Universidade Estadual de Londrina
Centro de Tecnologia e Urbanismo
Departamento de Engenharia Elétrica

Um Modelo de Caixeiro Viajante Generalizado para Minimizar o Tempo de Preparação de uma Máquina Tubeteira

Alexandre Junior Fenato

Universidade Estadual de Londrina

2008

UM MODELO DE CAIXEIRO VIAJANTE GENERALIZADO PARA MINIMIZAR O TEMPO DE PREPARAÇÃO DE UMA MÁQUINA TUBETEIRA

Alexandre Junior Fenato

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE INTEGRANTE DO PROGRAMA DE MESTRADO EM ENGENHARIA ELÉTRICA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO TÍTULO DE MESTRE EM ENGENHARIA ELÉTRICA PELA UNIVERSIDADE ESTADUAL DE LONDRINA.

Aprovada por:

_____

Prof. Dr. Robinson Samuel Vieira Hoto (orientador)
Laboratório de Simulação e Otimização de Sistemas (SimuLab)
Universidade Estadual de Londrina (UEL)

_____

Prof. Dr. Horácio Hideki Yanasse
Laboratório Associado de Computação e Matemática Aplicada (LAC)
Instituto Nacional de Pesquisas Espaciais (INPE)

_____

Prof. Dr. José Alexandre França
Universidade Estadual de Londrina (UEL)

_____

Profª. Drª. Silvia Galvão de Souza Cervantes
Universidade Estadual de Londrina (UEL)

LONDRINA, PR – BRASIL
2008

Dedico este trabalho a todos que cooperaram de alguma forma

para que eu pudesse desenvolvê-lo

Agradecimentos

Agradeço primeiramente a Deus, por ter dado-me força e oportunidades, através das quais me encaminhou até este importante momento da minha vida.

Ao meu orientador e amigo Robinson que dedicou, com paciência e atenção, parte do seu tempo para com minha orientação.

Ao professor José Alexandre e à professora Silvia por participarem da banca examinadora e dedicarem-se à análise e avaliação deste trabalho.

Ao professor Horácio pela colaboração e dedicação de seu tempo na análise e avaliação deste trabalho.

Aos meus pais por terem me criado em um ambiente de honestidade e perseverança, caminhos pelos quais eu tenho seguido rigorosamente.

A minha namorada Priscila, que sempre me incentivou nas horas mais difíceis.

A todos os pesquisadores citados na bibliografia deste trabalho, pois são os alicerces desta dissertação.

Este trabalho teve o apoio financeiro e motivador do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

Resumo da Dissertação apresentada ao Programa de Mestrado em Engenharia Elétrica da UEL como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica.

UM MODELO DE CAIXEIRO VIAJANTE GENERALIZADO PARA MINIMIZAR O TEMPO DE PREPARAÇÃO DE UMA MÁQUINA TUBETEIRA

Alexandre Junior Fenato

2008

Orientador: Robinson Samuel Vieira Hoto

Este trabalho consta de um estudo de caso prático, envolvendo o processo de fabricação de tubetes por uma máquina tubeteira, cujo tempo de preparação da máquina ocupa uma parcela considerável do seu tempo produtivo, podendo este ser minimizado pela redução do número de trocas de bolachas realizadas entre a confecção dos tubetes. Dois modelos para a minimização destas trocas são apresentados, um via Caixeiro Viajante e outro, via Caixeiro Viajante Generalizado. Os resultados obtidos por meio de simulações com o *solver Xpress-MP* foram consideravelmente melhores do que os utilizados pela empresa, com reduções de até 37% no número de trocas, indicando que a elaboração de uma heurística deve ser explorada em trabalhos futuros.

Abstract of dissertation presented to Program of Masters in Electrical Engineering of UEL as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

A GENERALIZED TRAVELING SALESMAN MODEL TO MINIMIZE THE SETUP OF A MACHINE TO MANUFACTURE TUBES

Alexandre Junior Fenato

2008

Advisor: Robinson Samuel Vieira Hoto

This work is a practical case study, involving the process of manufacturing tubes for a machine to manufacture tubes where the setup occupies a considerable portion of the productive time that can be minimized by reducing the number of exchanges of wafers between the confection of tubes. Two models for the minimization of these exchanges are presented, a Traveling Salesman and a Generalized Traveling Salesman. The results obtained by simulations with the solver Xpress-MP were satisfactory, with reductions of up to 37% in the number of exchanges, indicating that the development of a heuristic should be explored in future works.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

Introdução

A relevância deste trabalho advém da crescente pressão que as indústrias, em geral, têm recebido para otimizar seus processos. Esta pressão ocorre, entre outras causas, pela maior competitividade imposta pelas transformações que têm afetado a ordem econômica mundial.

O Brasil se enquadra também nesta tendência e tem experimentado profundas mudanças no seu setor produtivo no que tange a modernização de seus processos de produção, melhoria da qualidade de seus produtos e racionalização administrativa.

Variadas técnicas de otimização têm sido desenvolvidas e empregadas em indústrias com o intuito de reduzir ao máximo os custos de produção ou ainda aumentar o rendimento de processos produtivos.

A Tubeteira, máquina responsável pela confecção de tubetes (tubos rígidos e ocos feitos de papel reciclado), possui sua produtividade dependente, dentre outros fatores, do seu tempo de preparação, que corresponde ao tempo em que a máquina encontra-se fora de operação. Assim, uma maneira de melhorar o rendimento da tubeteira consiste em minimizar o seu tempo de preparação.

Durante o processo de confecção de um tubete, várias bolachas (fitas de papel reciclado acondicionadas em bobinas) são posicionadas em uma composição da máquina (estaleiro), de onde as fitas de papel desenroladas são encaminhadas até uma segunda composição (tanque de cola) e, por fim, uma terceira composição (máquina sobrepositora), que é responsável pela confecção e acabamento do tubete.

Nosso trabalho está baseado na Tubeteira de uma empresa, situada na cidade de Londrina, na qual segundo informações da gerência, seu tempo de

preparação representa em média 40% (quarenta por cento) do tempo produtivo da máquina, o que nos motivou fortemente ao estudo do problema.

Dentre os meios de redução do tempo de preparação da Tubeteira, identificamos a possibilidade de reduzir o número de trocas de bolachas durante a confecção dos tubetes, objetivo principal deste trabalho.

Por meio de uma pesquisa literária, nenhum trabalho que tratasse especificamente sobre o assunto foi encontrado, porém, de acordo com as características do problema verificamos a possibilidade de abordá-lo por meio de um Caixeiro Viajante, que consiste em determinar um ciclo hamiltoniano (Apêndice A) de custo mínimo (menor número de trocas de bolachas possível) que passe por todos os vértices (tubetes) de um grafo. A partir desta abordagem, desenvolvemos um modelo matemático para o Problema da Tubeteira a fim de minimizar o número de trocas de bolachas.

Durante nosso trabalho, determinamos, junto à gerência da empresa, a possibilidade de confeccionar os tubetes por meio de configurações que contenham intervalos vazios no sequenciamento de bolachas no estaleiro, fator que altera o espaço de busca do problema e sugere a utilização de uma nova estratégia, o Caixeiro Viajante Generalizado, no qual o caixeiro deve percorrer um conjunto de vértices (tubetes) e voltar para o vértice origem, passando por um vértice de cada *cluster* (conjunto de possíveis configurações das bolachas, conforme variação na posição do intervalo vazio), de modo que o custo total do percurso seja mínimo. Baseado nesta abordagem, desenvolvemos um segundo modelo matemático para o Problema da Tubeteira.

Como contribuições deste trabalho, destacamos a inclusão, no meio científico, de um estudo de caso prático, até então inexistente na literatura. Destacamos também, a apresentação de uma nova aplicação do Caixeiro Viajante Generalizado, o qual possibilitou boas soluções para o nosso problema. Ainda, os resultados numéricos obtidos corresponderam às expectativas, apresentando soluções com reduções consideráveis no número de trocas de bolachas, quando comparadas com as soluções práticas. Estes resultados foram

viabilizados por meio do pacote de otimização *Xpress-MP* (*Applications Of Optimization With Xpress-MP,* 2000), por meio da linguagem de programação *Mosel* (*Xpress-Mosel User Guide*, 2008).

Para a empresa, ainda não fornecemos benefícios concretos, porém, em vista dos resultados obtidos com as simulações, pretendemos desenvolver um sistema computacional, baseado numa heurística a ser desenvolvida, que permita uma programação rápida e eficiente para a confecção dos tubetes.

Destacamos também a possibilidade de utilizarmos nossos modelos em máquinas tubeteiras de empresas concorrentes, bem como em outros processos produtivos semelhantes.

O presente texto está organizado na seguinte estrutura:

Capítulo 1 Introduz o Problema da Tubeteira e suas principais características.

Capítulo 2 Apresenta o modelo matemático desenvolvido para o Problema da Tubeteira sem considerar os intervalos vazios, por meio de uma abordagem do Problema do Caixeiro Viajante.

Capítulo 3 Apresenta o modelo matemático desenvolvido para o Problema da Tubeteira que considera os intervalos vazios, por meio de uma abordagem do Problema do Caixeiro Viajante Generalizado. Introduz a estratégia utilizada para a consideração do mandril nos modelos desenvolvidos.

Capítulo 4 Relata os procedimentos computacionais e os resultados numéricos obtidos.

Capítulo 5 Conclusões e considerações para investigações futuras são apresentadas neste capítulo.

Apêndice A Resumo das nomenclaturas e definições básicas da Teoria dos Grafos.

Apêndice B Apresentação das principais características e funcionalidades do pacote de otimização *Xpress-MP*.

Anexo 1 Apresenta todas as configurações possíveis de bolachas divididas em *clusters* para o exemplo Carteira 1 e Carteira 2.

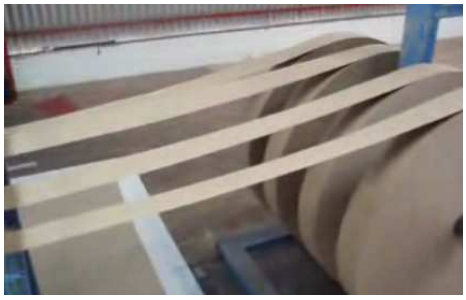
CAPÍTULO 1

O PROBLEMA DA TUBETEIRA

Este capítulo tem como principal finalidade, a apresentação do Problema relacionado com o tempo de preparação da Tubeteira (máquina utilizada para a confecção de tubetes) da empresa Sonoco de Londrina, no qual contextualizamos o problema, introduzindo as características básicas da tubeteira e dos tubetes, por fim, definimos o problema propriamente dito.

1.1 Os tubetes

Os tubetes são fabricados pela sobreposição ordenada de várias camadas de papel reciclado, oriundas das bolachas (fitas de papel reciclado enroladas em formato de grandes bobinas, conforme ilustra a figura 1a), coladas umas sobre as outras com tensões tecnicamente determinadas, gerando assim, um tubo rígido e oco, cuja espessura e rigidez dependem principalmente da quantidade de camadas e do tipo de papel utilizado em sua fabricação. A figura 1b ilustra o aspecto físico de um tubete após sua confecção e corte.



a)

b)

Figura 1 – a) Aspecto físico das bolachas. b) Aspecto físico de um tubete de papel.

Os tubetes são encomendados pelos clientes segundo sua resistência à pressão (rigidez), espessura da parede, diâmetro interno e comprimento. Dentre as suas principais finalidades, destaca-se a utilização por outras empresas como embalagens de proteção e conservação para seus produtos, além de servirem como meio de propaganda para suas marcas, pois são nos tubetes que os rótulos são afixados, por exemplo, as embalagens de fermento químico em pó, conforme mostra a figura 2.

Figura 2 – Exemplo de um tubete de papel com tampa e fundo de metal e rótulo afixado.

Outra utilização comum dos tubetes consiste no transporte de produtos frágeis, na qual atuam como barreira física e rígida para produtos com embalagens mais vulneráveis às pressões mecânicas, impedindo que determinados produtos sejam fisicamente danificados durante o seu deslocamento de um local para outro, como por exemplo, da fábrica para os respectivos pontos de distribuição.

De acordo com o cliente e da respectiva aplicação do tubete, o mesmo será composto por um número variável de camadas de papel, diferidas, entre si, segundo a largura, espessura, gramatura e composição. Este número variável de camadas restringe-se, por sua vez, à capacidade da tubeteira, que corresponde ao número máximo de bolachas que ela suporta durante a confecção de um tubete. De acordo com informações técnicas adquiridas

diretamente dos seus responsáveis operacionais, a tubeteira da Sonoco suporta um número máximo de 34 bolachas para a confecção de um tubete.

Na tabela 1, apresentamos a nomenclatura e respectivas posições de cada camada de papel na constituição de um tubete.

Tabela 1 – Camadas de papel constituintes dos tubetes.

Nome	*Inside Ply* (face interna)	Miolo	*Top Ply*	1ª cobertura	2ª cobertura (face externa)
Posição	1ª	2ª à 31ª	Antepenúltima	Penúltima	Última
Nº máximo de camadas	1	30	1	1	1

De acordo com a tabela 1, no sentido de dentro (face interna) para fora (face externa), o tubete é composto por um tipo de papel que corresponde à posição *inside ply*, por um número de camadas que varia de 1 à 30 referentes ao miolo do tubete, por um tipo de papel para a posição *top ply*, outro para a posição da 1ª cobertura e por fim, um para a 2ª cobertura, conferindo o acabamento externo do tubete. Cabe ressaltar que as posições das camadas de papel de um tubete coincidem com as posições das bolachas na tubeteira.

Segundo informações técnicas, na prática, o número mínimo de bolachas utilizadas na produção de qualquer tubete é igual a três: uma para a posição *inside ply*, outra para a posição *top ply* e por fim uma para a primeira cobertura. Em particular, a Sonoco domina uma tecnologia de produção de tubetes com apenas duas camadas de papel, utilizados por empresas de papel higiênico como eixo de suporte para enrolamento do seu produto.

1.2 A composição da tubeteira

A tubeteira é formada por três grandes composições: o Estaleiro, o Tanque de Cola e a Sobrepositora.

O Estaleiro, ilustrado pela figura 3, é o local onde as bolachas, já ordenadas segundo a composição do tubete, são depositadas para serem desenroladas até o tanque de cola.

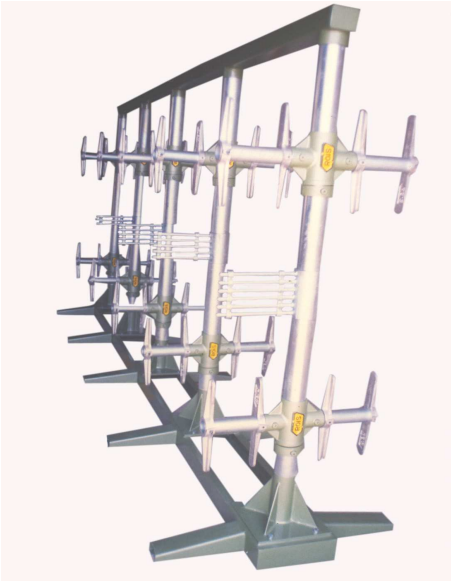


Figura 3 – Estaleiro.

O Tanque de Cola, conforme ilustrado pela figura 4, é composto por um reservatório em sua parte inferior, onde fica depositada a cola que será elevada e derramada sobre as fitas de papel que passam, uma a uma, por entre duas palhetas.

Figura 4 – Tanque de Cola.

A Sobrepositora, ilustrada pela figura 5, recebe as fitas de papel do tanque de cola e as sobrepõe, uma a uma, com uma tensão adequada, constituindo assim, um tubo de comprimento “infinito”, do qual serão cortados os tubetes numa serra circular acoplada à Sobrepositora.

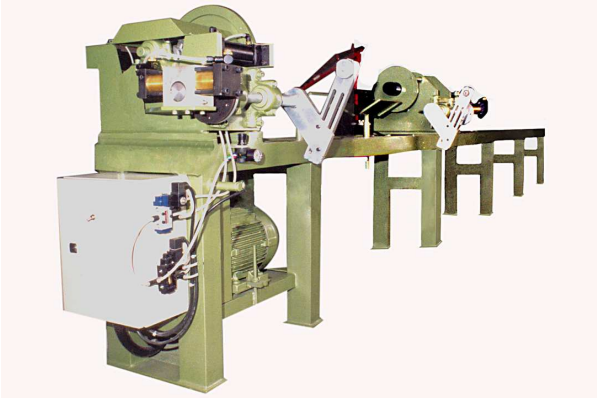


Figura 5 – Máquina Sobrepositora.

Veja na figura 6 uma foto da tubeteira da Sonoco de Londrina, com as suas três composições em destaque, onde o retângulo em verde representa o Estaleiro, o retângulo em amarelo o Tanque de Cola e o retângulo em vermelho a Máquina Sobrepositora. O retângulo em azul representa a Serra Circular acoplada na tubeteira.

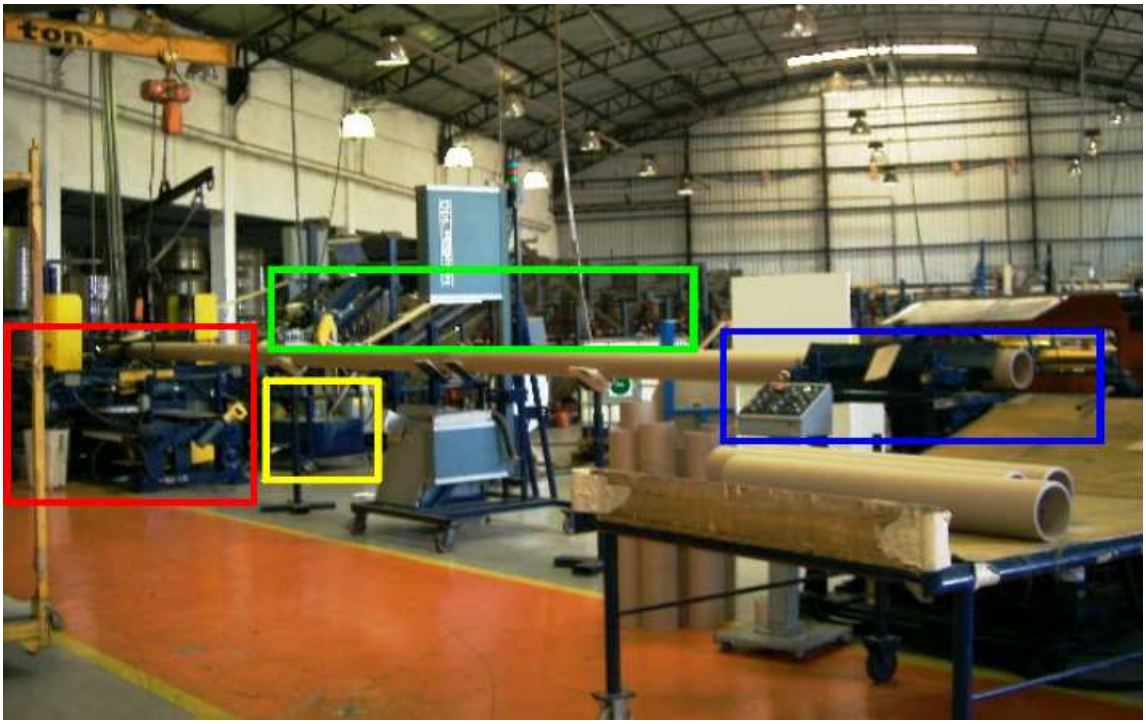


Figura 6 – Foto da tubeteira da Sonoco de Londrina.

1.3 Descrição do Problema

O tempo de preparação da tubeteira representa em média 40% (quarenta por cento) do tempo total de produção, conforme dados técnicos fornecidos pela empresa. Com base em uma pré-análise, definimos duas maneiras possíveis para a diminuição deste tempo ocioso: a otimização do tempo de preparação operacional e do tempo de preparação das bolachas.

O tempo de preparação operacional representa o período de ajuste da máquina e de dispositivos, limpeza da área de trabalho e manutenções preventivas. Sendo assim, este período depende diretamente das condições fornecidas pela empresa, tais como, o número e habilidade dos profissionais responsáveis pela execução deste processo.

O tempo de preparação das bolachas refere-se ao período necessário para a realização das trocas de bolachas entre a produção de um tubete e outro, que dividimos em dois processos de competências distintas:

Processo de competência prática: representa o tempo que os operadores levam para trocar as bolachas da máquina entre a produção de um tipo de tubete e outro. Note que este processo também compete à empresa, de modo que, optamos por não fazer recomendações;

Processo de competência lógica: consiste em encontrar uma seqüência de processamento para a produção de diferentes tipos de tubetes, que minimize o número total de trocas de bolachas. Este é o processo no qual concentramos nossos estudos.

Sendo assim, definimos nosso problema de uma forma mais específica, cujo objetivo consiste em encontrar um método que minimize trocas de bolachas na tubeteira. Veja um exemplo do nosso problema para a confecção de quatro tubetes:

Exemplo 1.1: Suponha que uma tubeteira tenha que produzir os quatro tubetes apresentados na tabela 2, na qual, os tubetes estão representados por letras e as bolachas por números. Considere que as bolachas deverão ser posicionadas no estaleiro, respeitando a ordem, da esquerda para a direita, a fim de compor os tubetes da posição *inside ply* até a posição de 1ª cobertura. Por exemplo, para a confecção do tubete A, as bolachas 200 (*inside ply*), 100, 102, 103 e 2000 (1ª cobertura) deverão estar alocadas, nesta ordem no estaleiro. Desta forma, nosso objetivo consiste em encontrar uma seqüência para execução dos quatro tubetes que minimize o número de trocas de bolachas na tubeteira.

Tabela 2 – Tubetes e respectivas bolachas referentes ao Exemplo 1.1.

		BOLACHAS					
		200	**100**	**101**	**102**	**103**	**2000**
TUBETES	**A**	X	X		X	X	X
	B	X	X	X		X	X
	C	X		X	X		X
	D	X		X	X	X	X

Durante várias visitas à empresa, um dos principais problemas discutidos com os responsáveis operacionais da tubeteira esteve relacionado com a questão das trocas de bolachas, sendo que a grande dificuldade encontrava-se em determinar se a retirada de bolacha poderia ser considerada como troca já que, o custo operacional para este procedimento, na prática, é menor do que o custo para a colocação de uma bolacha na tubeteira.

Não obstante, em ambos os casos, existe perda de tempo e ocupação de mão de obra, desta forma, por recomendação da gerência da empresa, definimos que tanto a colocação quanto a retirada de bolacha são computadas como uma troca e ainda, quando ambas as operações ocorrem para uma mesma posição no estaleiro, apenas uma troca deve ser computada.

Assim, o custo entre a confecção de um tubete e outro foi definido como o total de trocas de bolachas entre eles. Por exemplo, o custo C_{BD} entre o tubete B e D é dois (figura 7). Note que, em razão da definição de uma troca, $C_{BD} = C_{DB}$.

A figura 7 apresenta os custos entre os tubetes de uma seqüência escolhida como solução para o Exemplo 1.1 (tubete D, tubete B, tubete C, tubete A), na qual, cada um dos laços em vermelho refere-se a uma troca e, em azul, ao número de trocas necessário entre a produção de um tubete e outro.

Sendo assim, o número total de trocas necessário para a produção dos quatro tubetes é dado pelo somatório de todas as trocas, equação (1.1).

$$C_{total} = C_{DB} + C_{BC} + C_{CA} = 2 + 4 + 3 = 9 \qquad (1.1)$$

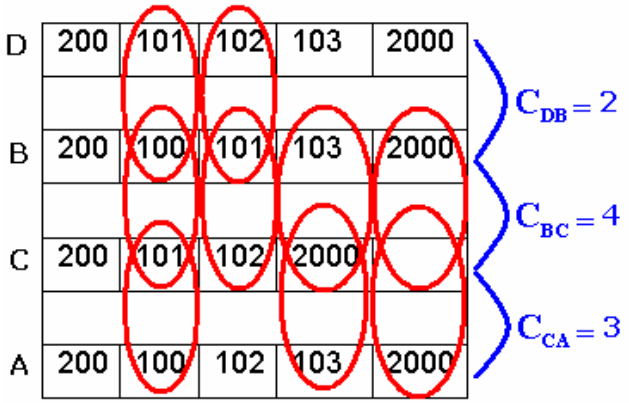


Figura 7 – Custos entre os tubetes da solução viável do Exemplo 1.1.

Em razão da simetria, podemos concluir que a sequência viável apresentada (D, B, C, A) possui um mesmo número de trocas do sentido inverso (A, C, B, D).

De acordo com os dados da tabela 2, podemos montar um grafo (figura 8) em que cada vértice representa um tubete e cada aresta possui um custo associado que equivale ao número de trocas de bolachas entre a confecção dos tubetes interligados por ela.

Neste grafo, incluímos um vértice origem, representado pela letra “O”, com o objetivo de possibilitar a busca de um ciclo hamiltoniano (Apêndice A) e, este por sua vez, representará uma solução para o problema. Para que o custo total do ciclo, referente à solução, não seja alterado, custos nulos devem ser atribuídos a todas as arestas que interligam o vértice origem.

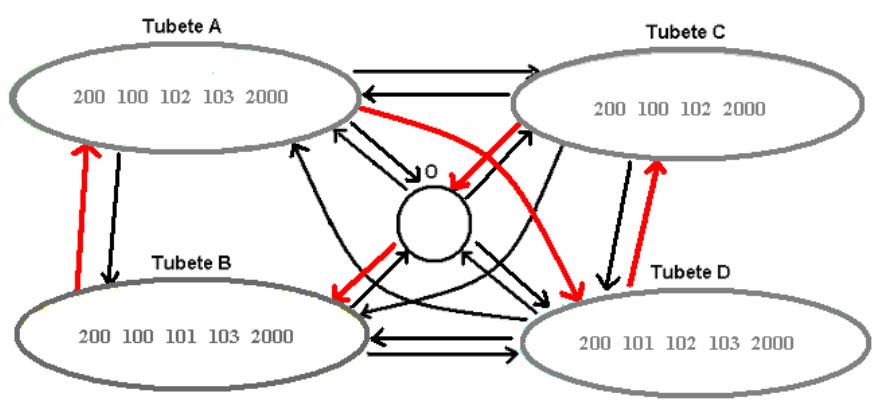


Figura 8 – Grafo para o Exemplo 1.1.

Como o número de vértices do grafo é relativamente pequeno, numa rápida inspeção, encontramos o ciclo hamiltoniano (O, B, A, D, C, O), fornecendo a seqüência (B, A, D, C) de tubetes como solução para o exemplo. A figura 9 ilustra esta solução, onde contabilizamos 4 trocas de bolachas.

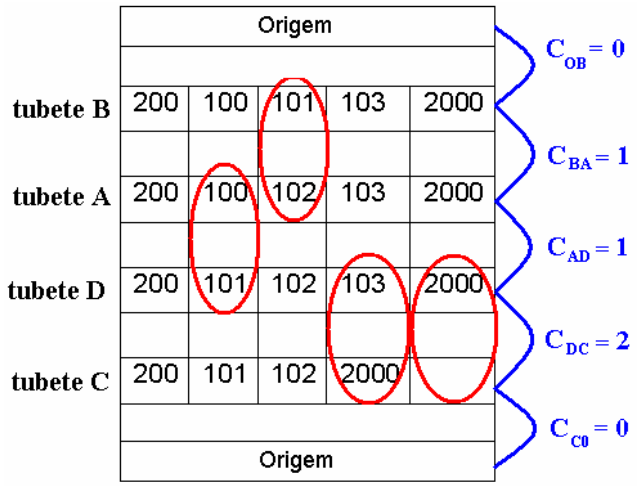


Figura 9 – Solução ótima e respectivos custos para o Exemplo 1.1.

Após uma ampla pesquisa bibliográfica, não encontramos nenhuma referência que tratasse sobre o problema do tempo de preparação de máquinas tubeteiras, porém, de acordo com as particularidades do problema que acabamos de apresentar, decidimos abordá-lo por meio de um Caixeiro Viajante, conhecido na literatura como PCV (Problema do Caixeiro Viajante).

No PCV, o caixeiro deve percorrer um conjunto de cidades e voltar para a sua cidade de origem, passando uma única vez em cada cidade, com o objetivo de minimizar o dispêndio atribuído ao trajeto. No próximo capítulo, apresentamos o modelo matemático para o Problema da Tubeteira, baseado numa abordagem por meio de um PCV.

CAPÍTULO 2

ABORDAGEM DE RESOLUÇÃO VIA PCV

Neste capítulo, apresentamos nossa primeira modelagem matemática para o Problema da Tubeteira, que está fortemente apoiada num Caixeiro Viajante com restrições para eliminação de sub-rotas propostas por Dantzig, Fulkerson e Johnson em 1954 (veja Lawer *et al*. 1990). Destacamos ainda, a possibilidade de permitir, no estaleiro, intervalos vazios (sem bolachas) entre as bolachas que serão utilizadas na confecção de um tubete. Esta possibilidade nos levou a tratar o problema por meio de um Caixeiro Viajante Generalizado.

2.1 O Problema do Caixeiro Viajante

O Problema do Caixeiro Viajante possui diversas aplicações em nosso cotidiano. Comumente, empresas de diversos setores necessitam determinar o melhor roteiro para a entrega de suas mercadorias em pontos de revenda. De fato, é mais econômico um roteiro que passe em cada ponto de revenda uma única vez. A coleta de leite em propriedades rurais é um outro exemplo em que é desejável encontrar um roteiro que saia da fábrica de laticínios, passe em cada ponto de coleta apenas uma vez e retorne ao ponto de partida.

Outra aplicação comum ocorre em processos de manufatura, em que a seqüência de execução de um determinado conjunto de tarefas pode contribuir para diminuir os custos da produção (Lawler *et al.*,1990).

No PCV, um caixeiro deseja visitar N cidades (vértices de um grafo), percorrendo rotas (arestas do grafo), pelas quais, ele pode viajar de uma cidade para outra. Cada rota tem um número associado, que representa o custo necessário para percorrê-la.

Este custo dependerá da aplicação do problema e pode representar a distância euclidiana entre cidades, o dispêndio com a viagem, o desperdício de matéria prima em processos industriais, o tempo de preparação de máquinas, que para o nosso problema é conseqüência do número de trocas de bolachas na tubeteira, entre outros.

Assim, o caixeiro viajante deseja encontrar um percurso fechado que passe por cada uma das N cidades apenas uma vez (ciclo hamiltoniano), cujo custo seja o mínimo dentre os custos de todos os possíveis percursos. Veja alguns métodos de solução e aplicações do PCV em Lawler *et al.*, 1995.

Na próxima seção, detalhamos o equacionamento do modelo matemático que desenvolvemos para o Problema da Tubeteira, cujas etapas de modelagem são baseadas em Rangel (2005), onde é apresentada uma metodologia básica de construção de modelos de otimização linear e linear inteira, trazendo inclusive, discussões sobre ferramentas computacionais úteis nas simulações de problemas reais.

2.2 Modelagem Matemática do Problema da Tubeteira como um PCV

Conforme definição do PCV apresentada, o Problema da Tubeteira pode ser definido por meio de um grafo valorado *G*(*V;A*) (Szwarcfiter, 1998), no qual *V* é o conjunto de vértices que representam os tubetes e *A* é o conjunto de arestas, às quais existe um custo associado que representa o número de trocas de bolachas entre a confecção dos tubetes interligados por ela. A solução deste problema consiste em determinar o ciclo hamiltoniano de custo mínimo.

Para a modelagem do problema vamos, primeiramente, definir a variável de decisão binária, x_{ij} para $i, j = 1,2,...,n$, $i \neq j$, em que, i e j são os índices que representam os tubetes e n correspondente ao número total de tubetes que deverá ser confeccionado. Desta forma, introduzimos a equação (2.1):

$$x_{ij} = \begin{cases} 1, & \textit{se o tubete i é confeccionado imediatamente antes do tubete } j \\ 0, & \textit{caso contrário} \end{cases} \quad (2.1)$$

A seguir, introduzimos o custo c_{ij} da aresta (i, j) do grafo, que representará o número de trocas de bolachas necessário para confeccionar o tubete j, logo após o tubete i, e, como nosso critério para tomada de decisão resume-se em obter o ciclo hamiltoniano de menor custo, a função objetivo pode ser definida conforme apresentado em (2.2).

$$\min \quad z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \quad (2.2)$$

Uma primeira restrição consiste em que cada tubete deve ser fabricado uma única vez. Por exemplo, considere um tubete representado pelo “vértice 3” de um grafo, se $x_{23} = 1$, incluímos o “vértice 3” $(j = 3)$ imediatamente após o “vértice 2” $(i = 2)$ na solução. Como cada vértice só deve aparecer uma única vez, as demais variáveis que representam a chegada no “vértice 3”, x_{i3}, $i = 1,2,...,n,\ i \neq 3$, devem ser nulas. Desta forma, como as variáveis de decisão são binárias, esta situação pode ser representada pela equação (2.3).

$$x_{13} + x_{23} + ... + x_{n3} = 1,\ i \neq 3 \quad (2.3)$$

Obviamente, partindo do “vértice 3”, da mesma forma que chegamos apenas uma vez até ele, devemos partir dele uma única vez, resultando assim, na equação representada por (2.4), que garante que apenas uma aresta saia do “vértice 3”.

$$x_{31} + x_{32} + ... + x_{3n} = 1,\ j \neq 3 \quad (2.4)$$

Este raciocínio se aplica a todos os tubetes, desta forma, definimos os conjuntos de restrições representados pelas equações (2.5) e (2.6), as quais garantem que, na solução, apenas uma aresta chegue e outra saia de cada vértice do grafo, respectivamente.

$$x_{1j} + x_{2j} + ... + x_{nj} = 1, \qquad j = 1,...,n \quad (2.5)$$

$$x_{i1} + x_{i2} + ... + x_{in} = 1, \qquad i = 1,...,n \tag{2.6}$$

Organizando o modelo construído até agora, temos:

$$\min \quad z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{2.7}$$

sujeito a:

$$x_{1j} + x_{2j} + ... + x_{nj} = 1, \qquad j = 1,...,n \tag{2.8}$$

$$x_{i1} + x_{i2} + ... + x_{in} = 1, \qquad i = 1,...,n \tag{2.9}$$

$$x_{ij} = 0 \ \ ou \ \ 1, \qquad i, j = 1,...,n, \ \ i \neq j \tag{2.10}$$

As restrições de saída (2.9) e chegada (2.8) podem gerar soluções que conduzem à sub-rotas (ciclos que não contém todos os vértices do grafo), veja (Goldbarg e Luna, 2000).

De fato, sub-rotas não são desejáveis em nosso problema, pois, correspondem a soluções que não possibilitam a confecção de todos os tubetes, assim, elas devem ser evitadas. Na literatura, existem várias propostas de inclusão de restrições para a eliminação de sub-rotas, por exemplo, restrições que computem as soluções associadas à sub-rotas como respostas inviáveis para o Problema do Caixeiro Viajante.

Duas formas de eliminação para as sub-rotas foram propostas por Dantzig, Fulkerson e Johnson em 1954 (veja Lawer *et al*., 1990). Uma delas aplicada ao nosso problema, baseia-se em limitar o número de variáveis associadas a um subconjunto de tubetes S, que pode receber valor diferente de zero. Este processo pode ser matematicamente representado pela seguinte restrição:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1, \quad \forall \, S \subset \{1, ..., n\} \tag{2.11}$$

Acrescentando as restrições (2.11) para eliminação de sub-rotas no modelo inicial (2.7 – 2.10), temos então o modelo matemático completo para o Problema da Tubeteira abordado como um PCV:

$$\min \quad z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{2.12}$$

sujeito a:

$$x_{1j} + x_{2j} + ... + x_{nj} = 1, \qquad j = 1,...,n \tag{2.13}$$

$$x_{i1} + x_{i2} + ... + x_{in} = 1, \qquad i = 1,...,n \tag{2.14}$$

$$\sum_{i\in S}\sum_{j\in S} x_{ij} \leq |S| - 1, \qquad \forall\, S \subset \{1, ..., n\} \tag{2.15}$$

$$x_{ij} = 0 \;\; ou \;\; 1, \qquad i, j = 1,...,n, \;\; i \neq j \tag{2.16}$$

Observe que o modelo (2.12 – 2.16) possui $2n$ restrições de designação e, $2^n - 2$ restrições para eliminação de sub-rotas, sendo que estas últimas crescem exponencialmente com o aumento do número de vértices do grafo. Como alternativa a este entrave, as inequações de sub-rotas podem ser geradas à medida que vão sendo necessárias, até que um ciclo hamiltoniano seja obtido. Esta técnica foi proposta por Dantzig, Fulkerson e Johnson em 1954 para resolver com sucesso um problema com 49 cidades.

Boa parte do sucesso na resolução de exemplares de grande porte do Problema do Caixeiro Viajante deve-se à integração de diversas metodologias. Para um estudo de algumas técnicas integradas, para a resolução de problemas de porte elevado envolvendo o PCV sugerimos o trabalho de Apllegate *et al.*, 2003.

No contexto de nosso problema, identificamos, junto à gerência da empresa, a possibilidade de permitir a existência de um intervalo vazio definido num sequenciamento definido no estaleiro da tubeteira. Para exemplificar isto, considere mais uma vez o Exemplo 1.1 e, observe na tabela 3, as possíveis

configurações de bolachas de cada tubete, onde o símbolo Ø representa os intervalos vazios.

Tabela 3 – Possíveis configurações de bolachas para a confecção dos tubetes do Exemplo 1.1, considerando os intervalos vazios.

			POSICIONAMENTO DAS BOLACHAS					
			1	2	3	4	5	6
CONFIGURAÇÕES	1	O						
	2	tubete A	200	100	102	103	2000	
	3		200	Ø	100	102	103	2000
	4		200	100	Ø	102	103	2000
	5		200	100	102	Ø	103	2000
	6		200	100	102	103	Ø	2000
	7	tubete B	200	100	101	103	2000	
	8		200	Ø	100	101	103	2000
	9		200	100	Ø	101	103	2000
	10		200	100	101	Ø	103	2000
	11		200	100	101	103	Ø	2000
	12	tubete C	200	101	102	2000		
	13		200	Ø	101	102	2000	
	14		200	101	Ø	102	2000	
	15		200	101	102	Ø	2000	
	16	tubete D	200	101	102	103	2000	
	17		200	Ø	101	102	103	2000
	18		200	101	Ø	102	103	2000
	19		200	101	102	Ø	103	2000
	20		200	101	102	103	Ø	2000

A partir dos dados da tabela 3, montamos o grafo contendo todas as configurações, vértices (2 a 20), e o vértice (1) de origem, conforme ilustrado pela figura 10. De acordo com o grafo, para que se cumpra o objetivo de confeccionar os quatro tubetes, o ciclo hamiltoniano de custo mínimo procurado deverá partir do vértice de origem, visitar pelo menos uma configuração de cada um dos subconjuntos (*clusters*) e voltar ao vértice de origem. Em vermelho, destacamos uma solução viável para o problema.

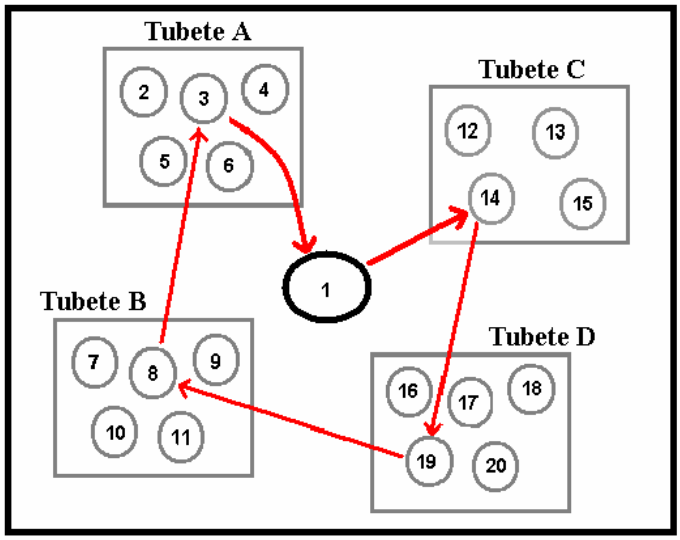


Figura 10 – Grafo que representa o Exemplo 1.1, considerando os intervalos vazios, com destaque para uma solução viável.

O procedimento para a análise de custo do problema, considerando os intervalos vazios, é análogo ao caso que não considera, conforme ilustra a figura 11. Observe que cada laço em vermelho refere-se a uma troca de bolacha.

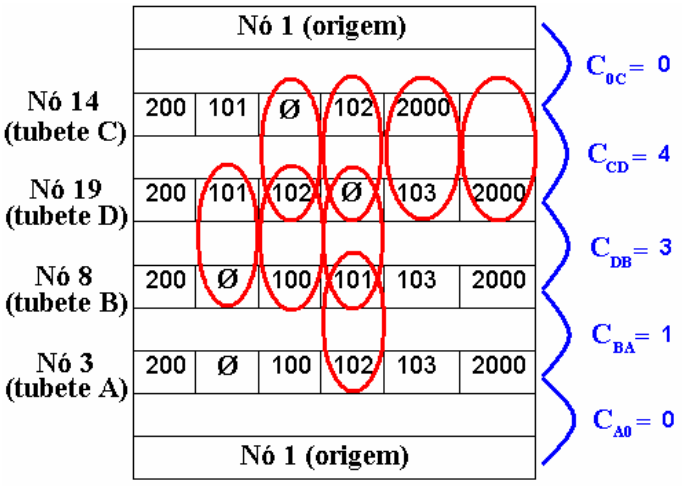


Figura 11 – Custos entre os tubetes da solução viável representada na figura 10.

A seqüência de configurações da solução viável (1, 14, 19, 8, 3, 1) equivale à seqüência de tubetes (C, D, B, A). Neste caso, o número de trocas de bolachas necessário para a confecção dos quatro tubetes é igual a 8.

O número de vértices para o Exemplo 1.1, neste novo contexto, aumentou de cinco para vinte, o que corresponde a um acréscimo considerável no poder combinatório do problema. Em razão disto, é provável que ocorra uma redução no número de trocas de bolachas na solução ótima, já que, as possibilidades, sem considerar os intervalos vazios, ainda fazem parte do novo espaço de busca. Contudo, devido à elevada ordem do novo grafo, a obtenção da solução ótima não é trivial.

Por meio de uma análise do problema representado pelo grafo da figura 10, verificamos que suas características correspondem à versão generalizada do Caixeiro Viajante, conhecida na literatura como Problema do Caixeiro Viajante Generalizado (PCVG), no qual o caixeiro deve visitar um único vértice de cada *cluster* (configurações viáveis de bolachas para um tubete) do grafo.

No próximo capítulo, apresentamos o modelo matemático para o Problema da Tubeteira baseado numa abordagem por meio de um Caixeiro Viajante Generalizado.

CAPÍTULO 3

ABORDAGEM DE RESOLUÇÃO VIA PCVG

Neste capítulo, apresentamos o modelo matemático desenvolvido para o Problema da Tubeteira, considerando intervalos vazios entre bolachas, por meio de um PCVG. Descrevemos também, a necessidade de contemplar, na abordagem de resolução do problema, o diâmetro interno de um tubete.

3.1 O Problema do Caixeiro Viajante Generalizado

O PCVG foi introduzido por Henry-Labordere (1969) e Srivastava *et al.* (1969), no primeiro caso, foi apresentado um estudo de uma aplicação no sequenciamento de arquivos de computador e no segundo os autores modelaram um problema de sequenciamento de visitas-auxílio a clientes por agências governamentais, utilizando programação dinâmica para solucioná-lo.

Laport e Nobert (1983) e Laport *et al.* (1987) apresentaram soluções para algumas aplicações do PCVG, utilizando a técnica *branch and bound*.

Aplicações do PCVG como em programação de processos de máquinas em indústrias, roteamento postal, *layout* de redes, entre outros, podem ser vistos em Noon (1988) e Noon e Bean (1991).

Vários autores estudaram o PCVG e suas aplicações a fim de desenvolverem métodos de redução para um PCV, motivados pela grande quantidade de estudos e heurísticas envolvendo este último.

A primeira técnica de transformação de um PCVG para um PCV foi apresentada por Lien e Ma (1993). Outros métodos podem ser vistos em Noon e Bean (1993) e Dimitrijevic e Saric (1997).

O PCVG é uma extensão simples e prática do PCV, no qual o conjunto N de vértices é dividido em m subconjuntos S_i (*clusters*), sendo que estes formam uma partição de N. O objetivo do problema consiste em encontrar o ciclo de custo mínimo que visita um único vértice de cada *cluster*.

O adjetivo “generalizado”, se justifica pelo fato de que o PCV é um caso específico do PCVG, quando cada *cluster* contém apenas um vértice.

A seguir, detalharemos o equacionamento do modelo matemático desenvolvido para o Problema da Tubeteira, considerando os intervalos vazios, por meio de um Caixeiro Viajante Generalizado.

3.2 Modelagem Matemática do Problema da Tubeteira como um PCVG

O Problema da Tubeteira, que considera os intervalos vazios, pode ser definido por meio de um grafo valorado *G*(*V;A*), no qual *V* é o conjunto de vértices, que representa todas as possíveis configurações de bolachas responsáveis pela confecção de todos os tubetes e *A* é o conjunto de arestas, às quais existe um custo associado que representa o número de trocas de bolachas entre a confecção dos tubetes interligados por ela.

Diferentemente do PCV, o grafo que representa o PCVG é dividido em *clusters*, em que cada *cluster* representa as possíveis configurações de bolachas, conforme variação no posicionamento do intervalo vazio, que permitem a confecção de um mesmo tubete.

Desta forma, sejam $S_1, S_2, ..., S_m$ os *clusters* que possibilitam a confecção dos tubetes $1,2,...,m$, respectivamente. De modo que, $S_1 \cup S_2 ... \cup S_m = N$ e $S_p \cap S_q = \emptyset$ para $p, q = 1,2,...,m$ e $p \neq q$, então, a solução deste problema consiste em determinar o ciclo de custo mínimo que passe por um único vértice de cada *cluster*.

Para a modelagem do problema vamos, primeiramente, definir a variável de decisão binária, x_{ij} para $i, j = 1,2,...,n$, equação (3.1), na qual, i e j são os índices que representam as possíveis configurações de bolachas (vértices do grafo) responsáveis pela confecção dos tubetes e n corresponde ao número total de configurações.

$$x_{ij} = \begin{cases} 1, & \textit{se a aresta } (i, j) \textit{ faz parte da solução} \\ 0, & \textit{caso contrário} \end{cases} \tag{3.1}$$

A fim de representar o número de trocas de bolachas necessário para confeccionar um tubete com a configuração de bolachas j, logo após a confecção de outro tubete com a configuração i, introduzimos c_{ij}, que equivale ao custo da aresta (i, j) do grafo.

Como a meta do problema consiste em encontrar a sequência de configurações (uma seqüência de tubetes), que minimiza o número total de trocas de bolachas, podemos definir a seguinte função objetivo:

$$\min z = \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{3.2}$$

Novamente, como cada tubete deve ser fabricado uma única vez, isto implica que apenas uma aresta deve chegar e apenas uma deve sair de cada *cluster*. Esta situação pode ser representada, respectivamente, pelas restrições (3.3) e (3.4).

$$\sum_{\substack{i\notin S_p \\ (i,j)\in A}} \sum_{\substack{j\in S_p \\ (i,j)\in A}} x_{ij} = 1, \quad \textit{para todos os conjuntos } S_p \tag{3.3}$$

$$\sum_{\substack{i\in S_p \\ (i,j)\in A}} \sum_{\substack{j\notin S_p \\ (i,j)\in A}} x_{ij} = 1, \quad \textit{para todos os conjuntos } S_p \tag{3.4}$$

Outra restrição necessária para o problema está representada pela equação (3.5), ela garante que a solução seja ininterrupta, o que significa que para qualquer vértice j do grafo, pertencente à solução, deve existir uma aresta chegando (i, j) e outra saindo (j, k) .

$$\sum_{\substack{i \in N \\ (i,j) \in A}} x_{ij} - \sum_{\substack{k \in N \\ (j,k) \in A}} x_{jk} = 0, \ \forall j \in N \tag{3.5}$$

As equações desenvolvidas até o momento, compõem um modelo que permite a obtenção de soluções com a presença de sub-rotas entre *clusters*, o que é um pouco diferente da formulação do problema via PCV. Assim, é necessário o acréscimo de restrições para a eliminação destas sub-rotas, representadas pelas equações (3.6), em que Ω refere-se aos subconjuntos de vértices, tais que, $2 \leq |\Omega| \leq m-2$ com $\Omega \subseteq N$ (Noon e Bean, 1991).

$$\sum_{\substack{p \in \Omega \\ (i,j) \in A}} \sum_{\substack{i \in S_p \\ (i,j) \in A}} \sum_{\substack{q \notin \Omega \\ (i,j) \in A}} \sum_{\substack{j \in S_q \\ (i,j) \in A}} x_{ij} \geq 1 \tag{3.6}$$

Balas (1989, 1993) apresenta um estudo detalhado envolvendo a formulação da equação (3.6) aplicado ao Problema do Caixeiro Viajante com Coleta de Prêmios.

Por fim, para que possamos garantir que apenas um vértice de cada *cluster* seja visitado, acrescentamos um novo conjunto de restrições ao modelo, uma para cada *cluster*, equações (3.7).

$$\sum_{(i,j) \in N_p} x_{ij} = 0, \quad para \ p = 1, ..., m \tag{3.7}$$

Organizando as equações desenvolvidas, obtemos o modelo completo (3.8 – 3.14) para o Problema da Tubeteira, considerando os intervalos vazios.

$$\min z = \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{3.8}$$

sujeito a:

$$\sum_{\substack{i \notin S_p \\ (i,j) \in A}} \sum_{\substack{j \in S_p \\ (i,j) \in A}} x_{ij} = 1, \quad para \ cada \ S_p \subseteq N \tag{3.9}$$

$$\sum_{\substack{i \in S_p \\ (i,j) \in A}} \sum_{\substack{j \notin S_p \\ (i,j) \in A}} x_{ij} = 1, \quad para \ cada \ S_p \subseteq N \tag{3.10}$$

$$\sum_{\substack{i\in N\\(i,j)\in A}} x_{ij} - \sum_{\substack{k\in N\\(j,k)\in A}} x_{jk} = 0,\ \forall j \in N \tag{3.11}$$

$$\sum_{p\in\Omega} \sum_{\substack{i\in S_p\\(i,j)\in A}} \sum_{\substack{q\notin\Omega\\(i,j)\in A}} \sum_{\substack{j\in S_q\\(i,j)\in A}} x_{ij} \geq 1,\ \ para\ cada\ \Omega\ /\ \ 2 \leq |\Omega| \leq m-2, com\ \Omega \subseteq N \tag{3.12}$$

$$\sum_{(i,j)\in N_p} x_{ij} = 0,\ \ para\ \ p = 1, ..., m \tag{3.13}$$

$$x_{ij} \in \{0,1\},\ \forall (i,j) \in A \tag{3.14}$$

Em uma das últimas visitas na empresa, a gerência sugeriu que considerássemos no problema o diâmetro do mandril, que define o diâmetro interno do tubete. Na prática, a troca de mandril é normalmente realizada após a confecção de todos os tubetes com diâmetros internos iguais, uma vez que o custo operacional para a troca do mandril é consideravelmente superior ao de uma troca de bolacha.

A fim de contemplar esta situação, propomos a estratégia de considerar o mandril como uma bolacha fictícia, cujo custo necessário para a troca é muito maior do que o custo de uma simples troca de bolacha, forçando um sequenciamento contínuo dos tubetes com o mesmo mandril.

No próximo capítulo, apresentamos os resultados numéricos obtidos por meio das simulações dos modelos desenvolvidos para o caso que considera e, para o caso que não considera os intervalos vazios. Apresentamos também, os resultados obtidos de acordo com a estratégia que acabamos de apresentar.

CAPÍTULO 4

SIMULAÇÕES

Apresentamos, neste capítulo, os procedimentos executados durante as simulações (modelagem computacional) e os resultados obtidos. O *software* utilizado foi o *Xpress-MP* (Apêndice B), no qual trabalhamos com a linguagem de programação *Mosel* (*Xpress-Mosel User Guide*, 2008), para a codificação de nossos modelos.

A fim de propiciar uma análise comparativa com a realidade prática do problema, optamos por executar carteiras reais da empresa ao invés de testes aleatórios exaustivos. No entanto, devido à política de sigilo da empresa, conseguimos, após grande resistência por parte da gerência, apenas duas carteiras diferentes executadas por ela em períodos distintos.

Na primeira, relatada neste trabalho como Carteira 1, todos os tubetes possuem o mesmo diâmetro interno e, na segunda, relatada como Carteira 2, existem tubetes com diâmetros internos distintos. Para ambas, apresentamos os resultados via PCV (capítulo 2) e PCVG (capítulo 3).

4.1 Modelagem Computacional

Durante as etapas computacionais desenvolvemos três programas principais, programas 1, 2 e 3.

Programa 1: programa em *Mosel* que calcula os custos entre todas as configurações de bolachas (tubetes), duas a duas, gerando uma matriz de custos;

Programa 2: programa em *Mosel* referente ao modelo matemático desenvolvido para o Problema da Tubeteira que não considera os intervalos vazios, equações

(2.12 – 2.16), que utiliza como dados de entrada a matriz de custos gerada pelo "programa 1";

Programa 3: programa em *Mosel* referente ao modelo matemático desenvolvido para o Problema da Tubeteira que considera os intervalos vazios, equações (3.8 – 3.14), que utiliza como dados de entrada a matriz de custos gerada no "programa 1".

Para a obtenção dos resultados numéricos, seguimos basicamente três protocolos de simulação, um para as carteiras sem considerar os intervalos vazios (protocolo 1), outro, no qual os intervalos vazios são considerados (protocolo 2) e, por fim, o protocolo 3 para a resolução da Carteira 2, na qual o problema foi dividido em quatro partes.

Protocolo 1:

Passo 1 – criação de um arquivo externo contendo as configurações das bolachas de todos os tubetes que compõem a referida carteira;

Passo 2 – alteração dos parâmetros de entrada do "programa 1", conforme dados do exemplo corrente;

Passo 3 – execução do "programa 1";

Passo 4 – alteração dos parâmetros de entrada do "programa 2", conforme dados do exemplo corrente;

Passo 5 – execução do "programa 2".

Protocolo 2:

Passo 1 – criação de um arquivo externo com os *clusters* (todas as configurações possíveis) de todos os tubetes da referida carteira;

Passo 2 – alteração dos parâmetros de entrada do "programa 1", conforme dados do exemplo corrente;

Passo 3 – execução do “programa 1”;

Passo 4 – alteração dos parâmetros de entrada do “programa 3”, conforme dados do exemplo corrente;

Passo 5 – execução do “programa 3”.

Protocolo 3:

Passo 1 – criação de quatro arquivos externos, contendo cada um deles, os *clusters* referentes aos tubetes de cada partição da Carteira 2;

Passo 2:

2.1 – alteração dos parâmetros de entrada do “programa 1”, conforme dados da primeira partição;

2.2 – execução do “programa 1”;

2.3 – alteração dos parâmetros de entrada do “programa 3”, conforme dados da primeira partição;

2.4 – execução do “programa 3”;

2.5 – repetição do “passo 2” para as partições remanescentes;

Passo 3 – criação de um novo arquivo externo composto pelas soluções (seqüências de tubetes) encontradas no “passo 2”;

Passo 4 – execução dos passos 2 ao 5 do “protocolo 1”, cujos dados de entrada são os tubetes do arquivo gerado no passo anterior.

A seguir apresentamos os resultados numéricos obtidos para as carteiras da empresa, por meio dos protocolos que acabamos de descrever.

4.2 Resultados Numéricos

Antes da apresentação das soluções referentes às carteiras, apresentamos o resultado obtido para o Exemplo 1.1 que considera os intervalos vazios, a fim de compararmos com o resultado encontrado para o mesmo exemplo sem considerá-los.

4.2.1 Resultado para o Exemplo 1.1 que considera os intervalos vazios

Para este caso, o número total de configurações de bolachas (vértices do grafo sem contar o vértice de origem) elevou-se de 4 (sem intervalos vazios) para 19 configurações (tabela 3, pág. 21).

Como se trata do caso que considera os intervalos vazios, utilizamos o "protocolo 2" para a obtenção da solução. Segue, na tabela 4, a seqüência encontrada (B,A,D,C), que totaliza 3 trocas de bolachas, uma a menos quando comparada ao caso que não considera os intervalos vazios. Em percentagem, esta redução equivale a 25%.

Tabela 4 – Solução para o Exemplo 1.1 com intervalos vazios.

Solução (tubete)	**Configuração das bolachas**
tubete B	(200, 100, 101, 103, 2000)
tubete A	(200, 100, 102, 103, 2000)
tubete D	(200, 101, 102, 103, 2000)
tubete C	(200, 101, 102, **Ø**, 2000)

4.2.2 Resultados para a Carteira 1 sem intervalos vazios

A Carteira 1 é composta por dez tubetes diferentes (tubete 1, tubete 2, ..., tubete 10), quantidade média produzida pela empresa em uma semana, cujas respectivas configurações de bolachas estão representadas por números, conforme tabela 5. Desta forma, nossa meta consiste em encontrar a seqüência

de execução para os 10 tubetes que minimiza o número total de trocas de bolachas.

Tabela 5 – Tubetes e respectivas configurações de bolachas da Carteira 1.

Tubetes	**Configuração das bolachas**
tubete 1	(10, 20, 40, 60, 80, 80, 90, 90, 90, 100)
tubete 2	(10, 20, 60, 60, 90, 90, 100)
tubete 3	(10, 30, 30, 30, 40, 40, 60, 90, 90, 100)
tubete 4	(10, 20, 40, 80, 80, 90, 90, 100)
tubete 5	(10, 70, 70, 80, 80, 90, 100)
tubete 6	(10, 70, 70, 70, 80, 80, 80, 90, 100)
tubete 7	(10, 20, 20, 60, 60, 70, 70, 70, 90, 100)
tubete 8	(10, 90, 90, 90, 100)
tubete 9	(10, 80, 80, 90, 90, 100)
tubete 10	(10, 20, 30, 40, 60, 90, 100)

Como se trata do caso que não considera os intervalos vazios, utilizamos os passos descritos no “protocolo 1” para a obtenção dos resultados, cuja seqüência encontra-se na tabela 6 e totaliza 44 trocas de bolachas.

Por outro lado, a seqüência de execução que foi utilizada, na prática, pela empresa é a seguinte: tubete 9, tubete 8, tubete 2, tubete 5, tubete 6, tubete 3, tubete 10, tubete 1, tubete 7 e tubete 4, que totaliza 60 trocas de bolachas.

Tabela 6 – Solução computacional para a Carteira 1 sem intervalos vazios.

Solução Ótima	**Configuração das bolachas**
tubete 8	(10, 90, 90, 90, 100)
tubete 9	(10, 80, 80, 90, 90, 100)
tubete 2	(10, 20, 60, 60, 90, 90, 100)
tubete 10	(10, 20, 30, 40, 60, 90, 100)
tubete 4	(10, 20, 40, 80, 80, 90, 90, 100)
tubete 5	(10, 70, 70, 80, 80, 90, 100)
tubete 6	(10, 70, 70, 70, 80, 80, 80, 90, 100)
tubete 1	(10, 20, 40, 60, 80, 80, 90, 90, 90, 100)
tubete 7	(10, 20, 20, 60, 60, 70, 70, 70, 90, 100)
tubete 3	(10, 30, 30, 30, 40, 40, 60, 90, 90, 100)

Comparando os valores encontrados computacionalmente com os valores práticos, verifica-se uma redução de 16 trocas de bolachas (60 → 44), que equivale a aproximadamente 27%. Apresentamos no próximo item os resultados para a Carteira 1, considerando os intervalos vazios.

4.2.3 Resultados para a Carteira 1 com intervalos vazios

Para este caso, o número total de configurações de bolachas elevou-se de 10 (sem intervalos vazios) para 79 configurações. No anexo 1, estão relacionadas todas as 79 configurações, divididas em dez *clusters* (um para cada tubete).

Utilizamos os passos descritos no “protocolo 2” para a obtenção dos resultados, cuja seqüência está relatada na tabela 7 e totaliza 38 trocas de bolachas. Este valor, comparado com o valor encontrado pelo mesmo exemplo sem considerar os intervalos vazios (44 trocas), apresenta uma redução de seis trocas de bolachas, que equivale a aproximadamente 14%.

Por outro lado, com relação à seqüência que foi utilizada na prática pela empresa (60 trocas), a redução foi de vinte e duas trocas (60 → 38), que equivale a aproximadamente 37%.

Tabela 7 – Solução computacional para a Carteira 1 com intervalos vazios.

Solução (tubete)	**Configuração das bolachas**
tubete 3	(10,30,30,30,40,40,60,90,90, **Ø**,100)
tubete 7	(10,20,20,60,60,70,70,70,90, **Ø**,100)
tubete 1	(10,20,40,60,80,80,90,90,90, **Ø**,100)
tubete 4	(10,20,40, **Ø**,80,80,90,90,100)
tubete 6	(10,70,70,70,80,80,80,90,100)
tubete 5	(10,70, **Ø**,70,80,80,90,100)
tubete 10	(10,20, **Ø**,30,40,60,90,100)
tubete 2	(10,20, **Ø**,60,60,90,90,100)
tubete 8	(10,90, **Ø**,90,90,100)
tubete 9	(10,80,80,90,90)

4.2.4 Resultados para a Carteira 2 sem intervalos vazios

A Carteira 2 é composta por dezesseis tubetes diferentes (tubete 1, tubete 2, ..., tubete 16), cujas respectivas configurações de bolachas estão representadas por números (de 1 a 31), conforme tabela 8. Desta forma, nossa meta consiste em encontrar a seqüência de execução para os 16 tubetes que minimiza o número total de trocas de bolachas.

Tabela 8 – Tubetes e respectivas bolachas da Carteira 2.

	Tubetes															
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Configuração das bolachas	**29**	29	**09**	24	**21**	29	**21**	25	**19**	17	**19**	17	**09**	09	**10**	10
	29	29	**09**	24	**21**	29	**22**	25	**19**	17	**19**	17	**09**	09	**10**	10
	29	29	**09**	28	**21**	29	**22**	25	**20**	17	**20**	17	**09**	09	**10**	10
	29	29	**09**	30	**07**	29	**22**	26	**20**	18	**20**	18	**09**	10	**10**	10
	29	29	**10**	06	**07**	29	**23**	14	**27**	18	**27**	18	**10**	10	**11**	11
	15	15	**10**	04	**07**	15	**23**		**27**	01	**27**	01	**10**	10	**11**	11
	15	15	**10**		**08**	15	**23**			19		19	**10**	10	**11**	11
	15	15	**10**		**08**	15	**24**			20		19	**10**	02	**12**	12
	15	15	**11**		**08**	15	**24**			20		20	**24**	11	**12**	12
	15	15	**11**		**08**	15	**24**			27		20	**24**	11	**12**	12
	15	15	**11**		**15**	30	**24**			27		20	**24**	11	**12**	03
	15	15	**11**		**15**	30	**15**					27	**29**	12	**13**	13
	15	15	**15**		**15**	30	**15**					27	**29**	12	**13**	30
	16	15	**15**		**16**	30	**15**						**29**	12	**30**	06
	16	16	**15**		**16**	30	**15**						**30**	12	**06**	31
	16	16	**15**		**16**	06	**30**						**30**	13	**31**	
	16	16	**16**		**16**	04	**30**						**30**	13		
	16	16	**16**		**06**		**30**						**6**	13		
	16	16	**16**		**04**		**30**						**4**	06		
	16	16	**16**				**06**							04		
	16	16	**16**				**04**									
	16	16	**06**													
	06	16	**04**													
	04	16														
		06														
		05														

Como esta carteira exige diferentes mandris durante o processo de confecção dos dezesseis tubetes que a constitui, reunimos estes conforme os respectivos mandris, na tabela 9.

Tabela 9 – Conjuntos de tubetes e respectivos mandris da Carteira 2.

Tubetes	1 e 2	6	3 e 5	4, 14, 15 e 16	7	13	8	9 e 10	11 e 12
Mandril (mm)	304,8	127	76,2	77,0	77,8	70,0	50,0	57,0	60,0

Como se trata do caso que não considera os intervalos vazios, utilizamos o “protocolo 1” para a obtenção dos resultados. Porém, a fim de considerar a estratégia proposta para a minimização do número de trocas de mandril, foi necessário acrescentar uma rotina no “programa 1”, na qual atribuímos custos elevados entre os tubetes que não possuem o mesmo mandril. Por exemplo, para o cálculo do custo entre a confecção do tubete 3 e 6, incrementamos um valor, relativamente alto, ao valor que representa o número de trocas de bolachas entre estes dois tubetes.

No “programa 2”, por sua vez, descontamos da solução obtida os custos acrescentados no “programa 1”, pois os mesmos são fictícios e foram utilizados apenas para forçar a minimização das trocas de mandris.

De acordo com as etapas que acabamos de descrever, obtivemos a seqüência apresentada na tabela 10, que totaliza 180 trocas de bolachas.

Com a finalidade de comprovar a coerência desta solução com a estratégia utilizada para a minimização das trocas de mandris, pode ser verificado, pela tabela 10, que todos os tubetes que utilizam o mesmo mandril apresentam-se consecutivamente na solução (linhas consecutivas da tabela).

Tabela 10 - Solução da Carteira 2 sem intervalos vazios.

Solução	Configuração das bolachas
tubete 2	(29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,16,6,5)
tubete 1	(29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
tubete 6	(29,29,29,29,29,15,15,15,15,15,30,30,30,30,30,6,4)
tubete 8	(25,25,25,26,14)
tubete 9	(19,19,20,20,27,27)
tubete 10	(17,17,17,18,18,1,19,20,20,27,27)
tubete 12	(17,17,17,18,18,1,19,19,20,20,20,27,27)
tubete 11	(19,19,20,20,27,27)
tubete 4	(24,24,28,30,6,4)
tubete 15	(10,10,10,10,11,11,11,12,12,12,13,13,30,6,31)
tubete 16	(10,10,10,10,11,11,11,12,12,12,3,13,30,6,31)
tubete 14	(9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4,)
tubete 3	(9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
tubete 5	(21,21,21,7,7,7,8,8,8,8,15,15,15,16,16,16,16,6,4)
tubete 13	(9,9,9,9,10,10,10,10,24,24,24,29,29,29,30,30,30,6,4,)
tubete 7	(21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)

A seqüência utilizada pela empresa, para a execução dos 16 tubetes foi a seguinte: tubete 1, tubete 2, tubete 6, tubete 5, tubete 3, tubete 4, tubete 16, tubete 15, tubete 14, tubete 7, tubete 13, tubete 8, tubete 10, tubete 9, tubete 12 e tubete 11, que totaliza 227 trocas. Comparando esta seqüência com o resultado obtido computacionalmente, obtivemos uma redução de 47 trocas (227 → 180), que em percentagem, equivale a aproximadamente 21%. No próximo item, apresentamos os resultados para a Carteira 2, considerando os intervalos vazios.

4.2.5 Resultados para a Carteira 2 com intervalos vazios

Para esta carteira, o número total de configurações de bolachas elevou-se de 16 (sem intervalos vazios) para 246 configurações. Veja, no anexo 1, todas as configurações, divididas em dezesseis *clusters* (um para cada tubete).

Como se trata do caso que considera os intervalos vazios, utilizamos o "protocolo 2" para a obtenção da solução, porém, após 100 horas de execução do "programa 2" em um computador com um processador *Intel Dual Core* de 1.6 *GHz* e 1 *GB* de memória *RAM,* nenhuma solução sem a presença de sub-rotas (solução viável) foi obtida. Uma nova tentativa foi feita numa máquina com um processador *Intel Core 2 Duo* de 3 *GHz* e 4 *GB* de *Ram*, porém após 100 horas de execução também não foi possível a obtenção de uma solução viável.

Como alternativa, decidimos dividir o problema em partes menores, com o prejuízo de obtermos um sub-ótimo. Tendo em vista os dados apresentados na tabela 9, dividimos a carteira em duas partes, de modo que cada uma delas viesse a ter oito *clusters*, conforme detalhamento a seguir:

Parte 1: composta pelos *clusters* referentes aos tubetes 1 e 2 (mandril 304,8 mm), pelos tubetes 3 e 5 (mandril 76,2 mm), pelos tubetes 9 e 10 (mandril 57,0 mm), e pelos tubetes 11 e 12 (mandril 60,0 mm);

Parte 2: composta pelos *clusters* referentes aos tubetes 6 (mandril 127,0 mm), 7 (mandril 77,8 mm), 8 (mandril 50,0 mm), 13 (mandril 70,0 mm) e pelos *clusters* referentes aos tubetes 4, 14, 15 e 16 (mandril de 77,0 mm).

A partir do problema particionado, utilizamos os passos descritos no "protocolo 3" para a obtenção dos resultados. O diagrama com as principais etapas de simulação está ilustrado na figura 12.

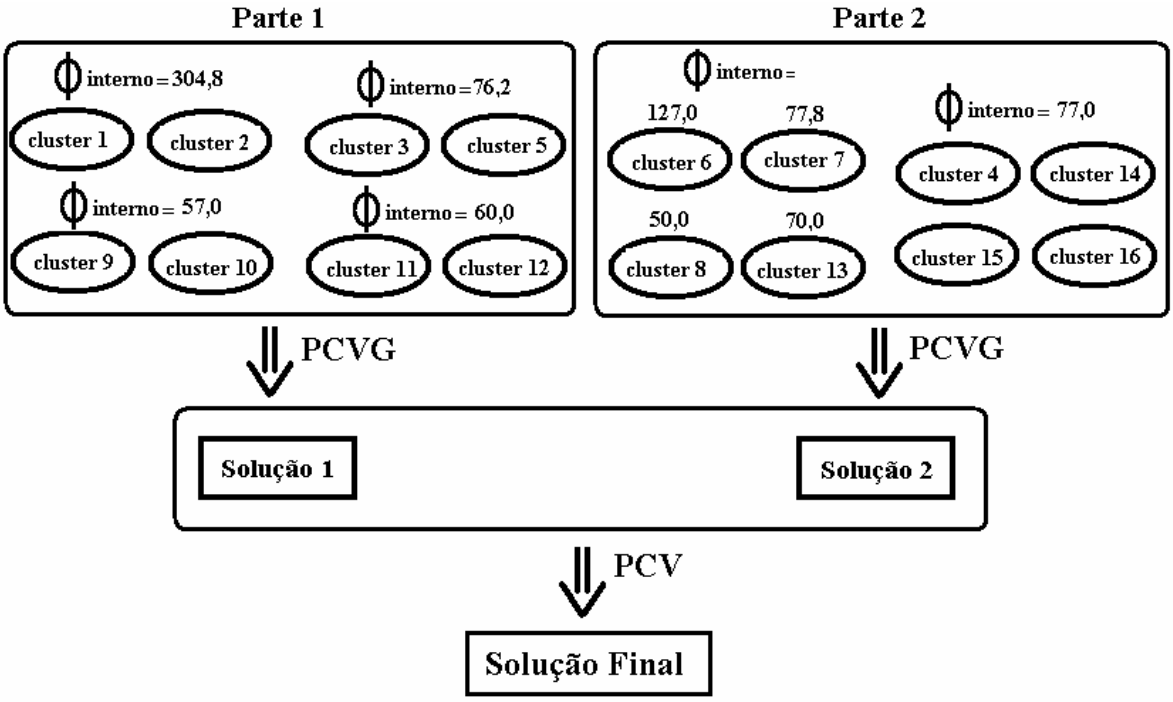


Figura 12 – Etapas de simulação da Carteira 2 com intervalos vazios dividida em 4 partes.

De acordo com a figura 12, primeiramente aplicamos a estratégia do PCVG (programa 3), para cada uma das partes do problema, por fim, utilizamos o PCV como estratégia para o novo espaço de busca, formado pela união das duas soluções (1 e 2).

A solução final obtida encontra-se na tabela 11, cuja sequência de processamento dos tubetes exige 173 trocas de bolachas. Comparando este resultado com a seqüência utilizada na prática pela empresa, obtivemos uma redução de 54 trocas (227 → 173), que em percentagem, equivale a aproximadamente 24%. Por outro lado, comparando a solução (173 trocas) com o resultado obtido pelo mesmo exemplo sem considerar os intervalos vazios (180 trocas), obtivemos uma redução de sete trocas de bolachas, que equivale a aproximadamente 4%.

Tabela 11 - Solução da Carteira 2 com intervalos vazios, dividida em 2 partes.

Solução (tubetes)	**Configuração das bolachas**
tubete 2	(29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,16,6, **Ø**,5)
tubete 1	(29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
tubete 6	(29,29,29,29,29,15,15,15,15,15,30,30,30,30,30,6, **Ø**,4)
tubete 12	(17,17,17,18,18,1,19,19,20,20, **Ø**,20,27,27)
tubete 11	(19,19,20,20,27,27)
tubete 9	(19,19,20,20,27,27)
tubete 10	(17,17,17,18,18,1,19,20,20,27,27)
tubete 8	(25,25, **Ø**,25,26,14)
tubete 4	(24,24, **Ø**,28,30,6,4)
tubete 15	(10,10, **Ø**,10,10,11,11,11,12,12,12,13,13,30,6,31)
tubete 16	(10,10, **Ø**,10,10,11,11,11,12,12,12,3,13,30,6,31)
tubete 14	(9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
tubete 13	(9,9,9,9,10,10,10,10,24,24,24,29, **Ø**,29,29,30,30,30,6,4)
tubete 7	(21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6, **Ø**,4)
tubete 5	(21,21,21,7,7,7,8,8,8,8, **Ø**,15,15,15,16,16,16,16,6,4)
tubete 3	(9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16, **Ø**,16,6,4)

Para este caso (Carteira 2 com intervalos vazios), apresentamos também os resultados referentes a uma outra simulação, no qual dividimos o problema em 9 partes, uma para cada mandril. Veja na figura 13, o diagrama referente às etapas de simulação utilizadas para obtenção dos resultados.

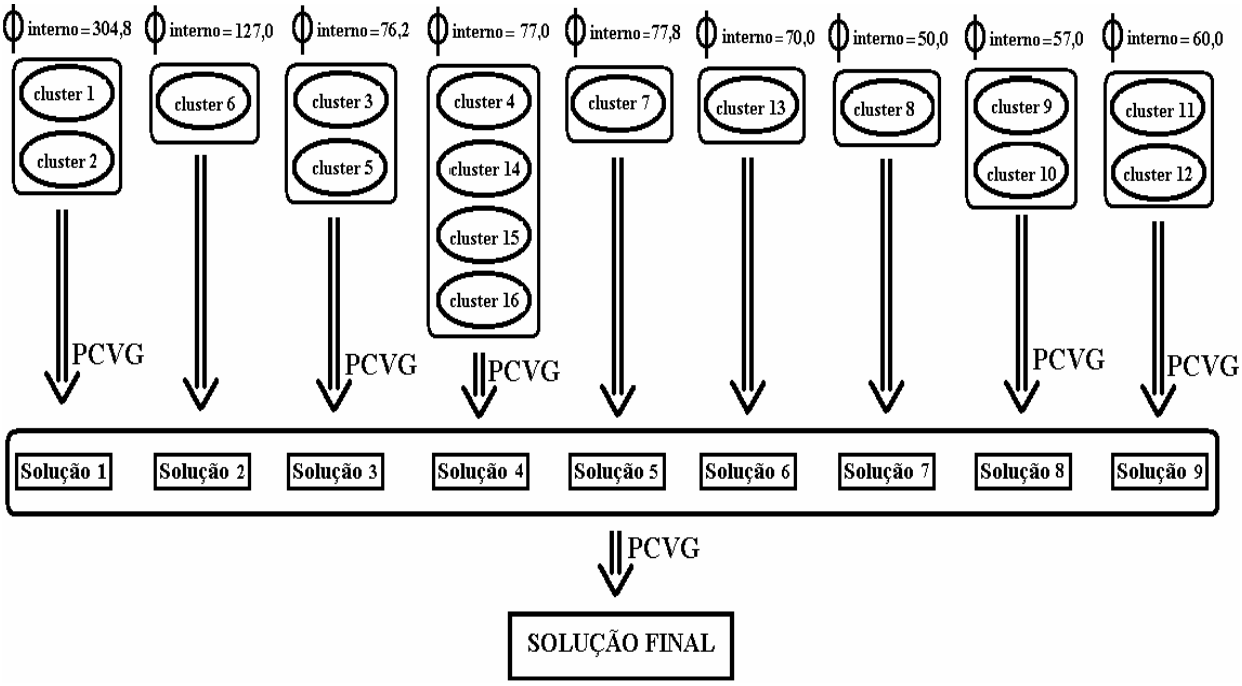


Figura 13 – Etapas de simulação da Carteira 2 com intervalos vazios dividida em 9 partes.

De acordo com a figura 13, utilizamos a abordagem do PCVG para cada um dos conjuntos de tubetes (*clusters*) que utilizam o mesmo mandril, com exceção dos mandris que são utilizados para a confecção de um único tubete, em que a solução é o próprio *cluster*. Por fim, para a obtenção da solução final, foi utilizada novamente a abordagem do PCVG, pois, no novo espaço de busca nem todos os *clusters* contém apenas um vértice (PCV), como no caso anterior (2 partes).

A seqüência obtida está apresentada na tabela 12 e totaliza 175 trocas de bolachas.

Tabela 12 - Solução da Carteira 2 com intervalos vazios, dividida em nove partes.

Solução (tubetes)	**Configuração das bolachas**
tubete 3	(9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,**Ø**,4)
tubete 5	(21,**Ø**,21,21,7,7,7,8,8,8,8,15,15,15,16,16,16,16,6,4)
tubete 7	(21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,**Ø**,4)
tubete 13	(9,9,9,9,10,10,10,10,24,24,24,29,29,29,**Ø**,30,30,30,6,4)
tubete 14	(9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
tubete 15	(10,**Ø**,10,10,10,11,11,11,12,12,12,13,13,30,6,31)
tubete 16	(10,**Ø**,10,10,10,11,11,11,12,12,12,3,13,30,6,31)
tubete 4	(24,**Ø**,24,28,30,6,4)
tubete 8	(25,25,25,**Ø**,26,14)
tubete 10	(17,17,17,18,18,1,19,20,**Ø**,20,27,27)
tubete 9	(19,19,20,20,27,27)
tubete 11	(19,19,20,20,27,27)
tubete 12	(17,17,17,18,18,1,19,**Ø**,19,20,20,20,27,27)
tubete 6	(29,29,29,29,29,15,15,15,15,15,30,30,30,30,30,6,4)
tubete 1	(29,29,29,29,29,15,15,15,15,15,15,15,15,**Ø**,16,16,16,16,16,16,16,16,16,6,4)
tubete 2	(29,29,29,29,29,15,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,16,6,**Ø**,5)

A fim de comprovar a coerência das soluções para o caso que considera os intervalos vazios (2 partes e 9 partes), com a estratégia utilizada para a minimização das trocas de mandris, pode ser verificado, pela tabela 11 e 12, que todos os tubetes que utilizam o mesmo mandril apresentam-se consecutivamente na solução (linhas consecutivas da tabela).

4.3 Considerações finais

Com a finalidade de organizar os resultados obtidos e facilitar uma análise comparativa com as respectivas seqüências utilizadas na prática pela empresa, reunimos na tabela 13, todas as soluções computacionais e práticas.

Tabela 13 – Soluções computacionais e práticas.

Exemplos	Modelos Utilizados	Estratégias	Custos totais (Computacional)	Custos totais (Prática)
Exemplo 1.1	Sem intervalos vazios	PCV	4 trocas	X
	Com intervalos vazios	PCVG	3 trocas	
Carteira 1 (um mandril)	Sem intervalos vazios	PCV	44 trocas	60 trocas
	Com intervalos vazios	PCVG	38 trocas	
Carteira 2 (nove mandris)	Sem intervalos vazios	PCV	180 trocas	227 trocas
	Com intervalos vazios (2 partes)	PCVG + PCV	173 trocas	
	Com intervalos vazios (9 partes)	PCVG	175 trocas	

Por meio de uma análise geral dos resultados, obtivemos computacionalmente, uma melhoria significativa em relação às soluções práticas da empresa, o que comprova a eficiência dos modelos desenvolvidos.

As conclusões referentes aos resultados obtidos estão apresentadas, com um maior nível de detalhes, no próximo capítulo, onde apresentamos também, algumas propostas para investigações futuras.

CAPÍTULO 5

CONCLUSÕES E INVESTIGAÇÕES FUTURAS

Reunimos neste capítulo, além das conclusões obtidas com base nos resultados numéricos, os principais tópicos que ainda necessitam atenção em análises futuras.

Primeiramente, após termos obtido uma razoável compreensão do problema, propusemos um modelo matemático, o qual validamos com dados reais ao invés de efetuarmos exaustivos testes aleatórios, não obstante, isto deverá ser feito no momento em que uma heurística for construída.

Como o número de trocas de bolachas está diretamente relacionado com o tempo de preparação da tubeteira, esperávamos que este tempo fosse reduzido quando estas trocas fossem minimizadas, o que de fato comprovamos, por meio das simulações efetuadas.

Não conseguimos uma quantificação exata, por parte da empresa, do tempo operacional gasto durante uma troca de bolacha, que foi atribuído como 5 minutos em média. Desta forma, por exemplo, para o caso da Carteira 2, a redução obtida foi de até 54 trocas (24%) de bolachas, que corresponde a uma diminuição de 270 minutos no tempo de preparação da tubeteira, aproximadamente quatro horas e meia gastas com mão de obra e com a máquina fora de produção.

De acordo com os resultados obtidos para os casos que não considera e que considera os intervalos vazios, verificamos um melhor desempenho no segundo, que pode ser explicado pelo aumento do poder combinatório do problema. Em contrapartida, ocorre um acréscimo do custo computacional, provocado pelo aumento considerável no número de vértices do grafo que representa o problema.

Dentre todos os resultados computacionais, não encontramos solução ótima apenas para o caso da Carteira 2, considerando os intervalos vazios. Alternativamente, obtivemos um sub-ótimo a partir de particionamentos do problema. Ainda assim, os resultados encontrados foram melhores do que os obtidos sem considerar os intervalos vazios.

A inviabilidade da solução ótima para o exemplo Carteira 2 via PCVG, nos sugere a investigação de uma heurística para resolver o problema. Uma das saídas, comumente tratada na literatura para aplicações práticas do PCVG é a redução do mesmo em um PCV equivalente, isto ocorre devido ao número de heurísticas potenciais envolvendo este último.

No trabalho de Yanasse e Pinto (2006) é apresentada uma técnica que reduz um PCVG a um PCV, aplicada ao Problema de Minimização de Trocas de Ferramentas, do inglês *Minimization of Tools Switches Problem* (*MTSP*) introduzido por Tang e Denardo (1988).

Ao examinar a técnica de Yanasse e Pinto (2006), acreditamos que a mesma possa ser adaptada ao Problema da Tubeteira, desta forma sugerimos que em trabalhos futuros isto seja investigado.

Concomitante ao desenvolvimento de uma heurística, outro potencial objetivo futuro é o desenvolvimento de um sistema computacional, que possibilite uma programação eficiente e rápida na determinação de boas seqüências de processamento para os tubetes.

APÊNDICE A

NOÇÕES BÁSICAS SOBRE GRAFOS

Tipicamente, um grafo é representado como um conjunto de pontos (vértices) ligados por retas (arestas). Dependendo da aplicação, as arestas podem ser direcionadas, e são representadas por "setas". Os grafos são muito úteis na representação de problemas da vida real, em vários campos profissionais. Os grafos podem possuir pesos (ou custos), quer nas arestas quer nos vértices, e o custo total em estudo será calculado a partir destes pesos. As notações e definições apresentadas neste apêndice foram baseadas no trabalho de Cardoso (2004).

1 Notações e noções básicas

Um grafo é representado por um conjunto de arestas e um conjunto de vértices, ou seja, um grafo G qualquer é representado por um conjunto $V(G)=\{v_1,...,v_n\}$ de vértices e por um conjunto $E(G)$ de arestas, sendo que cada uma delas possui um subconjunto de $V(G)$ de cardinalidade 2, isto é, $E(G)=\{e_1,...,e_n\}$ com $e_k=\{v_{ki},v_{kj}\}$, para $k\in\{1,...,m\}$. Por simplicidade de notação, uma aresta entre os vértices v_1 e v_2 será chamada de aresta v_1v_2.

Um grafo é considerado simples quando, nele, não existem arestas paralelas (mais do que uma aresta entre dois mesmos vértices) nem lacetes (arestas com ambos os extremos nos mesmos vértices).

1.1 Ordem e dimensão de um Grafo

A ordem e dimensão de um grafo referem-se ao número de vértices e arestas respectivamente. Normalmente os grafos são representados por linhas (arestas)

e pontos (vértices), por meio de uma figura plana. Veja na figura 14, um exemplo de grafo de ordem 6 e dimensão igual a 7.

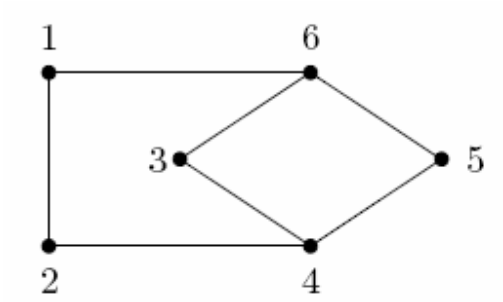


Figura 14 – Grafo de ordem e dimensão 6 e 7, respectivamente.

1.2 Sub-grafos e Super-grafos

Dados dois grafos G e G', diz-se que G' é um sub-grafo de G e que G é um super-grafo de G' se os conjuntos dos vértices e arestas de G' estão contidos em G, ou seja, $V(G') \subseteq V(G)$ e $E(G') \subseteq E(G)$. Veja na figura 15 uma representação gráfica exemplificando um sub-grafo cujos vértices são representados pelo sub-conjunto {3, 4, 5, 6} e seu respectivo super-grafo G representado pelo conjunto dos vértices {1, 2, 3, 4, 5, 6}

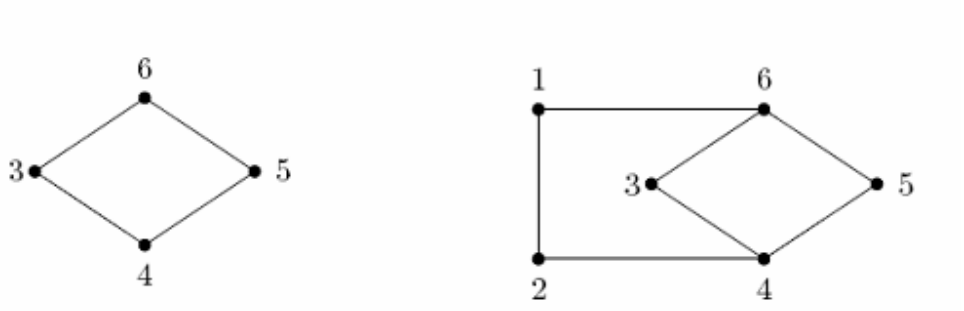


Figura 15 – Sub-grafo e respectivo super-grafo G .

1.3 Grau em grafos

Seja um grafo G qualquer e um vértice $v \in V(G)$, designa-se por grau ou valência de v e denota-se por *dG(v)* o número de arestas de G incidentes em v e ainda, denota-se por $\Delta(G)$ e $\delta(G)$ como sendo o número máximo e mínimo de vértices de G, respectivamente.

Um grafo é dito *p*-regular se e somente se todos os seus vértices têm grau p.

Quando se leva em conta os graus de todos os vértices de um grafo arbitrário G, cada aresta conta duas vezes, ficando claro então que:

$$\sum_{v \in V(G)} d_G(v) = 2\left|E(G)\right| \qquad \text{(A.1)}$$

Da equação (A.1) decorre que:

$$\delta(G) \leq \left\lfloor \frac{2\left|E(G)\right|}{n} \right\rfloor \leq \Delta(G) \qquad \text{(A.2)}$$

1.4 Vizinhança, passeio, trajeto e circuito de um grafo

Designa-se por vizinhança de um vértice qualquer v todos os vértices adjacentes a ele e denota-se este conjunto de vértices vizinhos por $N_G(v)$ e conseqüentemente, $d_G(v) = \left|N_G(v)\right|$.

Define-se um passeio em um grafo, entre dois vértices x e y quaisquer, como sendo toda a seqüência de vértices e arestas no formato da equação (A.3) com eventual repetição de vértices e arestas.

$$x = v_1, v_1v_2, v_2, ..., v_{k-1}v_k, v_k = y \qquad \text{(A.3)}$$

Para o caso representado pela equação (A.3) tem-se x e y como vértices extremos do passeio sendo x o vértice inicial e y o vértice final. Um trajeto, entre dois vértices quaisquer x e y, feito em um grafo é um passeio entre x e y sem arestas repetidas, podendo haver vértices repetidos. Já um caminho entre dois vértices é definido como sendo um trajeto sem vértices repetidos.

Um caso específico e muito usado na teoria de grafos é o circuito que é representado por um trajeto fechado, onde o vértice inicial coincide com o vértice final. Define-se por ciclos, todo trajeto fechado, em que somente os vértices inicial e final coincidem.

1.5 Comprimento e distância em grafos

Dado um caminho P (ciclo C) de um grafo G qualquer, designa-se por comprimento de $P(C)$ e denota-se por *comp(P) (comp(C)* o número de arestas que o constitui. Por exemplo, uma aresta é um caminho de comprimento 1 e um vértice é um caminho de comprimento nulo.

Seja um grafo G qualquer de ordem n com dois vértices quaisquer x e $y \in V(G)$ e seja $P_G(x, y)$ a notação dada ao conjunto de todos os caminhos de G entre x e y, designa-se então por distância entre vértices de G a seguinte função:

$$d_G : V(G) \times V(G) \quad \leftrightarrow \quad \{0, ..., n-1\} \cup \{\infty\} \qquad \text{(A.4)}$$

$$(x, y) \quad \rightarrow \quad d_G(x, y) = \begin{cases} \min\{comp(P) : P \in P_G(x, y)\} & se \quad P_G(x, y) \neq \emptyset \\ \infty & se \quad P_G(x, y) = \emptyset \end{cases}$$

Embora as notações designadas para o grau de um vértice e para a distância entre dois vértices sejam idênticas, torna-se muito simples a distinção entre elas a partir do contexto em que aparecem.

A menor distância entre dois vértices de um grafo G qualquer é designada como cintura de G e por sua vez a maior distância designa-se por diâmetro de G. Para os casos em que G é acíclico, ou seja, não possui ciclos, diz-se que o mesmo possui cintura infinita, veja equação (A.5) e para os casos em que G não é conexo, diz-se que o grafo tem diâmetro infinito, veja equação (A.6).

$$diam(G) = \infty \tag{A.5}$$

$$g(G) = \infty \tag{A.6}$$

2 Ciclos hamiltonianos

Designa-se por ciclo hamiltoniano, todo ciclo que contém todos os vértices de um grafo. Deste modo, todo caminho que contém todos os vértices do grafo pode ser chamado de caminho de Hamilton e por sua vez, todo grafo que contém um ciclo de Hamilton é dito grafo hamiltoniano.

A busca de um ciclo hamiltoniano em um grafo de pequena ordem é consideravelmente fácil, porém com o aumento do número de vértices e arestas do grafo, provar que não existe nenhum ciclo de Hamilton torna-se uma tarefa muito complexa.

Na busca de ciclos de Hamilton em grafos deve-se levar em conta as três regras básicas seguintes:

1. Para quaisquer vértices de grau dois, as arestas incidentes em tais vértices devem fazer parte do ciclo.
2. Não se deve percorrer nenhum ciclo que não contenha todos os vértices do grafo, ou seja, nenhum sub-ciclo deve ser obtido.

3. Deve-se eliminar todas as arestas não usadas incidentes em um determinado vértice já visitado.

Veja na figura 26 um exemplo de ciclo hamiltoniano.

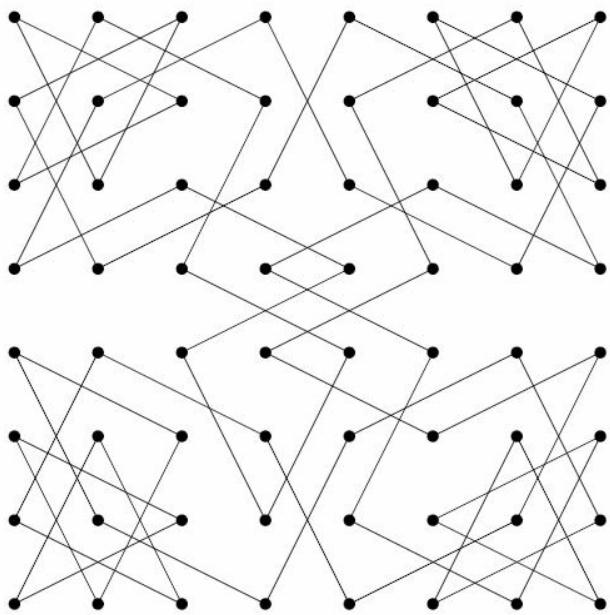


Figura 16 – Exemplo de um ciclo hamiltoniano.

APÊNDICE B

O PACOTE DE OTIMIZAÇÃO *XPRESS-MP*

O pacote *Xpress-MP* é composto por um conjunto de ferramentas de otimização e modelagem matemática utilizadas para resolver problemas de programação linear, inteira, quadrática, não-linear e estocástica. O *software* está disponível em todas as plataformas de computadores comuns e dependendo da necessidade da aplicação existem variadas versões que possibilitam o trabalho com problemas com mais de milhares de restrições e variáveis (*Applications Of Optimization With Xpress-MP,* 2000). Neste apêndice, apresentamos as principais características e funcionalidades do *Xpress-MP*.

1 Componentes e Interfaces

Basicamente, os dois componentes do *Xpress-MP* são o *Xpress-Mosel* e o *Xpress-Optimizer*. As ferramentas do pacote compreendem uma coleção de interfaces, objetivando atender às necessidades de diversos usuários e aplicações.

De maneira geral, pode-se discriminar do pacote as seguintes interfaces:

- ambiente de desenvolvimento gráfico *IVE;*

- livrarias comunicáveis com API_s para a maioria das linguagens de programação;

- versões de consoles autônomos com blocos e linhas de comandos específicos;

- API_s de programação com a linguagem *MOSEL*. Integração de componentes com a linguagem *MOSEL,* utilizando os módulo de conexão *MOSEL*.

2 *Xpress-Mosel*

O ambiente *Xpress-Mosel* permite formular o problema, resolver e analisar a solução, utilizando uma linguagem de programação completamente funcional, especificamente desenvolvida para tais finalidades. Este ambiente de desenvolvimento é composto basicamente da linguagem de programação *Mosel* com seu depurador, módulos e portas de entrada e saída para o acesso a componentes de outros *softwares* e dados de fontes externas, diretamente por meio da linguagem. Por exemplo, dados podem ser transferidos por meio de arquivos tipo texto e em formato *ODBC*.

O *Xpress-Mosel* possui também, uma interface aberta para extensões de escrita-usuário para a linguagem *Mosel*.

Programas em *Mosel* são compilados, o que os tornam rápidos e com a propriedade de não permitir que a lógica de comandos do programa torne-se disponível para usuários finais.

2.1 Linguagem Mosel

A linguagem *Mosel* segue padrões de linguagens básicas como *C*, *C*++ e *Pascal*. O desenvolvimento visual, por meio do ambiente *Xpress-IVE*, facilita o processo (*Xpress Mosel User Guide*, 2008).

Modelos de desenvolvimento na linguagem *Mosel* são auxiliados pelo depurador *Mose,l* que apura todos os erros típicos de programação, muito útil para o rastreamento e análise da execução de um modelo.

Um modelo desenvolvido por meio da linguagem *Mosel* pode ser executado e acessado por diferentes ambientes de programação, como *C*, *C*++ e *Java* por meio de livrarias do *Mosel*.

2.2 *Xpress-IVE*

O Xpress-IVE é um ambiente gráfico de desenvolvimento completo para o *Xpress-Mosel*. Ele incorpora um editor para programa em *Mosel*, um compilador e ambiente visual de execução.

Utilizando o *Xpress-IVE,* o usuário pode visualizar os objetos dentro do seu modelo por meio de árvore de entidades, que exibe a solução de todos os valores dos objetos de otimização quando disponíveis. O usuário pode também, organizar os arquivos fonte e arquivos de dados em projetos e ainda verificar o desempenho do otimizador por meio de sua visualização em tempo real de rodagem.

A tela principal do programa é composta, de acordo com a figura 17, dos seguintes blocos:

1 – No topo da tela, tem-se as opções básicas de manuseio de arquivos e ferramentas do programa.

2 – A primeira janela da esquerda para a direita refere-se à árvore de entidades que permite a visualização de parâmetros, restrições e demais entidades utilizadas pelo programador.

3 – A janela central refere-se ao editor, local onde encontra-se a lógica do processo de otimização, ou seja, o código fonte escrito em Linguagem *Mosel*.

4 – A última janela da direita tem como função principal amostrar os resultados obtidos após a compilação e execução do programa. Estes resultados podem ser amostrados por meio de gráficos, grafos, valores absolutos entre outras opções, de acordo com a necessidade do usuário.

5 – A janela inferior revela os dados referentes à compilação do código bem como os erros encontrados pelo compilador, possibilitando por meio de informações precisas, uma depuração rápida e eficaz do programa.

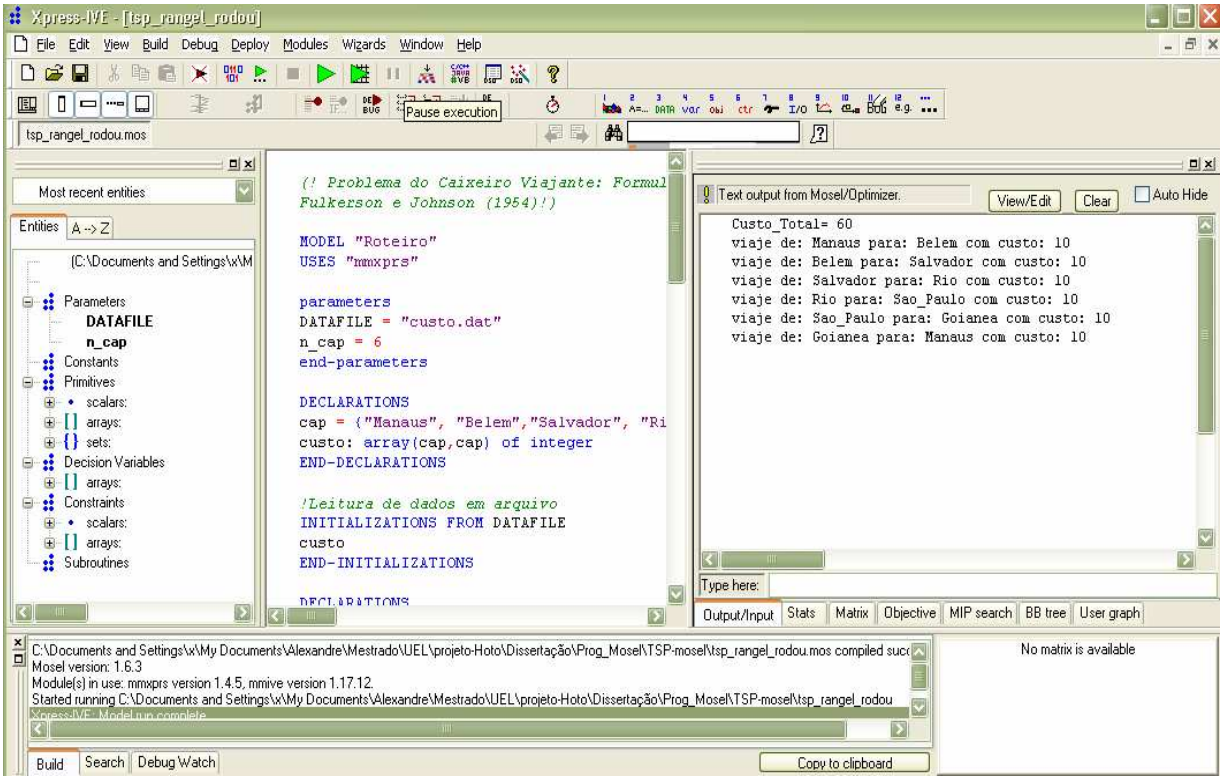


Figura 17 – Janela principal da interface Xpress-IVE.

3 Programação *BCL*

A filosofia por trás do *BCL* permite que, por meio de linguagens de programação de alto nível (*C, C++, Java, C#* ou *VB*), o usuário possa construir matrizes de forma muito flexível, passo a passo, sem que seja necessária uma determinada ordem pré-estabelecida.

A programação em *BCL* possui um elevado grau de sigilo, pois as linhas de comando são chamadas por conjuntos de rotinas, que após a compilação, tornam-se inacessíveis ao usuário final.

Uma vez concluído o problema, o mesmo é definido e resolvido utilizando a *Xpress-Optimizer*. Outras funções em *BCL* permitem o acesso e a análise da solução diretamente dentro de sua aplicação.

Maiores informações sobre a composição, funcionamento e principalmente sobre a linguagem de programação (mosel) podem ser obtidas por meio do Guia do Usuário *Xpress-Mosel* (*Xpress-Mosel User Guide*, 2008).

ANEXO 1

CLUSTERS DA CARTEIRA 1 E 2.

Neste anexo, estão apresentados os *clusters* com todas as configurações de bolachas possíveis para a confecção de todos os tubetes das Carteiras 1 e 2 da empresa.

Clusters da Carteira 1

Para esta carteira, as possíveis configurações estão numeradas de 1 a 79, nas quais as diferentes bolachas estão representadas pelos números (10, 20, 30, 40, 60, 70, 80, 90 e 100) e os intervalos vazios pelo símbolo *Ø.*

Configurações - Tubete 1

1 - (10,20,40,60,80,80,90,90,90,100)
2 - (10,Ø,20,40,60,80,80,90,90,90,100)
3 - (10,20, Ø,40,60,80,80,90,90,90,100)
4 - (10,20,40, Ø,60,80,80,90,90,90,100)
5 - (10,20,40,60, Ø,80,80,90,90,90,100)
6 - (10,20,40,60,80, Ø,80,90,90,90,100)
7 - (10,20,40,60,80,80, Ø,90,90,90,100)
8 - (10,20,40,60,80,80,90, Ø,90,90,100)
9 - (10,20,40,60,80,80,90,90, Ø,90,100)
10 - (10,20,40,60,80,80,90,90,90, Ø,100)

Configurações - Tubete 3

18 - (10,30,30,30,40,40,60,90,90,100)
19 - (10,Ø,30,30,30,40,40,60,90,90,100)
20 - (10,30,Ø,30,30,40,40,60,90,90,100)
21 - (10,30,30,Ø,30,40,40,60,90,90,100)
22 - (10,30,30,30,Ø,40,40,60,90,90,100)
23 - (10,30,30,30,40,Ø,40,60,90,90,100)
24 - (10,30,30,30,40,40,Ø,60,90,90,100)
25 - (10,30,30,30,40,40,60,Ø,90,90,100)
26 - (10,30,30,30,40,40,60,90,Ø,90,100)
27 - (10,30,30,30,40,40,60,90,90,Ø,100)

Configurações - Tubete 2

11 - (10,20,60,60,90,90,100)
12 - (10,Ø,20,60,60,90,90,100)
13 - (10,20,Ø,60,60,90,90,100)
14 - (10,20,60,Ø,60,90,90,100)
15 - (10,20,60,60,Ø,90,90,100)
16 - (10,20,60,60,90,Ø,90,100)
17 - (10,20,60,60,90,90,Ø,100)

Configurações - Tubete 4

28 - (10,20,40,80,80,90,90,100)
29 - (10,Ø,20,40,80,80,90,90,100)
30 - (10,20,Ø,40,80,80,90,90,100)
31 - (10,20,40,Ø,80,80,90,90,100)
32 - (10,20,40,80,Ø,80,90,90,100)
33 - (10,20,40,80,80,Ø,90,90,100)
34 - (10,20,40,80,80,90,Ø,90,100)

35 - (10,20,40,80,80,90,90,Ø,100)

Configurações - Tubete 5

36 - (10,70,70,80,80,90,100)
37 - (10,Ø,70,70,80,80,90,100)
38 - (10,70,Ø,70,80,80,90,100)
39 - (10,70,70,Ø,80,80,90,100)
40 - (10,70,70,80,Ø,80,90,100)
41 - (10,70,70,80,80,Ø,90,100)
42 - (10,70,70,80,80,90,Ø,100)

Configurações - Tubete 6

43 - (10,70,70,70,80,80,80,90,100)
44 - (10,Ø,70,70,70,80,80,80,90,100)
45 - (10,70,Ø,70,70,80,80,80,90,100)
46 - (10,70,70,Ø,70,80,80,80,90,100)
47 - (10,70,70,70,Ø,80,80,80,90,100)
48 - (10,70,70,70,80,Ø,80,80,90,100)
49 - (10,70,70,70,80,80,Ø,80,90,100)
50 - (10,70,70,70,80,80,80,Ø,90,100)
51 - (10,70,70,70,80,80,80,90,Ø,100)

Configurações - Tubete 7

52 - (10,20,20,60,60,70,70,70,90,100)
53 - (10,Ø,20,20,60,60,70,70,70,90,100)
54 - (10,20,Ø,20,60,60,70,70,70,90,100)
55 - (10,20,20,Ø,60,60,70,70,70,90,100)
56 - (10,20,20,60,Ø,60,70,70,70,90,100)
57 - (10,20,20,60,60,Ø,70,70,70,90,100)
58 - (10,20,20,60,60,70,Ø,70,70,90,100)
59 - (10,20,20,60,60,70,70,Ø,70,90,100)
60 - (10,20,20,60,60,70,70,70,Ø,90,100)
61 - (10,20,20,60,60,70,70,70,90,Ø,100)

Configurações - Tubete 8

62 - (10,90,90,90,100)
63 - (10,Ø,90,90,90,100)
64 - (10,90,Ø,90,90,100)
65 - (10,90,90,Ø,90,100)
66 - (10,90,90,90,Ø,100)

Configurações - Tubete 9

67 - (10,80,80,90,90,100)
68 - (10,Ø,80,80,90,90,100)
69 - (10,80,Ø,80,90,90,100)
70 - (10,80,80,Ø,90,90,100)
71 - (10,80,80,90,Ø,90,100)
72 - (10,80,80,90,90,Ø,100)

Configurações - Tubete 10

73 - (10,20,30,40,60,90,100)
74 - (10,Ø,20,30,40,60,90,100)
75 - (10,20,Ø,30,40,60,90,100)
76 - (10,20,30,Ø,40,60,90,100)
77 - (10,20,30,40,Ø,60,90,100)
78 - (10,20,30,40,60,Ø,90,100)
79 - (10,20,30,40,60,90,Ø,100)

Clusters da Carteira 2

Para esta carteira, as possíveis configurações estão numeradas de 1 a 245, nas quais as diferentes bolachas estão representadas pelos números de 1 a 31 e os intervalos vazios pelo símbolo *Ø*.

Configurações - Tubete 1

1 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
2 - (29,Ø,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
3 - (29,29,Ø,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
4 - (29,29,29,Ø,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
5 - (29,29,29,29,Ø,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
6 - (29,29,29,29,29,Ø,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
7 - (29,29,29,29,29,15,Ø,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
8 - (29,29,29,29,29,15,15,Ø,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
9 - (29,29,29,29,29,15,15,15,Ø,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
10 - (29,29,29,29,29,15,15,15,15,Ø,15,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
11 - (29,29,29,29,29,15,15,15,15,15,Ø,15,15,15,16,16,16,16,16,16,16,16,16,6,4)
12 - (29,29,29,29,29,15,15,15,15,15,15,Ø,15,15,16,16,16,16,16,16,16,16,16,6,4)
13 - (29,29,29,29,29,15,15,15,15,15,15,15,Ø,15,16,16,16,16,16,16,16,16,16,6,4)
14 - (29,29,29,29,29,15,15,15,15,15,15,15,15,Ø,16,16,16,16,16,16,16,16,16,6,4)
15 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,Ø,16,16,16,16,16,16,16,16,6,4)
16 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,Ø,16,16,16,16,16,16,16,6,4)
17 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,Ø,16,16,16,16,16,16,6,4)
18 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,Ø,16,16,16,16,16,6,4)
19 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,Ø,16,16,16,16,6,4)
20 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,Ø,16,16,16,6,4)
21 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,Ø,16,16,6,4)
22 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,Ø,16,6,4)
23 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,Ø,6,4)
24 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,Ø,4)

Configurações - Tubete 2

25 - (29,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
26 - (29,Ø,29,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
27 - (29,29,Ø,29,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
28 - (29,29,29,Ø,29,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
29 - (29,29,29,29,Ø,29,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
30 - (29,29,29,29,29,Ø,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
31 - (29,29,29,29,29,15,Ø,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
32 - (29,29,29,29,29,15,15,Ø,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
33 - (29,29,29,29,29,15,15,15,Ø,15,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
34 - (29,29,29,29,29,15,15,15,15,Ø,15,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
35 - (29,29,29,29,29,15,15,15,15,15,Ø,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
36 - (29,29,29,29,29,15,15,15,15,15,15,Ø,15,15,15,16,16,16,16,16,16,16,16,16,6,5)
37 - (29,29,29,29,29,15,15,15,15,15,15,15,Ø,15,15,16,16,16,16,16,16,16,16,16,6,5)
38 - (29,29,29,29,29,15,15,15,15,15,15,15,15,Ø,15,16,16,16,16,16,16,16,16,16,6,5)

39 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,Ø,16,16,16,16,16,16,16,16,16,16,6,5)
40 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,Ø,16,16,16,16,16,16,16,16,16,6,5)
41 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,Ø,16,16,16,16,16,16,16,16,6,5)
42 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,Ø,16,16,16,16,16,16,16,6,5)
43 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,Ø,16,16,16,16,16,16,6,5)
44 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,Ø,16,16,16,16,16,6,5)
45 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,Ø,16,16,16,16,6,5)
46 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,Ø,16,16,16,6,5)
47 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,Ø,16,16,6,5)
48 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,Ø,16,6,5)
49 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,16,Ø,6,5)
50 - (29,29,29,29,29,15,15,15,15,15,15,15,15,15,16,16,16,16,16,16,16,16,16,16,6,Ø,5)

Configurações - Tubete 3

51 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
52 - (9,Ø,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
53 - (9,9,Ø,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
54 - (9,9,9,Ø,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
55 - (9,9,9,9,Ø,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
56 - (9,9,9,9,10,Ø,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
57 - (9,9,9,9,10,10,Ø,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
58 - (9,9,9,9,10,10,10,Ø,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
59 - (9,9,9,9,10,10,10,10,Ø,11,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
60 - (9,9,9,9,10,10,10,10,11,Ø,11,11,11,15,15,15,15,16,16,16,16,16,6,4)
61 - (9,9,9,9,10,10,10,10,11,11,Ø,11,11,15,15,15,15,16,16,16,16,16,6,4)
62 - (9,9,9,9,10,10,10,10,11,11,11,Ø,11,15,15,15,15,16,16,16,16,16,6,4)
63 - (9,9,9,9,10,10,10,10,11,11,11,11,Ø,15,15,15,15,16,16,16,16,16,6,4)
64 - (9,9,9,9,10,10,10,10,11,11,11,11,15,Ø,15,15,15,16,16,16,16,16,6,4)
65 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,Ø,15,15,16,16,16,16,16,6,4)
66 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,Ø,15,16,16,16,16,16,6,4)
67 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,Ø,16,16,16,16,16,6,4)
68- (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,Ø,16,16,16,16,6,4)
69 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,Ø,16,16,16,6,4)
70 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,Ø,16,16,6,4)
71 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,Ø,16,6,4)
72 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,Ø,6,4)
73 - (9,9,9,9,10,10,10,10,11,11,11,11,15,15,15,15,16,16,16,16,16,6,Ø,4)

Configurações - Tubete 4

74 - (24,24,28,30,6,4)
75 - (24,Ø,24,28,30,6,4)
76 - (24,24,Ø,28,30,6,4)
77 - (24,24,28,Ø,30,6,4)
78 - (24,24,28,30,Ø,6,4)
79 - (24,24,28,30,6,Ø,4)

Configurações - Tubete 5

80 - (21,21,21,7,7,7,8,8,8,8,15,15,15,16,16,16,16,6,4)
81 - (21,Ø,21,21,7,7,7,8,8,8,8,15,15,15,16,16,16,16,6,4)

82 - (21,21,Ø,21,7,7,7,8,8,8,8,15,15,15,16,16,16,16,6,4)
83 - (21,21,21,Ø,7,7,7,8,8,8,8,15,15,15,16,16,16,16,6,4)
84 - (21,21,21,7,Ø,7,7,8,8,8,8,15,15,15,16,16,16,16,6,4)
85 - (21,21,21,7,7,Ø,7,8,8,8,8,15,15,15,16,16,16,16,6,4)
86 - (21,21,21,7,7,7,Ø,8,8,8,8,15,15,15,16,16,16,16,6,4)
87 - (21,21,21,7,7,7,8,Ø,8,8,8,15,15,15,16,16,16,16,6,4)
88 - (21,21,21,7,7,7,8,8,Ø,8,8,15,15,15,16,16,16,16,6,4)
89 - (21,21,21,7,7,7,8,8,8,Ø,8,15,15,15,16,16,16,16,6,4)
90 - (21,21,21,7,7,7,8,8,8,8,Ø,15,15,15,16,16,16,16,6,4)
91 - (21,21,21,7,7,7,8,8,8,8,15,Ø,15,15,16,16,16,16,6,4)
92 - (21,21,21,7,7,7,8,8,8,8,15,15,Ø,15,16,16,16,16,6,4)
93 - (21,21,21,7,7,7,8,8,8,8,15,15,15,Ø,16,16,16,16,6,4)
94 - (21,21,21,7,7,7,8,8,8,8,15,15,15,16,Ø,16,16,16,6,4)
95 - (21,21,21,7,7,7,8,8,8,8,15,15,15,16,16,Ø,16,16,6,4)
96 - (21,21,21,7,7,7,8,8,8,8,15,15,15,16,16,16,Ø,16,6,4)
97 - (21,21,21,7,7,7,8,8,8,8,15,15,15,16,16,16,16,Ø,6,4)
98 - (21,21,21,7,7,7,8,8,8,8,15,15,15,16,16,16,16,6,Ø,4)

Configurações -Tubete 6

99 - (29,29,29,29,29,15,15,15,15,15,30,30,30,30,30,6,4)
100 - (29,Ø,29,29,29,29,15,15,15,15,15,30,30,30,30,30,6,4)
101 - (29,29,Ø,29,29,29,15,15,15,15,15,30,30,30,30,30,6,4)
102 - (29,29,29,Ø,29,29,15,15,15,15,15,30,30,30,30,30,6,4)
103 - (29,29,29,29,Ø,29,15,15,15,15,15,30,30,30,30,30,6,4)
104 - (29,29,29,29,29,Ø,15,15,15,15,15,30,30,30,30,30,6,4)
105 - (29,29,29,29,29,15,Ø,15,15,15,15,30,30,30,30,30,6,4)
106 - (29,29,29,29,29,15,15,Ø,15,15,15,30,30,30,30,30,6,4)
107 - (29,29,29,29,29,15,15,15,Ø,15,15,30,30,30,30,30,6,4)
108 - (29,29,29,29,29,15,15,15,15,Ø,15,30,30,30,30,30,6,4)
109 - (29,29,29,29,29,15,15,15,15,15,Ø,30,30,30,30,30,6,4)
110 - (29,29,29,29,29,15,15,15,15,15,30,Ø,30,30,30,30,6,4)
111 - (29,29,29,29,29,15,15,15,15,15,30,30,Ø,30,30,30,6,4)
112 - (29,29,29,29,29,15,15,15,15,15,30,30,30,Ø,30,30,6,4)
113 - (29,29,29,29,29,15,15,15,15,15,30,30,30,30,Ø,30,6,4)
114 - (29,29,29,29,29,15,15,15,15,15,30,30,30,30,30,Ø,6,4)
115 - (29,29,29,29,29,15,15,15,15,15,30,30,30,30,30,6,Ø,4)

Configurações - Tubete 7

116 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
117 - (21,Ø,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
118 - (21,22,Ø,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
119 - (21,22,22,Ø,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
120 - (21,22,22,22,Ø,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
121 - (21,22,22,22,23,Ø,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
122 - (21,22,22,22,23,23,Ø,23,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
123 - (21,22,22,22,23,23,23,Ø,24,24,24,24,15,15,15,15,30,30,30,30,6,4)
124 - (21,22,22,22,23,23,23,24,Ø,24,24,24,15,15,15,15,30,30,30,30,6,4)
125 - (21,22,22,22,23,23,23,24,24,Ø,24,24,15,15,15,15,30,30,30,30,6,4)
126 - (21,22,22,22,23,23,23,24,24,24,Ø,24,15,15,15,15,30,30,30,30,6,4)
127 - (21,22,22,22,23,23,23,24,24,24,24,Ø,15,15,15,15,30,30,30,30,6,4)

128 - (21,22,22,22,23,23,23,24,24,24,24,15,Ø,15,15,15,30,30,30,30,6,4)
129 - (21,22,22,22,23,23,23,24,24,24,24,15,15,Ø,15,15,30,30,30,30,6,4)
130 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,Ø,15,30,30,30,30,6,4)
131 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,Ø,30,30,30,30,6,4)
132 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,Ø,30,30,30,6,4)
133 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,Ø,30,30,6,4)
134 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,Ø,30,6,4)
135 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,Ø,6,4)
136 - (21,22,22,22,23,23,23,24,24,24,24,15,15,15,15,30,30,30,30,6,Ø,4)

Configurações - Tubete 8

137 - (25,25,25,26,14)
138 - (25,Ø,25,25,26,14)
139 - (25,25,Ø,25,26,14)
140 - (25,25,25,Ø,26,14)
141 - (25,25,25,26,Ø,14)

Configurações - Tubete 9

142 - (19,19,20,20,27,27)
143 - (19,Ø,19,20,20,27,27)
144 - (19,19,Ø,20,20,27,27)
145 - (19,19,20,Ø,20,27,27)
146 - (19,19,20,20,Ø,27,27)
147 - (19,19,20,20,27,Ø,27)

Configurações - Tubete 10

148 - (17,17,17,18,18,1,19,20,20,27,27)
149 - (17,Ø,17,17,18,18,1,19,20,20,27,27)
150 - (17,17,Ø,17,18,18,1,19,20,20,27,27)
151 - (17,17,17,Ø,18,18,1,19,20,20,27,27)
152 - (17,17,17,18,Ø,18,1,19,20,20,27,27)
153 - (17,17,17,18,18,Ø,1,19,20,20,27,27)
154 - (17,17,17,18,18,1,Ø,19,20,20,27,27)
155 - (17,17,17,18,18,1,19,Ø,20,20,27,27)
156 - (17,17,17,18,18,1,19,20,Ø,20,27,27)
157 - (17,17,17,18,18,1,19,20,20,Ø,27,27)
158 - (17,17,17,18,18,1,19,20,20,27,Ø,27)

Configurações - Tubete 11

159 - (19,19,20,20,27,27)
160 - (19,Ø,19,20,20,27,27)
161 - (19,19,Ø,20,20,27,27)
162 - (19,19,20,Ø,20,27,27)
163 - (19,19,20,20,Ø,27,27)
164 - (19,19,20,20,27,Ø,27)

Configurações - Tubete 12

165 - (17,17,17,18,18,1,19,19,20,20,20,27,27)
166 - (17,Ø,17,17,18,18,1,19,19,20,20,20,27,27)
167 - (17,17,Ø,17,18,18,1,19,19,20,20,20,27,27)
168 - (17,17,17,Ø,18,18,1,19,19,20,20,20,27,27)
169 - (17,17,17,18,Ø,18,1,19,19,20,20,20,27,27)
170 - (17,17,17,18,18,Ø,1,19,19,20,20,20,27,27)
171 - (17,17,17,18,18,1,Ø,19,19,20,20,20,27,27)
172 - (17,17,17,18,18,1,19,Ø,19,20,20,20,27,27)
173 - (17,17,17,18,18,1,19,19,Ø,20,20,20,27,27)
174 - (17,17,17,18,18,1,19,19,20,Ø,20,20,27,27)
175 - (17,17,17,18,18,1,19,19,20,20,Ø,20,27,27)
176 - (17,17,17,18,18,1,19,19,20,20,20,Ø,27,27)
177 - (17,17,17,18,18,1,19,19,20,20,20,27,Ø,27)

Configurações - Tubete 13

178 - (9,9,9,9,10,10,10,10,24,24,24,29,29,29,30,30,30,6,4)
179 - (9,Ø,9,9,9,10,10,10,10,24,24,24,29,29,29,30,30,30,6,4)
180 - (9,9,Ø,9,9,10,10,10,10,24,24,24,29,29,29,30,30,30,6,4)
181 - (9,9,9,Ø,9,10,10,10,10,24,24,24,29,29,29,30,30,30,6,4)
182 - (9,9,9,9,Ø,10,10,10,10,24,24,24,29,29,29,30,30,30,6,4)
183 - (9,9,9,9,10,Ø,10,10,10,24,24,24,29,29,29,30,30,30,6,4)
184 - (9,9,9,9,10,10,Ø,10,10,24,24,24,29,29,29,30,30,30,6,4)
185 - (9,9,9,9,10,10,10,Ø,10,24,24,24,29,29,29,30,30,30,6,4)
186 - (9,9,9,9,10,10,10,10,Ø,24,24,24,29,29,29,30,30,30,6,4)
187 - (9,9,9,9,10,10,10,10,24,Ø,24,24,29,29,29,30,30,30,6,4)
188 - (9,9,9,9,10,10,10,10,24,24,Ø,24,29,29,29,30,30,30,6,4)
189 - (9,9,9,9,10,10,10,10,24,24,24,Ø,29,29,29,30,30,30,6,4)
190 - (9,9,9,9,10,10,10,10,24,24,24,29,Ø,29,29,30,30,30,6,4)
191 - (9,9,9,9,10,10,10,10,24,24,24,29,29,Ø,29,30,30,30,6,4)
192 - (9,9,9,9,10,10,10,10,24,24,24,29,29,29,Ø,30,30,30,6,4)
193 - (9,9,9,9,10,10,10,10,24,24,24,29,29,29,30,Ø,30,30,6,4)
194 - (9,9,9,9,10,10,10,10,24,24,24,29,29,29,30,30,Ø,30,6,4)
195 - (9,9,9,9,10,10,10,10,24,24,24,29,29,29,30,30,30,Ø,6,4)
196 - (9,9,9,9,10,10,10,10,24,24,24,29,29,29,30,30,30,6,Ø,4)

Configurações - Tubete 14

197 - (9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
198 - (9,Ø,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
199 - (9,9,Ø,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
200 - (9,9,9,Ø,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
201 - (9,9,9,10,Ø,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
202 - (9,9,9,10,10,Ø,10,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
203 - (9,9,9,10,10,10,Ø,10,2,11,11,11,12,12,12,12,13,13,13,6,4)
204 - (9,9,9,10,10,10,10,Ø,2,11,11,11,12,12,12,12,13,13,13,6,4)
205 - (9,9,9,10,10,10,10,2,Ø,11,11,11,12,12,12,12,13,13,13,6,4)
206 - (9,9,9,10,10,10,10,2,11,Ø,11,11,12,12,12,12,13,13,13,6,4)
207 - (9,9,9,10,10,10,10,2,11,11,Ø,11,12,12,12,12,13,13,13,6,4)
208 - (9,9,9,10,10,10,10,2,11,11,11,Ø,12,12,12,12,13,13,13,6,4)

209 - (9,9,9,10,10,10,10,2,11,11,11,12,Ø,12,12,12,13,13,13,6,4)
210 - (9,9,9,10,10,10,10,2,11,11,11,12,12,Ø,12,12,13,13,13,6,4)
211 - (9,9,9,10,10,10,10,2,11,11,11,12,12,12,Ø,12,13,13,13,6,4)
212 - (9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,Ø,13,13,13,6,4)
213 - (9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,Ø,13,13,6,4)
214 - (9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,Ø,13,6,4)
215 - (9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,Ø,6,4)
216 - (9,9,9,10,10,10,10,2,11,11,11,12,12,12,12,13,13,13,6,Ø,4)

Configurações - Tubete 15

217 - (10,10,10,10,11,11,11,12,12,12,13,13,30,6,31)
218 - (10,Ø,10,10,10,11,11,11,12,12,12,13,13,30,6,31)
219 - (10,10,Ø,10,10,11,11,11,12,12,12,13,13,30,6,31)
220 - (10,10,10,Ø,10,11,11,11,12,12,12,13,13,30,6,31)
221 - (10,10,10,10,Ø,11,11,11,12,12,12,13,13,30,6,31)
222 - (10,10,10,10,11,Ø,11,11,12,12,12,13,13,30,6,31)
223 - (10,10,10,10,11,11,Ø,11,12,12,12,13,13,30,6,31)
224 - (10,10,10,10,11,11,11,Ø,12,12,12,13,13,30,6,31)
225 - (10,10,10,10,11,11,11,12,Ø,12,12,13,13,30,6,31)
226 - (10,10,10,10,11,11,11,12,12,Ø,12,13,13,30,6,31)
227 - (10,10,10,10,11,11,11,12,12,12,Ø,13,13,30,6,31)
228 - (10,10,10,10,11,11,11,12,12,12,13,Ø,13,30,6,31)
229 - (10,10,10,10,11,11,11,12,12,12,13,13,Ø,30,6,31)
230 - (10,10,10,10,11,11,11,12,12,12,13,13,30,Ø,6,31)
231 - (10,10,10,10,11,11,11,12,12,12,13,13,30,6,Ø,31)

Configurações - Tubete 16

232 - (10,10,10,10,11,11,11,12,12,12,3,13,30,6,31)
233 - (10,Ø,10,10,10,11,11,11,12,12,12,3,13,30,6,31)
234 - (10,10,Ø,10,10,11,11,11,12,12,12,3,13,30,6,31)
235 - (10,10,10,Ø,10,11,11,11,12,12,12,3,13,30,6,31)
236 - (10,10,10,10,Ø,11,11,11,12,12,12,3,13,30,6,31)
237 - (10,10,10,10,11,Ø,11,11,12,12,12,3,13,30,6,31)
238 - (10,10,10,10,11,11,Ø,11,12,12,12,3,13,30,6,31)
239 - (10,10,10,10,11,11,11,Ø,12,12,12,3,13,30,6,31)
240 - (10,10,10,10,11,11,11,12,Ø,12,12,3,13,30,6,31)
241 - (10,10,10,10,11,11,11,12,12,Ø,12,3,13,30,6,31)
242 - (10,10,10,10,11,11,11,12,12,12,Ø,3,13,30,6,31)
243 - (10,10,10,10,11,11,11,12,12,12,3,Ø,13,30,6,31)
244 - (10,10,10,10,11,11,11,12,12,12,3,13,Ø,30,6,31)
245 - (10,10,10,10,11,11,11,12,12,12,3,13,30,Ø,6,31)
246 - (10,10,10,10,11,11,11,12,12,12,3,13,30,6,Ø,31)

Bibliografia

- D. APPLEGATE, R. BIXBY, V. CHVATAL e W. COOK. ***Implementing the Dantzig – Fulkerson – Johnson algorithm for large traveling salesman problems*****,** *Mathematical Programming*, Ser. B, 97, 91-153, 2003.

- E. BALAS. ***The Prize Collecting Traveling Salesman Problem****. Networks*, 19, 621-636, 1989.

- E. BALAS. **The *Prize Collecting Traveling Salesman Problem: II. Polyhedral Results*.** *Working Paper*, Carnegie Mellon University, 1993.

- D. M. CARDOSO. **Teoria dos Grafos e Aplicações.** Departamento de Matemática da Faculdade de Aveiro, Mestrado em Matemática, 2004/2005.

- *DASH OPTIMIZATION*. ***Xpress-Mosel User Guide***, 2008.

- DASH OPTIMIZATION, ***Applications Of Optimization With XpressMP***, Tradução para o inglês *de Programmation Linéaire* de C. Guéret, C. Prins E M. Sevaux, Dash Optimization Ltda, 2000.

- V. DIMITRIJEVIC *and* Z. SARIC. ***Efficient Transformation of the Generalized Traveling Salesman Problem into the Traveling Salesman Problem on Digraphs***, *Information Sci*., 102, 65-110, 1997.

- M.C. GOLDBARG e H.P.L. LUNA. **Otimização Combinatória e Programação Linear**, Editora Campus, 2000.

- A.L. HENRY-LABORDERE. ***The record balancing problem: a dynamic programming solution of a generalized traveling salesman problem***, Revue Francaise D Informatique DeRecherche Operationnelle 3 (NB2) 43-49, 1969.

- G. LAPORTE, H. MERCURE, Y. NOBERT. ***Generalized traveling salesman problem through n sets of nodes: The asymmetric case***, *Discrete Applied Mathematics*, 18, 185-197, 1987.

- G. LAPORTE, Y. NOBERT (1983). ***Generalized traveling salesman problem through n sets of nodes: An integer programming approach***, INFOR 21 (1), 61-75, 1983.

- E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN, *and* D.B SHMOYS. ***The traveling salesman problem***, *John Wiley and Sons, Chichester*, 1985.

- E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN, *and* D.B SHMOYS. ***The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization Problems***, *Wiley*, 1990.

- Y. LIEN, E. MA and B. W. WAH. ***Transformation of the Generalized Traveling Salesman Problem into the Standard Traveling Salesman Problem*****,** *Information Sci.* 74, 177-189, 1993.

- C.E. NOON. ***The Generalized Traveling Salesman Problem. Unpublished Dissertation*****,** *Department of Industrial and Operations Research, University of Tennessee*, 1988.

- C. E. NOON, J. C. BEAN, ***A Lagrangian Based Approach for the Asymmetric Generalized Traveling Salesman Problem****, Operations Research* 39, 623-632, 1991.

- S. RANGEL, **Notas em Matemática Aplicada - Introdução à Construção de Modelos de Otimização Linear e Inteira** - Sociedade Brasileira de Matemática Aplicada e Computacional, São Carlos - SP, Brasil, 2005.

- S.S. SRIVASTAVA, S. KUMAR, R.C. GARG, P. SEN. ***Generalized traveling salesman problem through n sets of nodes***, *CORS J.* 7 , 1969.

- J.L. SZWARCFTER. **Grafos e algoritmos computacionais**, Ed. Campos, 1988.

- C.S. TANG C.S. & E.V. DENARDO. ***Models Arising from a Flexible Manufacturing Machine, Part I: Minimization of the Number of Tool Switches.*** *Operations Research* Vol.36, p.767-777, 1988.

- H. H. YANASSE *and* M. J. P. LAMOSA. ***On solving the minimization of tool switches problem using graphs*****,** XII ICIEOM, Fortaleza, CE, 2006.