OTIMIZAÇÃO DO PROCESSO DE DISTRIBUIÇÃO DE LOCOMOTIVAS

Cristovão de Almeida Barra

MONOGRAFIA SUBMETIDA À COORDENAÇÃO DE CURSO DE ENGENHARIA DE PRODUÇÃO DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO.

Aprovada por:

____

Prof. Fernando Marques de Almeida Nogueira, D.Sc.

____

Eng. Marcelo Neder Machado

____

Prof. José Geraldo Ferreira, M.Sc.

JUIZ DE FORA, MG - BRASIL
NOVEMBRO DE 2008

BARRA, CRISTOVÃO DE ALMEIDA

Otimização do Processo de Distribuição de Locomotivas [Minas Gerais] 2008

IX, 43p. 29,7 cm (EPD/UFJF, Engenharia de Produção, 2008)

Monografia - Universidade Federal de Juiz de Fora, Departamento de Engenharia de Produção

1. Otimização de Processos
2. Alocação de Locomotivas

I. EPD/UFJF II. Título (série)

Agradecimentos

Agradeço ao professor Fernando, por todo o auxílio e orientação. Agradeço também a todos os amigos da MRS, em especial ao Marcelo Neder, grande motivador e parceiro no desenvolvimento do trabalho. E por último, um agradecimento a todos que de alguma forma ajudaram meu desenvolvimento pessoal e profissional durante o período acadêmico.

Resumo da monografia apresentada à Coordenação de Curso de Engenharia de Produção como parte dos requisitos necessários para a graduação em Engenharia de Produção.

OTIMIZAÇÃO DO PROCESSO DE DISTRIBUIÇÃO DE LOCOMOTIVAS

Cristovão de Almeida Barra

Novembro/ 2008

Orientador: Fernando Marques de Almeida Nogueira
Co-Orientador: Marcelo Neder Machado

Curso: Engenharia de Produção

Atualmente, a máxima utilização e aproveitamento dos recursos deixou de ser um diferencial empresarial e passou a ser uma exigência para que uma empresa seja competitiva diante o mercado. As empresas de transporte ferroviário estão inseridas neste contexto e, desta forma, a otimização da alocação e programação de locomotivas torna-se de fundamental importância para esse tipo de empresa, afinal a locomotiva é um dos seus recursos mais caros. O presente trabalho tem o objetivo de estudar o problema de programação de locomotivas, mostrando todas as variáveis de decisão e restrições impostas, e estudar uma modelagem teórica que faça a otimização dessa distribuição. Essa modelagem terá uma abordagem ideal, não sendo o objetivo considerar as implicações de sua resolução e implementação real. Ao final será elaborado um solver para resolução desse problema, mostrando que a modelagem proposta é capaz de resolver desde problemas simples até problemas complexos. O trabalho mostra que para a resolução de problema de programação de locomotivas de forma otimizada, a alternativa mais adequada é a utilização das ferramentas de Pesquisa Operacional.

Palavras-chaves: Locomotivas; Otimização; Pesquisa Operacional; Transporte Ferroviário

Abstract of Graduation Final Project presented to Production Engineering Department as a partial fulfillment of the requirements for the degree of Bachelor in Production Engineering

LOCOMOTIVE PROGRAMMING AND ASSIGNMENT OPTIMIZATION

Cristovão de Almeida Barra

November/ 2008

Advisor: Fernando Marques de Almeida Nogueira
Co- Advisor: Marcelo Neder Machado

Department: Production Engineering

Nowadays, increasing resource utilization and operations efficiency are no more considered a competitive advantage but an essential necessity to survive in a globalized market. In this context, rail transportation companies attempt to optimize one of their most expensive process: locomotive programming and assignment. This paper aims to study locomotive programming problem, presenting its decision variables and constraints and also aims to review some theoretical and mathematical models for optimizing this problem. Models presented here were formulated for general purpose only, not being considered particular constraints focused on practical implementation for a specific company. In the end a solver will be elaborated to crack this problem, showing that the proposed modeling is capable of solving both simpler and more complex problems. The study demonstrates that operations research tools are the most adequate options to answer locomotive programming and assignment problems in an optimized way.

Key-Words: Locomotive; Optimization; Operations Research; Rail Transportation

SUMÁRIO

CAPÍTULO I - INTRODUÇÃO 1

1.1. CONSIDERAÇÕES INICIAIS 1
1.2. OBJETIVOS 1
1.3. JUSTIFICATIVAS 1
1.4. ESCOPO DO TRABALHO 2
1.5. METODOLOGIA 2

CAPÍTULO II - REVISÃO BIBLIOGRÁFICA 3

2.1. PESQUISA OPERACIONAL 3
2.1.1. PROGRAMAÇÃO LINEAR (PL) 4
2.1.2. PROBLEMA DE TRANSPORTES (REDES) 5
2.1.3. PROGRAMAÇÃO INTEIRA 6
2.1.4. ALGORITMOS EXATOS X ALGORITMOS HEURÍSTICOS 7
2.2. PROGRAMAÇÃO DE LOCOMOTIVAS 9
2.2.1. DADOS DOS TRENS E DAS LOCOMOTIVAS 10
2.2.2. REDE ESPAÇO-TEMPO 11
2.2.3. VARIÁVEIS DE DECISÃO 13
2.2.4. FORMULAÇÃO MATEMÁTICA DO MODELO 14

CAPÍTULO III - DESCRIÇÃO 16

3.1. TRANSPORTE FERROVIÁRIO DE CARGA 16
3.1.1. MRS LOGÍSTICA 18
3.2. O PROBLEMA: PROGRAMAÇÃO DE LOCOMOTIVAS 20

CAPÍTULO IV - SOLUÇÃO 24

4.1. INTRODUÇÃO 24
4.2. DETALHAMENTO DO ALGORITMO 26
4.3. APLICAÇÃO EM ESCALA REAL 31

CAPÍTULO IV – CONSIDERAÇÕES FINAIS..32

REFERÊNCIAS BIBLIOGRÁFICAS 33

APÊNCICE 1 – ELABORAÇÃO DO ALGORITMO..35

APÊNCICE 2 – DADOS DE ENTRADA DO PRIMEIRO EXEMPLO....................................38

APÊNCICE 3 – RELATÓRIO DE SAÍDA DO PRIMEIRO EXEMPLO 39

APÊNCICE 4 – DADOS DE ENTRADA DO EXEMPLO EM ESCALA REAL 40

APÊNCICE 5 – RELATÓRIO DE SAÍDA DO EXEMPLO EM ESCALA REAL 43

ÍNDICE DE FIGURAS

Figura 1 – Problema de Transportes 6
Figura 2 - Exemplo de parte de uma rede espaço-tempo 13
Figura 3 - Percentual de cargas transportadas no Brasil 17
Figura 4 - Mapa do Sistema Ferroviário Nacional..18
Figura 5 - Mapa da malha da MRS Logística...19
Figura 6 - Exemplo do problema de programação de locomotivas 22
Figura 7 – Visualização do exemplo...26
Figura 8 – Representação Esquemática do exemplo 28
Figura 9 – Representação Esquemática do resultado 30

ÍNDICE DE TABELAS

Tabela 1 - Características dos modelos de programação de locomotivas 9
Tabela 2 – Resumo dos trens..27
Tabela 3 – Resumo das Locomotivas...27
Tabela 4 – Matriz Origem x Destinos – Custos de deslocamento...27
Tabela 5 – Matriz Cij 28

CAPÍTULO I - INTRODUÇÃO

1.1. CONSIDERAÇÕES INICIAIS

A capacidade de se adequar às mudanças, sempre atendendo o cliente e otimizando cada vez mais seus recursos, é um fator que pode determinar o sucesso de uma empresa. Sendo assim, o processo de programação e controle de uma empresa é peça fundamental para o perfeito andamento das atividades. Deste modo, cada vez mais a questão da otimização da programação vem sendo levantada e trabalhada pela maioria das empresas.

Aliado a esse cenário e a crescente demanda por transporte ferroviário, principalmente depois da privatização do setor em 1996, a questão que envolve a programação adequada de recursos se tornou não mais um diferencial, mas sim um requisito de produção em uma empresa de transporte.

A programação de locomotivas está entre as mais importantes atividades do processo de planejamento e programação da operação do transporte ferroviário de carga; uma programação adequada, além de ser economicamente vantajosa, gera uma melhor utilização dos recursos da empresa.

A programação de locomotivas consiste em alocar de maneira ótima um conjunto de locomotivas aos trens programados, que necessitam de tração em um dado horizonte de tempo.

A atividade de programação de locomotivas envolve a solução para as seguintes questões:

- Quais locomotivas devem ser alocadas para tracionar quais trens?
- Quais locomotivas devem ser alocadas a quais trens para serem rebocadas?
- Quais locomotivas devem viajar sem carga?

1.2. OBJETIVOS

O objetivo do trabalho é apresentar o problema de programação de locomotivas em uma empresa ferroviária, mostrando a existência de modelos matemáticos teóricos para a resolução do problema, e ao final propor e elaborar um modelo de forma simplificada, demonstrando a eficácia do mesmo.

1.3. JUSTIFICATIVAS

Sendo as locomotivas um dos recursos mais caros de uma empresa de transporte ferroviário, é de extrema importância o conhecimento das variáveis que envolvem o processo de programação desse recurso, o perfeito entendimento dessas informações é de grande importância e utilidade para a área operacional da empresa.

Outra fonte motivadora é a literatura nacional limitada. Mesmo sendo um assunto de extrema importância para uma empresa ferroviária, as fontes de pesquisa, principalmente nacionais, são bastante raras.

1.4. ESCOPO DO TRABALHO

Este trabalho envolve o estudo teórico de modelos matemáticos aplicados à programação de locomotivas. Posteriormente, serão analisados dados reais para entendimento do problema e aplicação prática do modelo, dados estes que serão coletados na MRS Logística, concessionária da malha ferroviária e uma das maiores transportadoras do Brasil.

1.5. METODOLOGIA

O início deste trabalho envolve o estudo detalhado do tema para fundamentação teórica. Sendo assim, tal estudo tem como fontes os livros, publicações, teses e quaisquer outros tipos de materiais disponíveis para consulta sobre este assunto. Vale ressaltar que esta é uma etapa de extrema importância, uma vez que será a base para todo o desenvolvimento do projeto.

A próxima fase é de levantamento de todas as variáveis e restrições do processo. Este é um trabalho que deve ser feito com bastante atenção, pois seus resultados serão as entradas para o processo de modelagem do processo.

Com todo esse processo mapeado e entendido, a próxima etapa será o estudo e elaboração do modelo matemático; esta é a principal fase do trabalho, afinal é nela que será colocado em prática todo o trabalho realizado anteriormente; é nesta que será feito um modelo, tornando capaz de ser rodado com resultados eficazes.

A última etapa é a análise dos resultados gerados no modelo elaborado, das quais serão tiradas as últimas conclusões.

CAPÍTULO II - REVISÃO BIBLIOGRÁFICA

2.1. PESQUISA OPERACIONAL

"A Pesquisa Operacional é uma ciência aplicada, voltada para a resolução de problemas reais. Tendo como foco a tomada de decisões, aplica conceitos e métodos de outras áreas científicas para concepção, planejamento ou operação de sistemas para atingir seus objetivos.

Através de desenvolvimentos de base quantitativa, a Pesquisa Operacional visa também introduzir elementos de objetividade e racionalidade nos processos de tomada de decisão, sem descuidar, no entanto dos elementos subjetivos e de enquadramento organizacional que caracterizam os problemas".

(http://www.sobrapo.org.br/sitesobrapo.htm).

Durante a Segunda Guerra Mundial, um grupo de cientistas foi convocado na Inglaterra para estudar problemas de estratégia e de tática associados com a defesa do país. O objetivo era decidir sobre a utilização mais eficaz de recursos militares limitados. A convocação deste grupo marcou a primeira atividade formal de Pesquisa Operacional. Os resultados positivos conseguidos pela equipe de pesquisa operacional inglesa motivaram os Estados Unidos a iniciarem atividades semelhantes. Apesar de ser creditada à Inglaterra a origem da Pesquisa Operacional, sua propagação deve-se principalmente à equipe de cientistas liderada por George B. Dantzig, dos Estados Unidos, convocada durante a Segunda Guerra Mundial. Ao resultado deste esforço de pesquisa, concluído em 1947, deu-se o nome de Método Simplex. (LISBOA, 2002).

Com o fim da guerra, a utilização de técnicas de pesquisa operacional atraiu o interesse de diversas outras áreas. A natureza dos problemas encontrados é bastante abrangente e complexa, exigindo portanto, uma abordagem que permita reconhecer os múltiplos aspectos envolvidos. Uma característica importante da pesquisa operacional e que facilita o processo de análise e de decisão é a utilização de modelos. Eles permitem a experimentação da solução proposta, isto significa que uma decisão pode ser mais bem avaliada e testada antes de ser efetivamente implementada. (LISBOA, 2002)

Um modelo é uma representação de um sistema real, pode já existir ou ser um projeto aguardando execução. No primeiro caso, o modelo pretende reproduzir o funcionamento do sistema, de modo a aumentar sua produtividade. No segundo caso, o modelo é utilizado para definir a estrutura ideal do sistema.

A confiabilidade da solução obtida através do modelo depende da validação do modelo na representação do sistema real. A validação do modelo é a confirmação de que ele realmente representa o sistema real. A diferença entre a solução real e a solução proposta pelo modelo depende diretamente da precisão do modelo em descrever o comportamento original do sistema. (LISBOA, 2002)

Um problema simples pode ser representado por modelos também simples e de fácil solução. Já problemas mais complexos requerem modelos mais elaborados, cuja solução pode vir a ser bastante complicada.

Em um modelo matemático, são incluídos três conjuntos principais de elementos:

- Variáveis de decisão e parâmetros: variáveis de decisão são as incógnitas a serem determinadas pela solução do modelo. Parâmetros são valores fixos no problema;
- Restrições: de modo a levar em conta as limitações físicas do sistema, o modelo deve incluir restrições que limitam as variáveis de decisão a seus valores possíveis (ou viáveis);
- Função objetivo: é uma função matemática que define a qualidade da solução em função das variáveis de decisão.

2.1.1. PROGRAMAÇÃO LINEAR (PL)

A Programação Linear (PL) é uma técnica de planejamento que vem se constituindo como uma das mais poderosas em quase todo ramo da atividade humana. Seus benefícios são exatamente aqueles procurados por qualquer empresa: diminuição dos custos e aumento dos lucros. Em algumas organizações ela está, inclusive, embutida em suas rotinas informatizadas de planejamento diário dos processos de operação.
(INDG - http://www.indg.com.br/po/definicao.asp)

Um modelo de Programação Linear é um modelo matemático de otimização no qual todas as funções são lineares (tanto a função objetivo, quanto as restrições). Para que um modelo de um determinado sistema possa ser representado por meio de PL, ele deve possuir as seguintes características:

- Proporcionalidade: a quantidade de recurso consumido por uma dada atividade deve ser proporcional ao nível dessa atividade na solução final do problema. Além disso, o custo de cada atividade é proporcional ao nível de operação da atividade;
- Não Negatividade: deve ser sempre possível desenvolver dada atividade em qualquer nível não negativo e qualquer proporção de um dado recurso deve sempre poder ser utilizado;
- Aditividade: o custo total é a soma das parcelas associadas a cada atividade;
- Separabilidade: pode-se identificar de forma separada o custo (ou consumo de recursos) específico das operações de cada atividade.

Segundo Lisboa (2002), podemos definir a programação linear como sendo o planejamento de atividades para obter um resultado ótimo, isto é, um resultado que atenda, da melhor forma possível, a um determinado objetivo. Embora alocação de recursos para atividades seja o tipo mais comum, a programação linear tem numerosos outros tipos de aplicação; assim, na prática, qualquer problema cujo modelo matemático se enquadre na forma geral de um modelo de PL, é um problema de programação linear. Um procedimento extremamente eficiente, chamado método simplex, está disponível para resolver problemas de PL, mesmo àqueles com milhares de variáveis. O objetivo da PL é encontrar o lucro máximo ou o custo mínimo em situações reais.

Mas qual é a magnitude deste benefício dentro das empresas? Segundo pesquisas efetuadas em empresas que têm utilizado esta ferramenta, a redução de custos se enquadra facilmente na faixa entre 1% e 5%, existindo casos que chegam até a 15%.
(INDG - http://www.indg.com.br/po/definicao.asp).

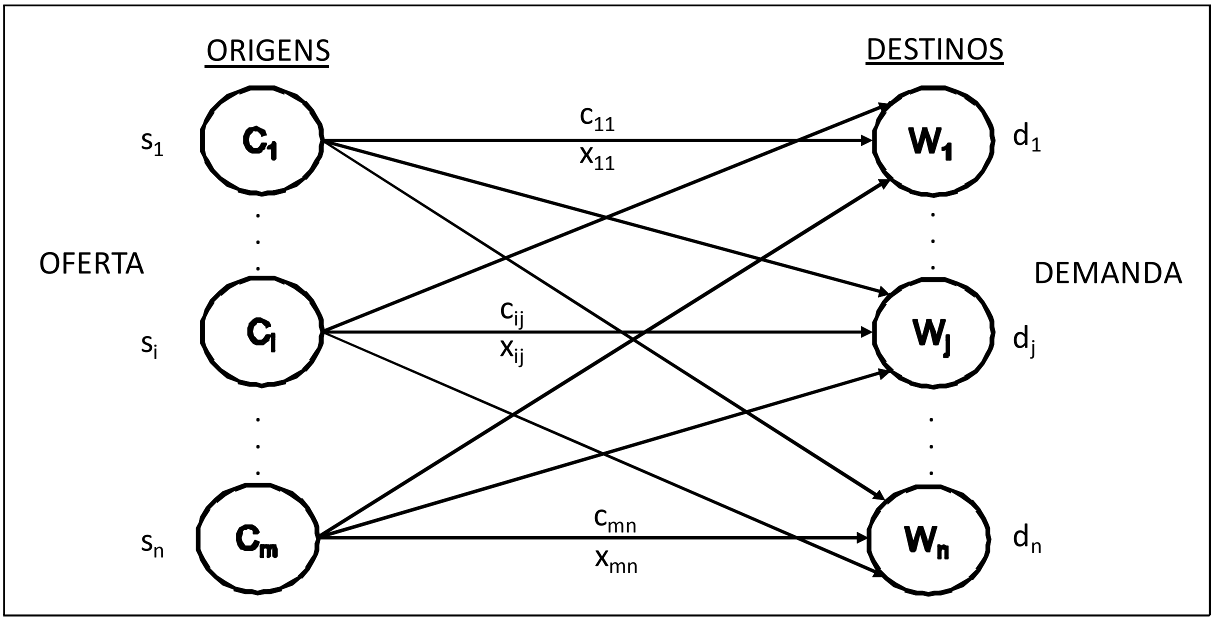
2.1.2. PROBLEMA DE TRANSPORTES (REDES)

Segundo Lins et al (2006), o Problema de Transporte é talvez o mais representativo dos Problemas de Programação Linear. É um problema de grande aplicação prática, tendo sido estudado por vários investigadores, embora tenha sido Dantzig o primeiro a estabelecer a sua formulação em PL e a propor um método sistemático de resolução.

O problema geral de transporte consiste em determinar a forma mais eficiente, isto é, mais econômica de enviar um bem disponível em quantidades limitadas em determinados locais para outros locais onde é necessário. Como qualquer problema de PL, este pode ser resolvido pelo método do simplex. Porém a sua estrutura particular permitiu a utilização de métodos que embora derivados do simplex, são bastante mais eficientes.

A figura seguinte ilustra o problema de transporte sob a forma de uma rede com *m* origens e *n* destinos representados por nós; os arcos que ligam as origens aos destinos representam os percursos através dos quais o produto pode ser transportado.

Figura 1 – Problema de Transportes



Fonte: LINS et al (2006). Modificado pelo autor

O modelo generalizado do problema é representado por:

$$\min Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \qquad (1)$$

sujeito a:

$$\sum_{j=1}^{n} x_{ij} = s_i \qquad \text{oferta} \qquad (2)$$

$$\sum_{i=1}^{m} x_{ij} = d_j \qquad \text{demanda} \qquad (3)$$

$$x_{ij} \geq 0 \qquad (i = 1,....,m\,;\, j = 1,...,n) \qquad (4)$$

2.1.3. PROGRAMAÇÃO INTEIRA (PI)

A Programação Inteira pode ser entendida como um caso específico da Programação Linear. Se as variáveis do problema pertencerem ao conjunto dos números inteiros (ou ao menos, parte destas variáveis), temos uma subclasse da Programação Linear chamada Programação Inteira (PI) ou programação Linear Inteira.

Tipos básicos:

- Programação Inteira Total - onde todas as variáveis de decisão são do tipo inteiro;

- Programação Inteira Mista - onde apenas uma parte das variáveis são do tipo inteiro, enquanto outras são do tipo real.

Uma abordagem freqüente ao problema PI consiste em resolver o problema PL associado, que resulta de relaxar as variáveis inteiras e depois arredondar as soluções não inteiras obtidas. No entanto, dificuldades podem surgir:

- Após arredondamento a solução pode ao ser admissível para o problema;
- Falta de garantia de que uma solução (conveniente) arredondada seja ótima para o problema. Aliás pode estar bem longe do ótimo.
- Portanto, seria útil termos um procedimento de solução eficaz para termos uma solução ótima para problemas de programação linear inteira.

Um problema de Programação Inteira limitado tem um número finito de soluções admissíveis, pelo que seria de se admitir um processo enumerativo para a determinação de uma solução ótima. No entanto, esse número pode facilmente tornar-se demasiado grande, tornando inviável a utilização de computadores, mesmo muito rápidos. Há, portanto toda a conveniência no recurso a algum método que apenas examine uma parcela do conjunto das soluções admissíveis.

Problemas de Programação Inteira (conjunto solução discreto) são geralmente muito mais difíceis de serem resolvidos quando comparados aos Problemas de Programação Linear ordinários (conjunto solução contínuo).

Um considerável número de algoritmos foi desenvolvido para este propósito, no entanto, estão limitados a problemas relativamente pequenos, tendo talvez poucas dúzias de variáveis. Alguns algoritmos aplicados com sucesso na PI são:

- branch and bound;
- branch and cut;
- branch and price.

2.1.4. ALGORITMOS EXATOS X ALGORITMOS HEURÍSTICOS

"O fato de diversos problemas de otimização possuírem alta complexidade computacional fortemente sugere que não existam algoritmos eficientes para resolverem esse problema de forma exata, isto é, provando a otimalidade, e por outro lado, se algoritmos exatos são utilizados somente problemas de pequeno e médio porte podem normalmente ser resolvidos.

O sucesso ou não desses algoritmos exatos está, entretanto, fortemente relacionado com a existência de uma estrutura especial do problema da qual o algoritmo possa tirar proveito. O limite de quão grande o problema pode ser resolvido de forma exata continua atrelado a natureza exponencial desses algoritmos e a teoria da complexidade computacional.

A experiência mostra que para alguns problemas, os algoritmos exatos realizam nada mais que a enumeração completa de todas as possíveis soluções, e como conseqüência, um limite de tempo tem de ser imposto para terminá-los antes de alcançar a otimalidade, ou às vezes, até mesmo uma solução viável. Além disso, muitas dessas abordagens exatas, tais como árvore de busca "*branch-and-bound*" e programação dinâmica, necessitam de uma quantidade crescente de memória a medida que o algoritmo progride. Por isso, em muitos casos, as buscas são abortadas quando o limite de memória dos computadores é atingido.

Para esses casos de maior complexidade existe o algoritmo heurístico, que é um método projetado para fornecer "boas" soluções, mas que não podem garantir otimalidade (nem viabilidade, em determinados casos). Sua intenção é resolver aproximadamente e eficientemente problemas de otimização grandes e/ou difíceis para os quais soluções não podem ser encontradas sem uma grande quantidade de tempo e memória computacional com os métodos exatos existentes.

Dentre as técnicas heurísticas tradicionais ou mais comumente utilizadas pode-se citar:

- Heurísticas gulosas ("*greedy heuristics*") - geram soluções pela consideração (inclusão e/ou exclusão) de componentes (por exemplo, variáveis), um de cada vez, até que uma solução viável seja encontrada. Algoritmos gulosos buscam ganhos imediatos a cada passo e geralmente uma solução viável é encontrada no final do procedimento. As heurísticas gulosas são quase sempre as de melhor desempenho computacional, todavia as soluções obtidas podem estar muito longe da ótima;
- Métodos de busca em vizinhanças - uma heurística de busca nas vizinhanças, ou busca local, começa com uma solução viável que é melhorada sucessivamente por uma série de trocas ou fusões, em uma busca local. A busca se dá movendo-se de uma solução viável para outra ou, para as suas vizinhanças, de forma a melhorar o resultado da função objetivo, mantendo-se evidentemente a viabilidade. Se nenhuma solução melhor é encontrada então um ponto de ótimo local é obtido. Muitos métodos de busca nas vizinhanças combinam uma heurística gulosa, para obter uma solução de partida, com uma busca local para encontrar um ótimo local;

- Relaxações matemáticas - nestes métodos, algumas restrições do problema original são relaxadas para conduzir a um problema solúvel através de técnicas de relaxação (tais como relaxação de programação linear, relaxação Lagrangeana, etc.). Uma solução para o problema relaxado pode ser então obtida usando-se técnicas de programação matemática. O resultado após o passo da relaxação inicial é geralmente não viável. "Um segundo passo então é necessário para se obter a viabilidade." (TORRACA, 2004)

2.2. PROGRAMAÇÃO DE LOCOMOTIVAS

Segundo Cordeau et al (1998), dado uma programação de trens, o problema de distribuição de locomotivas consiste em atribuir um conjunto de locomotivas para os trens programados, satisfazendo exigências pré-fixadas como o número de locomotivas necessárias e a potência necessária para tracionar o trem.

Cordeau et al (1998) cita também que muitas empresas ferroviárias no mundo desenvolveram sistemas de apoio à decisão para auxiliar os planejadores na programação das locomotivas. Estes sistemas basearam-se em grande parte, em técnicas de simulação e as regras ditadas pela experiência na tomada de decisão, mas foi com os métodos de otimização que se chegou aos melhores resultados.

Diante da importância deste tema, muitos trabalhos foram desenvolvidos, apresentando o problema e propondo soluções para a otimização do mesmo. Em Cordeau et al (1998) é apresentado uma tabela com os principais trabalhos desenvolvidos sobre o tema.

Tabela 1 - Características dos modelos de programação de locomotivas

Autores	Tipo de problema	Nível de Planejamento	Função Objetivo	Estrutura do modelo	Método de Solução
Forbes et al. (1991)	Apenas uma locomotiva por trem	Tático	Minimizar custos da operação	Problema de Alocação	Branch-and-bound
Fischetti and Toth (1997)	Apenas uma locomotiva por trem	Tático	Minimizar frota e locomotivas rebocadas	Problema de Alocação	Relaxação Lagrangiana
Florian et al. (1976)	Mais de uma locomotiva por trem	Estratégico	Minimizar investimento e manutenção	Multifluxo	Decomposição de Benders
Smith and Sheffi (1988)	Mais de uma locomotiva por trem	Estratégico	Minimizar custos da operação	Multifluxo	Heurística
Chih et al. (1990)	Mais de uma locomotiva por trem	Operacional	Maximizar lucro esperado	Multifluxo	Heurística de Decomposição
Ziarati et al. (1997)	Mais de uma locomotiva por trem	Operacional	Minimizar custos da operação	Multifluxo	Decomposição de Dantzig-Wolfe
Nõu et al. (1997)	Mais de uma locomotiva por trem	Tático	Minimizar custos da operação	Multifluxo	Decomposição de Dantzig-Wolfe
Ziarati et al. (1997)	Mais de uma locomotiva por trem	Operacional	Minimizar atrasos	Multifluxo	Decomposição de Dantzig-Wolfe
Cordeau et al. (1998)	Locomotivas e vagões	Tático	Minimizar custos da operação	Multifluxo	Decomposição de Benders

Fonte: CORDEAU et al (1998). Modificado pelo autor

Neste capítulo será apresentado um modelo para solução do problema de programação de locomotivas; tal modelo tem caráter apenas de referência e exemplificação do problema, não sendo aplicado para a solução do problema.

O modelo escolhido é proposto por AHUJA et al. (2005), ele adota a premissa de que os trens representam circulação regular diária; atrasos (variações nos tempos de circulação), abastecimentos e paradas para manutenção não são consideradas.

Antes de apresentar o modelo de otimização é necessário definir três tipos de estado da locomotiva em um trem:

- Locomotiva ativa: locomotiva ligada e que esteja acoplada e tracionando um trem;
- Locomotiva rebocada: locomotivas que ao invés de estarem tracionando o trem, estão sendo rebocadas com vagões. Isto permite que locomotivas extras possam ser levadas de locais onde se encontram em excesso para locais onde são demandadas.
- Locomotiva escoteira: locomotivas agrupadas entre si, formando um trem sem vagões. Esta formação geralmente apresenta velocidades maiores, contudo seu custo também é maior; afinal, além de ser necessário um maquinista para guiá-las, também não apresentam receita.

O objetivo deste modelo é a redução nos custos gerais envolvidos no processo, fazendo uma programação ótima de quais locomotivas estarão tracionando algum trem, quais não serão utilizadas, quais viajarão escoteiras e quais viajarão rebocadas.

Segundo Ahuja et al (2005), as variáveis utilizadas no modelo podem ser divididas em 3 categorias: Dados dos trens e das locomotivas; rede espaço-tempo; e variáveis de decisão.

2.2.1. DADOS DOS TRENS E DAS LOCOMOTIVAS

São considerados dados de entrada, as rotas e tempos de circulação dos trens, esses dados devem ser pré-estabelecidos e já estarem fixados. O termo "*l* "representa um trem especifico e "*L"* é o conjunto de todos os trens. Segundo Ahuja et al (2005) considera-se que um mesmo trem circulando diferentes dias da semana, seja representado por diferentes trens do conjunto *L*, um para cada dia e que para cada $l \in L$ são conhecidos sua origem e destino, assim como os tempos de partida e chegada.

- T_l: peso necessário para tracionar o trem *l*;

- β_l : potência por tonelada necessária para tracionar o trem *l*;
- H_l *:* potência exigida pelo trem *l*, que é definida como $H_l = T_l \beta_l$ *;*
- E_l : penalidade por alocar apenas uma única locomotiva ao trem.

Uma empresa ferroviária normalmente tem vários tipos diferentes de locomotivas, que apresentam diferentes características e número de eixos. Será denotado por "*K"* o conjunto de todos os tipos de locomotiva, e por "*k"* um determinado tipo de locomotiva. Serão associados os seguintes dados em cada trem $k \in K$:

- h^k *:* potência fornecida por uma locomotiva do tipo *k.;*
- a^k *: número* de eixos de uma locomotiva do tipo *k.;*
- G^k *:* custo semanal de possuir uma locomotiva do tipo *k;*
- B^k : tamanho da frota de locomotivas do tipo *k.*

Temos, ainda, os seguintes dados para as combinações de locomotivas nos trens:

- c^k_l : custo incorrido em atribuir um papel ativo para a locomotiva do tipo k em um trem *l*;
- d^k_l : custo incorrido em atribuir um papel de reboque para a locomotiva do tipo k em um trem *l*;
- t^k_l *:* capacidade de tração que cada locomotiva *k* faz para tracionar um trem *l*.

2.2.2. REDE ESPAÇO-TEMPO

O modelo de programação de locomotivas é formulado como um problema de fluxo em rede multiproduto de variáveis inteiras com restrições adicionais. Esta formulação de programação inteira mista representa o fluxo de locomotivas em uma rede espaço/tempo. Nesta rede, os arcos representam os trens e os nós representam os eventos (que podem ser as chegadas e partidas dos trens e locomotivas), e cada diferente tipo de locomotiva representa um produto. (MACHADO 2006).

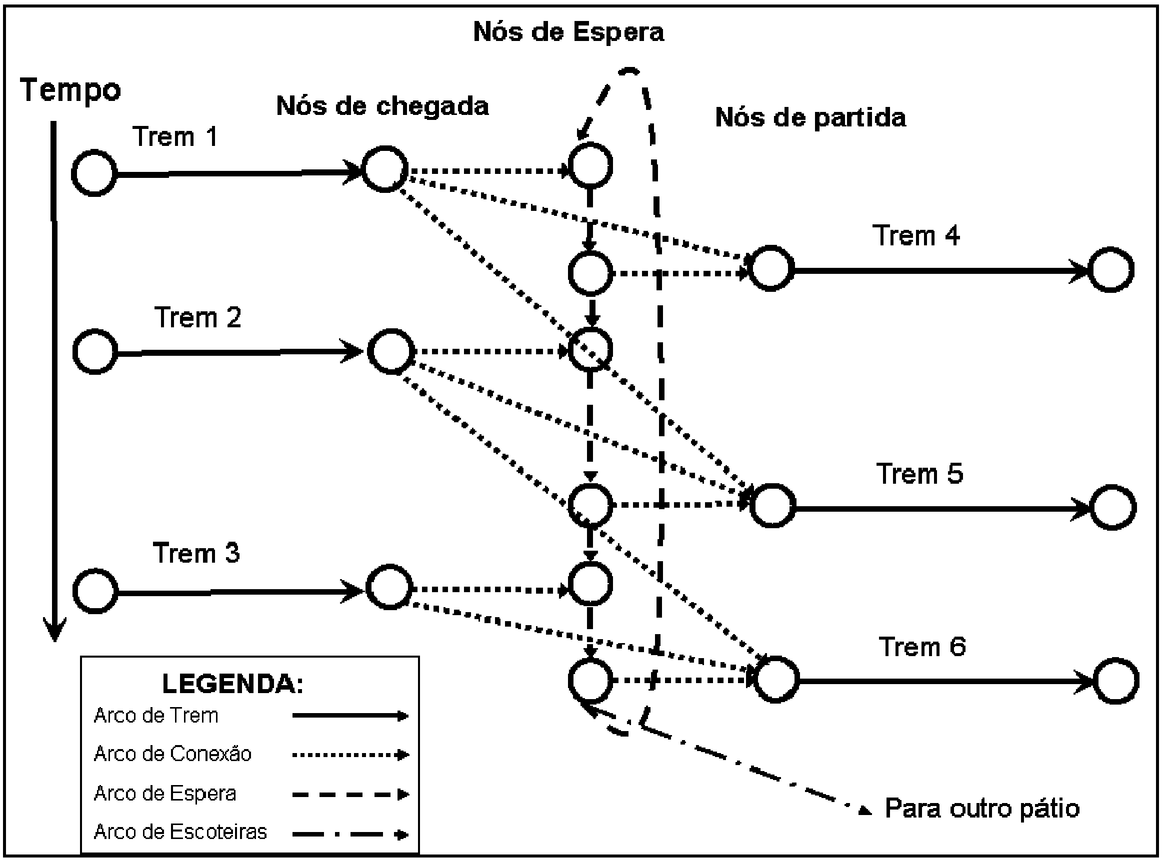
Podemos denotar a rede espaço-tempo como G = (N,A), em que N é o conjunto de nós, e A é o conjunto de arcos. Para cada evento de chegada, é criado um nó de chegada e para cada evento de partida, é criado um nó de partida. Cada nó de chegada ou de partida tem dois atributos: lugar e tempo.

O conjunto de nós referentes às chegadas e partidas dos trens são representados por *ArrNodes* e *DepNodes* respectivamente e os nós de espera (nos quais as locomotivas permanecem fora de trens) por *GrNodes*, *AllNodes = DepNodes U ArrNodes U GrNodes*.

O conjunto de arcos de trens (*TrArcs*) contém um arco para todos os trens programados. Seus pontos extremos representam os locais e tempos de sua chegada e partida. Cada nó de chegada e de partida é conectado a um nó de espera por um arco de conexão. Estas conexões pertencem ao conjunto de arcos de conexão (*CoArcs*). Em seguida são criados os nós de espera em cada pátio na ordem cronológica de seus atributos de tempo e estes são conectados aos nós de espera seguintes através dos arcos de espera, representados por *GrArcs*. Estes arcos permitem que as locomotivas que chegam possam permanecer em estoque para aguardar, até que sejam alocadas a algum trem de partida. Um determinado trem que repassa todo seu conjunto de locomotivas para o próximo trem de partida é representado pelo arco de conexão direta, que liga um nó de chegada diretamente a um nó de partida. Estes arcos também pertencem ao conjunto *CoArcs*. Finalmente, também se permite a utilização de locomotivas escoteiras. Um arco de escoteiras é criado de um nó de espera para outro nó de espera. Cada arco pertence ao conjunto *LiArcs* e tem um custo fixo F_l que representa o custo fixo de alocação de apenas uma locomotiva com maquinista da origem do arco de escoteiras para o destino. O arco de escoteiras também apresenta um custo variável que depende da quantidade de locomotivas viajando no grupo de escoteiras. E finalmente *AllArcs = TrArcs U CoArcs U LiArc.* (AHUJA et al 2005).

Essa descrição da rede espaço-tempo pode ser vista na figura a seguir, que ilustra uma parte de uma rede, com diversos tipos de arcos.

Figura 2 - Exemplo de parte de uma rede espaço-tempo



Fonte: MACHADO (2006) apud AHUJA et al (2005)

Vamos usar a seguinte notação adicional para a formulação da rede espaço-tempo:

- *I (i): Conjunto de arcos que chegam no nó i ∈ AllNodes.*
- *O (i): Conjunto de arcos que partem do nó i ∈ AllNodes.*

2.2.3. VARIÁVEIS DE DECISÃO

A formulação do modelo propõe as seguintes variáveis de decisão:

- X^k_l: número de locomotivas do tipo *k* para tracionar alocadas aos trens *l;*
- y^k_l: número de locomotivas do tipo *k* não tracionando (rebocadas, escoteiras ou ociosas) em todos os arcos;
- z_l: $z_l = 1$ se ao menos uma locomotiva circula no arco $l \in LiArcs \cup CoArcs$, $z_l = 0$ caso contrário;
- w_l: $w_l = 1$ caso apenas uma locomotiva seja alocada ao trem *l*, e $w_l = 0$ caso contrário;
- s^k: número de locomotivas do tipo *k* não utilizadas

2.2.4. FORMULAÇÃO MATEMÁTICA DO MODELO

Função Objetivo:

$$\min Z = \sum_{l \in TrArcs} \sum_{k \in K} c_l^k x_l^k + \sum_{l \in AllArcs} \sum_{k \in K} d_l^k y_l^k + \sum_{l \in LiArcs} F_l z_l + \sum_{l \in CB} B z_l + \sum_{l \in TrArcs} E_l w_l - \sum_{k \in K} F^k s^k \qquad (5)$$

Restrições:

$$\sum_{k \in K} t_l^k x_l^k \geq \mathrm{T_l} \qquad \forall l \in TrArcs \qquad (6)$$

$$\sum_{k \in K} h_l^k x_l^k \geq \beta_\mathrm{l} \mathrm{T_l} \qquad \forall l \in TrArcs \qquad (7)$$

$$\sum_{k \in K} \lambda_l^k x_l^k \leq 24 \qquad \forall l \in TrArcs \qquad (8)$$

$$\sum_{k \in K} \left(\mathrm{x}_l^k + y_l^k\right) \leq 12 \qquad \forall l \in TrArcs \cup LiArcs \qquad (9)$$

$$\sum_{l \in I(i)} \left(\mathrm{x}_l^k + y_l^k\right) = \sum_{l \in O(i)} \left(\mathrm{x}_l^k + y_l^k\right) \qquad \forall i \in \mathrm{A}ll\ Nodes, \forall k \in K \qquad (10)$$

$$\sum_{k \in K} \mathrm{y}_l^k \leq 12\mathrm{z_l} \qquad \forall l \in CoArcs \cup LiArcs \qquad (11)$$

$$\sum_{\mathrm{l} \in \mathrm{O(i)}} z_l = 1 \qquad \forall \mathrm{i} \in \mathrm{ArrNodes} \qquad (12)$$

$$\sum_{\mathrm{l} \in \mathrm{I(i)}} z_l = 1 \qquad \forall \mathrm{i} \in \mathrm{DepNodes} \qquad (13)$$

$$\sum_{k \in K} \left(\mathrm{x}_l^k + y_l^k\right) + w_l \geq 2 \qquad \forall l \in TrArcs \qquad (14)$$

$$\sum_{l \in S} \left(\mathrm{x}_l^k + y_l^k\right) + s^k \geq B^k \qquad \forall k \in K \qquad (15)$$

$$\mathrm{x_l^k} \in \mathrm{Z}^+ \qquad \forall l \in TrArcs,\ \forall k \in K \qquad (16)$$

$$\mathrm{y_l^k} \in \mathrm{Z}^+ \qquad \forall l \in TrArcs,\ \forall k \in K \qquad (17)$$

$$\mathrm{z_l} \in \{0,1\} \qquad \forall l \in CoArcs \cup LiArcs \qquad (18)$$

$$\mathrm{w_l} \in \{0,1\} \qquad \forall l \in TrArcs \qquad (19)$$

A função objetivo (5) contém seis termos. O primeiro termo designa o custo da locomotiva tracionando o trem; o segundo termo designa o custo da locomotiva sendo rebocada ou escoteira, o terceiro representa o custo fixo de alocação de apenas uma locomotiva ao trem, o quarto representa o custo fixo de quebra de tração. O quinto termo representa a penalidade por alocar apenas uma única locomotiva ao trem, e o sexto representa a economia acumulada de não usar todas as locomotivas.

As restrições (6) e (7) envolvem respectivamente a tonelagem e a potência necessária para tracionar o trem. A restrição (8) restringe o número máximo de eixos, neste caso limitando em 24. Na (9) restringe o número máximo de locomotivas por trem. Já na restrição (10) é conservado o balanço de fluxo nos nós, a (11), (12) e (13) garantem que pelo menos uma conexão direta ou quebra de tração ocorra para cada trem que chega a seu destino. A alocação de apenas uma única locomotiva é representada pela restrição (14). As demais restrições são utilizadas para garantir as condições básicas de um modelo de fluxo em rede. (AHUJA et al 2005).

Este modelo apresentado se aplicado de forma literal em uma empresa ferroviária, iria criar milhares de variáveis e restrições, não podendo ser resolvido de forma ótima ou, então, sendo resolvido em um tempo muito grande (opção fora de cogitação em qualquer empresa). Sendo assim a única forma de resolução deste modelo, e de todos os outros formulados para esse tipo de problema é a formulação de um algoritmo heurístico. Cada autor apresenta, deste modo, seu próprio método para essa “simplificação” do modelo ideal. A boa elaboração deste algoritmo heurístico é que mede a funcionalidade e aplicação do modelo elaborado.

CAPÍTULO III - DESCRIÇÃO

3.1. TRANSPORTE FERROVIÁRIO DE CARGA

Será apresentada uma breve introdução sobre o transporte ferroviário de carga, que irá ajudar a compreender melhor a extensão do problema de programação de locomotivas. Logo em seguida, será feita a apresentação da MRS Logística, empresa de transporte ferroviário, em que alguns dados foram coletados para exemplificar melhor o problema.

A primeira ferrovia brasileira foi construída pela Imperial Companhia de estradas de ferro, fundada pelo Visconde de Mauá, ligando o Porto de Mauá, na Baía de Guanabara, a Serra da Estrela, no caminho de Petrópolis. Tinha uma extensão de 14,5 km e bitola de 1676 mm. Logo depois, outras surgiram no Nordeste, Recôncavo Baiano e, principalmente, em São Paulo, para servir à economia cafeeira, então em franco desenvolvimento (Estrela do Café).

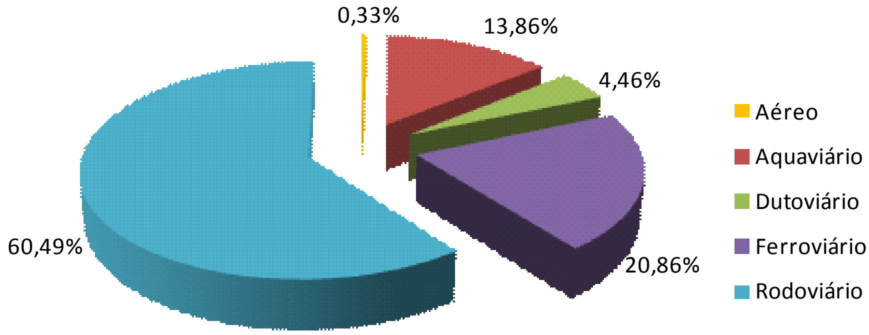
Eram, em geral, construídas ou financiadas por capitais ingleses que visavam somente à satisfação de seus interesses comerciais, sem o mínimo de planejamento.

Entre 1870 e 1920, vivíamos uma verdadeira “Era das Ferrovias”, sendo que o crescimento médio desta era de 6.000 km por década. Após 1920, com o advento da era do automóvel, as ferrovias entraram numa fase de estagnação, não tendo se recuperado até os dias atuais.

Atualmente, o Brasil é um país pobre em ferrovias; e estas se encontram irregularmente distribuídas pelo território, pois enquanto a Região Sudeste concentra quase metade (47%) das ferrovias do país, as Regiões Norte e Centro-oeste, juntas, concentram apenas 8%. O país possui hoje 30.000 km de ferrovias para tráfego, o que dá uma densidade ferroviária de 3,1 metros por km²; esta é bem pequena em relação aos EUA (150m/km²) e Argentina (15m/km²).

A capacidade de transportar grandes volumes de carga com considerável eficiência energética é a característica marcante do modal ferroviário. Esta característica é ainda mais vantajosa quando as distâncias envolvidas no transporte são médias ou grandes (dado que os principais custos deste transporte são fixos), o que faz, também, com que os custos de transporte por tonelada/quilômetro reduzam-se, na medida em que se aumenta o número de unidades de transporte.

Figura 3 - Percentual de cargas transportadas no Brasil



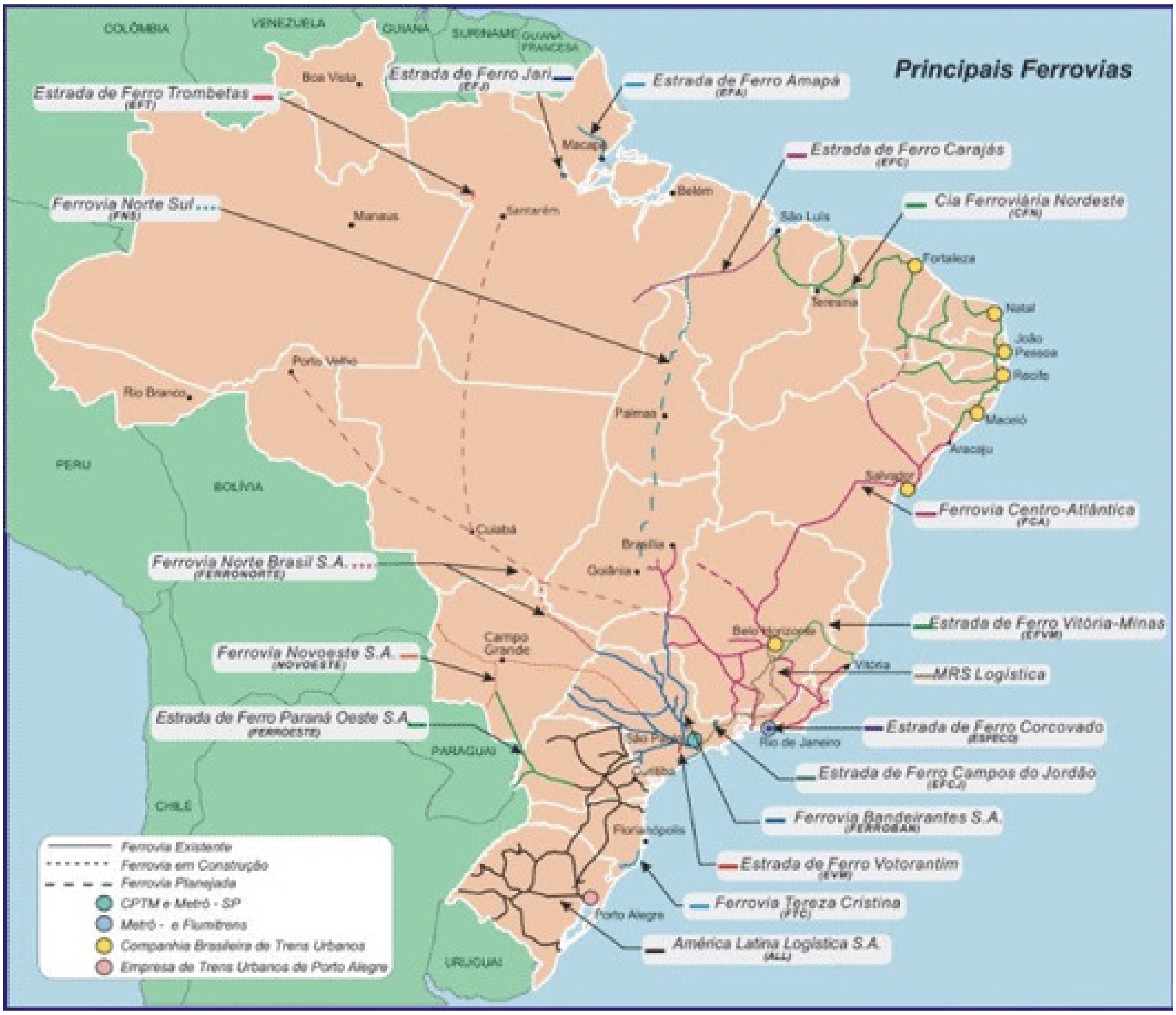
Fonte: ANTT (2006). Modificado pelo autor

Uma das limitações do transporte ferroviário é o maior tempo de viagem, quando comparado com o rodoviário, o que faz com que o cliente mantenha um maior estoque nas extremidades, embora essa modalidade também possa ser utilizada temporariamente como armazéns, além da baixa flexibilidade de acesso. Mediante a essa limitação, as concessionárias ferroviárias vêm cada vez mais buscando a prática da multi e intermodalidade.

São cargas típicas do modal ferroviário:

- Produtos Siderúrgicos;
- Grãos;
- Minério de Ferro;
- Cimento e Cal;
- Adubos e Fertilizantes;
- Derivados de Petróleo;
- Calcário;
- Carvão Mineral e Clinquer;
- Contêineres.

Figura 4 - Mapa do Sistema Ferroviário Nacional



Fonte: ANTT (2007)

3.1.1. MRS LOGÍSTICA

Com o intuito de aumentar a oferta e melhoria de serviços, o governo federal colocou em prática ações voltadas para a privatização, concessão e delegação de serviços públicos de transporte a Estados, Municípios e iniciativa privada. A Lei n.º 8.031/90, de 12/04/90, e suas alterações posteriores, instituiu o Programa Nacional de Desestatização - PND. O processo de desestatização do setor ferroviário foi iniciado em 10/03/92, a partir da inclusão da Rede Ferroviária Federal S.A. - RFFSA no PND, pelo Decreto n.º 473/92.

Foi neste contexto que surgiu a MRS; constituída em agosto de 1996, assumindo a concessão no dia 1º de dezembro do mesmo ano, após a obtenção por cessão dos direitos adquiridos pelo Consórcio MRS Logística, através do leilão de privatização, realizado em 20/09/96, na Bolsa de Valores do Rio de Janeiro, pelo valor de R$888,9 milhões.

As linhas da MRS abrangem a mais desenvolvida região do país interligando as cidades de Belo Horizonte, São Paulo e Rio de Janeiro. Assim, além de ser o sistema que une os maiores centros consumidor e produtores do país, suas linhas constituem-se no acesso ferroviário a importantes portos brasileiros: Rio de Janeiro, Sepetiba e Santos, além de atender ao terminal privativo de embarque de minério de ferro de propriedade da MBR, na Ilha de Guaíba na Baía de Angra dos Reis.

Os trechos que foram concedidos para a exploração do transporte ferroviário de cargas são aqueles que pertenceram às antigas ferrovias, Estrada de Ferro Central do Brasil, nas linhas que ligam Rio de Janeiro a São Paulo e a Belo Horizonte, bem como a Ferrovia do Aço e aquelas pertencentes à Estrada de Ferro Santos-Jundiaí, excluídas, em ambos os casos, as linhas metropolitanas de transporte de passageiros no Rio de Janeiro e em São Paulo. Desta forma, suas linhas abrangem a mais desenvolvida região do país, com uma participação média aproximada de 65% do PIB brasileiro.

Figura 5 - Mapa da malha da MRS Logística

Fonte: MRS (2007)

O mix de produtos da empresa é subdividido em duas classes: o heavy-haul, transportado através de trens unitários (mesmo produto e cliente compondo a formação do trem) devido aos grandes volumes das cargas demandadas; e a carga geral, composta por vários tipos de carga que atendem a vários segmentos de mercados, possuindo assim, diferentes origens e destino para a mesma formação do trem.

Os segmentos de mercados atendidos pelo mix de produtos são: mineração; siderurgia (aços, metalurgia); construção; químico; papel; comoditties; automotivo; além de contêineres, embarcadores e operadores logísticos.

3.2. O PROBLEMA: PROGRAMAÇÃO DE LOCOMOTIVAS

Os problemas de transporte ferroviário podem ser classificados quanto ao horizonte de planejamento. Os problemas estratégicos tratam de decisões mais estruturais da ferrovia, com impacto de longo prazo e efeito duradouro, como por exemplo, a aquisição de novos equipamentos; construção, melhoria ou mesmo desativação de trechos, e outros. No nível tático, os problemas têm impacto de curto e médio prazos e são também chamados de problemas de planejamento; normalmente, esses problemas consistem em definir estratégias a serem adotadas na ferrovia, modos de operação e planos válidos por um período razoável de tempo (por exemplo, um mês). A curto prazo, existem os problemas operacionais que dizem respeito a decisões que devem ser tomadas imediatamente, ou no máximo em dois dias. Esse último tipo de problema deve considerar o maior número de detalhes possível e serve para ajustar os planos previamente definidos.

O problema de programação de locomotivas consiste em alocar um conjunto de locomotivas para atender aos trens programados em um dado horizonte de tempo e determinar as rotas de todas as locomotivas na malha ferroviária. De acordo com a situação da frota de locomotivas no momento e com a necessidade de tração em cada trecho da malha, determina-se a composição de locomotivas que formará cada trem, de modo a maximizar o atendimento com o menor custo possível.

Diferentemente dos EUA e Europa, onde os problemas de otimização decidem quais dos seus recursos serão utilizados no atendimento das demandas, no Brasil a decisão, em geral, é sobre como utilizar da melhor maneira todos os recursos disponíveis, procurando atender ao máximo as demandas.

Como já mencionado, as locomotivas podem ser usadas de três maneiras diferentes: locomotivas ativas, são aquelas usadas para tracionar os trens; locomotivas rebocadas, são as que são carregadas desligadas, como se fossem um vagão; e por último, as locomotivas escoteiras, que são locomotivas agrupadas entre si, formando um trem sem vagões.

Ao alocar locomotivas aos trens, há outro tipo de problema a ser considerado. Quando um trem chega ao seu destino, todo o conjunto de locomotivas pode ser acoplado a outro trem que está de partida (conexão direta); ou as locomotivas do conjunto são separadas e reagrupadas com outras provenientes de outros trens, para que então possam ser anexadas a trens de partida (quebra de tração). A conexão direta é sempre mais desejada, pois no outro caso, por depender de locomotivas provenientes de mais de um trem, a vulnerabilidade ao atraso é maior.

Desta forma, o desafio da atividade de programação de locomotivas consiste em tomar as seguintes decisões:

- Quais locomotivas devem ser alocadas para tracionar quais trens?
- Quais locomotivas devem ser alocadas a quais trens para serem rebocadas?
- Quais locomotivas devem viajar “escoteiras”?

Existem muitas restrições que dificultam a fácil programação dessas locomotivas, dentre as restrições com foco operacional, podemos citar:

- Potência exigida para tracionar o trem (está vinculado ao peso do trem e ao perfil de rampa a ser enfrentado no trecho);
- Tipos de locomotivas que podem ser acopladas uma a outra;
- Número máximo de locomotivas que podem ser acopladas no mesmo trem;
- Disponibilidade de certo tipo de locomotiva;
- Impossibilidade de andar em certos trechos, causadas por restrições físicas;
- A alocação deve buscar reduzir o consumo de combustível.

Existem, também outras restrições que são de níveis estratégicos e envolvem geralmente, acordos comerciais ou regras de circulação da empresa, podemos citar também:

- Certas locomotivas não podem andar em certos trechos (acordos entre empresas);
- Alguns fluxos trabalham com locomotivas cativas;
- A alocação deve levar em conta questões de equipagem (como conforto, por exemplo);

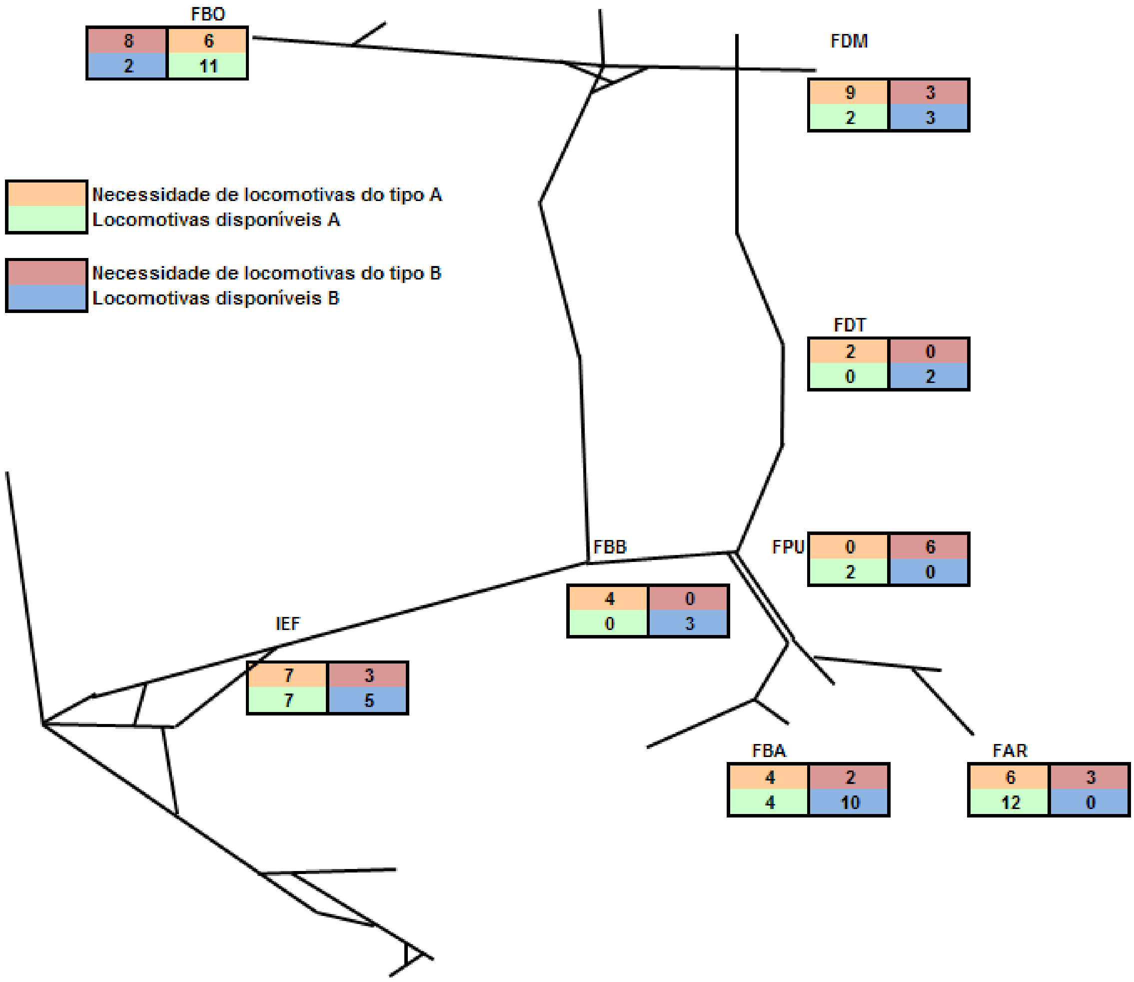
Um exemplo de quão complexo pode se tornar este problema em uma empresa, pode ser demonstrado abaixo. Estes dados foram coletados da empresa MRS Logística e representam a realidade na alocação de locomotivas, mostrando, apenas um pouco da quantidade de variáveis que este problema pode gerar.

A empresa possui hoje cerca de 300 pátios, em uma malha de 1674 km de extensão. Possui pouco mais de 500 locomotivas (de 21 modelos diferentes), que tracionam cerca de 160 trens por dia.

Para o cálculo da quantidade de locomotivas necessárias para tracionar um trem é preciso conhecer o trecho por onde o trem irá passar, isso é importante, pois é necessário ter o conhecimento do perfil da via (relacionado a % da rampa, que na MRS são discretizados em 12 perfis). Além disto, é preciso conhecer o esforço trator e a velocidade de regime de todas as locomotivas. Com esses dados, juntamente com as outras restrições citadas, é possível dimensionar os trens (número de locos e vagões). Esse é um dos dados de entrada do problema, que tende fazer a programação das locomotivas de acordo com a demanda de trens em um horizonte temporal, com as demandas pré-fixadas.

A figura abaixo apresenta com um pequeno exemplo do problema de programação de locomotivas. Nele são apresentados demandas e disponibilidades de locos, essa demanda pode ser entendida como o número de locomotivas necessárias para tracionar os trens programados para sair de determinado pátio, e as locomotivas disponíveis são locomotivas que estão, ou irão chegar ao pátio com algum trem programado.

Figura 6 - Exemplo do problema de programação de locomotivas



Fonte: Autor

Nesse caso é preciso definir quais locomotivas irão para quais pátios, qual o percurso a ser realizado por elas e de que maneira elas serão transportadas. Isso resume bem o que é o problema de programação de locos.

Os dados apresentados são apenas para ilustrar a complexidade e importância da programação de locomotivas, e não serão necessariamente usadas na elaboração e solução do modelo a ser apresentado no próximo capítulo.

CAPÍTULO IV - SOLUÇÃO

4.1. INTRODUÇÃO

O trabalho em questão se propõe a apresentar uma solução para o problema de alocação de locomotivas em uma malha ferroviária. Como descrito anteriormente esse problema pode ser classificado em tático e operacional, dependendo do nível de aprofundamento e horizonte de solução a ser buscado.

O problema aqui proposto será de nível tático e dessa forma não vai abordar uma das principais variáveis de um problema operacional, a questão da alocação no tempo. Essa é uma das maiores dificuldades de se trabalhar no nível operacional, pois a questão do tempo é de difícil resolução e aumenta significativamente o tempo de resolução do problema.

Dessa forma o algoritmo a ser elaborado terá uma função tática, ele é utilizado no planejamento dos recursos, para se ter uma visão global das possibilidades de alocação das locomotivas em um certo horizonte. É lógico que essa solução deverá ser trabalhada posteriormente com a variável tempo, pois as locomotivas em geral mudam suas posições com o passar do tempo, mas isso não tira a importância da análise tática de alocação.

Serão feitas algumas simplificações no modelo, essas simplificações nem sempre poderão ser realizadas na resolução de um problema real, mas se tratando de um trabalho acadêmico que tem o objetivo de apresentação da ferramenta de otimização e de entendimento global do problema de locomotivas, essas simplificações não terão grande relevância para a resolução.

Dentre as principais simplificações que serão feitas para esse modelo tático, podemos citar:

- O modelo não trabalhará com várias rotas, para ir de um ponto a outro, só haverá uma rota;
- Caracterização das locomotivas apenas pela potência (HP);
- Locomotiva só se movimenta escoteira (não haverá loco rebocada);
- Não será considerada a questão de auxílio;
- Não será considerada a questão de manutenção das locomotivas;
- Não haverá atividades das locomotivas nos pátios;
- Não serão consideradas paradas para abastecimento;
- Não serão consideradas paradas para troca de equipagem;
- Não serão considerados cruzamentos na malha;
- Não será considerada a capacidade de cada pátio;
- As locomotivas podem andar em todos os trechos.

A modelagem foi desenvolvida para resolver o seguinte problema: Alocar as locomotivas disponíveis para os trens programados, levando em consideração os pátios de origens das locomotivas, os pátios de partida dos trens, os custos de deslocamento das locomotivas de pátio a pátio, a potência (HP) oferecido por cada locomotiva e a potência (HP) exigida para cada trem. O objetivo do algoritmo é encontrar a solução de alocação que represente o menor custo de deslocamento das locomotivas.

Pelas características desse problema, a melhor forma de modelá-lo é como uma variação do Problema de Transportes e com todas as suas variáveis inteiras. O modelo é representado da seguinte maneira:

- Função Objetivo: minimizar o custo de alocação de todas as locomotivas aos trens programados;
- Restrição de oferta: garante que as locomotivas irão ser utilizadas apenas uma vez, e que algumas locomotivas poderão não ser alocadas (quando a oferta for maior que a demanda)
- Restrição de demanda: garante que será alocada uma quantidade mínima de locomotivas para atender a demanda de potência do trem.
- Restrição de alocação: garante a escolha de qual locomotiva será alocada para qual trem.

Matematicamente:

$$\min\ Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{20}$$

sujeito a:

$$\sum_{j=1}^{n} x_{ij} \leq 1 \qquad para \quad i = 1, 2, ..., m \tag{21}$$

$$\sum_{i=1}^{m} x_{ij} * s_i \geq d_j \qquad \text{para} \quad j = 1, 2, ..., n \tag{22}$$

$$x_{ij} = \begin{cases} 1 & \text{se i designado para j} \\ 0 & \text{caso contrário} \end{cases} \qquad (i = 1, ..., m ; j = 1, ..., n) \tag{23}$$

Onde:

i = locomotiva disponível

j = trem

m = número total de locomotivas

n = número total de trens

s_i = oferta de potência (HP) de cada locomotiva i

d_j = demanda de potência (HP) de cada trem j

C_{ij} = custo de transportar a locomotiva i para o pátio onde está o trem j

X_{ij} = variável binária que indica se a locomotiva i foi designado para o trem j

Existem 3 diferenças nas restrições dessa modelagem e a modelagem clássica do problema de transportes, uma delas é que neste caso a variável X_{ij} é binária (assume apenas 0 ou 1), e as outras duas dizem respeito ao balanceamento da rede, nesta modelagem proposta, a oferta não é necessariamente igual a demanda.

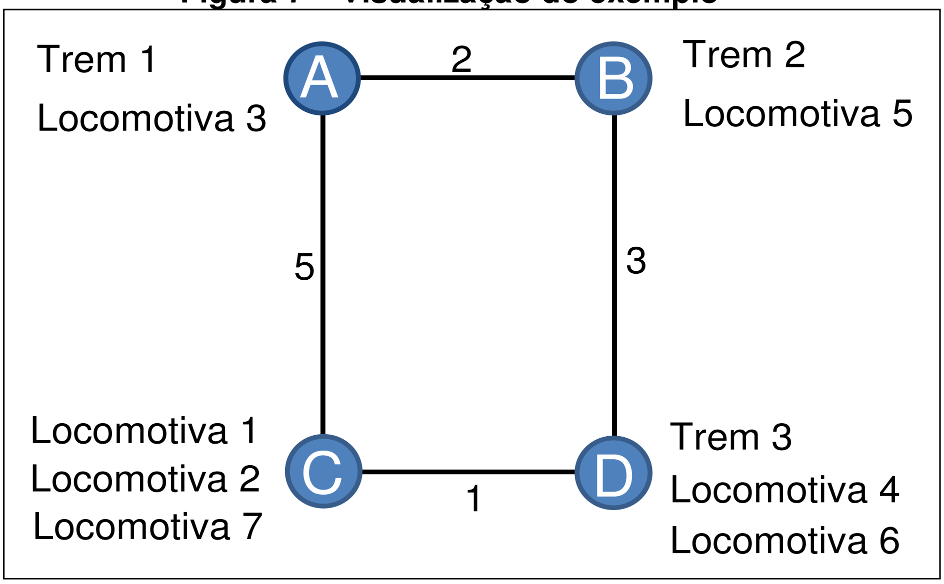
Para a resolução desse problema é necessário informar como entrada, os seguintes dados:

- As locomotivas disponíveis, informando a quantidade total, em qual pátio estão cada uma delas e qual é a potência de cada uma;
- Trens programados, informando a quantidade total, em qual pátio eles estão e qual é a potência demandada de cada um;
- Os custos de deslocamento das locomotivas de um pátio a outro.

4.2. DETALHAMENTO DO ALGORITMO

Para exemplificar a resolução do problema pelo algoritmo proposto, será apresentado um pequeno exemplo do problema, mostrando passo a passo sua resolução. Para esse exemplo serão considerados os seguintes dados:

Figura 7 – Visualização do exemplo



Fonte: Autor

- 4 pátios $\begin{cases} A \\ B \\ C \\ D \end{cases}$

- 3 trens:

Tabela 2 – Resumo dos trens

TREM	PÁTIO DE ORIGEM	POTÊNCIA DEMANDADA (HP)
1	A	2000
2	B	2500
3	D	4000

- 7 locomotivas:

Tabela 3 – Resumo das Locomotivas

LOCOMOTIVA	PÁTIO DE ORIGEM	POTÊNCIA DISPONIBILIZADA (HP)
1	C	1500
2	C	1000
3	A	1500
4	D	1000
5	B	1500
6	D	1000
7	C	1500

- Custo de deslocamento entre os pátios:

Tabela 4 – Matriz Origem x Destinos – Custos de deslocamento

	A	B	C	D
A	0	2	5	5
B	2	0	4	3
C	5	4	0	1
D	5	3	1	0

O primeiro passo para a resolução desse problema é a organização dos dados de entrada para que o algoritmo possa interpretá-los, em geral são usadas matrizes para representar essas entradas.

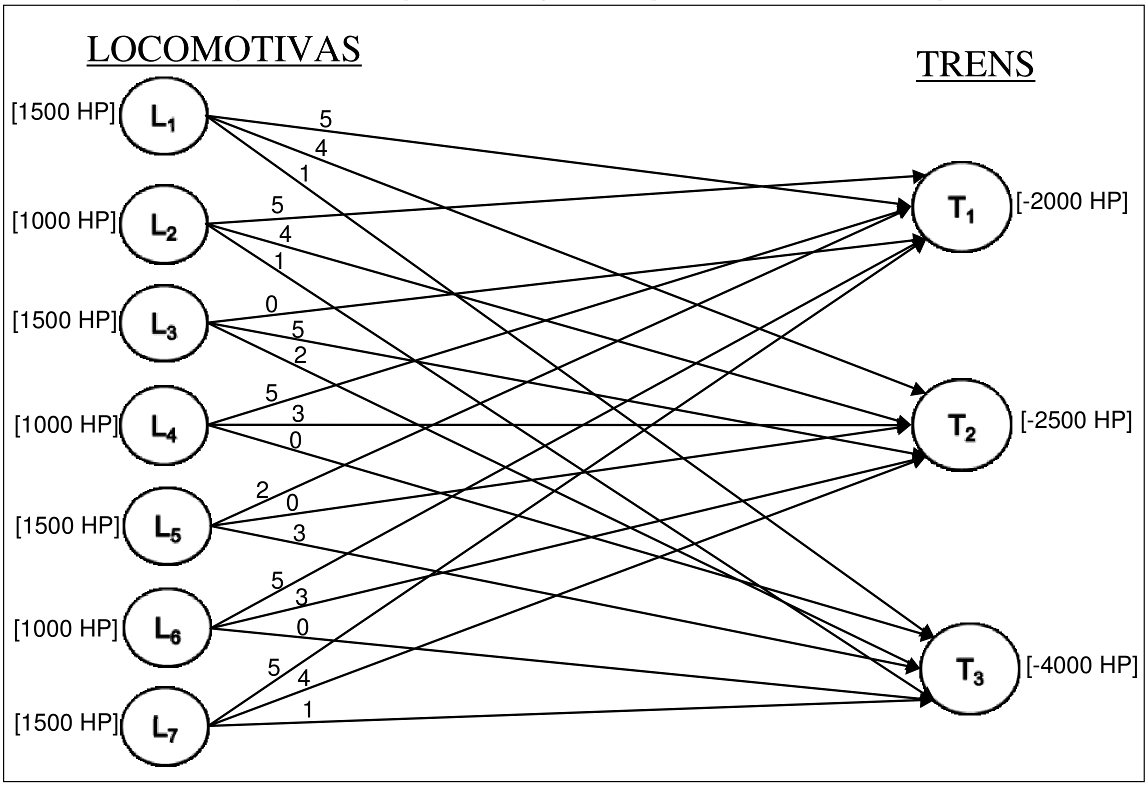
O passo seguinte para esse exemplo é criar a matriz C_{ij} (custo de alocar a locomotiva i para o pátio onde está o trem j), essa matriz é calculada através da matriz de custo de deslocamento entre os pátios, matriz localização das locomotivas e a matriz localização dos trens. Esse cálculo pode ser encarado como um pré-processamento para o algoritmo apresentado, afinal seus dados serão utilizados diretamente para a resolução do problema. Segue a tabela com o resultado deste pré-processamento:

Tabela 5 – Matriz C_{ij}

TREM / LOCO	1	2	3
1	5	4	1
2	5	4	1
3	0	2	5
4	5	3	0
5	2	0	3
6	5	3	0
7	5	4	1

A figura seguinte resume o problema através da representação em forma de uma rede:

Figura 8 – Representação Esquemática do exemplo



Fonte: Autor

Com todos os dados de entrada organizados, só resta escrever matematicamente o problema:

Função Objetivo a ser minimizada:

$$Z = 5x_{11} + 4x_{12} + 4x_{13} + 5x_{21} + 4x_{22} + 4x_{23} + 0x_{31} + 5x_{32} + 2x_{33} + 5x_{41} + 3x_{42} + 0x_{43} + 2x_{51} + 0x_{52} + 3x_{53} + 5x_{61} + 3x_{62} + 0x_{63} + 5x_{71} + 4x_{72} + 1x_{73} \quad (23)$$

Sujeito a:

$$x_{11} + x_{12} + x_{13} \leq 1 \quad (24)$$

$$x_{21} + x_{22} + x_{23} \leq 1 \quad (25)$$

$$x_{31} + x_{32} + x_{33} \leq 1 \quad (26)$$

$$x_{41} + x_{42} + x_{43} \leq 1 \quad (27)$$

$$x_{51} + x_{52} + x_{53} \leq 1 \quad (28)$$

$$x_{61} + x_{62} + x_{63} \leq 1 \quad (29)$$

$$x_{71} + x_{72} + x_{73} \leq 1 \quad (30)$$

$$1500x_{11} + 1000x_{21} + 1500x_{31} + 1000x_{41} + 1500x_{51} + 1000x_{61} + 1500x_{71} \geq 2000 \quad (31)$$

$$1500x_{12} + 1000x_{22} + 1500x_{32} + 1000x_{42} + 1500x_{52} + 1000x_{62} + 1500x_{72} \geq 2500 \quad (32)$$

$$1500x_{13} + 1000x_{23} + 1500x_{33} + 1000x_{43} + 1500x_{53} + 1000x_{63} + 1500x_{73} \geq 4000 \quad (33)$$

Com:

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases} \qquad (i = 1,....,7\ ;\ j = 1,2,3) \quad (34)$$

A solução encontrada para esse problema foi:

Menor Custo de alocação: 10 unidades

Alocação correspondente:

Loco 1 => Trem 3

Loco 2 => Trem 1

Loco 3 => Trem 1

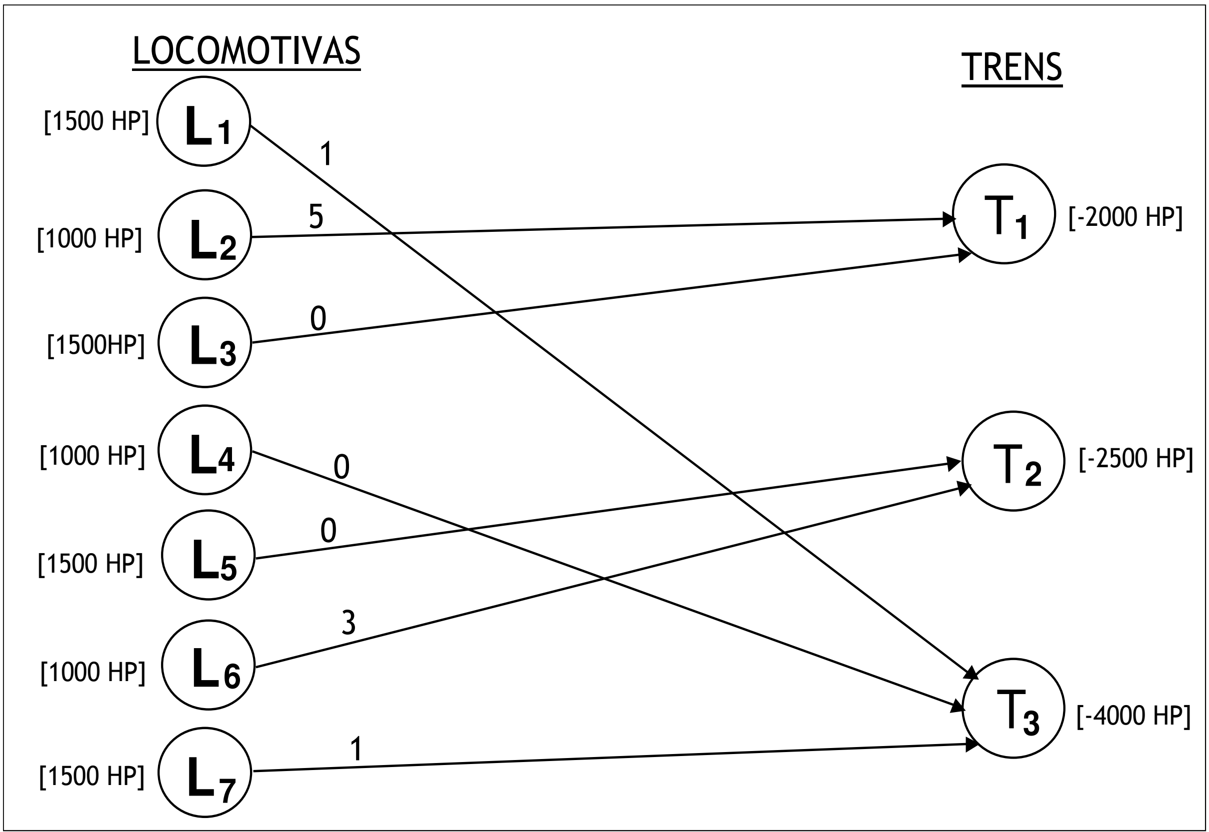
Loco 4 => Trem 3

Loco 5 => Trem 2

Loco 6 => Trem 2

Loco 7 => Trem 3

Figura 9 – Representação Esquemática do resultado



Fonte: Autor

Todas as restrições foram cumpridas, com as locomotivas sendo utilizadas apenas uma única vez e os trens recebendo a potência necessária.

Foi elaborado um algoritmo para resolução desse problema, o software escolhido para rodar esse algoritmo foi o Xpress, que utiliza linguagem de modelagem MOSEL. Não serão apresentados mais detalhes sobre o software, nem sobre a linguagem do modelo, pois esses não fazem parte do escopo desse estudo. Para maiores informações, acessar o site oficial do desenvolvedor do software: *http://www.dashoptimization.com*.

O programa elaborado está todo parametrizado e busca todos os dados de entrada de um arquivo externo, desta forma ele pode ser aplicado para a resolução de problemas de todos os tamanhos possíveis, basta alterar apenas os dados de entrada nesse arquivo externo.

Para a resolução deste problema o software utilizou o método Simplex Dual e foram realizadas 8 iterações, em um tempo inferior a 1 segundo.

Ao final desse trabalho, na área reservada para apêndices, está a montagem do problema no software Xpress, os dados de entrada para a resolução do problema e o relatório de resposta.

Cabe ressaltar que foi tomado como premissa que o algoritmo só resolveria os problemas onde o número de locomotivas fosse suficiente para atender toda a demanda de trens, quando isso não acontecesse o algoritmo daria como resposta a seguinte frase "Quantidade de Locomotivas não suficientes para atender a demanda dos trens".

4.3. APLICAÇÃO EM ESCALA REAL

Para demonstrar que esse algoritmo elaborado é capaz de resolver problemas maiores, faremos um exemplo utilizando uma quantidade de variáveis reais de um problema. Para isso foram coletados alguns dados reais da MRS Logística, sendo esses: a quantidade de trens de carga geral, a quantidade de locomotivas disponíveis para esses trens e a quantidade de pátios de partida das locomotivas e trens. Para a resolução foram estimados os custos de deslocamento entre os pátios, as potências requeridas por cada trem, a potência oferecida por cada locomotiva e a localização de todos os trens e locomotivas.

Foi utilizado como dados de entrada, 40 trens, 75 locomotivas e 30 pátios. Todos os outros dados coletados e estimados podem ser vistos no apêndice 4, que contém todos os dados de entrada necessários para a resolução do problema.

O algoritmo utilizado nesse caso é o mesmo do exemplo anterior, e encontrou como o menor custo, 64 unidades. Para essa resolução o software utilizou o método Simplex Dual e foram realizadas 180 iterações, em um tempo de 3,2 segundos.

Esse problema gerou 3000 variáveis de decisão (multiplicação do número de locomotivas, pelo número de trens), tornando a resolução do problema mais complexa computacionalmente. Ao final foi encontrada a solução ótima, que previa a alocação de 74 locomotivas aos trens e apenas uma locomotiva (número 69) não precisou ser utilizada. Essa não utilização da locomotiva, ocorreu pela oferta de locomotivas maior que a demanda de potência dos trens e pela otimização na utilização das mesmas.

Os resultados detalhados deste problema podem ser vistos no apêndice 5.

Esses resultados mostram a forma mais econômica para a alocação das locomotivas para atender a demanda dos trens, mas seu resultado final deve ser interpretado pelo tomador de decisão, afinal essa é apenas uma ferramenta de apoio e alguns casos específicos não podem ser modelados e resolvidos computacionalmente.

Assim fica comprovado que o algoritmo é capaz de resolver desde problemas pequenos, até problemas em níveis reais. A solução do problema, então, atinge o objetivo esperado, validando o modelo e o algoritmo usado.

CAPÍTULO IV – CONSIDERAÇÕES FINAIS

Nesse trabalho foi apresentada uma descrição completa do problema de alocação ótima de locomotivas na malha ferroviária. Foram apresentados os conceitos relacionados ao problema, as principais variáveis levadas em consideração na prática, os métodos de resolução do problema propostos por alguns autores, e por fim foi elaborada uma formulação matemática que soluciona o problema de locomotivas em um nível tático.

A ferramenta de otimização criada cumpriu seu objetivo, que é de apresentar os conceitos de Pesquisa Operacional na resolução de problemas cotidianos nas empresas. Cabe ressaltar que o programa proposto não tem o objetivo de ser aplicado na realidade de nenhuma ferrovia, pois foram realizadas muitas simplificações, que em âmbito real não poderiam ser realizadas.

Fica a confirmação que para a resolução do problema de programação de locomotivas de forma otimizada a alternativa mais adequada é a utilização das ferramentas de Pesquisa Operacional. Podemos concluir também que mesmo sendo um problema de alta complexidade, a programação de locomotivas pode ser resolvido através da implementação de um sistema que inclua todas as variáveis de decisão possíveis e que utilize um algoritmo específico para sua resolução.

É interessante citar também que a implementação de um sistema de suporte a decisão para esse tipo de análise é fundamental para melhorar o nível de trabalho em uma ferrovia, atualmente com a demanda crescente de transporte e a enorme quantidade de recursos a disposição, não podemos apenas confiar na habilidade das pessoas na tarefa de alocação de recursos, apesar de ser muito importante esse conhecimento pessoal, ele deve ser complementado com um apoio computacional, só dessa forma poderemos garantir melhores resultados.

Dessa forma, conclui-se que o trabalho desenvolvido cumpriu com o escopo delineado e proporcionou um aprendizado importante para o autor dessa monografia.

REFERÊNCIAS BIBLIOGRÁFICAS

AHUJA R.K., CUNHA C.B. and SAHIN G., 2005, *Network Models in Railroad Planning and Scheduling*, chapter in J.C. Smith, editor, *Tutorials in Operations Research*, 54-101. INFORMS.

AHUJA R.K., LIU J., ORLIN J. B., SHARMA D., SHUGHART L. A., 2002, *Solving Real-Life Locomotive Scheduling Problems*

CORDEAU J. F., TOTH P., VIGO D., 1998, *A Survey of Optimization Models for Train Routing and Scheduling,* Transportation Science, vol 42, nº 4, PP. 380-404

MACHADO, M. N., 2006, *Pesquisa Operacional Aplicada ao Processo de Planejamento e Programação da Operação do Transporte Ferroviário de Carga*. Tese IME/MRS, Rio de Janeiro, RJ, Brasil.

TORRACCA, N. A., 2003, *Alocação Dinâmica de Tripulações de Trens de Carga*. Tese - Universidade Federal do Rio de Janeiro, COPPE

LINS, M. P. E.; CALÔBA, G. M.; 2006, *Programação Linear com aplicações em teoria dos jogos e avaliação de desempenho (data envelopment analysis)*, Editora Interciência.

TEIXEIRA, M. C. C.; 2005; *Investigação Operacional*; Licenciatura em Florestal e Rural - Instituto Politécnico de Castelo Branco.

GOLDBARG, M. C.; LUNA, H. P. L.; *Otimização Combinatória e Programação Linear*, 2ª edição, Editora Campus LTDA.

Sociedade Brasileira de Pesquisa Operacional (SOBRAPO). Disponível em: <http://www.sobrapo.org.br/sitesobrapo.htm> Acesso em: Maio/2008

LISBOA, Erico, *Apostila de Pesquisa Operacional*, 2002. Disponível em: <http://www.decom.ufop.br/prof/rduarte/CIC271/apostila_po.pdf> Acesso em: Maio/2008

Instituto de Desenvolvimento Gerencial (INDG). Disponível em: <http://www.indg.com.br/po/definicao.asp> Acesso em: Maio/2008

ALBAN, MARCUS, 2002*, Os Modais e os Desafios da Multimodalidade na Bahia,* Cadernos Flem IV - Transportes e Logística

Agência Nacional de Transportes Terrestres (ANTT). Disponível em: <http://www.antt.gov.br/carga/ferroviario> Acesso em: Maio/2008

Associação Nacional dos Transportadores Ferroviários (ANTF). Disponível em: <http://www.antf.org.br> Acesso em: Maio/2008

APÊNCICE 1 – ELABORAÇÃO DO ALGORITMO

```
!MONTAGEM DO PROBLEMA NO SOFTWARE XPRESS;

model "Alocação de Locomotivas"
uses "mmxprs"

parameters
ARQUIVO_INPUT = "input.txt"
end-parameters

declarations
Qtd_Locos : integer !Define variavel que armazena quantidade de locos
Qtd_Trens : integer !Define variavel que armazena quantidade de trens
Qtd_Patios: integer !Define variavel que armazena quantidade de patios
end-declarations

initializations from ARQUIVO_INPUT
  Qtd_Locos Qtd_Trens Qtd_Patios
end-initializations

declarations

        !Número de trens
        Numero_do_Trem = 1..Qtd_Trens

        !Número de Pátios
        Numero_do_Patio = 1..Qtd_Patios

        !Número de locos
        Numero_da_Loco = 1..Qtd_Locos

        !Matriz de custos de deslocamento das locomotivas de um pátio até outro
        Custo_Deslocamento : array (Numero_do_Patio,Numero_do_Patio) of integer

        !Local (pátio) inicial da locomotiva
```

```
        Local_Loco : array(Numero_da_Loco) of integer

        !Local (pátio) inicial do trem
        Local_Trem : array(Numero_do_Trem) of integer

        !Disponibilidade de HP da locomotiva
        HP_Loco : array(Numero_da_Loco) of integer

        !Disponibilidade de HP do trem
        HP_Trem : array(Numero_do_Trem) of integer

        !Matriz de custos de deslocamento das locomotivas até o pátio do trem
        Custo_LocoXTrem : array (Numero_da_Loco,Numero_do_Trem) of integer

        !Matriz com as variáveis de decisão
        Alocacao_LocoXTrem : array(Numero_da_Loco,Numero_do_Trem) of mpvar
end-declarations

initializations from ARQUIVO_INPUT
  Custo_Deslocamento HP_Loco HP_Trem Local_Loco Local_Trem
end-initializations

!Cria a matriz Custo_LocoXTrem
writeln("Matriz Custo LocoXTrem:")
forall(i in Numero_da_Loco) do
        forall(j in Numero_do_Trem) do
                Custo_LocoXTrem(i,j) := Custo_Deslocamento(Local_Loco(i), Local_Trem(j))
                write(Custo_LocoXTrem(i,j), "-")
        end-do
        writeln
end-do
writeln
writeln
```

```
!Declaração da Função Objetivo
FUNCAO_OBJETIVO := SUM(i in Numero_da_Loco, j in Numero_do_Trem)
Custo_LocoXTrem(i,j) * Alocacao_LocoXTrem(i,j)

!Restrições
!Restricao de Integralidade (define que as variaveis de decisao sejam inteiras e binarias: 0
ou 1)
forall(i in Numero_da_Loco, j in Numero_do_Trem) Alocacao_LocoXTrem(i,j) is_binary

!Restrição de utilizacao de apenas uma vez cada loco
forall(i in Numero_da_Loco) sum(j in Numero_do_Trem) Alocacao_LocoXTrem(i,j) <= 1

!Restrição de atendimento da HP requisitada por trem
forall(j in Numero_do_Trem) sum(i in Numero_da_Loco) HP_Loco(i) *
Alocacao_LocoXTrem(i,j)>= HP_Trem(j)

!Executa o otimizador para resolver o problema
 minimize(FUNCAO_OBJETIVO)

!Escreve a resposta encontrada
writeln("RESULTADOS: ")
if getobjval < 0 then
        writeln("Quantidade de Locomotivas não suficientes para atender a demanda dos
        trens")
end-if
if getobjval > 0 then
        writeln("Menor Custo: ", getobjval)
        forall (i in Numero_da_Loco, j in Numero_do_Trem) do
                if getsol(Alocacao_LocoXTrem(i,j)) = 1 then
                writeln("Loco ", i, " => Trem ", j)
                end-if
        end-do
end-if

end-model
```

APÊNCICE 2 – DADOS DE ENTRADA DO PRIMEIRO EXEMPLO

!Input do modelo do TCC (otimização da distribuição de locos)
Qtd_Locos: 7
Qtd_Trens: 3
Qtd_Patios: 4

!Criação da matriz de custo de deslocamento entre os pátios
Custo_Deslocamento : [
0, 2, 5, 5,
2, 0, 4, 3,
5, 4, 0, 1,
5, 3, 1, 0]

!Vetor com disponibilidade de HP da locomotiva
HP_Loco : [1500, 1000, 1500, 1000, 1500, 1000, 1500]

!Vetor com disponibilidade de HP do trem
HP_Trem : [2000, 2500, 4000]

!Vetor com local (pátio) inicial da locomotiva
Local_Loco : [3, 3, 1, 4, 2, 4, 3]

!Vetor com local (pátio) inicial do trem
Local_Trem : [1, 2, 4]

APÊNCICE 3 – RELATÓRIO DE SAÍDA DO PRIMEIRO EXEMPLO

Alocacao_LocoXTrem has 2 dimension(s) and contains a total of 21 elements of type 'decision variable'

Alocacao_LocoXTrem(1,1)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(1,2)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(1,3)=Solution: 1, Reduced cost: 0
Alocacao_LocoXTrem(2,1)=Solution: 1, Reduced cost: 0
Alocacao_LocoXTrem(2,2)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(2,3)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(3,1)=Solution: 1, Reduced cost: 0
Alocacao_LocoXTrem(3,2)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(3,3)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(4,1)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(4,2)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(4,3)=Solution: 1, Reduced cost: 0
Alocacao_LocoXTrem(5,1)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(5,2)=Solution: 1, Reduced cost: 0
Alocacao_LocoXTrem(5,3)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(6,1)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(6,2)=Solution: 1, Reduced cost: 0
Alocacao_LocoXTrem(6,3)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(7,1)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(7,2)=Solution: 0, Reduced cost: 0
Alocacao_LocoXTrem(7,3)=Solution: 1, Reduced cost: 0

APÊNCICE 4 – DADOS DE ENTRADA DO EXEMPLO EM ESCALA REAL

!Input do modelo do TCC (otimização da distribuição de locos)

Qtd_Locos: 75

Qtd_Trens: 40

Qtd_Patios: 30

!Criação da matriz de custo de deslocamento entre os pátios

Custo_Deslocamento : [00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01,

01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02,

02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03,

03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04,

04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05,

05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06,

06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07,

07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08,

08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09,

09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10,

10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11,

11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12,

12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13,

13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14,

14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15,
15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14,
14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13,
13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12,
12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11,
11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10,
10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08, 09,
09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07, 08,
08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06, 07,
07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05, 06,
06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04, 05,
05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03, 04,
04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02, 03,
03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01, 02,
02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00, 01,
01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 14, 13, 12, 11, 10, 09, 08, 07, 06, 05, 04, 03, 02, 01, 00]

!Vetor com disponibilidade de HP da locomotiva

HP_Loco : [3000, 2000, 2000, 2600, 2600, 3000, 3000, 2000, 3000, 2000, 2600, 2600, 3000, 3000, 2600, 2600, 2600, 3000, 2600, 3000, 3000, 2600, 2000, 2000, 2600, 2600, 2600, 2600, 2000, 3000, 2000, 2000, 2000, 2000, 3000, 2000, 3000, 3000, 2600, 2600, 3000, 2000, 2000, 2000, 3000, 2600, 2000, 2000, 2600, 2000, 3000, 2000, 3000, 3000, 3000, 2000, 2600, 3000, 3000, 3000, 2600, 2600, 2600, 2600, 2600, 3000, 3000, 3000, 2600, 3000, 2600, 3000, 3000, 3000, 3000]

!Vetor com disponibilidade de HP do trem
HP_Trem : [1500, 7000, 1500, 4200, 2000, 6000, 4200, 7000, 6000, 1500, 1500, 2000, 7000, 4200, 6000, 2000, 7000, 6000, 7000, 2000, 1500, 6000, 7000, 4200, 2000, 1500, 4200, 6000, 1500, 2000, 7000, 4200, 6000, 1500, 7000, 4200, 2000, 6000, 4200, 2000]

!Vetor com local (pátio) inicial da locomotiva
Local_Loco : [13, 7, 28, 17, 12, 17, 13, 29, 30, 18, 18, 13, 19, 9, 8, 20, 5, 15, 28, 15, 29, 25, 3, 5, 10, 23, 7, 25, 17, 10, 15, 5, 27, 27, 29, 19, 1, 22, 24, 22, 6, 10, 28, 17, 22, 9, 4, 12, 7, 27, 15, 21, 27, 30, 17, 27, 10, 29, 12, 17, 2, 30, 22, 8, 9, 9, 16, 12, 12, 28, 22, 3, 16, 9, 23]

!Vetor com local (pátio) inicial do trem
Local_Trem : [11, 21, 28, 14, 25, 22, 14, 3, 19, 11, 23, 16, 15, 26, 3, 17, 2, 6, 4, 23, 25, 9, 29, 16, 20, 6, 15, 20, 1, 22, 10, 7, 9, 20, 23, 28, 5, 27, 2, 13]

APÊNCICE 5 – RELATÓRIO DE SAÍDA DO EXEMPLO EM ESCALA REAL

RESULTADOS

Menor Custo: 64

Loco 1 => Trem 7
Loco 2 => Trem 32
Loco 3 => Trem 36
Loco 4 => Trem 24
Loco 5 => Trem 10
Loco 6 => Trem 28
Loco 7 => Trem 7
Loco 8 => Trem 17
Loco 9 => Trem 15
Loco 10 => Trem 34
Loco 11 => Trem 2
Loco 12 => Trem 40
Loco 13 => Trem 28
Loco 14 => Trem 33
Loco 15 => Trem 26
Loco 16 => Trem 2
Loco 17 => Trem 8
Loco 18 => Trem 13
Loco 19 => Trem 36
Loco 20 => Trem 13
Loco 21 => Trem 29
Loco 22 => Trem 21
Loco 23 => Trem 8
Loco 24 => Trem 37
Loco 25 => Trem 31
Loco 26 => Trem 35
Loco 27 => Trem 8
Loco 28 => Trem 5
Loco 29 => Trem 16
Loco 30 => Trem 18
Loco 31 => Trem 27
Loco 32 => Trem 19
Loco 33 => Trem 38
Loco 34 => Trem 38
Loco 35 => Trem 23
Loco 36 => Trem 25
Loco 37 => Trem 39
Loco 38 => Trem 6
Loco 39 => Trem 35
Loco 40 => Trem 20
Loco 41 => Trem 19
Loco 42 => Trem 31
Loco 43 => Trem 3
Loco 44 => Trem 12
Loco 45 => Trem 6
Loco 46 => Trem 22
Loco 47 => Trem 19
Loco 48 => Trem 4
Loco 49 => Trem 32
Loco 50 => Trem 38
Loco 51 => Trem 27
Loco 52 => Trem 2
Loco 53 => Trem 14
Loco 54 => Trem 17
Loco 55 => Trem 9
Loco 56 => Trem 14
Loco 57 => Trem 31
Loco 58 => Trem 23
Loco 59 => Trem 4
Loco 60 => Trem 9
Loco 61 => Trem 39
Loco 62 => Trem 17
Loco 63 => Trem 30
Loco 64 => Trem 22
Loco 65 => Trem 22
Loco 66 => Trem 33
Loco 67 => Trem 13
Loco 68 => Trem 1
Loco 69 => Não utilizada
Loco 70 => Trem 23
Loco 71 => Trem 11
Loco 72 => Trem 15
Loco 73 => Trem 24
Loco 74 => Trem 18
Loco 75 => Trem 35