$, caso contrário. A função objetivo (4.1) é a soma ponderada das variáveis relacionadas ao fluxo e à construção das ligações. Em alguns casos, os custos fixos representam plenamente o custo real da rede, ou seja, não há nenhum custo associado com o volume (fluxo) de uma mercadoria em um link. No entanto, a fim de modelar adequadamente alguns problemas, deve-se incluir, por vezes, custos de volume baseado. Na função, c_{ij}^k é o custo linear associado ao fluxo de uma unidade do produto k através do link (i,j) e f_{ij} é o custo fixo associado à seleção da ligação (i,j) na solução final.

O conjunto de restrições (4.2) assegura que o fluxo do produto k deixa o nó de origem $O(k)$ e chega ao destino nó $D(k)$, e são, ainda, responsáveis pela conservação do fluxo. As restrições (4.3) limitam o fluxo de produtos aos arcos selecionados; as restrições

(4.4) garantem que as variáveis x_{ij}^k são não-negativas, e as restrições (4.5) definem que as variáveis y_{ij} são binárias.

4.2.2. Problema de Projeto de Redes Capacitadas

Consideram-se agora Problemas de Projeto de Redes Capacitadas. Do mesmo modo como foi citado anteriormente, o fluxo de produtos em um link só é permitido se um custo fixo é pago. No entanto, a quantidade deste fluxo é limitada a uma certa capacidade. Exemplos de aplicações deste tipo de problema ocorrem com frequência em projeto de rede.

A diferença na formulação matemática deste problema para o anterior é que uma única restrição do tipo (4.6) mostrada a seguir é gerada para cada arco, em vez de k restrições por arco como anteriormente na restrição (4.3). Além disso, nesta restrição (4.6), d_k foi alterado para C , que representa o fluxo máximo que pode circular no arco.

$$\sum_{k \in K} x_{ij}^k \leq C y_{ij}, \quad \forall (i, j) \in A. \quad (4.6)$$

4.2.3. Um novo modelo de projeto de redes com restrições de HOP

Um novo modelo, que passa agora a levar em conta a questão da confiabilidade da rede, foi proposto neste trabalho. Este fato foi tratado através de restrições de saltos ou hops (Costa et al. 2009), que limitam o número de conexões ou links entre uma central e cada nó ou consumidor servido. No caso de alguns tipos de rede (de telecomunicações, por exemplo), um caminho muito grande na transmissão de dados interfere diretamente na confiabilidade destas redes, já que pode resultar em perda de informações. Por este motivo, tornou-se interessante analisar os resultados com um modelo que possui um número máximo de caminhos que o produto k pode percorrer, número este que será denotado por H . Este modelo é apresentado a seguir:

$$\text{Minimizar } \sum_{(i,j) \in A} \left(\sum_{k \in K} \sum_{h=1}^H c_{ijh}^k x_{ijh}^k + f_{ij} y_{ij} \right) \quad (4.7)$$

sujeito a

$$\sum_{j(O(k),j) \in A} x_{O(k)j1}^k = d_k, \quad \forall k \in K, \quad (4.8)$$

$$\sum_{j(i,j) \in A} x_{ij(h+1)}^k - \sum_{j(j,i) \in A} x_{jih}^k = 0, \quad \forall k \in K, i \notin \{O(k), D(k)\}, h = 1 \dots H, \quad (4.9)$$

$$\sum_{j(j,D(k)) \in A} \sum_{h=1}^H x_{jD(k)h}^k = d_k, \quad \forall k \in K, \quad (4.10)$$

$$\sum_{k \in K} \sum_{h=1}^H x_{ijh}^k \leq Cy_{ij}, \quad \forall (i,j) \in A, \quad (4.11)$$

$$x_{ijh}^k \geq 0, \quad \forall (i,j) \in A, h = 1 \dots H, \quad \forall k \in K, \quad (4.12)$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A. \quad (4.13)$$

Como antes, a função objetivo (4.7) é a soma ponderada das variáveis relacionadas ao fluxo (x_{ijh}^k) e à construção das ligações (y_{ij}). Nota-se que agora é considerado também o parâmetro h , relacionado ao número do arco em que o produto está passando, levando em conta o caminho desde o nó origem até o nó destino. A restrição (4.8) garante que a soma de tudo que sai do nó origem seja igual à demanda do produto em questão e, ainda, que cada variável x saia com $h = 1$. A restrição (4.9) garante a conservação do fluxo em todos os nós que não são origem nem destino e, ainda, que cada variável x entre no nó com h e saia com $h+1$. A restrição (4.10) garante que a soma de tudo que chega no nó destino seja igual à demanda do produto em questão. A restrição de capacidade (4.11) garante que a soma de tudo que passa em um arco não exceda sua capacidade e as restrições das variáveis (4.12) e (4.13) garantem que todas as variáveis x e y sejam inteiras e binárias, respectivamente.

Só depois da análise dos resultados deste modelo através de programação em linguagem C (que será detalhada posteriormente), foi concluído que havia a necessidade da adição de novas restrições, as quais são apresentadas a seguir:

$$\sum_{j(i,j) \in A} x_{ij1}^k = 0, \quad \forall k \in K, i \notin \{O(k)\}, \quad (4.14)$$

$$\sum_{j(j,O(k)) \in A} \sum_{h=1}^H x_{jO(k)h}^k = 0, \quad \forall k \in K, \quad (4.15)$$

$$\sum_{j(D(k),j) \in A} \sum_{h=1}^H x_{D(k)jh}^k = 0, \quad \forall k \in K. \quad (4.16)$$

A restrição (4.14) garante que, se o nó não for o nó origem, o fluxo daquele produto não pode começar por ele. A restrição (4.15) garante que nada entre no nó origem de cada produto e a restrição (4.16) garante que nada saia do nó destino de cada produto.

Com mais estas três restrições, o modelo de projeto de redes com restrições de HOP fica completo.

4.2.4. Modelo de Projeto de Redes com Restrições “soft” de HOP

O modelo elaborado anteriormente de projeto de redes com restrições de HOP foi modificado de modo que o custo da rede passa a não ser mais o fator a ser minimizado e começa a fazer parte das restrições (continuando, assim, de forma ativa no modelo). Isso acontece com a ajuda dos resultados obtidos nos testes para redes com HOP (resultados estes que serão representados por \hat{z}). Sobre estes, é colocada uma tolerância e o valor da soma ponderada das variáveis relacionadas ao fluxo e à construção das ligações, ou seja, o custo total da rede, deverá ser menor que o produto de \hat{z} por um coeficiente de tolerância α (um pouco maior que 1), simulando um determinado custo que não se deseja ultrapassar.

Além disso, o antigo parâmetro H passa a ser designado por h forte - h_{fo} . Essa mudança ocorre porque um novo parâmetro irá aparecer, denominado h fraco - h_{fr} , que representa o número de arcos do fluxo no qual se deseja evitar ultrapassar. Por fim, uma última variável é criada para a conclusão do novo modelo: p_k . Esta variável recebe o valor 1 se o produto k necessita um percurso com número de arcos maior que h_{fr} e 0 caso contrário. É nesta variável, ainda, que se baseia a função objetivo. Com esta variável e mais os dois parâmetros citados anteriormente (h_{fo} e h_{fr}), torna-se possível a ideia básica deste modelo, que é de minimizar o número de produtos que utilizem em seu fluxo mais arcos do que um valor desejado (h_{fr}), respeitando um limite de custo conhecido. Neste modelo, portanto, a questão da confiabilidade é tratada de uma forma um pouco diferente da anterior.

O modelo completo fica da seguinte maneira:

$$\text{Minimizar } \sum_{k \in K} p_k, \quad (4.17)$$

sujeito a

$$\sum_{j|(O(k),j) \in A} x_{O(k),j}^k = d_k, \quad \forall k \in K, \quad (4.18)$$

$$\sum_{j|(i,j) \in A} x_{ij(h+1)}^k - \sum_{j|(j,i) \in A} x_{jih}^k = 0, \quad \forall k \in K, i \notin \{O(k), D(k)\}, h = 1 \dots H, \quad (4.19)$$

$$\sum_{j|(j,D(k)) \in A} \sum_{h=1}^H x_{jD(k)h}^k = d_k, \quad \forall k \in K, \quad (4.20)$$

$$\sum_{j|(i,j) \in A} x_{ij1}^k = 0, \quad \forall k \in K, i \notin \{O(k)\}, \quad (4.21)$$

$$\sum_{j|(j,O(k)) \in A} \sum_{h=1}^H x_{jO(k)h}^k = 0, \quad \forall k \in K, \quad (4.22)$$

$$\sum_{j|(D(k),j) \in A} \sum_{h=1}^H x_{D(k)jh}^k = 0, \quad \forall k \in K, \quad (4.23)$$

$$\sum_{k \in K} \sum_{h=1}^H x_{ijh}^k \leq C y_{ij}, \quad \forall (i,j) \in A, \quad (4.24)$$

$$\sum_{(i,j) \in A} \left(\sum_{k \in K} \sum_{h=1}^H c_{ijh}^k x_{ijh}^k + f_{ij} y_{ij} \right) \leq \bar{z}^* \alpha, \quad (4.25)$$

$$\sum_{(i,j) \in A} x_{ijh_{fr}}^k \leq d_k p_k, \quad \forall k \in K, \quad (4.26)$$

$$x_{ijh}^k \geq 0, \quad \forall (i,j) \in A, \quad h=1 \dots H, \quad \forall k \in K, \quad (4.27)$$

$$y_{ij} \in \{0,1\}, \quad \forall (i,j) \in A, \quad (4.28)$$

$$p_k \in \{0,1\}, \quad \forall k \in K. \quad (4.29)$$

A função objetivo (4.17) representa agora a soma da variável p de cada produto, que é o que se deseja minimizar. As restrições (4.18)-(4.24) e (4.27)-(4.28) são as mesmas do modelo de projeto de redes com restrições de HOP. A restrição (4.25) garante que a soma ponderada das variáveis relacionadas ao fluxo e à construção das ligações seja menor que o produto do coeficiente de tolerância α pelo valor ótimo referente aos testes para redes com HOP, com o h_{fo} considerado. Na restrição (4.26) é onde encaixamos a variável p_k , assegurando-se que, para cada produto k , a soma de todo o fluxo seja menor que o produto da demanda do produto por p_k , considerando-se $h = h_{fr}$. Por fim, a restrição (4.29) define que as variáveis p_k são binárias.

5. Implementações Básicas

Nesta seção, será mostrado como os modelos apresentados anteriormente foram implementados e testados. Três formas de resolução foram analisadas. A primeira delas usa o software comercial Excel e seu *solver* embutido. A segunda utiliza programação em C para gerar um arquivo .lp. A terceira, que se baseia nos três últimos modelos estudados, utiliza programação em linguagem C e, ainda, o pacote comercial IBM ILOG CPLEX a fim de se obter resultados otimizados para diversas instâncias de dados.

5.1. Usando o *solver* do Microsoft Excel

Este tópico será dividido em quatro etapas: começando extraindo os dados relativos à rede (número de produtos e arcos, capacidade, custo fixo e variável, etc) para um exemplo proposto e colocando-os em tabelas no Excel. Em seguida, é escrito a fórmula da função objetivo e colocamos o conjunto de restrições. Depois, o *solver* é utilizado a fim de se alcançar a solução ótima e, por fim, o resultado é obtido e analisado.

5.1.1. Inserção dos Dados no Excel

Para ilustrar a resolução do modelo via o *solver* do Excel, será utilizado um exemplo simples de Redes Capacitadas (seção 4.4.2) com os seguintes dados:

- 2 produtos;
- 5 nós;
- 7 arcos;

Os dados do grafo e das demandas de cada produto são apresentados nas tabelas 5.1 e 5.2, respectivamente.

Tabela 5.1: Dados do Grafo para o exemplo implementado no Excel

<i>Nó Origem</i>	<i>Nó Destino</i>	<i>Custo Variável</i>	<i>Custo Fixo</i>	<i>Capacidade</i>
1	2	10	100	15
1	3	15	100	15
1	4	8	100	15
2	5	25	20	30
3	4	7	40	10
3	5	9	80	20
4	5	10	80	10

Tabela 5.2: Dados dos produtos para o exemplo implementado no Excel

	<i>Nó Origem</i>	<i>Nó Destino</i>	<i>Demanda</i>
<i>produto 1</i>	1	5	25
<i>produto 2</i>	2	5	15

Colocando estes dados em tabelas no Excel e reservando um espaço para as variáveis a serem otimizadas, tem-se o começo da planilha, que poder ser visualizada na Figura 5.1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	DADOS													
2														
3	<i>Nó Origem</i>	<i>Nó Destino</i>	<i>Custo Variável</i>	<i>Custo Fixo</i>	<i>Capacidade</i>									
4	1	2	10	100	15		VARIÁVEIS							
5	1	3	15	100	15									
6	1	4	8	100	15		<i>Arco -></i>	12	13	14	25	34	35	45
7	2	5	25	20	30		<i>x (prod 1)</i>							
8	3	4	7	40	10		<i>x (prod 2)</i>							
9	3	5	9	80	20		<i>y</i>							
10	4	5	10	80	10									
11														
12		<i>Nó Origem</i>	<i>Nó Destino</i>	<i>Demanda</i>										
13	<i>produto 1</i>	1	5	25										
14	<i>produto 2</i>	2	5	15										

Figura 5.1: Inserção dos dados na planilha para o exemplo implementado no Excel

Na figura, x é o fluxo real que vai de um nó a outro e y é a variável binária associada à construção de casa arco. O espaço em branco na tabela que contém estas variáveis será preenchido posteriormente com os valores correspondentes à solução ótima, após a utilização do *solver*.

5.1.2. Inserção da Função Objetivo e do Conjunto de Restrições

Nesta etapa, será adicionado à planilha a função objetivo e o conjunto de restrições.

-Função Objetivo:

Esta é formada pela soma da multiplicação de cada fluxo x (tanto do produto 1 quanto do produto 2) pelo custo variável relativo a cada arco e da multiplicação da variável binária y pelo custo fixo, também relativo a cada arco, fato que é visto na equação (4.1) na seção 4.2.1. Esta soma, a qual se deseja minimizar, é representada de acordo com as linhas e colunas da planilha pela fórmula:

$$Fobj = C4 \cdot H7 + C5 \cdot I7 + C6 \cdot J7 + C7 \cdot K7 + C8 \cdot L7 + C9 \cdot M7 + C10 \cdot N7 + C4 \cdot H8 + C5 \cdot I8 + C6 \cdot J8 + C7 \cdot K8 + C8 \cdot L8 + C9 \cdot M8 + C10 \cdot N8 + D4 \cdot H9 + D5 \cdot I9 + D6 \cdot J9 + D7 \cdot K9 + D8 \cdot L9 + D9 \cdot M9 + D10 \cdot N9$$

-Conjunto de Restrições:

Este é formado pelas restrições (4.2) e (4.6) das seções 4.2.1 e 4.2.2, respectivamente. O conjunto de restrições (4.2) assegura que o fluxo de cada produto deixa o nó de origem $O(k)$ e chega ao destino nó $D(k)$, e são, ainda, responsáveis pela conservação do fluxo. A restrição (4.6) garante que o fluxo total que circula no arco não ultrapasse a capacidade do mesmo. As demais restrições são feitas no momento que se utiliza o *solver*, como será explicado mais adiante. A Figura 5.2 mostra as restrições inseridas na planilha.

17	RESTRIÇÕES															
18																
19																
20	Produto 1						Produto 2									
21		Sai	Entra	Soma	Restrição			Sai	Entra	Soma	Restrição					
22	Nó 1	0	0	0	=	25	Nó 1	0	0	0	=	0				
23	Nó 2	0	0	0	=	0	Nó 2	0	0	0	=	15				
24	Nó 3	0	0	0	=	0	Nó 3	0	0	0	=	0				
25	Nó 4	0	0	0	=	0	Nó 4	0	0	0	=	0				
26	Nó 5	0	0	0	=	-25	Nó 5	0	0	0	=	-15				
27													Arco	Fluxo	Restrição	
													12	0	<=	0
													13	0	<=	0
													14	0	<=	0
													25	0	<=	0
													34	0	<=	0
													35	0	<=	0
													45	0	<=	0

Figura 5.2: Inserção do conjunto de restrições na planilha para o exemplo implementado no Excel

As duas primeiras partes são referentes à restrição (4.2) e já possuem as fórmulas inseridas nas respectivas células. Por exemplo, o nó 1 é o nó origem do produto 1 e, por este motivo, o somatório de tudo que sai dele (variáveis x) deverá ser igual à demanda deste produto, que é 25 no exemplo utilizado. Já o nó 2, não é origem e nem destino do produto 1, o que resulta na conservação do fluxo: a diferença entre o que sai e o que entra neste nó é 0. Este mesmo raciocínio é utilizado nessas duas primeiras tabelas de restrições, sempre de acordo com a restrição (4.2).

A última parte é referente à restrição (4.6), sendo esta formada de um lado da desigualdade pela soma dos fluxos dos produtos 1 e 2 em cada arco e do outro pela multiplicação da variável binária y pela capacidade de cada arco. Esta restrição mostra a principal característica de Redes Capacitadas, nas quais a quantidade do fluxo em cada arco é limitada a uma certa capacidade.

Após a inserção destas restrições, tem-se a planilha completa e pronta para a utilização do *solver*, apresentada na Figura 5.3.

1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
2	DADOS																	
3	Nó Origem	Nó Destino	Custo Variável	Custo Fixo	Capacidade													
4	1	2	10	100	15													
5	1	3	15	100	15													
6	1	4	8	100	15													
7	2	5	25	20	30													
8	3	4	7	40	10													
9	3	5	9	80	20													
10	4	5	10	80	10													
11																		
12		Nó Origem	Nó Destino	Demanda														
13	produto 1	1	5	25														
14	produto 2	2	5	15														
15																		
16																		
17	RESTRIÇÕES																	
18																		
19																		
20	Produto 1						Produto 2											
21		Sai	Entra	Soma	Restrição			Sai	Entra	Soma	Restrição		Arco	Fluxo	Restrição			
22	Nó 1	0	0	0	=	25		Nó 1	0	0	0	=	12	0	<=	0		
23	Nó 2	0	0	0	=	0		Nó 2	0	0	0	=	13	0	<=	0		
24	Nó 3	0	0	0	=	0		Nó 3	0	0	0	=	14	0	<=	0		
25	Nó 4	0	0	0	=	0		Nó 4	0	0	0	=	25	0	<=	0		
26	Nó 5	0	0	0	=	-25		Nó 5	0	0	0	=	34	0	<=	0		
27													45	0	<=	0		

Figura 5.3: Planilha completa e pronta para a utilização do *sol/ver* no exemplo implementado no Excel

5.1.3. Utilização do Solver

O *add-in solver* do *Microsoft Excel* é uma ferramenta para resolução de modelos matemáticos lineares. Com ele, é possível obter uma solução ótima para o problema em questão. O mesmo se encontra no menu “ferramentas” → “solver” e sua tela principal é mostrada na Figura 5.4.

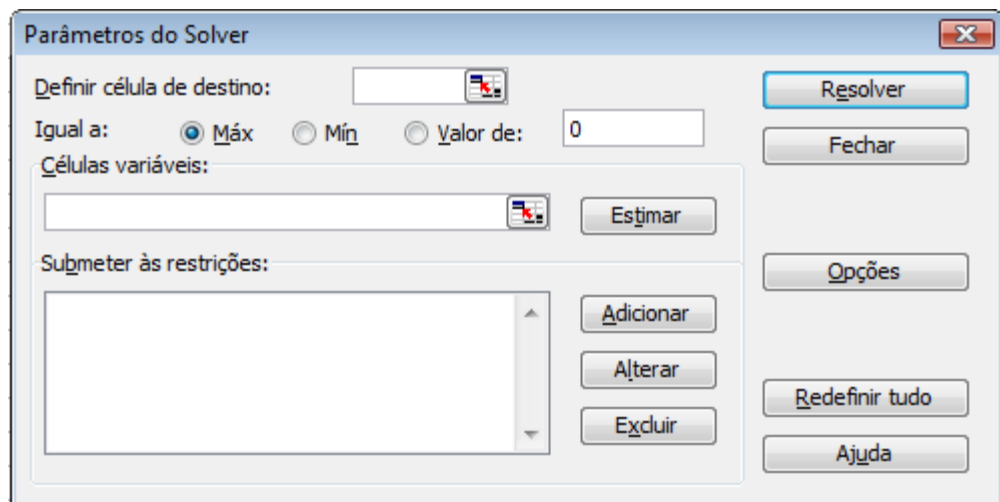


Figura 5.4: Tela Principal do Solver do Excel

Na célula de destino, coloca-se o que se deseja otimizar, ou seja, a função objetivo, que está na célula K13.

No item “igual a”, coloca-se o tipo de otimização, que neste caso é “min”, já que deseja-se minimizar a função objetivo.

Na parte de “Células Variáveis”, coloca-se as variáveis x relativas a cada produto e, ainda, a variável binária y . Assim, essas células variam de H7 até N9.

No caso das restrições, basta seguir o que já foi relatado nas tabelas anteriormente, ou seja, as células de D22 a D26 devem ser iguais às células de F22 às células F26, as células de K22 a K26 devem ser iguais às células de M22 às células M26 e as células de P20 a P26 devem ser menores ou iguais às células de R20 a R26.

Além disso, devemos adicionar as restrições das variáveis x e y , correspondentes às equações (4.4) e (4.5) da seção 4.2.1. Para a primeira, selecionamos as células de H7 a N8 e as fazemos maiores ou iguais a 0. Para a variável y , selecionamos as células de H9 a N9 e as definimos como binárias. A tela do *solver* ficará, então, de acordo com a Figura 5.5.

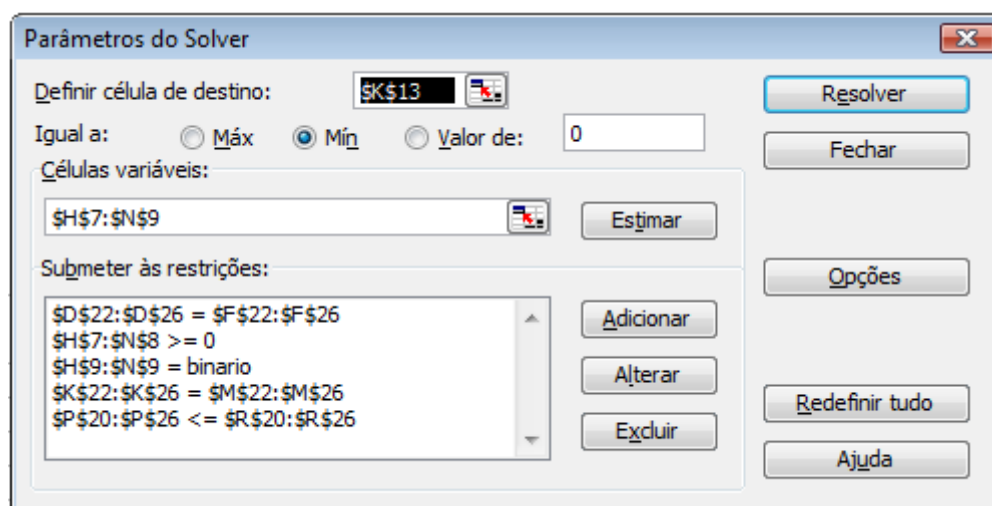


Figura 5.5: Adição dos Parâmetros no Solver para o exemplo implementado no Excel

Finalmente, seleciona-se a opção “Resolver” e a solução ótima é obtida com os valores referentes a esta otimização aparecendo nas células que estavam em branco (células das variáveis x e y , bem como a célula da função objetivo).

5.1.4. Resultado

O *solver* preenche, portanto, as células dedicadas às variáveis com a solução ótima e encontra, assim, um resultado para a função objetivo. O resultado final do problema está apresentado na Figura 5.6.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	DADOS																	
2																		
3	Nó Origem	Nó Destino	Custo Variável	Custo Fixo	Capacidade													
4	1	2	10	100	15													
5	1	3	15	100	15													
6	1	4	8	100	15													
7	2	5	25	20	30													
8	3	4	7	40	10													
9	3	5	9	80	20													
10	4	5	10	80	10													
11																		
12		Nó Origem	Nó Destino	Demanda														
13	produto 1	1	5	25														
14	produto 2	2	5	15														
15																		
16																		
17																		
18	RESTRICÕES																	
19																		
20	Produto 1						Produto 2						Arco					
21		Sai	Entra	Soma	Restrição			Sai	Entra	Soma	Restrição			Fluxo	Restrição			
22	Nó 1	25	0	25	=	25		Nó 1	0	0	0	=	0	12	0	<=	0	
23	Nó 2	0	0	0	=	0		Nó 2	15	0	15	=	15	13	15	<=	15	
24	Nó 3	15	15	0	=	0		Nó 3	0	0	0	=	0	14	10	<=	15	
25	Nó 4	10	10	0	=	0		Nó 4	0	0	0	=	0	25	15	<=	30	
26	Nó 5	0	25	-25	=	-25		Nó 5	0	15	-15	=	-15	34	0	<=	0	
27														35	15	<=	20	
														45	10	<=	10	

Figura 5.6: Modelo Completo Resolvido relacionado ao exemplo implementado no Excel

Analisando-se o resultado, vê-se que o valor da função objetivo é um valor factível, e que o fluxo confere com as regras de nó origem e destino bem como de demandas. A Figura 5.7 ilustra este fluxo relativo à solução ótima do problema.

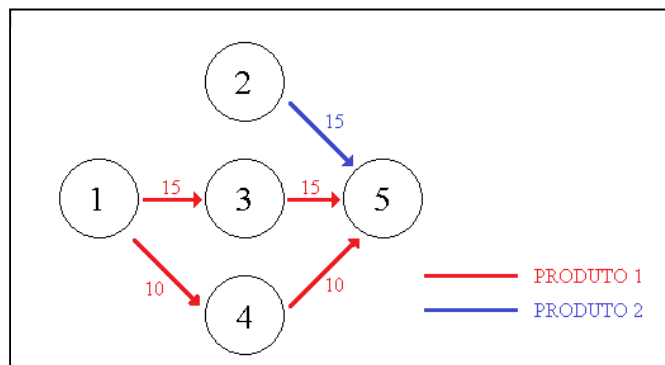


Figura 5.7: Fluxo correspondente à solução ótima do exemplo implementado

5.2. Gerando um arquivo .lp através de um programa em C

Após a montagem do modelo no Excel, o objetivo era utilizar uma linguagem de programação para gerar um arquivo de texto que pudesse ser lido por um *solver* do tipo IBM ILOG CPLEX. Foi escolhida a linguagem C.

Assim, foi elaborado um programa que lia um arquivo de texto contendo os dados relativos à rede, armazenava estes em vetores e matrizes dentro do programa e, após manipular estes dados, gerava um outro arquivo de texto numa linguagem que pudesse ser entendida pelo *solver* a ser utilizado mais tarde. O formato de saída escolhido foi o formato .lp, que basicamente escreve literalmente a função objetivo e todas as restrições do problema. Com o intuito de ilustrar este fato, o mesmo exemplo anterior foi utilizado. A seguir é apresentado o modo como o arquivo que contém os dados é disposto e, depois, o arquivo gerado pelo programa elaborado em C.

5.2.1. Arquivos de Dados

O arquivo de dados consiste em apenas números, os quais representam as diversas características da rede. Ele está organizado da seguinte maneira:

- uma linha contendo o número de nós, arcos e produtos, nesta sequência;
- uma matriz contendo os dados de cada arco, em que cada linha apresenta a seguinte sequência: nó origem, nó destino, custo variável, capacidade, custo fixo e mais dois dados que não terão importância no programa;
- uma matriz contendo os dados dos produtos, em que cada linha apresenta a seguinte sequência: nó origem, nó destino e demanda.

Esta formatação segue o padrão das instâncias descritas por Ghamlouche et al. (2004). Como foi utilizado o mesmo exemplo do tópico anterior, os dados já são conhecidos e a Figura 5.8 mostra como eles aparecem no arquivo de texto a ser lido, de acordo com a organização citada acima.

5	7	2		
1	2	10	15	100
1	3	15	15	100
1	4	8	15	100
2	5	25	30	20
3	4	7	10	40
3	5	9	20	80
4	5	10	10	80
1	5	25		
2	5	15		

Figura 5.8: Dados utilizados como exemplo para a geração de um arquivo .lp seguindo o padrão das instâncias descritas por Ghamlouche et al. (2004).

5.2.2. Arquivo Gerado

O arquivo gerado, o qual deverá ser lido posteriormente por um *solver*, representa a formulação apresentada no item 4.2.1, trocando-se a restrição (4.3) pela restrição (4.6), ou seja, trata-se de um modelo de Redes Capacitadas. Nesta representação, os símbolos de somatórios foram abertos e, portanto, a expressão algébrica fica um tanto grande. As expressões contidas neste arquivo são apresentadas a seguir:

Minimize

$$10x_{1_2_1} + 15x_{1_3_1} + 8x_{1_4_1} + 25x_{2_5_1} + 7x_{3_4_1} + 9x_{3_5_1} + 10x_{4_5_1} + \\ 10x_{1_2_2} + 15x_{1_3_2} + 8x_{1_4_2} + 25x_{2_5_2} + 7x_{3_4_2} + 9x_{3_5_2} + 10x_{4_5_2} + \\ 100y_{1_2} + 100y_{1_3} + 100y_{1_4} + 20y_{2_5} + 40y_{3_4} + 80y_{3_5} + 80y_{4_5}$$

subject to:

$$(x_{1_2_1} + x_{1_3_1} + x_{1_4_1}) - (0) = 25 \\ (x_{2_5_1}) - (x_{1_2_1}) = 0 \\ (x_{3_4_1} + x_{3_5_1}) - (x_{1_3_1}) = 0 \\ (x_{4_5_1}) - (x_{1_4_1} + x_{3_4_1}) = 0 \\ (0) - (x_{2_5_1} + x_{3_5_1} + x_{4_5_1}) = -25$$

$$(x_{1_2_2} + x_{1_3_2} + x_{1_4_2}) - (0) = 0 \\ (x_{2_5_2}) - (x_{1_2_2}) = 15 \\ (x_{3_4_2} + x_{3_5_2}) - (x_{1_3_2}) = 0 \\ (x_{4_5_2}) - (x_{1_4_2} + x_{3_4_2}) = 0 \\ (0) - (x_{2_5_2} + x_{3_5_2} + x_{4_5_2}) = -15$$

$$x_{1_2_1} + x_{1_2_2} \leq 15y_{1_2} \\ x_{1_3_1} + x_{1_3_2} \leq 15y_{1_3} \\ x_{1_4_1} + x_{1_4_2} \leq 15y_{1_4} \\ x_{2_5_1} + x_{2_5_2} \leq 30y_{2_5} \\ x_{3_4_1} + x_{3_4_2} \leq 10y_{3_4} \\ x_{3_5_1} + x_{3_5_2} \leq 20y_{3_5} \\ x_{4_5_1} + x_{4_5_2} \leq 10y_{4_5}$$

$$x_{1_2_1} \geq 0 \\ x_{1_3_1} \geq 0 \\ x_{1_4_1} \geq 0 \\ x_{2_5_1} \geq 0 \\ x_{3_4_1} \geq 0 \\ x_{3_5_1} \geq 0 \\ x_{4_5_1} \geq 0 \\ x_{1_2_2} \geq 0 \\ x_{1_3_2} \geq 0 \\ x_{1_4_2} \geq 0 \\ x_{2_5_2} \geq 0 \\ x_{3_4_2} \geq 0 \\ x_{3_5_2} \geq 0 \\ x_{4_5_2} \geq 0$$

$$y_{1_2} = \{0,1\} \\ y_{1_3} = \{0,1\} \\ y_{1_4} = \{0,1\} \\ y_{2_5} = \{0,1\} \\ y_{3_4} = \{0,1\} \\ y_{3_5} = \{0,1\} \\ y_{4_5} = \{0,1\}$$

A nomenclatura $x_{4_5_1}$, por exemplo, representa o fluxo x que vai do nó 4 ao nó 5 relativo ao produto 1. Já a nomenclatura y_{1_2} , por exemplo, representa a variável binária y relativa ao arco que vai do nó 1 ao nó 2.

Com isto em mente e analisando-se o resultado obtido no arquivo gerado, pode-se entender mais claramente como funciona o modelo de Redes Capacitadas.

Esta etapa do trabalho foi a primeira em contato com a linguagem C e muito ajudou a entender mais profundamente os modelos estudados. Serviu, também, de preparação para uma parte de programação mais difícil, utilizando comandos complexos e algumas lógicas bem complicadas, assunto abordado no próximo tópico.

5.3. Usando a tecnologia CONCERT do IBM ILOG CPLEX

Após a implementação de um programa que gerasse um arquivo .lp, o próximo passo foi usar a tecnologia CONCERT do IBM ILOG CPLEX para, ao invés de gerar um arquivo com o modelo a ser otimizado, gerar um arquivo pronto, já com os resultados da otimização para várias instâncias de dados.

Para isto, fez-se necessário o aprendizado de alguns recursos da tecnologia citada acima. A seguir, buscou-se apresentar alguns exemplos, com o objetivo de mostrar de que maneira se utiliza esta tecnologia em linguagem C. Os trechos foram retirados do programa elaborado referente ao modelo de restrições com HOP (seção 4.2.3). Os comentários são precedidos de duas barras (//).

5.3.1. Declaração de Variáveis

```
// Declaração da variável x
typedef IloArray<IloNumVarArray> VarMatrix;
typedef IloArray<VarMatrix> VarMatrix2;
typedef IloArray<VarMatrix2> VarMatrix3;

VarMatrix3 x;
x = VarMatrix3(env,a);
for (i=0;i<a;i++)
{
    x[i] = VarMatrix2(env,a);
    for (j=0;j<a;j++)
    {
        x[i][j] = VarMatrix(env,c);
        for (k=0;k<c;k++)
        {
            x[i][j][k] = IloNumVarArray(env,h);
            for (t=0;t<h;t++)
            {
                stringstream ss;
                ss << "x" << i+1 << "_" << j+1 << "_" << k+1 << "_" << t+1 ;
                string name=ss.str();
                x[i][j][k][t] = IloNumVar(env,0,IloInfinity, name.c_str());
            }
        }
    }
}
// Obs: a = número de nós; c = número de produtos
```

5.3.2. Função objetivo

```
lloExpr objFn(env);

for (i=0;i<b;i++)
{
    objFn += mat1[i][4]*y[mat1[i][0]][mat1[i][1]];
}
for (i=0;i<b;i++)
{
    for (k=0;k<c;k++)
    {
        for (t=0;t<h;t++)
        {
            objFn += mat1[i][2]*x[mat1[i][0]][mat1[i][1]][k][t];
        }
    }
}

model.add(lloMinimize(env, objFn));
// Obs: b = número de arcos; mat1 = matriz com informações da rede
```

5.3.3. Restrições

```
// Restrição da capacidade
for (i=0;i<b;i++)
{
    lloExpr lhs(env);
    for (k=0;k<c;k++)
    {
        for (t=0;t<h;t++)
        {
            lhs += x[mat1[i][0]][mat1[i][1]][k][t];
        }
    }
    model.add(lhs - mat1[i][3]*y[mat1[i][0]][mat1[i][1]] <= 0);
}
```

5.3.4. Extração e Resolução do Modelo

```
cplex.extract(model);
cplex.solve();
```

5.3.5. Obtenção do Valor da Função Objetivo

```
double fo = cplex.getValue(objFn);
```

As implementações nesta etapa foram feitas para três modelos diferentes e exigiram uma grande parte do tempo destinado à produção deste trabalho, devido à extensão e complexidade dos programas em C. A apresentação dos resultados foi registrada e pode ser conferida no próximo capítulo.

Como os programas implementados possuem a mesma lógica, apenas um será apresentado nesta monografia e este se encontra no Apêndice.

6. Resultados e Análise

Esta etapa dedicou-se à exposição e análise dos resultados referentes aos programas elaborados usando a tecnologia CONCERT do IBM ILOG CPLEX. As primeiras implementações foram descritas usando o exemplo das redes capacitadas composto pelas restrições (4.1)-(4.2), (4.4)-(4.6). Depois, utilizou-se o modelo de redes com HOP, formado pelas restrições (4.7)-(4.16) e, por fim, os novos testes de restrições “soft” de HOP formados pelas restrições (4.17)-(4.29). As duas primeiras partes foram apresentadas numa mesma tabela para efeito de comparação de resultados. Os testes foram realizados tomando como referência uma série de instâncias de dados, que têm o formato descrito na seção 5.2.1.

6.1. Redes Capacitadas e Modelos com HOP

Com relação ao problema de redes capacitadas, foi obtido um valor ótimo da função objetivo deste modelo para cada instância. Posteriormente, com o programa elaborado para o modelo de redes com HOP, foram analisados os mesmos resultados, só que para diversos valores de H .

A Tabela 6.1 mostra os resultados comentados anteriormente para cinco instâncias de dados que apresentam 10 nós, 35 arcos e 10 produtos, mas com valores diferentes para custos (fixos e variáveis) e capacidades. Os valores marcados com um traço significam que não foi possível obter uma solução factível.

Tabela 6.1: Resultados para Redes Capacitadas e Modelos com HOP para as 5 instâncias de dados.

<i>Instância</i>	<i>Valor Ótimo</i>	<i>H = 1</i>	<i>H = 2</i>	<i>H = 3</i>	<i>Desvio</i>	<i>H = 5</i>	<i>Desvio</i>	<i>H = 10</i>	<i>Desvio</i>
1	74079	-	-	78219	5,59%	74079	0	74079	0
2	92403	-	-	97424	5,43%	92403	0	92403	0
3	115304	-	-	120075	4,14%	115304	0	115304	0
4	84908	-	-	86242	1,57%	84908	0	84908	0
5	113036	-	-	115251	1,96%	113036	0	113036	0

Pode-se notar na tabela 6.1, que quando limitamos o número máximo de arcos que o produto pode passar em $H = 3$, o valor ótimo sofre um desvio de até 5,59%. É interessante notar que não é possível encontrar um caminho factível se limitamos o número máximo de arcos em 1 ou 2. Ainda, para $H = 5$, é possível verificar que o valor da função objetivo se iguala ao valor ótimo e, portanto, para qualquer H maior o mesmo ocorrerá.

O programa foi simulado para várias outras instâncias. A Tabela 6.2 mostra os resultados obtidos para cinco outras instâncias, que possuem o mesmo número de nós e de produtos que as instâncias anteriores, mas com um número de arcos igual a 60.

Tabela 6.2 - Resultados para Redes Capacitadas e Modelos com HOP para 5 outras instâncias de dados

<i>Instância</i>	<i>Valor Ótimo</i>	<i>H = 1</i>	<i>H = 2</i>	<i>Desvio</i>	<i>H = 3</i>	<i>Desvio</i>	<i>H = 5</i>	<i>Desvio</i>
6	31730	-	33521	5,64%	31730	0	31730	0
7	48920	-	51348	4,96%	48920	0	48920	0
8	63767	-	69681	9,27%	63767	0	63767	0
9	33740	-	34604	2,56%	33740	0	33740	0
10	53790	-	56548	5,13%	53790	0	53790	0

Verifica-se na tabela 6.2 que é possível estabelecer soluções factíveis para $H = 2$. Isso acontece pois nestas instâncias o número de possibilidade de construção de arcos é maior para uma mesma quantidade de nós. Assim, mesmo se limitamos o número de arcos que o produto pode passar em um valor baixo, é mais provável de se encontrar um caminho com os dados destas instâncias do que nas anteriores.

Os resultados foram analisados para mais de 70 instâncias de dados e foram bem parecidos com os já apresentados. Para várias instâncias, foi possível a obtenção de soluções factíveis para $H = 2$, assim como mostrado anteriormente. No entanto, o valor do desvio com relação ao valor ótimo é bem alto, chegando até a 30%, o que não é interessante em um projeto de redes real. Apesar desse fato, para $H = 3$, os valores do desvio dificilmente ultrapassam 1%, o que é um fato excelente na questão de custo e confiabilidade das redes. Para $H = 5$ ou mais, praticamente todos os resultados correspondem ao valor ótimo (desvio = 0).

O tempo de obtenção de cada solução, ou seja, tempo que o CPLEX demora para encontrar o valor ótimo, também foi analisado. Para as instâncias que não apresentam um número tão grande de produtos e arcos, o tempo é por volta de 1 segundo. Porém, com um número maior de produtos e arcos, o tempo chega a ser dado em minutos. Algumas instâncias chegaram a necessitar de mais de uma hora.

É interessante citar, também, que o programa elaborado fornece, além dos custos totais de cada rede simulada (apresentados nas tabelas anteriores), os valores das variáveis x e y , o que torna possível a análise de como o fluxo de cada produto se comporta na rede. Isso serviu, ainda, de recurso para correção de alguns erros presentes nos códigos durante a elaboração dos programas.

6.2 - Modelos com Restrições “soft” de HOP

Com relação ao modelo de projeto de redes com restrições “soft” de HOP (seção 4.2.4), os testes foram realizados fixado-se $h_{fo} = 5$ (que representa o H no modelo que teve os resultados apresentados no último tópico) e variando-se h_{fr} (número de arcos do fluxo do produto que se deseja evitar ultrapassar) entre 2, 3 e 4. Isso significa que, para $h_{fr} = 3$, por exemplo, o programa elaborado garante que nenhum produto passará por mais de 5 arcos e que o menor número possível de produtos passarão por mais de 3 arcos. Além disso, o valor do coeficiente de tolerância α foi variado entre 1,05, 1,10 e 1,20.

Neste caso, os resultados obtidos para cada uma das variações de h_{fr} e α citadas acima correspondem ao custo total da rede e ao número de produtos cujo fluxo ultrapassa h_{fr} (valor do somatório de p_k).

Estes resultados são apresentados na Tabela 6.3, tomando como referência as mesmas cinco instâncias iniciais do tópico anterior.

Tabela 6.3: Resultados para o Modelo com Restrições “soft” de HOP para as 5 primeiras instâncias

<i>Inst.</i>	Valores								
1	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 2$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 0$
	77782,95	77782,95	76782,00	81486,91	79649,00	78876,00	86891,00	88689,00	87592,00
2	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 2$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 0$
	97023,15	96966,00	94323,00	101643,3	101643,3	101643,3	106340,0	110883,6	109162,0
3	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 2$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 1$	$\Sigma p = 0$	$\Sigma p = 0$
	121015,0	120075,0	115304,0	125435,0	126501,0	125972,0	136782,0	138073,0	138364,8
4	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 2$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 2$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 2$	$\Sigma p = 0$	$\Sigma p = 0$
	89153,40	89153,40	89153,40	92433,00	91289,00	93398,80	101889,6	101889,6	98669,00
5	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 2$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 2$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 2$	$\Sigma p = 0$	$\Sigma p = 0$
	118687,8	118687,8	116841,0	123901,0	124339,6	124339,6	135643,2	129606,5	135643,2

Nota-se na Tabela 6.3 que fixando um determinado valor para o coeficiente de tolerância α , o valor do somatório de p_k sempre decresce ou permanece em 0 ao aumentarmos o valor de h_{fr} . Além disso, verifica-se que não são muitos produtos que necessitam mais que h_{fr} arcos em seu percurso, já que as instâncias acima possuem número de produto igual a 10 e o valor máximo do somatório de p_k é 2. Este fato mostra que o modelo em questão apresenta um bom resultado na questão da confiabilidade.

Um fato interessante a se notar é que em determinadas situações, em que o h_{fr} passa de 3 para 4 e o somatório de p_k se mantém em 0, o valor do custo pode aumentar ou diminuir. Este fato ocorre pois a única variável presente na função objetivo é p_k . A soma ponderada relativa aos custos está presente somente nas restrições. Com isso, torna-se interessante, num trabalho futuro, realizar uma análise do problema fazendo a intersecção de vários objetivos, como as restrições de HOP, o lucro, entre outros.

Os resultados obtidos para outras instâncias foram similares aos anteriores, como mostra a tabela 6.4, para instâncias com mesmo número de nós e arcos (10 e 35, respectivamente), mas com número de produtos igual a 25.

Tabela 6.4: Resultados para o Modelo com Restrições “soft” de HOP para 5 outras instâncias

<i>Inst.</i>	<i>Valores</i>								
11	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$
	243850,9	243850,9	243850,9	255462,9	255462,9	255462,9	273405,0	264878,0	278686,8
12	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$
	338575,6	338575,7	338575,7	354698,3	354698,3	354698,3	386943,6	386943,6	386943,6
13	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 6$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$	$\Sigma p = 5$	$\Sigma p = 0$	$\Sigma p = 0$
	438580,0	440478,2	440398,0	458558,0	461453,3	461453,3	503403,6	494623,0	483897,0
14	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 8$	$\Sigma p = 3$	$\Sigma p = 1$	$\Sigma p = 8$	$\Sigma p = 3$	$\Sigma p = 1$	$\Sigma p = 8$	$\Sigma p = 3$	$\Sigma p = 1$
	332258,8	332258,8	332258,9	348080,7	348080,7	343759,0	358437,0	341543,0	379724,4
15	$\alpha = 1,05$			$\alpha = 1,10$			$\alpha = 1,20$		
	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$	$h_{fr} = 2$	$h_{fr} = 3$	$h_{fr} = 4$
	$\Sigma p = 8$	$\Sigma p = 3$	$\Sigma p = 1$	$\Sigma p = 8$	$\Sigma p = 3$	$\Sigma p = 1$	$\Sigma p = 8$	$\Sigma p = 3$	$\Sigma p = 1$
	452812,5	452812,5	452812,5	462587,0	470643,0	474375,0	505375,0	469741,0	494625,0

Verifica-se na Tabela 6.4 que, o valor do somatório de p_k aumentou em comparação com a Tabela 6.3, o que era de se esperar já que o número de produtos de cada rede simulada aumentou.

Essas simulações também foram feitas para várias outras instâncias, com a obtenção de resultados similares, o que leva a conclusão de que este último modelo estudado e implementado fornece resultados bem satisfatórios para um projeto de redes, pois, respeitando um limite de custo desejado, ele permite um controle relativo da quantidade de arcos que o produto passa dentro da rede e mostra uma boa resposta com relação à confiabilidade.

7. Considerações Finais

Ao final deste trabalho de conclusão de curso, diversos modelos matemáticos foram estudados e vários implementados na prática. Para isso, fez-se necessário o aprendizado de como analisar esses modelos, de como implementá-los em linguagem de programação C e, ainda, como utilizar os pacotes presentes no mercado, como os disponíveis no Excel e no software IBM ILOG CPLEX.

Com tudo isso, os resultados obtidos para os três modelos simulados apresentam respostas bem satisfatórias de acordo com alguns requisitos que se deseja atender em diversos projetos de redes hoje em dia, como o custo e a confiabilidade.

O primeiro modelo implementado, que trata de Redes Capacitadas, se aplica quando o único interesse na rede é a minimização do custo. Com relação às redes em que a questão da confiabilidade é levada em conta, os modelos com restrições de HOP e com restrições “soft” de HOP apresentam resultados bem satisfatórios, em que o último permite um controle maior do fluxo do produto na rede, mas não tem como principal objetivo a minimização do custo, como o modelo de restrição com HOP. A escolha de um deles depende, então, de qual é o principal objetivo na rede e os testes com os programas elaborados poderão servir como ajuda nessa escolha.

O interessante deste projeto como um todo foi a mistura das áreas de atuação, desde a matemática com os conceitos dos métodos e modelos, passando pela parte de linguagem de programação juntamente com a utilização de softwares e finalizando com a grande área de aplicação de projetos de redes.

REFERÊNCIAS BIBLIOGRÁFICAS

- Arenales, M.; Armentano, V.; Morabito, R.; Yanasse, H. Pesquisa operacional para cursos de engenharia. Editora Campus, 2007.
- Baldo, T.A. Pesquisa Operacional I (Apostila).
<http://d.yimg.com/kq/groups/20909156/1047948747/name/apostila>.
- Cabral, E. A., Erkut, E., Laporte, G. & Patterson, R. A. (2008). Wide area telecommunication network design: application to the Alberta SuperNet, Journal of the Operational Research Society (Forthcoming).
- Corte-Real, M. & Gouveia, L. (2007). Network flow models for the local access network expansion problem, Computers & Operations Research 34: 1141–1157.
- Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. Computers & Operations Research, 32, 1429-1450.
- Costa, A. M., Cordeau, J.-F. & Laporte, G. (2009). Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints, Networks 53: 141–159.
- Ghamlouche I, Crainic TG, Gendreau M (2004) Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. Annals of Operations Research
- Goldbarg, M. C.; Luna, H. P. L. Otimização combinatória e programação linear. Editora Campus, 2000.
- Gouveia, L. (1999). Using hop-indexed models for constrained spanning and Steiner tree models, in B. Sansò and P. Soriano (eds), Telecommunications Network Planning, Kluwer, Boston : 21–32.
- Mirshawka, V. Aplicações de pesquisa operacional. Editora Nobel, 1981.

Ruffolo, M., Daskin, M. S., Sahakian, A. V. & Berry, R. A. (2007). Design of a large network for radiological image data, *IEEE Transactions on Information Technology in Biomedicine* 11: 25–39.

Tornatore, M., Maier, G. & Pattavina, A. (2007). Wdm network design by ilp models based on flow aggregation, *IEEE/ACM Transactions on Networking* 15: 709–720.

APÊNDICE

Foram elaborados quatro programas diferentes em linguagem C, todos com o mesmo princípio. A seguir é apresentado o último destes programas, relativo à seção 4.2.4 - Modelo de Projeto de Redes com Restrições “*soft*” de HOP.

```
#include <ilcplex/ilocplex.h>
#include <string>
#include <stdio.h>
#include <cstdlib>
#include <iostream>
#include <time.h>

ILOSTLBEGIN

typedef IloArray<IloNumVarArray> VarMatrix;
typedef IloArray<VarMatrix> VarMatrix2;
typedef IloArray<VarMatrix2> VarMatrix3;

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

//DECLARAÇÃO DAS VARIÁVEIS
FILE *input,*output,*input_ot;
int valores[3000], i, j, k, a, b, c, mat1[400][7], mat2[200][3], x,
y, hfo, hfr, t, cont=0, cont2=0, alfa_aux, valor;
char arquivo[10],buff[50];
float valores_otimos[100],result,alfa,resultado[12];

//LEITURA DOS VALORES ÓTIMOS DE RESTRIÇÕES DE HOP PARA H = 5
input_ot = fopen("valores_otimos.txt", "r");
for(i=0;i<72;i++) fscanf(input_ot, "%f", &valores_otimos[i]);
fclose(input_ot);

//CRIAÇÃO DO ARQUIVO DE SAÍDA
output = fopen("output.txt", "w");

if(output==NULL) printf("Erro no output\n");
else {

//FORMAÇÃO DO NOME DOS ARQUIVOS DE ENTRADA
for(x=1;x<2;x++) {

for(y=1;y<10;y++) {

if(((x<=3)&&(y>=7)));
else {

strcpy(arquivo,"r");
if(x<10) strcat(arquivo,"0");
sprintf(buff, "%s%d", "", x);
strcat(arquivo,buff);
strcat(arquivo,".");
sprintf(buff, "%s%d", "", y);
strcat(arquivo,buff);
strcat(arquivo,".dow");
printf("\n\n---%s---\n\n",arquivo);
```

```
//LEITURA DOS ARQUIVOS DE ENTRADA
input = fopen(arquivo, "r");
if(input==NULL) printf("Erro no input\n");
else
{
printf(output,"alfa=1.05\t\t\talfa=1.10\t\t\talfa = 1.15\t\t\talfa
= 1.20\n");
for (i=0;i<4;i++) fprintf(output,"hfr = 2\tthfr = 3\tthfr = 4\t");
fprintf(output,"\n");

for(hfo=5;hfo<6;hfo++){
if((hfo==4)||((hfo==6)||((hfo==7)||((hfo==8)||((hfo==9)));
else {

for (alfa=1.05;alfa<1.21;alfa=alfa+0.05){

for(hfr=2;hfr<5;hfr++){

printf("\n\n\n---H forte = %d---\n\n\n",hfo);
printf("---H fraco = %d---\n\n\n",hfr);

fscanf(input,"%s",buff);

i=0;
while(!feof(input))
//ler todos os valores até que o arquivo chegue ao fim
{

fscanf(input, "%d", &valores[i]);
i++;
}

//nesse ponto todos os valores do arquivo estao gravados em ordem no vetor 'valores'

a=valores[0];
b=valores[1];
c=valores[2];

k=3;

for(i=0;i<b;i++)
{

for(j=0;j<7;j++)
{

mat1[i][j]=valores[k];
k++;
}
}

for(i=0;i<c;i++){

for(j=0;j<3;j++){

mat2[i][j]=valores[k];
k++;
}

}
```

```

}

//Correção dos índices de mat1 e mat2 para começarem do 0
for(i=0; i<b; i++){
mat1[i][0]--;
mat1[i][1]--;
}

for(i=0; i<c; i++){
mat2[i][0]--;
mat2[i][1]--;
}

//MODELO
IloEnv env;
IloModel model(env);
IloCplex cplex(env);

IloNumVarArray p;
VarMatrix y;
VarMatrix3 x;

//DECLARAÇÃO DAS VARIÁVEIS
p = IloNumVarArray(env,c);
for (k=0;k<c;k++){
stringstream ss;
ss << "p" << k+1;
string name=ss.str();
p[k]=IloNumVar(env,0,1,ILOBOOL,name.c_str());
}

y = VarMatrix(env,a);
for (i=0;i<a;i++) {
y[i] = IloNumVarArray(env,a);
for (j=0;j<a;j++) {
stringstream ss;
ss << "y" << i+1 << "_" << j+1;
string name=ss.str();
y[i][j] = IloNumVar(env,0,1,ILOBOOL,name.c_str()); //, name.c_str());
}
}

x = VarMatrix3(env,a);
for (i=0;i<a;i++) {
x[i] = VarMatrix2(env,a);
for (j=0;j<a;j++) {
x[i][j] = VarMatrix(env,c);//, name.c_str());
for (k=0;k<c;k++) {
x[i][j][k] = IloNumVarArray(env,hfo);
for (t=0;t<hfo;t++){
stringstream ss;
ss << "x" << i+1 << "_" << j+1 << "_" << k+1 << "_" << t+1 ;
string name=ss.str();
x[i][j][k][t] = IloNumVar(env,0,IloInfinity, name.c_str());
}
}
}
}

//FUNÇÃO OBJETIVO

```

```

lloExpr objFn(env);

for (k=0;k<c;k++){
objFn += p[k];
}

model.add(lloMinimize(env, objFn));

//RESTRIÇÕES

//Novas Restrições

lloExpr lhs8(env);
for (i=0;i<b;i++) {
lhs8 += mat1[i][4]*y[mat1[i][0]][mat1[i][1]];
}

for (i=0;i<b;i++)
{

for (k=0;k<c;k++) {

for (t=0;t<hfo;t++){
lhs8 += mat1[i][2]*x[mat1[i][0]][mat1[i][1]][k][t];
}
}
}

model.add(lhs8-alfa*valores_otimos[cont]<= 0);

for (k=0;k<c;k++) {
lloExpr lhs9(env);
for (i=0;i<b;i++) {
lhs9 += x[mat1[i][0]][mat1[i][1]][k][hfr];
}
model.add(lhs9-mat2[k][2]*p[k]<=0);
}

// Restrição da capacidade
for (i=0;i<b;i++) {
lloExpr lhs(env);
for (k=0;k<c;k++){
for (t=0;t<hfo;t++) {
lhs += x[mat1[i][0]][mat1[i][1]][k][t];
}
}

model.add(lhs - mat1[i][3]*y[mat1[i][0]][mat1[i][1]] <= 0);
}

//Restrição dos nós
for (k=0;k<c;k++){

for(i=0;i<a;i++){
int no = i;

if (no != mat2[k][0]){
lloExpr lhs5(env);
for(j=0;j<b;j++){
if (mat1[j][0]==no)

```

```

lhs5 += x[i][mat1[j][1]][k][0];
}
model.add(lhs5==0);
}

if(no == mat2[k][1]){
lloExpr lhs6(env);
for(j=0;j<b;j++){
if (mat1[j][0]==no){
for(t=0;t<hfo;t++)lhs6 += x[no][mat1[j][1]][k][t];
}
}

model.add(lhs6==0);
}

if(no == mat2[k][0]){
lloExpr lhs7(env);
for(j=0;j<b;j++){

if (mat1[j][1]==no){

for(t=0;t<hfo;t++) lhs7+=x[mat1[j][0]][no][k][t];
}
}

model.add(lhs7==0);
}

if(no == mat2[k][0]){
lloExpr lhs1(env);
for(j=0;j<b;j++){

if (mat1[j][0]==no) lhs1 += x[i][mat1[j][1]][k][0];
}

model.add(lhs1==mat2[k][2]);
}

if(no == mat2[k][1]){
lloExpr lhs2(env);
for(j=0;j<b;j++){

if (mat1[j][1]==no){

for(t=0;t<hfo;t++) lhs2+=x[mat1[j][0]][i][k][t];
}
}

model.add(lhs2==mat2[k][2]);
}

if((no != mat2[k][0])&&(no != mat2[k][1])){

for(t=0;t<hfo-1;t++){
lloExpr lhs3(env);
lloExpr lhs4(env);
for(j=0;j<b;j++){

if (mat1[j][1]==no) lhs3+=x[mat1[j][0]][i][k][t];

```

```

if (mat1[j][0]==no) lhs4 += x[i][mat1[j][1]][k][t+1];
}

model.add(lhs3-lhs4==0);
}
}

}
}

//EXTRAÇÃO DO MODELO E OBTENÇÃO DOS RESULTADOS
cplex.extract(model);
cplex.exportModel("modeloOrig.lp");
cplex.solve();

double fo = cplex.getValue(objFn);
cout << endl << "Valor da solucao: " << fo << endl << endl;

result = 0;

for (i=0;i<b;i++) {
double valor = cplex.getValue(y[mat1[i][0]][mat1[i][1]]);
result += mat1[i][4]*valor;
}

for(k=0;k<c;k++){

for (i=0;i<b;i++){

for(t=0;t<hfo;t++){

double valor = cplex.getValue(x[mat1[i][0]][mat1[i][1]][k][t]);
result += mat1[i][2]*valor;
}
}
}

printf("Resultado = %.5f",result);
resultado[cont2]=result;
cont2++;

env.end();

fprintf(output,"p = %.0f\t",fo);
} // for hfr
} // for alfa
} // condicao hfo
} // for hfo
cont2=0;
fprintf(output,"\n");

for(i=0;i<12;i++) fprintf(output,"%.2f\t",resultado[i]);

fprintf(output,"\n\n");
cont++;
fclose(input);
} // input
} // condicao para x<=3 e y>=7
} // y

```

```
} // x
fclose(output);
} // output
system("pause>>null");
return 0;

} // int main
```