

UNIVERSIDADE FEDERAL DO CEARÁ
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE MESTRADO EM LOGÍSTICA E PESQUISA OPERACIONAL

RODRIGO FRANK DE SOUZA GOMES

**APLICAÇÃO DA METAHEURÍSTICA TABU SEARCH NA OTIMIZAÇÃO DE
ROTAS DE MANUTENÇÃO PREVENTIVA EM CAMPO**

FORTALEZA – CE

2011

RODRIGO FRANK DE SOUZA GOMES

Dissertação apresentada ao Programa de Mestrado em Logística e Pesquisa Operacional da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Mestre em Ciência (MSc.) em Logística e Pesquisa Operacional.

Área de concentração: Pesquisa Operacional

Orientador: Prof. José Lassance de Castro Silva,
D.Sc.

FORTALEZA – CE

2011

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca de Pós-Graduação em Engenharia - BPGE

-
- G617a Gomes, Rodrigo Frank de Souza
 Aplicação da metaheurística tabu search na otimização de rotas de manutenção
preventiva em campo / Rodrigo Frank de Souza Gomes. – 2011.
 108 f.: il. Color. enc. ; 30 cm.
- Dissertação (Mestrado) – Universidade Federal do Ceará, Pró-Reitoria de Pesquisa e
Pós- Graduação. Programa de Mestrado em Logística e Pesquisa Operacional, Fortaleza,
2011.
 Área de Concentração: Pesquisa Operacional
 Orientação: Prof. Dr. José Lassance de Castro Silva
1. Otimização combinatória. 2. Manutenção. 3. PCV. 4. Busca Tabu I. Título.

CDD 003

RODRIGO FRANK DE SOUZA GOMES

**APLICAÇÃO DA METAHEURÍSTICA TABUSEARCH NA OTIMIZAÇÃO DE
ROTAS DE MANUTENÇÃO PREVENTIVA EM CAMPO**

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Logística e Pesquisa Operacional em 08 de dezembro de 2011, da Universidade Federal do Ceará, como requisito parcial para obtenção do grau de Mestre em Logística e Pesquisa Operacional.

Área de Concentração: Pesquisa Operacional.

Aprovado em 09/12/2011.

BANCA EXAMINADORA

Prof. Dr. José Lassance de Castro Silva (Orientador)

Universidade Federal do Ceará – UFC

Prof. Dr. João Bosco Furtado Arruda

Universidade Federal do Ceará – UFC

Prof. Dr. Plácido Rogério Pinheiro

Universidade de Fortaleza

Aos meus pais, esposa e filhas.

AGRADECIMENTOS

- Em primeiro lugar à Deus por me possibilitar alcançar esse sonho em minha vida;
- Aos meus pais Francisco e Lucy por todo o amor a mim dedicado durante toda a vida.
- À minha família, irmãos e amigos, pelo apoio e compreensão durante todos esses meses de dedicação e restrição de tempo;
- À minha querida esposa Josy por compreender que esse sonho precisava ser concretizado.
- Ao Professor Dr. José Lassance de Castro Silva pela excelente orientação deste trabalho e por quem tenho especial apreço.
- Ao Professor Dr. João Bosco Furtado Arruda por seus belíssimos ensinamentos em sala de aula e pelo desenvolvimento de nossa capacidade crítica.
- Ao amigo Denis Carvalho Rocha Freire pelo grande apoio computacional nas aplicações dos algoritmos.
- À todos os colegas, professores e funcionários do Programa de Mestrado em Logística e Pesquisa Operacional/GESLOG-UFC pela convivência harmoniosa durante todo o período do curso.

"A ciência humana de maneira nenhuma nega a existência de Deus. Quando considero quantas e quão maravilhosas coisas o homem compreende, pesquisa e consegue realizar, então reconheço claramente que o espírito humano é obra de Deus, e a mais notável."

Galileu Galilei

RESUMO

O objetivo deste trabalho foi propor uma aplicação baseada na metaheurística Busca Tabu (TS) para ser utilizada em serviços de manutenção preventiva em campo (FPMS) a fim de obter maior eficiência logística, através do roteamento de setores de manutenção. Ao contrário dos serviços realizados na indústria, onde todos os sistemas, máquinas e equipamentos estão localizados praticamente no mesmo local, serviços de manutenção em campo requerem um componente adicional diretamente relacionado ao custo, que se refere exatamente a diferença entre a unidade de base e local de trabalho. Serviços em campo podem ser considerados uma variação do Problema do Caixeiro Viajante (PCV) e suas diferentes abordagens, como o Problema Dinâmico do Reparador Viajante (DTRP - *Dynamic Travelling Repairman Problem*) proposto por Bertsimas e Van Ryzin. Em situações práticas do dia-a-dia existe uma enorme demanda por serviços de manutenção a serem realizados em campo, demonstrando sua relevância: elevadores, escadas rolantes, aparelhos segurança eletrônica residencial, suporte de TI à *hardwares*, entre outros. O método foi implementado e testado em problemas da biblioteca TSP-LIBRARY variando de 17 a 280 pontos. Boas soluções foram encontradas em um tempo de processamento aceitável. O *input* do problema leva em consideração duas formas: coordenadas geográficas ou coordenadas cartesianas. Para uma aplicação prática do mundo real, foi considerada uma empresa de manutenção em elevadores e os resultados também foram eficientes, reduzindo bastante os custos de transporte e a logística empregada na operação.

Palavras-chave: otimização, manutenção em campo, PCV, Busca Tabu.

ABSTRACT

The aim of this paper was to propose an application based on the Metaheuristic Tabu Search (TS) to be used on FIELD PREVENTIVE MAINTENANCE SERVICES (FPMS) in order to get more logistics efficiency by routing maintenance sectors. Unlike services performed in industry, where all systems, machines and equipment are located practically in the same location, maintenance services in the field require an additional component directly related to cost, which refers to exactly offset between the base unit and jobsite. Services in the field can be considered a variation of the Travelling Salesman Problem (TSP) and its different approaches, like the DTRP (Dynamic Travelling Repairman Problem) proposed by Bertsimas and Van Ryzin. There is a huge demand for maintenance in the field, demonstrating its relevance: elevators, escalators, electronic devices for home-security, IT hardware support and others. The method was designed, implemented and tested in problems of the TSP-LIBRARY ranging from 17 up to 280 points. Good solutions were found in a acceptable processing time. The input data can be made by geographical coordinates or 2D-coordinates. For a real-world application, it was considered an Elevator Company and the results were also efficient, greatly reducing transportation cost and logistics used in the operation.

Key-words: optimization, maintenance on field, TSP, Tabu Search.

LISTA DE FIGURAS

Figura 1.1 – Fatores de Influência na gestão da manutenção	20
Figura 2.1 – A função manutenção na perspectiva do tempo	27
Figura 2.2 – Níveis de maturidade da manutenção	28
Figura 2.3 – Relação entre manutenção e logística para peças de reposição	29
Figura 3.1 – Uma solução para o PCV com 13 cidades	34
Figura 3.2 – Solução ótima para PCV com 13 cidades	34
Figura 3.3 – Ilustração para o PCV com 44 cidades	35
Figura 3.4 – Ilustração de um $G = (N, A)$	36
Figura 3.5 – Heurística ‘vizinho mais próximo’	38
Figura 3.6 – Movimento <i>swap</i>	39
Figura 3.7 – Representação gráfica do método r-OPT da troca de nós da rota	41
Figura 3.8 – Pseudo-código do método HP	44
Figura 3.9 – Pseudo-código para metaheurística SA	45
Figura 3.10 – Pseudo-código para metaheurística GA	46
Figura 3.11 – Ilustração do KTR	47
Figura 4.1 – Trajetória de uma metaheurística TS	54
Figura 4.2 – Ilustração do movimento na fila <i>tabu</i>	58
Figura 4.3 – Estruturas de memória utilizadas na TS	59
Figura 5.1 – Organização em zonas e rotas (setores) de manutenção preventiva	66
Figura 5.2 – Otimização de uma rota de manutenção para o TRP	67
Figura 5.3 – Distância linear x Distância Real	74
Figura 5.4 – Variações entre distância linear e real para capitais brasileiras	75
Figura 5.5 – Tela principal do aplicativo r-TS	76
Figura 5.6 – Parâmetros de calibragem do r-TS	82
Figura 5.7 – Demonstração do critério de aspiração no r-TS	83
Figura 5.8 – Intensificação no r-TS para o problema gr24 (TSPLIB)	84
Figura 5.9 – Evolução gráfica do resultado do problema gr24 (TSPLIB)	85

Figura 6.1 – Gráfico comparativo entre soluções obtidas para problemas simétricos do TSPLIB até 51 pontos	87
Figura 6.2 – Desvios % x Pontos para diversas instâncias do TSPLIB	88
Figura 6.3 – Tempo de processamento	89
Figura 6.4 – Tempo de processamento comparativo entre r-TS e AGHC para problemas TSPLIB de 0 a 280 pontos	91
Figura 6.5 – Ilustração da subdivisão manual de rotas pelo Google maps	93
Figura 6.6 – Tela do aplicativo GPS Tool	95
Figura 6.7 – Matriz D_{ij} para o setor 3 / zona 2	97
Figura 6.8 – Calibração do r-TS para problema prático (setor 3 / zona 2)	97
Figura 6.9 – Resultado do r-TS para problema prático (setor 3 / zona 2)	98
Figura 6.10 – Comparação entre rota inicial e rota gerada pelo r-TS	100
Figura 6.11 – Redução percentual da distância total das rotas pelo r-TS	101

LISTA DE TABELAS

Tabela 3.1 – Explosão combinatória	32
Tabela 3.2 – Número de permutações (soluções) avaliadas pelo método HP	42
Tabela 3.3 – Comparativo entre permutações geradas pelos métodos HP e HP*	43
Tabela 5.1 – Tipos de serviços de manutenção em campo para uma empresa de elevadores	64
Tabela 6.1 – Resultados obtidos por r-TS	87
Tabela 6.2 – Calibrações utilizadas no r-TS	89
Tabela 6.3 – Resultados comparativos entre r-TS e AGHC	90
Tabela 6.4 – Organização da <i>LIFTECH</i> em zonas e setores de manutenção	94
Tabela 6.5 – Seqüência estabelecida no sistema da empresa para o setor 3 / zona 2	96
Tabela 6.6 – Resultados gerais obtidos na <i>LIFTECH</i>	99

LISTA DE QUADROS

Quadro 4.1 – Estrutura básica de um código TS	56
Quadro 5.1 – <i>Procedure</i> ‘calcula_dist_euclidiana’	72
Quadro 5.2 – <i>Procedure</i> ‘calcula_dist_geográfica’	73
Quadro 5.3 – Variáveis utilizadas no r-TS	77
Quadro 5.4 – Funções e procedimentos do código r-TS	77
Quadro 5.5 - <i>Procedure</i> ‘Tform1.IniciaIteracao’	78
Quadro 5.6 – Função ‘Tform1.DistanciaTotalSequenciaAtual’	79
Quadro 5.7 – Função e procedimento ‘FilaTabu’	80
Quadro 5.8 – Função ‘Tform1.InformaSequenciaAtual’	80
Quadro 5.9 – Função ‘Tform1.InformaListaTabu’	80

LISTA DE ABREVIATURAS E SIGLAS

AHC	: Algoritmos Heurísticos Construtivos
AHM	: Algoritmos Heurísticos de Melhoramento
$Asp()$: Função do critério de aspiração
BLP	: <i>Bounded Latency Problem</i>
BT	: Busca <i>Tabu</i>
Delta_f	: $\text{Min} \{ f_i - f_{i-1} \}$
F	: Fila <i>tabu</i>
GA	: <i>Genetic Algorithms</i>
GKTRP	: <i>Generalized k-Travelling Salesman Problem</i>
MLP	: <i>Minimum Latency Problem</i>
NMÁX	: Número máximo de iterações
NV	: Número de vizinhos
M	: Iteração em que foi obtida a melhor iteração
f	: Função objetivo
f*	: Limite de aceitação da solução
i, j	: Variáveis de <i>loops</i> .
$Min()$: Função que determina o menor valor de um conjunto de dados (para problemas de minimização)
PCV	: Problema do Caixeiro Viajante
POCP	: Problema de Otimização Combinatória Permutacional
PSP	: Provedor de Serviços Logísticos
S	: Configuração final da solução encontrada pelo TS
S^*	: Melhor configuração de uma determinada iteração
SA	: <i>Simulated Annealing</i>
S_0	: Configuração da solução inicial viável
S_{ij}	: Configuração da solução “j” vizinha da “i”.
$Swap$: Função que realiza uma perturbação em relação a uma solução existente
T	: Tamanho da fila <i>tabu</i>
TS	: <i>Tabu Search</i>
TSP	: <i>Travelling Salesman Problem</i>
KTSP	: <i>Multi(k)-Travelling Salesman Problem</i>

KTRP : *Multi(k)-Travelling Repairman Problem*
TRP : *Travelling Repairman Problem*
VRP : *Vehicle Routing Problem*

SUMÁRIO

Resumo	viii
<i>Abstract</i>	ix
Lista de Figuras	x
Lista de Tabelas	xii
Lista de Quadros	xiii
Lista de Abreviaturas e Siglas	xiv
1. INTRODUÇÃO	
1.1 Justificativa e relevância	18
1.2 O problema do caixeiro viajante aplicado às soluções logísticas	20
1.3 Objetivos	21
1.4 Delineamento e organização do trabalho	22
2. LOGÍSTICA NAS OPERAÇÕES DE SERVIÇOS DE MANUTENÇÃO	
2.1 Logística nas operações de serviços	24
2.2 Uma breve perspectiva da manutenção e sua relação com a Logística	27
3. PROBLEMA DO CAIXEIRO VIAJANTE	
3.1 Considerações Iniciais	31
3.2 O modelo matemático do PCV	35
3.3 Métodos gerais de solução para POCP e PCV	37
3.3.1 Heurística <i>NN- Nearest neighbor</i> ('vizinho mais próximo')	37
3.3.2 <i>Swap</i>	38
3.3.3 <i>Savings</i> de Clarke e Wright	39
3.3.4 <i>r – OPT</i> de Lin-Kernighan	40
3.3.5 Heurística Permutacional (<i>HP</i>)	42
3.3.6 Metaheurísticas	44
3.4 Variações do PCV: <i>k-Travelling Repairman Problem</i> (KTR)	47
3.5 Aplicações logísticas do PCV	48
4. METODOLOGIA	
4.1 Heurísticas e Metaheurísticas	50
4.2 <i>Tabu Search</i> (TS)	52
4.2.1 Introdução	52
4.2.2 Características de um algoritmo TS	55
4.2.3 Elementos utilizados em um algoritmo TS	57
4.2.4 A Fila <i>Tabu</i>	57
4.2.5 A função <i>Asp()</i> – Critério de Aspiração	58
4.2.6 Conceitos básicos acerca de estruturas de memória	59
4.2.7 Diversificação e Intensificação	59
4.3 Utilização do TS em soluções de problemas na área de Logística	60
	xvi

5. CARACTERIZAÇÃO DO PROBLEMA E ESTRATÉGIA DE SOLUÇÃO

5.1 Serviços de manutenção periódica em campo	61
5.2 Tipos de serviço de manutenção em campo	62
5.3 A manutenção periódica de elevadores	64
5.3.1 A organização da manutenção de campo em zonas e setores	65
5.3.2 A periodicidade de alguns serviços de manutenção	67
5.3.3 O Sistema de atendimento em empresas de manutenção	68
5.3.4 Diferenças no atendimento de manutenções	69
5.3.5 A delimitação do estudo	70
5.4 Estratégia de solução para o problema	70
5.4.1 Visão geral da solução proposta	70
5.4.2 O cálculo das distâncias	72
5.4.2.1 Distâncias euclidianas (lineares) e reais	73
5.4.3 O algoritmo r-TS	75
5.4.3.1 O algoritmo básico do r-TS	77
5.4.3.2 Calibragem inicial do r-TS	81
5.4.3.3 Critério de aspiração e a Intensificação no r-TS	82

6. EXPERIMENTOS COMPUTACIONAIS

6.1 Aplicação aos problemas clássicos da literatura TSPLIB	86
6.1.1 Comparativo com resultados obtidos com outra metaheurística ..	90
6.2 Aplicação a um caso prático: manutenção de elevadores	91
6.2.1 O problema da empresa Elevadores “ <i>LIFTECH</i> ”	91
6.2.2 “ <i>LIFTECH</i> ” em Fortaleza-CE	93
6.2.3 Procedimentos realizados	95
6.2.3.1 O r-TS adaptado ao problema prático	96
6.2.4 Resultados obtidos	98

7. CONSIDERAÇÕES FINAIS

7.1 Conclusões	102
7.2 Sugestões para trabalhos futuros	103

REFERÊNCIAS BIBLIOGRÁFICAS	104
---	------------

CAPÍTULO 1 – INTRODUÇÃO

1. Justificativa e relevância do trabalho

A produtividade e a otimização de processos dentro das organizações industriais e de serviços são temas relevantes dentro do cenário de alta competitividade global. Ao observarmos as tendências das economias atuais e dos negócios dentro das organizações, é visível a importância atual e a relevância de estudos na área de logística urbana relacionada a *serviços*. No Brasil, as estatísticas confirmam a tendência já observada nos países desenvolvidos desde a década de 90, onde se constata o aumento gradativo da participação do setor de serviços nas economias. Conforme Corrêa (2002), quase 60% da população economicamente ativa dedica-se à atividade de serviço, o que torna a participação desse segmento fundamental para a composição do PIB nacional. Em 2010, de acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE), a participação do setor de serviços no PIB superou a casa dos 65%, com mais de R\$ 2,12 trilhões em valor adicionado.

Larrañaga (2008) destaca que um dos maiores desafios atuais da logística está relacionado ao ganho de competitividade empresarial em termos de redução de custos e aumento de produtividade, com consequência direta sobre o nível de serviço e geração de valor agregado aos clientes. Sistemas logísticos eficientes e eficazes, ao permitir a redução do custo final dos produtos, significam um melhor padrão de vida para toda a sociedade. Para organizações que atuam fortemente no segmento de *serviços*, tais como as operações de coleta, distribuição e serviços de manutenção periódica de máquinas e equipamentos, por exemplo, torna-se mais evidente a relevância de estudos relacionados à designação e otimização de rotas.

O uso de técnicas baseadas em Pesquisa Operacional (PO) auxilia o tomador de decisão na geração de vantagem competitiva, tendo em vista a complexidade associada aos problemas de logística urbana, notadamente nas grandes metrópoles brasileiras. A complexidade de um sistema real resulta do fato de que seu comportamento é influenciado por um número muito grande de elementos ou variáveis. Exemplos de aplicações da PO podem ser observados na determinação do custo mínimo de produção, maximização de lucros, otimização de rotas, resolução de problemas de transporte, designação e sequenciamento, entre outros. Segundo Silva (2010), o ponto chave da

Pesquisa Operacional reside na construção de modelos matemáticos, a partir dos quais escolhe-se uma técnica adequada, eficiente e eficaz para resolvê-lo.

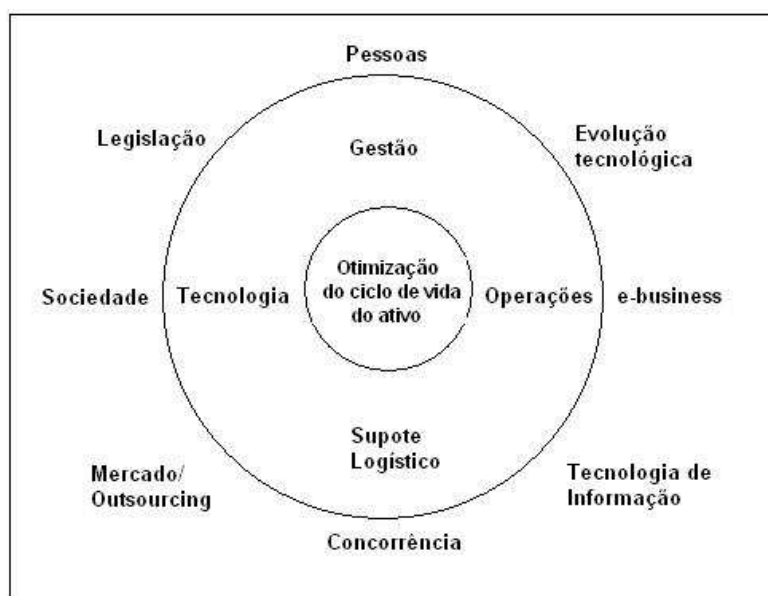
A ausência ou incerteza em relação às informações disponíveis sobre cada variável que influencia o sistema real prejudica diretamente a formulação de um plano de trabalho ou de um planejamento de serviços, reduzindo a produtividade e normalmente incorrendo em custos adicionais elevados. Segundo Andrade (2000), o esforço despendido para a modelagem de um problema leva a uma compreensão mais profunda do próprio problema, identificando melhor seus elementos internos, suas variáveis principais, suas interações com o ambiente externo, as informações necessárias e os resultados possíveis de obter. Essa abordagem quantitativa dos problemas fornece uma estrutura de raciocínio e análise que permite desenvolver a visão sistêmica do processo.

No caso específico da atividade de manutenção periódica de equipamentos em rota mensal, por exemplo, o roteamento otimizado contribui para o ganho de produtividade, permitindo uma gestão mais eficiente das operações logísticas da organização. Segundo Pintenlon e Parodi-Herz (2008), a maior parte da literatura sobre Gestão da Manutenção aborda que manutenção é um conjunto de atividades requeridas para manter ativos fixos numa condição desejada de operação ou restabelecê-los a esta condição. Ainda de acordo com os autores, em um ponto de vista pragmático, o objetivo-chave da gestão da manutenção é a otimização do ciclo de vida total do ativo, conforme ilustra a Figura 1.1 dada a seguir, cujo processo é complexo, visto que envolve diversas expertises, tais como tecnologia, técnicas de gerenciamento, operações e suporte logístico.

Neste contexto, a Pesquisa Operacional, através de técnicas de resolução exata ou aproximada, como as obtidas através de otimização combinatória permutacional, pode contribuir diretamente para ganho de eficiência operacional. Ainda de acordo com Silva (2010), a PO também engloba um conjunto de técnicas direcionadas a problemas complexos voltados para a tomada de decisões em empresas.

Este trabalho trata exclusivamente do desafio logístico de se planejar e organizar rotas de serviços de manutenção preventiva em campo, a fim de obter-se o melhor planejamento possível ao menor custo de deslocamento. À medida que o número de clientes (ou equipamentos) a serem visitado mensalmente cresce significativamente, a gestão direta sobre o processo torna-se complexa e exige o devido suporte de técnicas adequadas de roteirização.

Figura 1.1 – Fatores de influência na Gestão da Manutenção.



Fonte: Kobbacy (2008).

1.2 O problema do caixeiro viajante aplicado às soluções logísticas

O Problema clássico do Caixeiro Viajante (PCV), denominado na literatura de TSP (*Travelling Salesman Problem*) consiste em um dos mais tradicionais e conhecidos problemas da Pesquisa Operacional. Seu objetivo é definir a melhor rota (menor distância, menor custo ou menor tempo, por exemplo) de um circuito de pontos ou localidades para que determinado agente percorra todos os pontos a serem visitados uma única vez, sem repetição, de modo a minimizar um determinado critério.

Reinet (1994) *apud* Cunha *et. al.* (2002) afirma que o PCV é um dos mais proeminentes dentre um amplo conjunto de problemas de otimização combinatória. Segundo o autor, o PCV tem atraído pesquisadores de diferentes campos, entre os quais pesquisa operacional, matemática, física, biologia, inteligência artificial entre outros. Apesar de sua definição aparentemente simples, o PCV até hoje é um desafio da Pesquisa Operacional e centenas de artigos já foram publicados sobre o tema (CUNHA *et. al.*, 2002).

O PCV tem sua relevância consolidada ao longo dos anos por sua grande aplicação prática e pela sua dificuldade na obtenção de soluções exatas. O PCV pertence à classe de problemas considerada difícil ou intratável. Viana (1998), aborda a classificação dos

problemas de otimização combinatória permutacional baseada na teoria fundamentada por Cook (1971). De acordo com sua complexidade de tempo, os problemas poderão enquadrar-se nas classes P-polinomial ou NP-não polinomial. O PCV está contido na classe NP-*hard*, que contém problemas de maior dificuldade dentre todos os NP. O alcance de soluções exatas, à medida que o número de pontos ou localidades aumenta, exige recursos significativos de tempo computacional, sendo, às vezes, inviável em virtude do seu processamento nos computadores. Para problemas mais complexos (notadamente os práticos), são utilizadas técnicas aproximativas que normalmente obtêm boas soluções em um tempo computacional aceitável. Tais técnicas são conhecidas como heurísticas ou metaheurísticas.

Uma heurística é uma técnica que procura boas soluções com um tempo computacional razoável sem garantir necessariamente nem viabilidade nem soluções exatas ou ótimas. Já a metaheurística possui, além das características anteriormente citadas, a capacidade adicional de diversificar o campo de soluções e “escapar” de um ótimo local.

Ruiz-Vanoye e Díaz-Parra (2010) descrevem que a meta-heurística Tabu Search (TS) tem por base três princípios:

- i) Uso de uma estrutura de dados para memorizar o histórico da evolução do processo de busca;
- ii) Uso de um mecanismo de controle para fazer um balanceamento entre a aceitação, ou não, de uma nova configuração, com base nas informações registradas na fila tabu referentes às restrições e aspirações desejadas;
- iii) Procedimentos de estratégia de intensificação e diversificação utilizando memórias de curto e longo prazos.

Com base nestas afirmações, propomos resolver o problema específico aqui abordado através desta técnica.

1.3 Objetivos

O objetivo principal do trabalho é estudar e resolver o problema logístico dos serviços de manutenção preventiva em campo através da metaheurística *Tabu Search* (TS). Aplicamos o Problema clássico do Caixeiro Viajante ao contexto do problema

específico, a fim de obter boas soluções práticas através da simulação de rotas a serem percorridas (menor distância total) para cada setor de trabalho. As soluções práticas poderão ser comparadas com os dados do sistema real usado em uma empresa elencada no contexto desse trabalho. O estudo de caso aborda uma empresa multinacional de prestação de serviços de manutenção em elevadores.

Os objetivos específicos elencados abaixo representam o caminho para alcance do objetivo principal previamente definido:

- i) Caracterizar o PCV, suas particularidades, modelo e relevância para problemas da área de logística;
- ii) Contextualizar o PCV para aplicações voltadas à otimização de rotas de manutenção, destacando sua influência na eficiência logística da operação;
- iii) Desenvolver um procedimento específico para solução de problemas práticos de otimização de rotas de manutenção preventiva em campo;
- iv) Criar um sistema computacional para fins didáticos (com entrada de dados em coordenadas 2D ou Matricial) e práticos (entrada de dados em coordenadas geográficas);
- v) Avaliar os resultados do procedimento desenvolvido comparando com as soluções obtidas em problemas de referência publicados por Beasley (TSP-LIBRARY);
- vi) Comparar os resultados com soluções obtidas por outros procedimentos;
- vii) Aplicar o método a um problema real de otimização de rotas de manutenção preventiva, comparando os resultados obtidos com os dados anteriormente coletados na empresa objeto do estudo.

1.4 Delineamento e organização do trabalho

Com base nos objetivos definidos, a metodologia científica utilizada classifica-se como uma pesquisa exploratória com estudo de caso. Segundo Gil (2008), o estudo de caso é uma modalidade de pesquisa amplamente utilizada e consiste no estudo profundo e exaustivo de um ou poucos objetos, de maneira que permita seu amplo e detalhado conhecimento. Esta metodologia, de acordo com Yin (2001) *apud* Gil (2008), é

encarada como o delineamento mais adequado para a investigação de um fenômeno contemporâneo dentro de seu contexto real.

O trabalho está dividido em sete capítulos. O Capítulo 1 é a introdução, onde constam a relevância do tema, a contextualização do problema, o objetivo geral, os objetivos específicos e a metodologia utilizada para o desenvolvimento do estudo realizado. O Capítulo 2 refere-se ao estudo da importância da logística em operações do segmento de serviços e sua contribuição para a área de manutenção. No Capítulo 3 há uma revisão da literatura científica acerca do Problema clássico do Caixeiro Viajante, onde estão apresentadas as principais características do problema, seu modelo matemático e as mais recentes técnicas de resolução. O Capítulo 4 trata da técnica de solução de problemas de otimização combinatória através da metaheurística Tabu Search. São apresentadas neste capítulo as principais definições, aplicações e particularidades do método. A caracterização do problema e a estratégia da solução são apresentados no Capítulo 5. Os experimentos computacionais constam no Capítulo 6, onde foram realizados testes com dados científicos de referência na TSLIB e uma aplicação a um problema real de otimização de manutenção preventiva periódica. As conclusões do trabalho e a proposição de estudos continuados a serem desenvolvidos são apresentados no Capítulo 7.

CAPÍTULO 2 – LOGÍSTICA EM OPERAÇÕES DE SERVIÇOS DE MANUTENÇÃO

Este capítulo está dividido em três seções. A primeira seção (2.1) apresenta os aspectos característicos da Logística nas operações de serviços, sua evolução e aplicações no setor de serviços. Na segunda seção (2.2) é apresentada uma breve perspectiva evolucionária do conceito de manutenção nas organizações e sua relação com a área de logística. Na terceira seção (2.3) é caracterizado o problema da pesquisa através da definição dos principais elementos de um serviço de manutenção periódica de campo, com destaque para o setor de manutenção de elevadores.

2.1 Logística em operações de Serviços

A palavra logística tem a origem no verbo francês *loger*, que significa alojar e era relacionada ao abastecimento militar de grandes exércitos com tudo o que era necessário para a batalha na linha de frente. O desenvolvimento histórico da logística é comumente dividido na literatura em três períodos de características diferentes: antes de 1950, de 1950 a 1980, e posterior a 1980. Já é possível, inclusive, destacar alguns fatores recentes que caracterizam a logística no século XXI. No período que antecedeu 1950, não houve avanços significativos no campo da logística empresarial. Porém, em razão da 2ª Guerra Mundial, muitas operações militares utilizaram estratégias logísticas eficazes, que posteriormente fundamentaram o início do que hoje é a Pesquisa Operacional. No período seguinte (51-80), começaram a aparecer os primeiros livros sobre logística e em 1963 foi criado o CLM – *Council of Logistics Management*. Nesta época já se registravam as tendências de reduções de custos e avanços tecnológicos incipientes. Ballou (2010) destaca esses anos como o período do desenvolvimento, tendo em vista que representou a decolagem para a teoria e a prática da logística.

Posteriormente a 1980, os avanços em tecnologia e nas telecomunicações levaram a uma ampliação do conceito de logística até o conceito de SCM – *Supply Chain Management*. O século XXI testemunha a transformação digital das empresas e o rearranjo de integração logística. Organizações que não se enquadrarem rapidamente no EDI (*Electronic Interchange Data*) e no SPED (Sistema de Processamento Eletrônico e

Digital) brasileiro certamente estarão à margem da evolução imposta pela tecnologia à cadeia de suprimentos.

Peres (2006) disserta sobre a evolução da logística, demonstrando a dinâmica de seu conceito e aplicação:

A logística é um conceito em constante evolução, atrelado à busca de ganhos de competitividade e níveis de custos reduzidos em função do desafio global e da necessidade de agir de modo rápido frente às alterações ambientais. [...] Era vista como função de apoio, não vital ao sucesso dos negócios. Em uma velocidade impressionante, tal visão vem sendo alterada em direção ao reconhecimento da logística como elemento estratégico.

Decorridos mais de sessenta anos do fim da guerra, a Logística apresentou uma evolução continuada, sendo considerada atualmente como um dos elementos-chave na estratégia competitiva das empresas.

A logística empresarial tem como objetivo prover o cliente com os níveis de serviços desejados e a meta de nível de serviço logístico é providenciar bens ou serviços corretos, no lugar certo, no tempo exato e na condição desejada ao menor custo possível (BALLOU, 2010).

Em todos os países desenvolvidos, o setor de serviços ocupa posição de destaque na economia. De acordo com Corrêa (2002), o principal indicador utilizado para evidenciar esse fato tem sido a participação do setor de serviços na ocupação da mão-de-obra e na geração de riqueza, refletida pelo Produto Interno Bruto. No Brasil, as estatísticas não se mostram diferentes da tendência apresentada no mundo industrializado. Em termos de ocupação de mão-de-obra, o setor de serviços vem aumentando sua participação nas últimas décadas e praticamente 60% da população brasileira dedica-se a atividade de serviços. Os dados do IBGE classificam, no setor de serviços, as atividades de comércio, transportes, comunicações, instituições financeiras, administração pública, aluguéis e outros serviços. Em relação ao Produto Interno Bruto, o setor de serviços já é responsável por 65% das riquezas geradas no país.

Notadamente em organizações que atuam essencialmente com prestação de serviços, a logística deve ser administrada como um esforço integrado para atingir a satisfação do cliente pelo menor custo total. Esse processo de geração de valor está diretamente

relacionado com o nível de serviço logístico desempenhado pela empresa. Segundo Bowersox *et. al.*(2007), a chave para alcançar a liderança logística é conhecer a fundo a arte de equilibrar a competência operacional e o compromisso com o atendimento às expectativas e solicitações fundamentais dos clientes. O autor destaca também que o desempenho logístico básico é medido em termos de disponibilidade, desempenho operacional e confiabilidade do serviço logístico. A disponibilidade envolve dispor de recursos (estoque, no caso de produtos, ou pessoal, para serviços) de modo a atender as constantes necessidades dos clientes. O desempenho operacional trata do tempo necessário para entregar um pedido ou serviço demandado pelo cliente. O desempenho é afetado por outros fatores como velocidade, consistência e flexibilidade no atendimento às solicitações dos clientes. Por fim, a confiabilidade do serviço, que envolve os atributos relacionados à qualidade da logística e que representa a medição precisa da disponibilidade e do desempenho operacional.

Ballou (2010) destaca que as organizações de serviços têm muitos problemas logísticos e que o campo da logística, até hoje, pouco fez para ter reconhecida importância na administração de empresas em geral. A expectativa é que, à medida que a logística amadureça, muito mais será escrito sobre problemas logísticos em empresas (firmas) de serviços. Problemas tais como a localização de facilidades e o atendimento a serviços de manutenção telefônica são exemplos destacados pelo autor.

Como se observa na literatura, apesar de representar 65% do PIB brasileiro, ainda não há disponível um volume significativo de publicações voltadas especificamente à área de logística em serviços. É provável que este seja um dos fatores para justificar o vasto campo de melhora para os serviços prestados pelas empresas brasileiras. A título de exemplo, estão listadas abaixo algumas empresas e organizações de serviços, com destaque para a sociedade em geral:

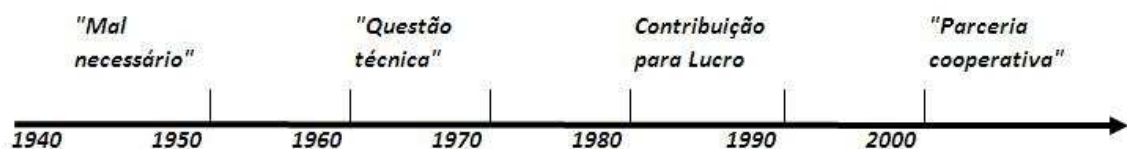
- i. Empresas de transporte em geral;
- ii. Empresas especializadas em manutenção de máquinas e equipamentos;
- iii. Correios e postagens;
- iv. Empresas intermediárias de coleta e distribuição;
- v. Serviços públicos de segurança e saúde paramédica;
- vi. Outros.

Como se observa pelos diferentes tipos de empresas de serviços, é demasiado complexo definir estratégias logísticas unificadas para serviços, visto que existem variáveis e níveis de intensidade de fatores diferentes para cada empresa. Tais fatores, como o nível de personalização (customização do serviço), intensidade do contato entre empresa e cliente, ênfase em pessoas ou tecnologia (grau de relação do serviço) e a predominância de atividades de *back* ou *front Office*, precisam ser cuidadosamente analisados para um estudo estratégico de logística em serviços. Para o exemplo de empresas especializadas em serviços de manutenção, é fundamental uma grande confiabilidade no contato entre empresa e cliente, visto que o *front office* predomina na avaliação do serviço pelo cliente.

2.2 Uma breve perspectiva da manutenção e sua relação com a Logística

Assim como a área de logística, a manutenção também passou por uma evolução significativa dentro das organizações empresariais. Pintenlon e Parodi-Herz (2008) sugerem que o conceito de manutenção foi alterado dramaticamente desde a década de 40, conforme ilustra a Figura 2.1 dada a seguir. Nessa época, a manutenção era nada mais que uma inevitável parte necessária ao processo produtivo. A evolução ocorreu passo-a-passo, com grandes avanços a cada vinte anos, como se pode observar nas décadas de 60 e 80, quando a manutenção passou a ter um conceito mais técnico e, posteriormente, como uma função que contribuía para o lucro das organizações. Nos anos 2000, a atividade já era parte do conceito cooperativo de resultado, interagindo diretamente com a produção. Atualmente, com a complexidade das instalações, a alta tecnologia e o necessário ganho contínuo de produtividade como forma de garantia de competitividade em um mercado altamente globalizado, a manutenção tornou-se um elemento estratégico que suporta os objetivos do negócio de uma organização produtiva.

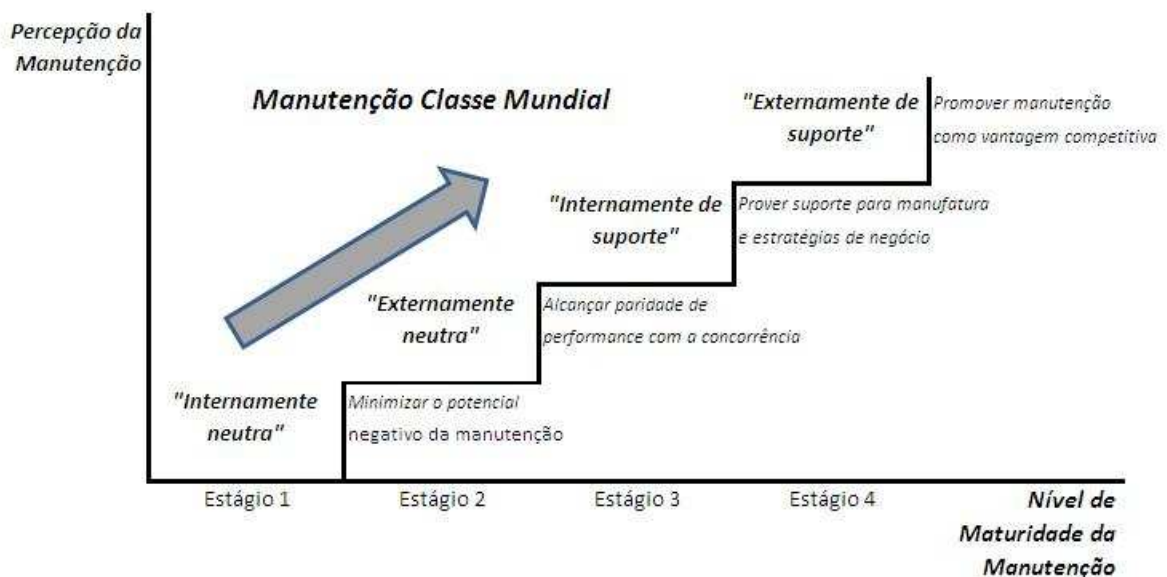
Figura 2.1 – A função manutenção na perspectiva do tempo.



Fonte: Pintenlon e Parodi-Herz (2008).

A maioria dos autores de livros e artigos sobre gestão da manutenção concorda com a definição de que manutenção é o “conjunto de atividades requeridas para manter ativos físicos na condição de operação desejada ou restabelecê-los a esta condição”. Em uma visão mais pragmática, Pintenlon e Parodi-Herz (*op. cit.*) destacam que o objetivo-chave da manutenção é a otimização do ciclo de vida total dos ativos, ou seja, em outras palavras é a maximização da disponibilidade e confiabilidade de todo o sistema produtivo, a fim de atender às expectativas de produção com a qualidade requerida e em um tempo adequado.

Figura 2.2 – Níveis de maturidade da manutenção.

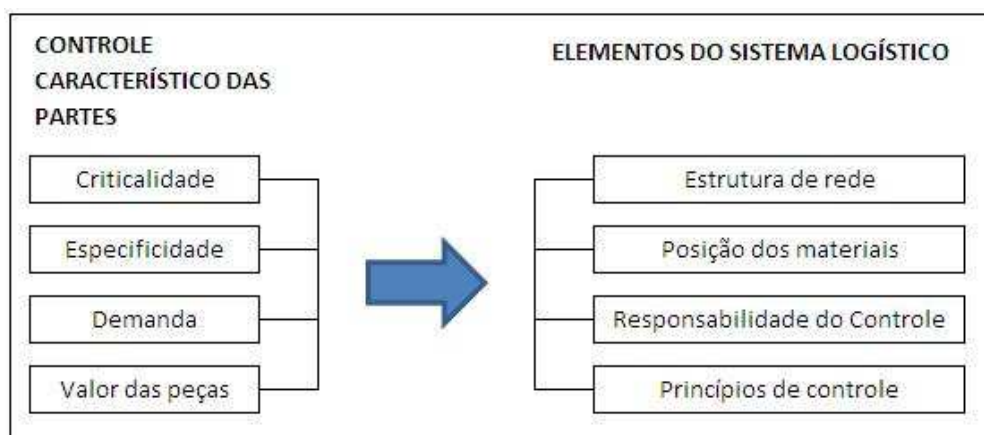


Fonte: Pintenlon e Parodi-Herz (2008).

A Figura 2.2, dada acima, apresenta os quatro estágios da evolução da maturidade da manutenção, iniciando de um estágio interno neutro, onde o objetivo era minimizar seu próprio “potencial negativo”, passando por outros dois estágios (2 e 3) onde a função manutenção evoluiu para o conceito de alcance de *performance* semelhante à concorrência e, em seguida, sendo tratada como uma área de suporte à produção e à estratégia do negócio, até chegar ao nível de manutenção classe mundial, onde a função passa a exercer um papel de vantagem competitiva do negócio, alicerçada por tecnologia, estratégias em logística e gestão operacional.

Na literatura, diversos trabalhos têm sido publicados de forma a possibilitar esse ganho de maturidade através de uma evolução operacional e de otimização da logística da manutenção. Huiskonen (2001) abordou a relação existente entre as características de controle de peças de reposição para uma eficaz gestão da manutenção e a estratégia logística a ser adotada para alcance dos objetivos. Esta relação está ilustrada na Figura 2.3, dada abaixo.

Figura 2.3 – Relação entre manutenção e logística para peças de reposição.



Fonte: Huiskonen (2001).

Dhaliwal (1986) foi um dos primeiros a abordar técnicas de Inteligência Artificial (AI) na solução de problemas de manutenção e sistemas complexos de engenharia. Kobaccy (2008) publicou diversas técnicas de AI aplicadas à manutenção, tais como: KBS (*Knowledge based systems*), CBR (*Case based reasoning*), GAs (*Genetic Algorithms*), NNs (*Neural Networks*), FL (*Fuzzy Logic*), entre outras. Kobbacy já havia publicado em 1992 um artigo sobre KBS aplicado à melhoria de rotinas de manutenção.

Outros trabalhos na área de otimização de serviços de manutenção foram desenvolvidos por Blakeley *et. al.* (2003), Biggs e Zuo (2006), Sheo e Kuo (2006), Damaso e Garcia (2009) e Berkholz (2009). Nestes artigos são apresentados modelos para otimização de manutenção preventiva, modelos de planejamento de rotas (*routing-scheduling*) e resultados de redução de custos.

Blakeley *et.al.* (2003) demonstra, através de um modelo matemático, a complexidade logística para a solução do problema de roteirização com um caso prático de uma empresa multinacional de manutenção de elevadores e escadas rolantes. O artigo apresenta o passo-a-passo do desenvolvimento de um sistema de planejamento logístico automatizado, desenvolvido em parceria com a ESRI (*Environmental Systems Research Institute*) e utilizando técnicas computacionais e a metaheurística Tabu Search. Nele é apresentado um resultado com otimização de mais de US\$ 1 milhão de economia com *payback* em menos de um ano da implantação do novo sistema.

Berkholz (2009) destaca que a logística de manutenção é complexa quando os dados de demanda são voláteis ou incertos. Sua abordagem é voltada à necessidade de um modelo de previsão de capacidade através de pré-requisitos para ganho de confiabilidade e tempo de resposta. Essa ótica complementa a visão anteriormente relatada por Huiskonen (2001).

Diversos outros artigos poderiam ser citados de forma a enaltecer a relação direta entre a logística em operação de serviços e a gestão de manutenção nas organizações, tais situações críticas de *performance* como a logística na manutenção de aeronaves e logística de manutenção de frota de veículos de um Provedor de Serviços Logísticos (PSP) (Huang e Yao, 2008). Isto se deve ao vasto campo de estudo dessas duas grandes áreas, que atualmente são tratadas como funções estratégicas nas organizações por contribuírem diretamente no nível de serviço e para a adição de valor econômico ao serviço, garantindo mais competitividade no mercado para as organizações. Porém, ainda percebe-se um distanciamento entre o campo da ciência e as empresas, o que nos leva a perceber que apenas uma seleta parcela de empresas utiliza técnicas ou modelos de otimização em seus processos logísticos ou na gestão da manutenção.

CAPÍTULO 3 – PROBLEMA DO CAIXEIRO VIAJANTE

Este Capítulo encontra-se dividido em cinco seções. A seção 3.1 contém as considerações iniciais sobre o problema do caixeiro viajante (PCV), tais como o conceito, as características do problema, sua complexidade de solução e ilustrações exemplificativas. A segunda seção apresenta o modelo matemático do problema e traz comentários sobre a função objetivo e suas restrições. Os principais métodos de solução para os problemas de otimização combinatória permutacional (POCP) e para o PCV são apresentados na seção 3.3. Nesta parte do texto são explicadas as diferenças entre métodos exatos e aproximativos. Na penúltima seção (3.4) são tratadas superficialmente algumas variações para o PCV, em destaque o KTR (*k-Travelling Repairman Problem*). Por fim, na seção 3.5, são apresentados abordagens científicas recentes que demonstram a aplicação do PCV como solução de problemas logísticos reais.

3.1 Considerações Iniciais

O problema do caixeiro viajante é um problema clássico de otimização combinatória. Dada a posição geográfica de um número de cidades distintas, o PCV se resume em achar o menor percurso passando por cada cidade uma única vez e retornar ao ponto de partida inicial.

Segundo Cunha *et. al.* (2002), sua origem é creditada a Willian Rowan Hamilton, que inventou um jogo cujo objetivo era o de traçar um roteiro através dos vértices de um dodecaedro (vértices que equivalem a cidades) que iniciasse e terminasse no mesmo vértice (cidade) sem, contudo, repetir uma visita.

O problema consiste basicamente em analisar todos os circuitos (rotas) hamiltonianos existentes para n pontos a fim de obter a distância mínima ou o menor tempo do percurso (função objetivo). Um circuito hamiltoniano é aquele que passa por todos os vértices de um grafo. Os vértices podem ser representados pelos n pontos geográficos de localização das cidades, enquanto as arestas são as vias que ligam diretamente as cidades.

Considerando que o máximo de caminhos ou rotas distintas em um circuito hamiltoniano é dado por $n!$ constata-se que é inviável a análise de todas as soluções possíveis (distintas) mesmo para problemas de pequena complexidade, ou seja, para valores de $n > 10$ conforme apresentado na Tabela 3.1, dada a seguir, que demonstra a explosão combinatorial de soluções para problemas desta natureza. O tempo estimado está calculado com base numa máquina hipotética da ordem de 10^{-9} segundos, para execução das instruções de controle do programa, de acesso aos dados, cálculo das distâncias, comparações, chamadas a subprogramas etc. (VIANA, 1998).

Tabela 3.1 – Explosão combinatória

n	n!	Tempo
5	120	0,00012 segundos
10	3628800	3,62880 segundos
12	479001600	8 minutos
15	1307674368000	15 dias
20	2432902008176640000	77.147 anos
50	3.0414093201713378043612E+0064	∞
100	9.3326215443944152681699E+0157	∞
500	1.2201368259911100687912E+1134	∞
1000	4.0238726007709377354362E+2567	∞

Fonte: Viana (1998)

É válido ressaltar que o tempo computacional para análise de todas as soluções distintas e possíveis é **impraticável**, visto que, mesmo para um pequeno número de pontos ($n=10$), o número de soluções possíveis é maior que 3 (três) milhões.

O PCV faz parte dos problemas de otimização classificados como *NP-hard* (não-polinomiais difíceis) tendo em vista sua complexidade e o tempo exponencial de processamento à medida que o número de pontos a serem visitados cresce. Rabak (1999) destaca que o único algoritmo que se conhece que resolve todas as instâncias do problema é a completa enumeração de todas as permutações possíveis, avaliando a distância (ou custo, ou tempo) de cada uma delas e selecionando-se a melhor. Para obter

esta solução, a ordem do tempo necessário para rodar esse algoritmo é $O(n!)$, dando ênfase a impraticidade do método (tempo e memória).

Silva *et. al.* (2005) destacam que problemas da classe *NP-hard* justificam o uso de técnicas refinadas aplicadas na resolução de problemas como o PCV, com o intuito de encontrar boas soluções viáveis a um baixo custo em termos de recursos computacionais.

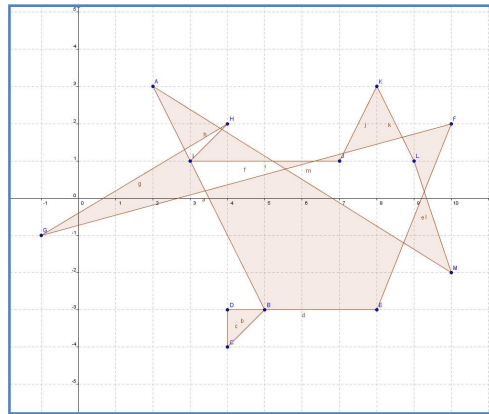
O PCV é considerado simétrico quando a distância de um ponto i a um ponto j é a mesma de j para i , ou seja, $d_{ij} = d_{ji}, \forall i, j = 1, 2, 3, \dots, n$. Usualmente, a consideração de simetria é uma simplificação para a solução do PCV, visto que, em problemas complexos e reais, faz-se necessária a consideração de sentido de malha viária, o que torna o problema assimétrico e conseqüentemente mais difícil.

Problemas de otimização tendem a determinar boas soluções através de algoritmos eficientes e eficazes, que procuram pela solução ótima dentro do espaço de busca com uma estratégia bastante racional.

As heurísticas abordadas no Capítulo 4 reduzem bastante o espaço de busca com o intuito de diminuir o tempo computacional de execução. Porém, suas soluções não necessariamente são exatas, ou seja, métodos heurísticos não garantem encontrar a(s) solução(es) ótima(s).

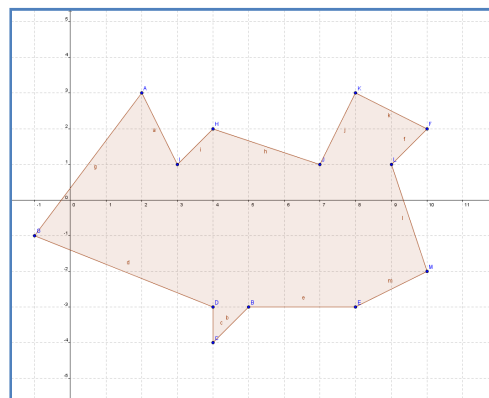
As Figuras 3.1 e 3.2 apresentam ilustrações do PCV com duas soluções distintas para uma mesma instância ($n=13$). A primeira apresenta um circuito hamiltoniano com diversos cruzamentos, cujo resultado total da rota foi igual a 58,22 km. Já a figura 3.2, apresenta a solução ótima obtida através de métodos heurísticos, onde não há cruzamentos entre os vértices do grafo e o valor final foi de 33,90 km.

Figura 3.1 – Uma solução para o PCV com 13 cidades.



Fonte: Elaborado pelo autor (2011). Plotado no *software* GeoGebra – *Dynamic Mathematics for Schools*.

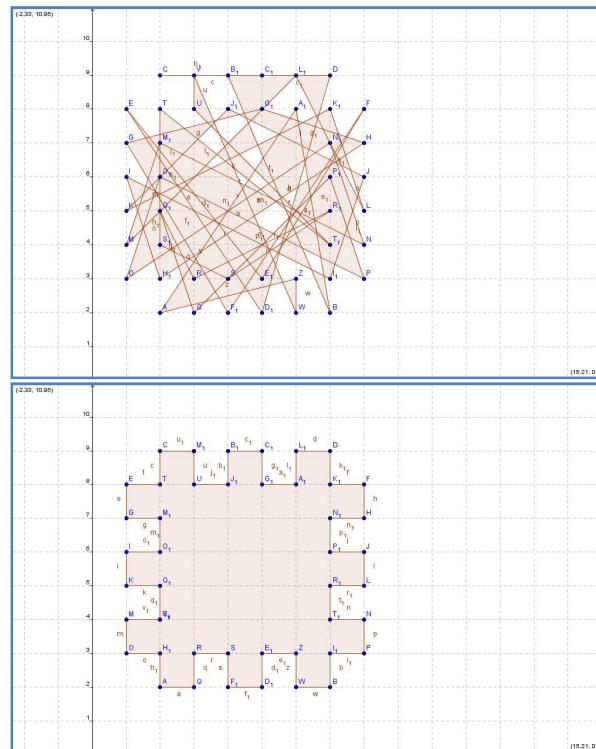
Figura 3.2 – Solução ótima do PCV com 13 cidades.



Fonte: Elaborado pelo autor (2011). Plotado no *software* GeoGebra – *Dynamic Mathematics for Schools*.

A figura 3.3 representa, respectivamente acima e abaixo, a solução inicial e a solução ótima para o problema proposto por Viana (1998) com $n=44$. A solução inicial obteve o valor de 209,76 km e a solução ótima obtida através de uso de metaheurística foi igual a 44 km, demonstrando a capacidade de evolução da otimização do ciclo hamiltoniano.

Figura 3.3 – Ilustração para o PCV com 44 cidades.



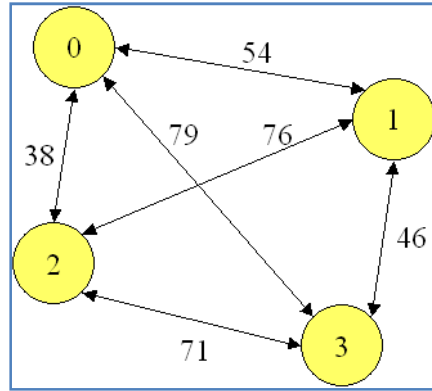
Fonte: Adaptado de Viana (1998). Ilustrações elaboradas pelo autor (2011).

3.2 O modelo matemático do PCV

A simplicidade aparente do problema do caixeiro viajante esconde à primeira vista a dificuldade de sua solução (RABAK, 1999). Até hoje, encontram-se publicados na literatura e em sítios específicos da *internet* desafios de soluções voltadas à instâncias do PCV para mais de 2000 pontos.

Conforme comentado anteriormente, o problema do caixeiro viajante é um problema de otimização associado ao da determinação dos caminhos hamiltonianos em um grafo. A Figura 3.4 dada abaixo, ilustra o Grafo $G = (N, A)$, onde $N = \{0, 1, 2, 3\}$ e $A = \{(0, 1), (0, 2), (0, 3), (1, 0), (1, 2), (1, 3), (2, 0), (2, 1), (2, 3), (3, 0), (3, 1), (3, 2)\}$.

Figura 3.4 – Ilustração de um $G = (N, A)$



Fonte: Zanuz (2007). <http://www.zanuz.com/2007/09/o-problema-do-caixeiro-viajante.html>.

Na ilustração acima, a solução $0 \rightarrow 1 \rightarrow 3 \rightarrow 2 \rightarrow 0$ é uma solução ótima do problema, visto que representa a menor distância total possível para o percurso.

Matematicamente o problema poderia ser formulado da seguinte maneira:

$$\min Z = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot x_{ij} \quad (1)$$

Sujeito às restrições:

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, \quad \forall i = 1, \dots, n \quad (3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad \forall S \subset N \text{ e } |S| \in \left[\frac{N}{2} \right] \quad (4)$$

$$x_{ii} = 0, \quad \forall i = 1, 2, 3, \dots, n \quad (5)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j = 1, 2, 3, \dots, n \quad (6)$$

A equação (1) é a função objetivo do problema. Consiste em minimizar a distância (ou tempo) para que o caixeiro viajante percorra n pontos distintos uma única vez e retornar ao ponto de partida inicial. As equações de (2) a (6) são as restrições do PCV.

As equações (2) e (3) representam, respectivamente, as restrições de que o caixeiro somente pode passar uma única vez em cada ponto, seja na entrada, ou na saída, devendo partir para o próximo ponto até que todo o percurso seja realizado.

A equação (4) restringe a possibilidade de ciclos parciais, ou seja, evita que qualquer ciclagem menor que n seja solução viável para o problema. Esta restrição eleva substancialmente o número de restrições ($2^n - 2$), visto que cresce exponencialmente com n .

A equação (5) determina que não é possível ir de uma cidade para ela mesma, enquanto que a equação (6) representa a definição da variável binária x_{ij} de confirmação de deslocamento, tal que $x_{ij} = 1$, se o caixeiro irá se deslocar da cidade i para a cidade j , ou $x_{ij} = 0$, caso contrário.

O problema formulado exclusivamente pelas equações (1) a (6) tratam-se de uma formulação de programação linear inteira, visto que as variáveis de decisão x_{ij} são inteiras (binárias) e a função objetivo Z é linear.

3.3 Métodos gerais de solução para POCP e PCV:

Conforme destacado na seção 3.1, a obtenção de solução exata para problemas de otimização *NP-hard* como o PCV somente é garantida através de uma varredura completa em todo o espaço de soluções possíveis. Logicamente, este fator limita a aplicação de métodos exatos para problemas mais complexos.

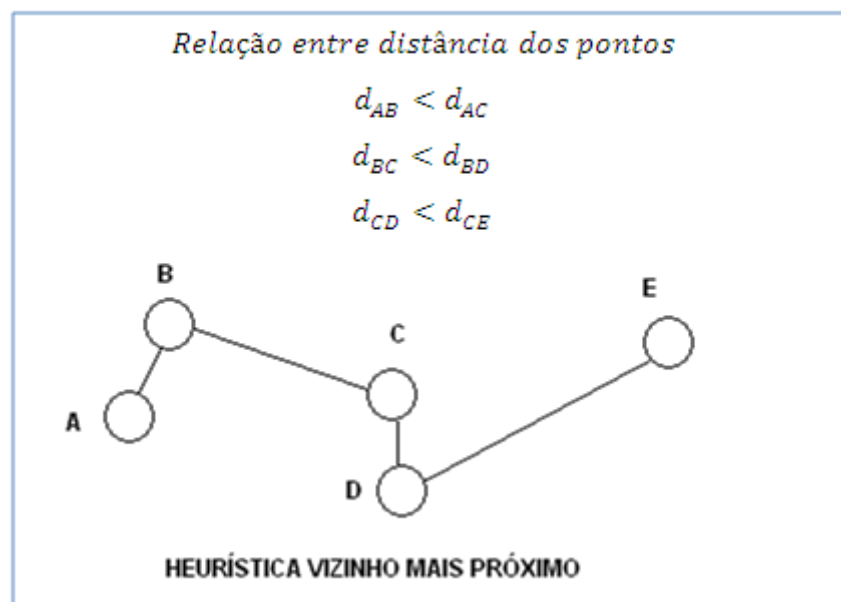
São encontrados na literatura diversos artigos publicados com métodos aproximativos de resolução do problema do caixeiro viajante. Como abordado anteriormente, mesmo que para valores pequenos de n , soluções através de algoritmos exatos ou através de programação linear são bastante dispendiosas, e muitas vezes requerem um tempo computacional impraticável. A seguir, apresentam-se algumas das soluções mais tradicionais e clássicas da literatura.

3.3.1 Heurística NN- *Nearestneighbor* ('vizinho mais próximo')

A heurística *Nearestneighbor* (vizinho mais próximo) foi descrita por Solomon (1987). Por tratar-se de um método aproximativo desenvolvido para resolver um problema em tempo polinomial, as soluções geradas não necessariamente são exatas ou ótimas. Este método, como o próprio nome diz, possui uma concepção bastante simples e intuitiva.

Através de uma matriz de distâncias entre diversos pontos, o percurso é construído a partir de um primeiro ponto (o mais próximo da origem) e os demais pontos adicionados sequencialmente, sendo sempre o mais próximo do último adicionado à sequência. Na ilustração da figura 3.5, dada a seguir, em relação ao ponto inicial “A” da sequência, o vizinho mais próximo é o ponto “B”; em relação ao ponto “B”, “C” está mais próximo que “D” e assim por diante. Esse processo se repete enquanto o limite da capacidade da rota é respeitado.

Figura 3.5 – Heurística vizinho mais próximo



Fonte: Elaborado pelo autor (2011). Plotado no IOR Tutorial.

Schepke *et. al.* (2004) descrevem uma variação da heurística ‘vizinho mais próximo’ que é a ‘vizinho mais próximo dá chance’. Basicamente a diferença está na situação em que, em vez de encerrar a rota quando a demanda é extrapolada, ele escolhe o ponto mais próximo seguinte ao atual. Caso a demanda deste for maior que o espaço ainda existente para a rota, repete-se esse passo sucessivamente até encontrar um ponto que respeite a capacidade ainda disponível. O objetivo é reduzir o número de rotas.

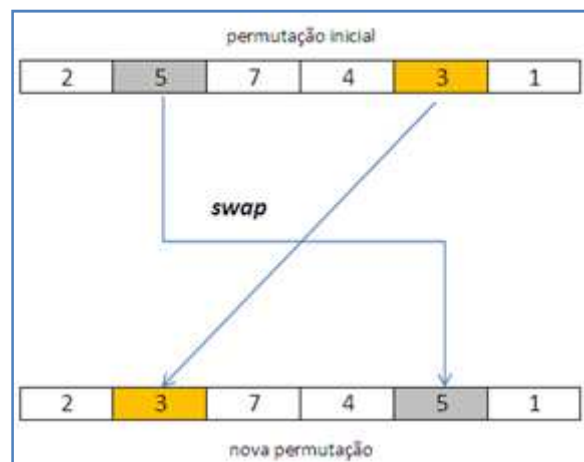
3.3.2 Swap

O algoritmo *swap* foi descrito por Corberán (2000). O método possui uma concepção bastante simples: a partir de uma solução inicial, são realizadas trocas simples (*swaps*) e

aleatórias entre duas cidades de forma a avaliar se a troca reduziu o custo total das rotas. Caso ocorra redução, a troca é realizada. O processo termina a partir de um critério de parada, tal como número máximo de iterações ou tempo de processamento computacional.

A Figura 3.6, dada a seguir, ilustra o método *swap*. Dada uma permutação inicial tal como a sequência $2 \rightarrow 5 \rightarrow 7 \rightarrow 4 \rightarrow 3 \rightarrow 1$, o processo ocorre através de trocas simples entre dois pontos da sequência, conforme ilustrado abaixo. Os pontos 5 e 3 foram trocados entre si resultando em uma nova permutação $2 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 5 \rightarrow 1$ cujo resultado é avaliado. As trocas são aleatórias e o procedimento termina com o critério de parada definido.

Figura 3.6 – Movimento *swap*



3.3.3 Savings de Clarke e Wright

O algoritmo das economias foi proposto por Clarke e Wright (1964). O método é baseado nos conceitos e noções da Economia. Um algoritmo de *savings* inicia com um processo iterativo que visa percorrer todas as cidades duas a duas, de maneira a calcular as economias deste deslocamento considerando ainda, o custo de retornar ao ponto inicial. Trata-se de um Algoritmo Heurístico Construtivo (AHC) baseado numa função gulosa de inserção. Este nome deve-se ao fato de que durante o processo iterativo, a função escolhe sempre a maior economia dentre as possíveis.

Schepke *et.al.* (2004) destacam que a noção de economia pode ser definida como o custo da combinação, ou união, de duas sub-rotas existentes. Liu (1999) *apud* Schepke

et.al. (op.cit.) define que duas rotas contendo os clientes i e j podem ser combinadas, desde que i e j estejam ou na primeira ou na última posição de suas respectivas rotas.

Matematicamente as economias S_{ij} podem ser definidas como:

$$S_{ij} = C_{Oi} + C_{jO} - C_{ij} , onde:$$

C_{Oi} é o custo entre o ponto inicial O e a cidade i ,

C_{jO} é o custo entre a cidade j e o ponto inicial O ,

C_{ij} é o custo entre as duas cidades i e j

S_{ij} é a economia da decisão de seguir essa rota

Este procedimento permite ter o conhecimento das cidades com distâncias mais próximas e, em segunda instância, outro processo iterativo busca o maior *saving*, considerando as duas cidades mais próximas uma da outra e as roteia em relação à cidade-base.

Diversas aplicações na literatura são encontradas com o uso deste método de solução. Trata-se de um modelo econômico-iterativo de busca de soluções para problemas de roteirização. O método de *Savings* é bastante aplicado a problemas práticos baseados no PCV por tratar o aspecto econômico na construção da rota. Para problemas práticos, como por exemplo rotas de distribuição de mercadorias, o algoritmo das economias percorre todos os clientes dois a dois, sempre comparando o custo do retorno ao depósito inicial. Dessa forma, por tratar-se de um método construtivo, a solução sempre visa minimizar o custo, daí o nome *savings*.

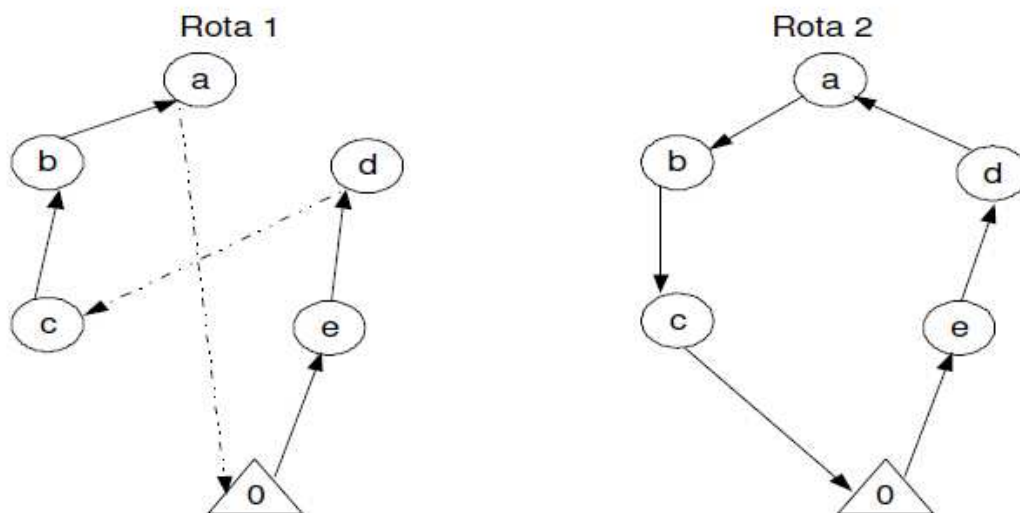
3.3.4 $r - OPT$ de Lin-Kernighan

Esse método foi proposto por Lin e Kernighan (1973) e baseia-se na ideia da troca de trechos entre pontos para otimizar o percurso. A primeira abordagem publicada sobre esse método ainda incipiente foi em 1958, conforme Croes (1958), que tratou da troca

de 2 trechos não consecutivos. Posteriormente, em 1965 ele abordou esse método para 3 trechos. O número de trechos designados para cada troca é definido por r . Caldas e Santos (2008) destacam que quanto maior o valor de r , melhor será a solução obtida, porém, maior será o tempo computacional empregado.

As abordagens mais comuns são 2-OPT e 3-OPT. Segundo Schepke *et. al.* (*op.cit.*) o método de melhoramento 2-OPT intra-rotas consiste na possível troca de dois trechos, não consecutivos, de uma rota e refazer as conexões. O custo da nova rota é avaliado e se for menor que o custo da rota original, a nova rota é mantida. Um critério de parada é definido para terminar o processo, sendo normalmente, a definição de um número máximo de iterações sem que haja melhora na solução. Existe ainda uma outra abordagem para a heurística 2-OPT inter-rotas, que consiste na possível troca de dois trechos de duas rotas diferentes, um de cada rota e refazer as conexões, criando duas novas rotas. Assim como a abordagem anterior, as novas rotas somente são mantidas se o custo delas for menor que o custo das rotas originais. A Figura 3.7 ilustra esse processo para um movimento 2-OPT.

Figura 3.7 – Representação gráfica do método 2-OPT da troca de nós da rota.



Fonte: Caldas e Santos (2008).

3.3.5 Heurística Permutacional (HP)

A heurística permutacional é um método desenvolvido por Silva e Soma (2001) aplicado aos problemas de otimização combinatória permutacional (POCP), entre os quais o PCV. O método avalia um conjunto fixo de soluções, construído através de permutações das n cidades, que diversifica a busca dentro do conjunto de soluções viáveis para o problema.

O método *HP* consiste basicamente em dividir o conjunto de soluções viáveis S em n vizinhanças $N(S_i)$ distintas entre si, onde para cada $1 \leq i \leq n$, s_i é uma permutação que inicia com o ponto i , e cada uma destas vizinhanças será particionada em quatro subvizinhanças $N(S_{ij}) \subset N(S_i)$, $1 \leq j \leq 4$. O número total de permutações geradas na HP é $[2 \cdot n \cdot (n - 1) \cdot (n - 2) + 4n]$.

A Tabela 3.2 apresenta o número de soluções visitadas para diferentes valores de n , onde o valor de $n!$ representa a totalidade de soluções possíveis a serem avaliadas. Conforme dados apresentados abaixo, é possível verificar que o método HP reduz substancialmente o espaço de soluções avaliadas, o que permite ser aplicada a diversos problemas práticos.

Tabela 3.2 – Número de permutações (soluções) avaliadas pelo método *HP*

n	$n!$	<i>HP</i>
10	3.628.800	1.480
20	2,4329 E+18	13.760
30	2,65253E+32	48.840
40	8,15915E+47	118.720
50	3,04141E+64	235.400
60	8,32099E+81	410.880
70	1,1979E+100	657.160
80	7,1569E+118	986.240
90	1,4857E+138	1.410.120
100	9,3326E+157	1.940.800

Silva *et. al.* (2005) expandiram o número de soluções visitadas a fim de verificar como a diversificação poderia reduzir os desvios percentuais entre as soluções encontradas e as soluções ótimas da biblioteca TSPLIB (1995). Essa reformulação foi tratada como

HP^* , e o número de permutações geradas passou de $[2.n.(n-1).(n-2) + 4n]$ para $[12.n.(n-1).(n-2) + 24n]$ conforme observa-se na Tabela 3.3, dada a seguir.

Tabela 3.3 – Comparativo entre permutações geradas pelos métodos HP e HP^*

n	HP	HP^*
10	1.480	8.880
20	13.760	82.560
30	48.840	293.040
40	118.720	712.320
50	235.400	1.412.400
60	410.880	2.465.280
70	657.160	3.942.960
80	986.240	5.917.440
90	1.410.120	8.460.720
100	1.940.800	11.644.800

A Figura 3.8, dada a seguir, apresenta um pseudo-código para implementação computacional do método HP .

Figura 3.8 – Pseudo-código do método *HP*

```

void HP( n )
{
    int i, j, k, p1, s, limite, cont;
    int s1[MAXITEMS], s2[MAXITEMS];

    limite= ((n / 35) + 1) * 250; /* Limita o num. de soluções geradas por subvizinhança */
    for(i=1; i<=n; i++) s1[i]=i; /* s1 está associada a solução si */
    for(p1=1; p1<=n; p1++) { /* p1 identifica a primeira posição de si */
        if (p1>1) { s1[1]=p1; s1[p1]=1; }
        for(i=1; i<=4; i++) { /* Procedimento para gerar as subvizinhanças */
            switch (i) {
                case 1: for(k=1; k<=n; k++) s2[k]=s1[k]; break;

                case 2: j=(n+1)/2; for(k=2; k<=j; k++) {
                    s=s2[k]; s2[k]=s2[n+2-k]; s2[n+2-k]=s; } break;

                case 3: for(k=1; k<=n; k++) s2[k]=s1[k]; k=2;
                    while (k<=(n-1)) {
                        s=s2[k]; s2[k]=s2[k+1]; s2[k+1]=s; k=k+2; } break;

                case 4: j=(n+1)/2; for(k=2; k<=j; k++) {
                    s=s2[k]; s2[k]=s2[n+2-k]; s2[n+2-k]=s; } break;
            }
            cont=1;
            fo_g( n, s2); /* Avalia o desempenho da primeira solução(s2) da subvizinhança */

            for(k=2; k<=(n-1); k++) {
                for(j=k+1; j<=n; j++) {
                    s=s2[k]; s2[k]=s2[j]; s2[j]=s;
                    /* Avalia o desempenho das soluções da subvizinhança, quando */
                    cont=cont+1; /* o número de soluções(cont) não ultrapassar o limite. */
                    if (cont<=limite) fo_g( n, s2);

                    s2[j]=s2[k]; s2[k]=s;
                }
            }
            if (p1>1) { s1[1]=1; s1[p1]=p1; }
        }
    }
    return;
}

```

Fonte: Silva e Soma (2001).

3.3.6 Metaheurísticas

O conceito fundamental de Metaheurística está apresentado no Capítulo 4. Trata-se de um método de resolução geral que fornece tanto uma estrutura quanto diretrizes de estratégia gerais para desenvolver um método heurístico específico que se ajuste a um tipo de problema particular. A característica fundamental de uma metaheurística é sua habilidade de diversificar o campo de soluções e “escapar” de um ótimo local. No

processo iterativo de busca de soluções, as metaheurísticas ao atingirem ótimos locais executam mecanismos de “fuga” dessa região de soluções de diversas maneiras diferentes, dentre as quais destaca-se a permissão de soluções inferiores em casos de maximização ou superiores em caso de funções-objetivo de minimização. As metaheurísticas mais conhecidas e utilizadas são: *SA (Simulated Annealing)*, *GA (Genetic Algorithms)* e *TS (Tabu Search)*.

O termo *Simulated Annealing (SA)* pode ser traduzido por ‘têmpera simulada’, correspondendo a uma simulação algorítmica do processo físico de têmpera de certos materiais, que consiste em submetê-los inicialmente a altas temperaturas e reduzi-las gradualmente até atingirem, com aumentos e reduções do estado de energia, o equilíbrio térmico, tornando-os assim, consistentes e rígidos.

Um pseudo-código para implementação de um algoritmo SA está apresentado na Figura 3.9:

Figura 3.9 – Pseudo-código para meta-heurística SA.

```

{Início A3.1}
[Módulo Principal]
  ler (S0, M, P, L,  $\alpha$ )           {configuração inicial/parâmetros}
  S = S0
  T0 = TEMP_INIC ( )              {temperatura inicial}
  T = T0
  j = 1
  repita
    i = 1
    repita
      Si = PERTURBA ( Si-1 )
       $\Delta f_i = f(S_i) - f(S_{i-1})$ 
      se ( $\Delta f_i \leq 0$ ) ou ( $\exp(-\Delta f_i / T) > \text{RND}()$ ) então
        S = Si
        nsucc = nsucc + 1

      fim-se
      i = i + 1
    até ( nsucc  $\geq$  L ) ou ( i  $\geq$  P )   {equilíbrio}
    T =  $\alpha \cdot T$ 
    j = j + 1
  até ( nsucc = 0 ) ou ( j  $\geq$  M )      {condição de parada}
  imprima ( S )                        {melhor solução}
{Fim - Módulo Principal}
[ PERTURBA ]
  {Início} . . . . {Fim}
[ TEMP_INIC ]
  {Início} . . . . {Fim}
[ RND ]
  {Início} . . . . {Fim}
{Fim - A3.1}

```

Fonte: Viana (1998).

Os algoritmos genéticos (GA) foram criados por Holland (1975) e são baseados nos processos observados na evolução natural das espécies. O GA é formado por uma população de indivíduos que representam as soluções do problema. Gomes (2008) descreve que os indivíduos são avaliados por uma função que atribui um valor chamado aptidão a cada indivíduo da população segundo sua qualidade em relação a função objetivo do problema. Os indivíduos são escolhidos por um procedimento inspirado na seleção natural para passarem por operações genéticas que resultam em descendentes que comporão a nova população. Segundo Mitchell (1998) e Haupt (2004) *apud* Gomes (2008), os estudos mostram que a nova população tem a tendência de ter aptidões melhores do que a população anterior. O processo de gerar novas populações é chamado de geração e as operações genéticas são conhecidas como *crossover*. A figura 3.10 apresenta um pseudo-código básico para implementação de um GA.

Figura 3.10 - Pseudo-código para metaheurística GA.

```
Escolha de uma população inicial
enquanto o critério de parada não é satisfeito faça
  repita
    se a condição do crossover é satisfeita então
      início
        seleciona os cromossomos pais;
        escolhe os parâmetros do crossover;
        executa o crossover;
      fim
    se a condição da mutação é satisfeita então
      início
        seleciona o(s) cromossomo(s) para a mutação;
        escolhe os parâmetros da mutação;
        executa a mutação;
      fim
    avalia a aptidão dos descendentes;
    até a quantidade de descendentes necessária;
    atualiza nova população;
  fim_enquanto
```

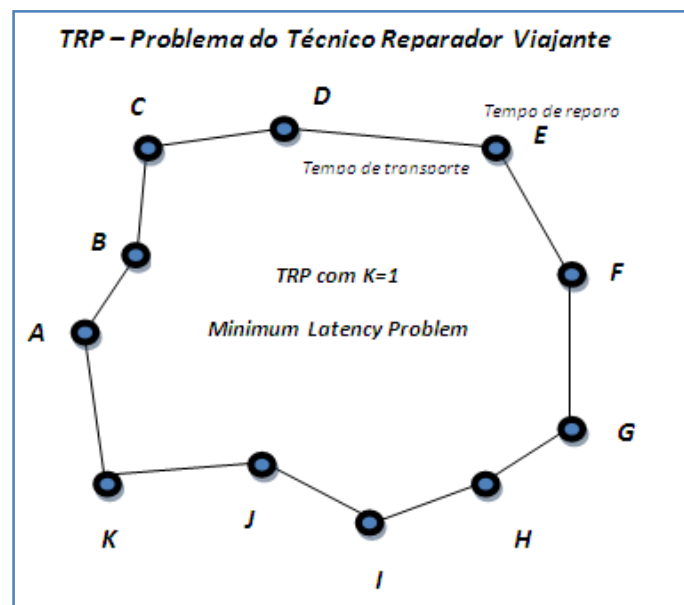
Fonte: Reeves e Rowe (2002) *apud* Gomes (2008)

No capítulo 4 (seção 4.2) são tratados em maior profundidade os aspectos relativos à metaheurística *Tabu Search* (TS), suas principais características e implementação.

3.4 Variações do PCV: *k-Travelling Repairman Problem* (KTR)

O problema do caixeiro viajante possui variações de sua abordagem na literatura científica. Um exemplo claro é o Problema do Multi-Reparador Viajante, intitulado na literatura internacional de *k-Travelling Repairman Problem* (KTR). É também conhecido como *Minimum Latency Problem* (MLP, quando $k=1$), *Deliveryman Problem* ou *School Bus-Driver Problem*.

Figura 3.11 – Ilustração do KTR



Fonte: Elaborado pelo autor.

Em todas as suas variações, o problema permanece sob a classificação NP-hard, frente à sua complexa solução. O problema consiste em um grafo $G = (V, E)$ não direcionado, onde a função objetivo visa minimizar a soma do tempo de espera (latência) de todos os clientes a serem visitados em todas as k rotas (k -tours). Inicialmente o KTR considerava o tempo de reparo (*repair time*) em cada cliente igual a zero, o que tornava a abordagem um problema clássico do PCV. Posteriormente, o KTR foi desenvolvido, generalizado e denominado *Generalized Travelling Repairman Problem* (GKTR) para tempos de reparo diferentes de zero, o que o torna um problema mais aplicável a problemas reais.

Trata-se de situações práticas do dia-a-dia de empresas que atuam com serviços de manutenção, por exemplo. Para que sejam atendidas as demandas de todos os clientes, um número determinado de técnicos reparadores percorre rotas distintas a fim de que sejam atendidas as demandas no menor espaço de tempo e custo possíveis.

Um exemplo clássico para ilustrar o GKTR é o serviço de manutenção de elevadores, onde cada empresa possui k rotas de manutenção preventiva a serem percorridas por determinado número de técnicos, cujo tempo de reparo em cada cliente é diferente de zero. A Figura 3.11, dada acima, ilustra o KTR para $k=1$, ou seja, um problema simplificado de um único técnico reparador viajante. A rota $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H \rightarrow I \rightarrow J \rightarrow K \rightarrow A$ é percorrida de forma a minimizar a distância e o tempo de espera de cada cliente, visto que em cada visita ocorre o registro da parada do técnico e o tempo necessário para execução do reparo,

A definição de latência tanto no KTR como no GKTR representa o montante de tempo que determinado cliente espera para ser servido e não inclui o tempo p inerente ao reparo. De acordo com Jothi e Raghavachari (2006), uma variação complementar do GKTR possibilita estimar o número mínimo de técnicos necessários para o cumprimento da função objetivo, que passa a ser a minimização da soma dos períodos de latência considerando a existência de uma restrição quanto ao valor máximo de L (tempo máximo de latência). Essa abordagem é conhecida como *Bounded Latency Problem* (BLP) e representa uma aplicação para problemas reais de logística de manutenção, entre outras aplicações. Outras abordagens têm sido amplamente desenvolvidas para modelos em tempo real (*real time models*), tal como destacado por Bonifaci e Stougie (2008).

3.5 Aplicações logísticas do PCV

De forma clássica, as aplicações mais comuns do PCV estão relacionadas às áreas de distribuição, vendas externas e transporte, visto que a função objetivo do problema está diretamente relacionada à minimização dos custos de deslocamentos ou tempos para cumprir determinado circuito de clientes ou cidades. Para exemplificar, citamos os trabalhos recentes de Goldberg e Luna (2000) que listaram várias aplicações práticas, tais como: a programação de operações de máquinas e manufatura, a programação de

transporte entre células de manufatura, otimização dos movimentos das ferramentas de corte, trabalhos administrativos entre outros; Costa *et. al.* (2001) mostram técnicas para otimização de serviços postais; Baldacci *et. al.* (2003), propõem um algoritmo exato para o problema de roteamento com coletas e entregas; Laporte *et.al* (2010) abordaram além das variáveis coletas e entregas, o custo do manuseio e movimentação das cargas (*handling costs*); Silva *et. al.* (2008) abordaram o PRV para coleta de gêneros alimentícios na cidade de Fortaleza-CE; Caldas e Santos (2008) apresentaram uma heurística do PCV para roteamento de uma frota de veículos para distribuição de bebidas em uma região urbana; Rabak (2009), apresenta técnicas de otimização para encontrar uma boa solução para o problema da inserção automática de componentes eletrônicos. Jothi e Ragavachari (2007) e Bonifaci e Stougie (2009) abordaram o KTR através da proposição de soluções aproximadas para o problema logístico de rotas de manutenção; Raduan (2009) abordou a roteirização parcialmente dinâmica de serviços a serem realizados em campo, entre muitos outros.

Após vistas todas essas aplicações do PCV, resolvemos adotar a metaheurística TS por não ter sido tão empregada nesta classe de problemas.

CAPÍTULO 4 – METODOLOGIA

Este capítulo é composto por três seções. A primeira seção (4.1) apresenta os conceitos introdutórios e as principais diferenças entre heurísticas e metaheurísticas. A segunda seção (4.2) tem o objetivo de apresentar especificamente a metaheurística *Tabu Search*, sua estrutura, modelo, complexidade e características principais. A terceira seção (4.3) está relacionada a aplicações da técnica na literatura recente, em especial a problemas da área de Logística.

4.1 Heurísticas e Metaheurísticas

Heurística deriva da palavra grega *heuriskein* que significa encontrar ou descobrir (REEVES, 1995). Na área de Pesquisa Operacional, o termo possui uma conotação relacionada a métodos que procuram soluções viáveis para problemas de otimização combinatória. O desenvolvimento de algoritmos ou métodos heurísticos se deu basicamente em decorrência da dificuldade, ou mesmo impossibilidade, de obtenção de soluções exatas para problemas combinatórios de larga escala.

Heurística representa qualquer método ou técnica criada ou desenvolvida, para solução de determinado tipo de problema. Trata-se de um método aproximativo desenvolvido especificamente para resolver um problema em tempo polinomial, conforme Viana (1998).

Reeves (1995) conceitua heurística como uma técnica que procura boas soluções com um tempo computacional razoável sem garantir nem viabilidade nem otimalidade, e ainda em muitos casos, sem definir quão próxima uma solução viável encontra-se da solução ótima de determinado problema específico.

Hillier e Lieberman (2010) definem um método heurístico como um procedimento que provavelmente vai encontrar uma excelente solução viável, mas não necessariamente uma solução ótima, para um determinado problema em análise.

“Métodos heurísticos em geral se baseiam em idéias relativamente simples de senso comum de como buscar uma boa solução. Essas idéias precisam ser cuidadosamente adaptadas para se adequar ao problema de interesse específico.

Portanto, os métodos heurísticos tendem a ser específicos por natureza”.

De acordo com Sousa (2009), entre as vantagens que podem ser elencadas para justificar a importância e a utilização de métodos heurísticos, destacam-se:

- i. Possibilitam a flexibilização para manejar as variáveis ou características do problema;
- ii. Podem apresentar mais de uma solução, possibilitando ao analista verificar qual tem melhor qualidade para o problema analisado;
- iii. A maioria dos métodos consegue alcançar soluções de boa qualidade sem recorrer ao formalismo matemático aumentando a facilidade de implementação.

Os métodos heurísticos podem ser subdivididos em: AHC (Algoritmos Heurísticos Construtivos) e AHM (Algoritmos Heurísticos de Melhoramento). Segundo Schepke *et.al.* (2004), as heurísticas construtivas utilizam técnicas de adição na construção do problema e o processo é contínuo e gradativo. Exemplos de heurísticas construtivas são: “algoritmo vizinho mais próximo” desenvolvido por Solomon (1987), “algoritmo vizinho mais próximo dá chance”, e o “algoritmo de *Savings*” proposto por Clarke e Wright (1964).

As heurísticas de melhoramento ou evolutivas são técnicas em que, a partir de uma solução inicial, são feitas trocas com o objetivo de melhorá-la (SCHEPKE *et. al., op.cit.*). A cada passo são feitas trocas a fim de diminuir o custo original da rota. As tentativas de troca são realizadas até um determinado critério de parada pré-estabelecido, tal como um determinado número de iterações, tempo de processamento ou não existência de soluções melhores a partir de determinado número de tentativas.

Já as metaheurísticas são consideradas heurísticas de uso e aplicação geral que nos fornecem boas soluções para vários tipos diferentes de problemas. Viana (2008) destaca que como as metaheurísticas trabalham com funções de probabilidade, não é sempre possível garantir a obtenção de um mesmo resultado para uma determinada instância de um problema, o que não ocorre com as heurísticas. Complementando a abordagem acima, Sousa (2009) destaca que o processo de busca de uma meta-heurística precisa ser eficiente para que possa encontrar a solução ótima ou uma solução de boa qualidade.

Nesse contexto, segundo o autor, a grande diferença entre as metaheurísticas está no algoritmo de pesquisa da vizinhança.

Hillier e Lieberman (2010) conceituam metaheurística como um método de resolução geral que fornece tanto uma estrutura quanto diretrizes de estratégia gerais para desenvolver um método heurístico específico que se ajuste a um tipo de problema particular.

A característica fundamental de uma metaheurística é sua habilidade de diversificar o campo de soluções e “escapar” de um ótimo local. No processo iterativo de busca de soluções, as metaheurísticas ao atingirem ótimos locais executam mecanismos de “fuga” dessa região de soluções de diversas maneiras diferentes, dentre as quais se destacam a permissão de soluções inferiores em casos de maximização ou superiores em caso de funções-objetivo de minimização.

Hillier e Lieberman (2010) ainda destacam que toda vez que um problema prático puder ser resolvido por algoritmo capaz de garantir otimalidade, deve-se usar então esse processo. “O papel da meta-heurística é lidar com problemas que são muito grandes e complexos de serem resolvidos por algoritmos exatos”.

Heurísticas e Metaheurísticas possuem largo campo de aplicação na solução de problemas típicos de otimização combinatória, sendo, portanto, técnicas importantes da área de Pesquisa Operacional para aplicações práticas em diversas áreas do conhecimento, notadamente Logística. Vale ressaltar que não é garantida a existência de soluções melhores que aquela encontrada através do uso de heurísticas ou meta-heurísticas. Porém, em geral os resultados são satisfatórios.

4.2 *Tabu Search*

4.2.1 Introdução

Tabu Search (TS) ou Busca *Tabu* (BT) é uma meta-heurística largamente empregada na Pesquisa Operacional que usa algumas idéias de senso comum para permitir ao processo de busca escapar de um ótimo local (HILLIER; LIEBERMAN, 2010). A TS teve sua origem a partir de uma solução de Glover (1989) e é considerada uma estratégia para solução de problemas complexos de otimização combinatória, tendo em vista que

possui uma estrutura flexível que permite a utilização de outros métodos de programação linear ou de heurísticas especializadas com o objetivo de realizar buscas iterativas no espaço de soluções possíveis.

Apesar de a técnica ter sido inicialmente publicada há mais de vinte anos, até hoje muitos trabalhos científicos nas áreas de Engenharia, Computação e Logística têm sido realizados com a utilização da meta-heurística Busca *Tabu*, demonstrando seu grau de eficiência em problemas de otimização combinatória. Glover (1990) apresentou uma série de aplicações publicadas já nos primeiros anos de desenvolvimento do método, demonstrando seu vasto campo de aplicação. Entre os quais, cita: *Employee Scheduling* (GLOVER; McMILLIAN, 1986), *Neural networks* (WERRA; HERTZ, 1989), *Machine scheduling* (LAGUNA; BARNES; GLOVER, 1989), *Job shop scheduling* (ECK, 1989), *Telecommunications path assignments* (OLIVEIRA; STROUD, 1989), *Network topology design* (LEE, 1989) entre outros. Podemos citar também estudos recentes realizados por Pires (2006), Sousa (2009) e Moreira (2011).

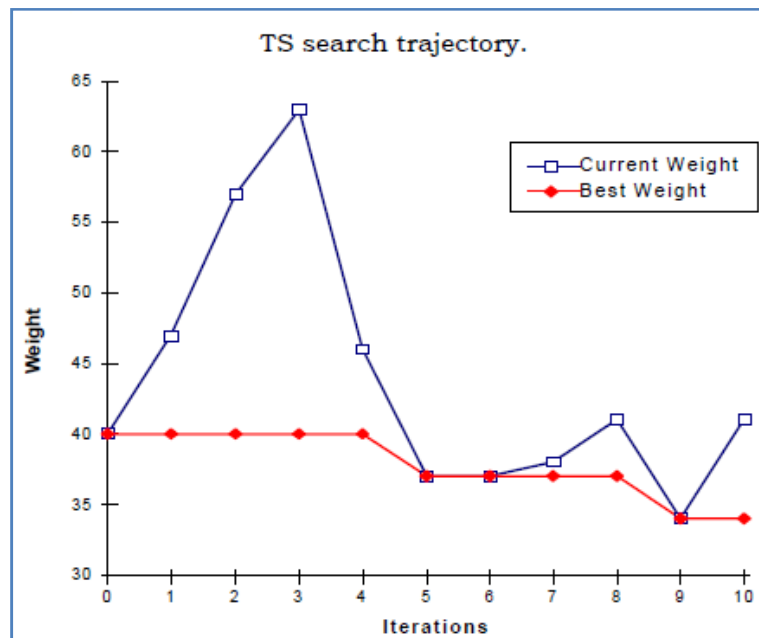
De acordo com Viana (1998), Ruiz-Vanoye e Díaz-Parra (2010) a meta-heurística TS tem por base três princípios:

- i. Uso de uma estrutura de dados para memorizar o histórico da evolução do processo de busca;
- ii. Uso de um mecanismo de controle para fazer um balanceamento entre a aceitação, ou não, de uma nova configuração, com base nas informações registradas na fila tabu referentes às restrições e aspirações desejadas;
- iii. Procedimentos de estratégia de intensificação e diversificação utilizando memórias de curto e longo prazos.

Sousa (2009) destaca que a Busca *Tabu* está fundamentada nos conceitos de movimento, vizinhança e memória. O movimento está relacionado à alteração de uma dada configuração que representa uma solução possível para o problema. Assim, a vizinhança é gerada a partir de movimentos ou perturbações que modificam a estrutura de uma solução inicial de modo a criar outras soluções. A memória refere-se à capacidade desta metaheurística de registrar o histórico de soluções visitadas a fim de “escapar” de ótimos locais.

A partir de uma solução inicial, a TS opera usando movimentos de troca (permutações) em uma determinada vizinhança, avaliando os resultados destes movimentos. Alguns destes movimentos são colocados em uma fila *tabu*, que representa uma região que não poderá ser visitada por um determinado número de iterações (equivalente ao tamanho “t” da fila *tabu*). Uma exceção se dá quando um determinado movimento, mesmo que pertencente à fila *tabu*, produz uma solução que atenda a um critério de aspiração, que normalmente é definido como uma função que compara diversas soluções com a melhor solução já encontrada. A fila *tabu* é alterada ao longo do processo iterativo, onde alguns elementos são introduzidos, outros retirados.

Figura 4.1 – Trajetória de uma metaheurística TS



Fonte: Glover (1990)

Diferentemente de outros métodos de otimização, a TS permite durante o processo de diversificação (que será tratado a seguir), que o processo iterativo percorra soluções piores que a imediatamente anterior, buscando com isso, fugir de ótimos locais. Essa característica do algoritmo é também conhecida como “escalada da montanha” (*climbing mountain*), que significa o movimento de percorrer soluções piores (mais altas do ponto de vista de um problema de minimização) a fim de encontrar em outras regiões do espaço de solução, uma solução melhor do que até então encontrada conforme ilustra a Figura 4.1 dada acima.

4.2.2 Características de um algoritmo TS

Dentre as variáveis definidas no passo de inicialização do TS, as definições do Número de Vizinhos (NV) e do Número Máximo de Iterações (NMÁX) estão diretamente relacionadas ao tempo de processamento para execução do algoritmo. Hertz e Werra (1987) *apud* Viana (1998) sugerem que “experimentos são necessários para determinar de que forma estes parâmetros estão relacionados com o tamanho do problema”. Com base na referência verificada na literatura, sugere-se $NV = n/2$ e $n \leq NMAX \leq 3n$, onde n é uma instância do problema a ser atacado.

Baseado ainda nas características apresentadas de um algoritmo TS, nota-se que algumas questões ficam sem resposta, visto que, conforme a própria definição de uma metaheurística, o TS traz as diretrizes de estratégia gerais para desenvolver um método heurístico específico que se ajuste a um tipo de problema particular.

Adaptando de Hillier e Lieberman (2010), as questões fundamentais são:

- i) Qual procedimento de busca local deve ser adotado?
- ii) Como definir a estrutura de vizinhança a fim de intensificarmos a busca de soluções ótimas em determinada região promissora?
- iii) Qual mecanismo de “fuga” de ótimos locais deve ser usado a fim de garantir maior diversificação?
- iv) Como dimensionar o tamanho da fila tabu (T) e qual deve ser a forma de movimentação da fila a cada iteração?
- v) Por quanto tempo uma movimentação de tabus deve permanecer na fila?
- vi) Qual regra de parada deve ser adotada?

Essas questões são fundamentais para um bom desenvolvimento do TS, tendo em vista que representam as peculiaridades de cada técnica e diferenciam as várias heurísticas criadas e desenvolvidas na literatura especializada. Além disso, representam fatores, como dito anteriormente, diretamente relacionados à *performance* da técnica, dentre as quais o tempo de processamento, grau de diversificação e intensificação.

Conforme o Quadro 4.1, apresentado a seguir, um algoritmo TS deve possuir a seguinte característica:

Quadro 4.1 – Estrutura básica de um código TS

1. Inicialização:

- 1.1 Defina uma solução inicial viável.
- 1.2 Defina o tamanho da “fila tabu” (F), a característica de sua movimentação e número de iterações (NMÁX).
- 1.3 Defina o critério de aspiração (Asp).
- 1.4 Defina o tamanho da vizinhança (NV) a cada iteração.
- 1.5 Estabeleça a função *swap* de perturbação da vizinhança.

2. Iterações:

- 2.1 Use um procedimento de busca local para definir as movimentações viáveis na vizinhança local da solução atual.
- 2.2 Desconsidere qualquer movimentação na lista de tabus atual, a menos que a movimentação resulte em uma solução melhor que a melhor solução encontrada até então.
- 2.3 Determine quais das soluções restantes fornecem a melhor solução.
- 2.4 Adote essa solução como solução para a próxima iteração independentemente de ela ser melhor ou pior que a solução atual.

3. Atualização da lista de tabus:

- 3.1 Insira a última movimentação para impedir o retorno a uma solução já adotada anteriormente;
- 3.2 Se a lista estiver completa, elimine o membro mais antigo da lista.

4. Regra de parada:

- 4.1 Defina um critério de parada, como por exemplo, um número fixo de iterações, tempo máximo de processamento do computador ou número fixo de iterações consecutivas sem uma melhoria no melhor valor encontrado para a função objetivo.

5. Saída (impressão de S – configuração final).

Fonte: Viana (1998). Texto adaptado.

4.2.3 Elementos utilizados em um algoritmo TS

Os identificadores básicos de uma metaheurística Busca Tabu são:

S0:	configuração da solução inicial viável
S:	configuração final da solução encontrada pelo TS
S*:	melhor configuração de uma determinada iteração
S _{ij} :	configuração da solução “j” vizinha da “i”
F:	fila tabu
T:	tamanho da fila tabu
NV:	número de vizinhos
NMÁX:	número máximo de iterações
M:	iteração em que foi obtida a melhor iteração
f:	função objetivo
f*:	limite de aceitação da solução
Delta_f:	$\min \{ f_i - f_{i-1} \}$
i, j	variáveis de <i>loops</i> .

Funções:

<i>Asp()</i>	função do critério de aspiração
<i>Min()</i>	função que determina o menor valor de um conjunto de dados (para problemas de minimização)
<i>Swap</i>	função que realiza uma perturbação em relação a uma solução existente

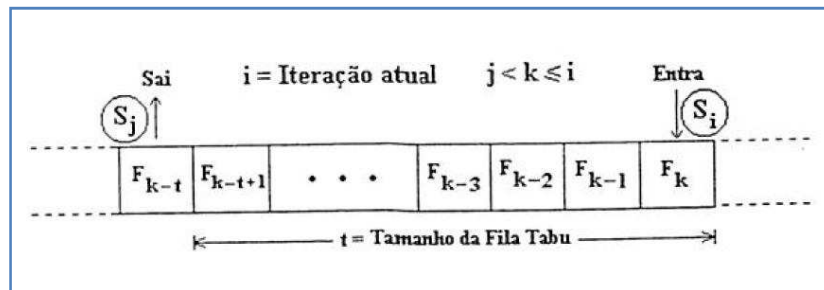
4.2.4 A Fila Tabu

A Fila *Tabu*, conforme ilustração 4.2 dada a seguir, pode assumir diversas configurações. Glover e Laguna (1991) estabelecem que na maioria das aplicações sejam definidas filas *tabu* fixas (*a single tabu list*). Porém, a fila *tabu* também pode ser dinâmica, ou seja, permitir um número variável de elementos de acordo com o processo iterativo. Nesse caso, para cada iteração, a Fila *Tabu* tem um “*tabu_size(i)*”, cujo valor pode ser determinado aleatoriamente em função de um parâmetro *k_i*, pertencente a um intervalo mínimo e máximo pré-definido.

O tamanho “*t*” da fila *tabu* é um dos fatores mais importantes para a *performance* do método, sendo, portanto, um dos parâmetros a serem alterados no processo de “calibragem” do algoritmo. A definição do tamanho da fila é bastante relevante porque impacta nos resultados de diversificação e intensificação. De forma geral, recomenda-se que o tamanho da fila seja proporcional ao tamanho do problema; problemas grandes favorecem filas maiores, enquanto que problemas pequenos requerem filas pequenas.

Na Figura 4.2, dada abaixo, pode-se observar a dinâmica da fila *tabu* através da entrada e saída de um movimento S , de acordo com o tamanho t da fila.

Figura 4.2 – Ilustração do movimento na fila *tabu*



Fonte: Viana (1998)

Vale ressaltar que: se o tamanho da fila *tabu* é pequeno, a busca pode ser excessivamente atraída para ótimos locais, de forma análoga ao que aconteceria se não houvesse a fila. Ao contrário, se a fila *tabu* for muito grande em relação ao tamanho do problema, são criadas muitas restrições e as melhores soluções podem não ser encontradas.

4.2.5 A função $Asp()$ – Critério de Aspiração

O critério de aspiração desempenha um importante papel no método iterativo utilizado na metaheurística TS. Segundo Viana (1998), seu objetivo é fazer com que um estado *tabu* possa ser utilizado para se movimentar para ele, caso determinado critério de aspiração seja atendido.

Glover (1990) destaca que um critério de aspiração pode ser aplicado a um atributo *tabu*, durante o período em que ele permanece na Fila *Tabu* e sobrepondo seu *status*, se o resultado obtido por este movimento melhora o resultado anteriormente obtido no processo.

O funcionamento deste critério indica que, se S^* está na Fila *Tabu* (estado proibido), é possível mover-se para o estado corrente S , caso:

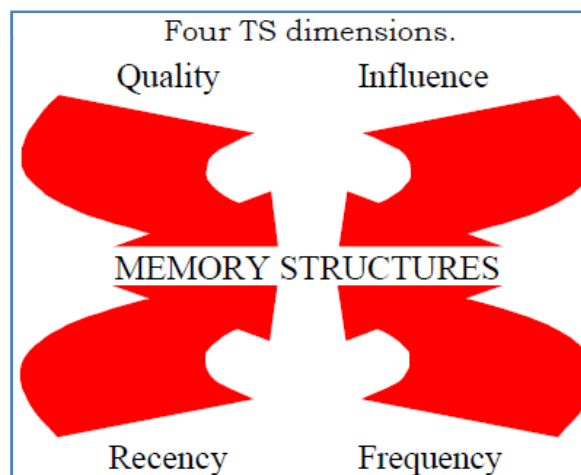
$$f(S^*) \leq Asp(F(S^*))$$

4.2.6 Conceitos básicos acerca de estruturas de memória

Conforme abordado anteriormente, a estrutura de memória refere-se à capacidade desta metaheurística de registrar o histórico de soluções visitadas a fim de “escapar” de ótimos locais. Reeves (1995) aborda as principais estruturas de memória utilizadas pela TS, classificando-as basicamente em quatro tipos:

- i. Memória baseado em atributos, que trata-se de uma estrutura de memória de curta duração;
- ii. Memória baseada na frequência;
- iii. Memória baseada na qualidade;
- iv. Memória baseada na influência.

Figura 4.3 – Estruturas de memória utilizadas na TS



Fonte: Glover (1997).

4.2.7 Diversificação e Intensificação

Segundo Sousa (2009), o objetivo da diversificação é impedir a convergência prematura do algoritmo, forçando-o a visitar soluções desconhecidas até o momento, e evitar que a busca entre em ciclo. Na prática, a diversificação é um procedimento para explorar novas regiões e muitas técnicas podem ser utilizadas para executar esta rotina. O método *swap*, por exemplo, é um dos mais básicos e clássicos. Existem muitos outros

métodos já aplicados com sucesso neste processo, tal como a heurística permutacional de Silva e Soma (*op.cit.*), por exemplo.

A intensificação é um procedimento de refinamento da solução, uma vez que é aplicado em regiões onde a solução já está muito próxima da solução ótima.

4.3 Utilização da TS em soluções de problemas na área de Logística

Desde a proposição da técnica Busca Tabu por Glover em 1986, muitas pesquisas foram realizadas acerca de sua aplicação em Problemas de Otimização Combinatória Permutacional (POCP). A literatura demonstra o vasto campo de aplicação desta técnica, visto o considerável volume de artigos publicados anualmente em periódicos e revistas científicas em todo o mundo nas mais diversas áreas e para diferentes tipos de problemas, entre os quais, a logística.

Segundo Reeves (1995), a grande maioria das aplicações da TS ocorreu a partir de 1989 e, desde então, tem encontrado com sucesso boas soluções para uma variedade significativa de problemas. Dentre as principais aplicações, podemos destacar:

- i. Aplicações na área de planejamento: *employee scheduling, flow shop, single machine scheduling, multiple-machine scheduling, deadlines and setup lines, just-in-time scheduling.*
- ii. Aplicações na área de transportes: *travelling salesman (TSP), traveling repairman (TRP), vehicle routing problem (VRP).*
- iii. Aplicações em *lay-outs* e projetos de circuitos;
- iv. Telecomunicações: *path assignment.*
- v. Redes neurais;
- vi. Outros.

CAPÍTULO 5 – CARACTERIZAÇÃO DO PROBLEMA E ESTRATÉGIA DE SOLUÇÃO

Este capítulo é composto por quatro seções e tem por objetivos caracterizar o problema da pesquisa e propor uma estratégia de solução. A primeira seção apresenta a definição e as principais características dos serviços de manutenção em campo (*Field Service*). A segunda seção tem o objetivo de apresentar os tipos mais característicos deste tipo de serviço, traçando um comparativo de suas complexidades. A terceira seção detalha o caso das empresas de manutenção de elevadores, que servirá de base para a aplicação prática proposta por este trabalho. Por fim, a seção 5.4 trata da estratégia proposta para a solução do problema e a delimitação do trabalho.

5.1 Serviços de manutenção periódica em campo

Para diversos ramos de prestação de serviços de manutenção, torna-se necessário que periodicamente sejam realizadas intervenções nos equipamentos para a realização de manutenção preventiva, corretiva (reparação) ou procedimentos de troca de peças. Exemplos comuns destes serviços são: manutenção de elevadores e escadas rolantes, inspeções mensais em sistemas de segurança eletrônica, tais como sensores, câmeras e unidades de monitoramento, manutenção periódica de portões automáticos eletrônicos, reabastecimento ou reparação de caixas eletrônicos, vistorias em torres de energia eólica e subestações, manutenção de terminais de telecomunicações etc.

Raduan (2009) argumenta que são considerados serviços de atendimento e manutenção em campo, aqueles derivados do Problema do Caixeiro Viajante (TSP) tanto em sua forma estática como dinâmica e do Problema Dinâmico do Reparador Viajante (DTRP), introduzido por Bertsimas e Van Ryzin (1991).

Ocorre que, diferentemente de serviços realizados na indústria, onde todos os sistemas, máquinas e equipamentos encontram-se localizados praticamente no mesmo local, os serviços de manutenção em campo exigem um componente adicional diretamente relacionado com custo, que se refere exatamente ao deslocamento entre a unidade-base e o local da realização do serviço. Desta forma, é fundamental que seja realizado um planejamento logístico para atendimento dos locais onde há demanda por determinado serviço de reparação.

De uma maneira mais generalizada, serviços de atendimento em campo exigem o deslocamento de agentes, sejam técnicos de manutenção, policiais, bombeiros, paramédicos ou especialistas até o local da ocorrência onde o serviço se faz necessário, podendo este deslocamento ser realizado por diversas formas como, por exemplo, através do uso de veículos ou através de uma caminhada, tal como fazem os carteiros (RADUAN, 2009).

É válido salientar que alguns tipos de serviços de atendimento em campo são considerados críticos, visto que possuem função social de alta relevância como, por exemplo, paramédicos do SAMU, polícia e bombeiros. Estes casos diferem significativamente dos serviços de manutenção tradicionais, visto que praticamente não permitem programação, sendo o atendimento em caráter de emergência o registro mais comum.

5.2 Tipos de serviço de manutenção em campo

As demandas por serviços de manutenção em campo podem ser divididas basicamente em três grupos com características bem distintas:

- i. Manutenção preventiva programada;
- ii. Manutenção corretiva ou reparação;
- iii. Emergências.

Na manutenção preventiva programada, os serviços a serem realizados em campo são definidos por um procedimento operacional padrão (POP) onde são determinadas todas as rotinas de serviços a serem realizadas nos equipamentos, tais como a inspeção, ajuste e a lubrificação de peças e componentes. Este tipo de serviço em geral é periódico e visa prevenir a ocorrência de falhas; portanto, a visita do técnico ocorre independentemente do registro de uma falha ou defeito.

Já na manutenção corretiva, a demanda é incerta, ou seja, não há previsibilidade da ocorrência de demanda por reparação. Neste caso, o acionamento do serviço é uma consequência da existência de uma parada no sistema de produção, ou um defeito em determinado equipamento. A manutenção corretiva é um procedimento de gestão reativo, visto que espera pela falha para que um processo de reparo seja realizado.

Representa ainda, uma forma mais cara de serviço que a manutenção preventiva programada, o que faz empresas e especialistas em gestão de manutenção dedicarem-se a técnicas mais eficientes de solução e com menor custo empregado ao processo.

Dependendo do tipo de máquina a que se refere o serviço de reparação demandado, o tempo de espera pelo serviço de manutenção em campo é uma variável altamente relevante e incorre em custos elevados. Portanto, faz-se necessário comparar os custos de interrupção com o custo do atendimento. Raduan (2009) define o custo do atendimento como a soma dos custos de mobilização e de deslocamento até o local do serviço de reparo. Para os casos onde o custo de interrupção do serviço ou da parada de uma máquina em produção é mais elevado que o custo do atendimento, o tempo de resposta da empresa prestadora de serviços deverá ser o menor possível a fim de restabelecer o serviço à sua normalidade.

Para os demais casos onde o custo da interrupção é inferior ou equivalente ao custo do atendimento, é possível adotar uma estratégia de programação de atendimento, priorizando-se a redução dos custos na tomada de decisão quanto à forma de atendimento.

Exceções são aquelas onde há uma contratação de nível de serviço, também conhecido como contrato de *performance*. Neste caso, os custos do atendimento diferenciado já são previamente cobertos pela remuneração de contrato, atendendo aos fatores de competitividade normalmente característicos deste tipo de serviço.

Para casos de urgência e emergência, tratados como exceção, a mobilização de pessoal para atender o local de demanda de serviço deve ser imediata, uma vez que o tempo de resposta nestes casos é um fator bastante relevante.

Normalmente, é comum que as rotas de serviços das empresas especializadas em serviços de manutenção em campo sejam dimensionadas para absorver uma parcela de serviços programados (manutenções preventivas, por exemplo) e outra parcela para urgências e emergências. Outra alternativa é manter equipes distintas, tal como ocorre na maioria das empresas de manutenção e assistência técnica de elevadores e escadas rolantes.

Na Tabela 5.1, dada a seguir, apresenta-se um resumo característico que diferencia esses dois tipos característicos de demanda de serviço em campo.

Tabela 5.1 – Tipos de serviços de manutenção em campo para elevadores.

Característica	Manutenção Preventiva	Manutenção corretiva
Frequência de demanda do serviço	Programada	Incerta
Tempo dedicado ao serviço	Definido	Incerto
Plano de deslocamento	Roteirização estática (programada)	Roteirização dinâmica
Status do equipamento no momento da demanda pelo serviço	Em operação	Fora de operação ou com operação deficiente
Volume de ordens de serviços	1 (uma) OS para cada equipamento sob manutenção	Incerto; em geral, ¼ ou 25% do volume de manutenções preventivas
Regimento de atendimento	Normal	Normal, urgência ou emergência

Pela Tabela 5.1, é possível verificar que em termos de planejamento logístico, os serviços de manutenção em campo exigem diferentes estratégias logísticas, visto que possuem características distintas quando há ou não previsibilidade na frequência da demanda e no tempo requerido para o serviço.

5.3 A manutenção periódica de elevadores

De acordo com a ABNT – Associação Brasileira de Normas Técnicas, em 2008 existiam mais de 300.000 elevadores instalados no Brasil.

Elevadores são equipamentos eletro-mecânicos que demandam manutenção periódica e serviços de assistência técnica contínuos para a garantia de sua funcionalidade e segurança para todos os usuários. Tendo em vista a sua importância frente à necessidade de deslocamento de pessoas em edifícios privados, públicos e comerciais, hospitais e centros de convenções e entretenimento, a eficácia e o tempo de resposta dos serviços de manutenção são essenciais para o equilíbrio do dia-a-dia da população que utiliza este tipo de transporte.

Por questões legais, no Brasil, os elevadores devem possuir um plano de manutenção mensal, ou seja, é obrigatória a periodicidade do serviço de manutenção/conservação a cada mês para verificação de itens de segurança, ajustes em portas e no freio da máquina de tração, limpeza de trincos e testes gerais nos sensores existentes na caixa de corrida.

Dessa forma, mais de 320.000 elevadores passam mensalmente por manutenção periódica no Brasil, sem contar com os atendimentos corretivos, de urgência e emergência, que são atendidos num prazo médio de até 4 horas do momento do acionamento do chamado na Central de Atendimento das empresas.

O mercado é basicamente dividido entre as empresas multinacionais fabricantes e as empresas locais que prestam serviços unicamente de conservação, manutenção e reparos em elevadores.

Por tratar-se de um setor muito específico, encontram-se na literatura brasileira apenas um discreto número de artigos e dissertações publicadas na área de otimização deste tipo de serviço de manutenção, o que torna o setor bastante incipiente em termos do uso de modelos otimizados de logística de manutenção preventiva ou corretiva. Mesmo as empresas multinacionais fabricantes, apesar de serem referências mundiais, ainda demandam planejamento logístico nesta área operacional.

O presente trabalho aborda justamente a otimização de rotas de manutenção preventiva, que, juntamente com o trabalho publicado por Raduan (2009) na área de roteirização parcialmente dinâmica, podem contribuir para adoção de novas práticas logísticas neste setor empresarial.

5.3.1 A organização da manutenção de campo em zonas e setores

Como forma de gerenciar de forma mais eficaz a logística do atendimento aos serviços de manutenção em campo, a grande maioria das empresas agrupa os clientes em setores, dimensionando a equipe de técnicos reparadores de acordo com o volume de clientes de cada setor específico.

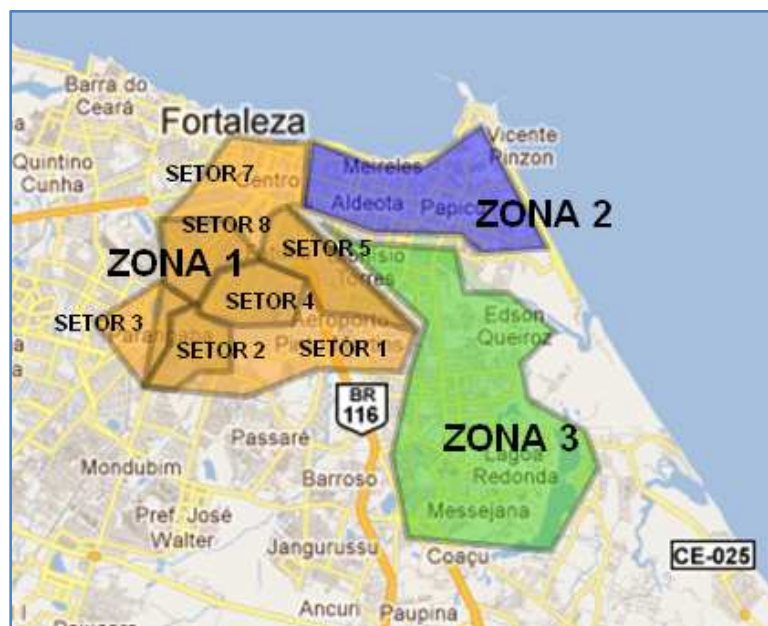
Quando o número de setores é elevado, é normal que haja um agrupamento dos setores em zonas. Com isso, é possível dimensionar a estrutura funcional indireta, tais como supervisores e consultores de contrato.

A Figura 5.1, dada a seguir, demonstra hipoteticamente uma divisão geográfica em zonas (ou áreas) e setores (ou rotas) na cidade de Fortaleza-CE. No caso do exemplo, apenas a zona 1 está subdividida em setores, a fim de que seja possível destacar essa característica comum dos serviços de manutenção em campo.

Entre os critérios mais usuais de segmentação ou agrupamento de clientes por parte das empresas, podemos destacar:

- i. Posicionamento geográfico, ou seja, agrupamento de clientes próximos uns dos outros de forma a minimizar o deslocamento dos agentes;
- ii. Segmentação técnica, que considera o agrupamento de clientes que possui equipamentos com características iguais ou muito semelhantes;
- iii. Segmentação por potencial, que está relacionado à decisão de agrupar clientes com tipo de atendimento requerido, por exemplo, um setor somente com hospitais e centros médicos, outro formado somente por edifícios comerciais de alto padrão, residenciais etc.
- iv. Métodos estatísticos, *clusterização*.

Figura 5.1 Organização em zonas e rotas (setores) de manutenção preventiva.



Para os serviços de manutenção em campo em áreas urbanas como as grandes capitais brasileiras, o critério geográfico geralmente tem sido o mais adotado por facilitar a logística de deslocamento, fator este decisivo quando o tempo de espera está diretamente relacionado com o nível de serviço pretendido.

Esse problema real das empresas em relação à dificuldade de sequenciar as rotas de manutenção em campo é típico do problema clássico do reparador viajante (TRP), onde o objetivo é minimizar o deslocamento total de cada rota de manutenção. Um exemplo real é apresentado na Figura 5.2 dada a seguir.

Figura 5.2 Otimização de uma rota de manutenção para o TRP.



67

preventiva é periódica, ou seja, ocorre dentro de um ciclo finito e pré-determinado de tempo no qual o serviço deve ser novamente realizado.

A periodicidade na qual um serviço de manutenção de campo deve ser realizado depende de uma série de fatores que não fazem parte do escopo do presente trabalho; porém, à medida que o serviço é essencial e que uma falha tem por consequência problemas graves, esse intervalo de tempo normalmente é bastante curto.

A manutenção de aeronaves comerciais, por exemplo, passam por inspeções diárias de manutenção, além de rotinas mensais, trimestrais e anuais. Elevadores possuem uma obrigatoriedade no Brasil para que a manutenção seja realizada ao menos uma vez por mês, a fim de que sejam verificados os itens de segurança necessários ao funcionamento do equipamento. Outros equipamentos como bombas de drenagem automática, geradores de médio porte a diesel, entre outros, podem possuir um intervalo maior de tempo entre uma manutenção preventiva e outra.

5.3.3 O Sistema de atendimento em empresas de manutenção

Algumas empresas privadas que são notadamente dedicadas aos serviços de manutenção de elevadores na cidade de Fortaleza-CE, foram pesquisadas no intuito de perceber se existe correlação entre os seus diferentes sistemas de atendimento. O objetivo foi caracterizar semelhanças no processo de manutenção em campo a fim de melhor contextualizar o problema a ser tratado nesse trabalho. As seguintes empresas foram elencadas para análise:

- Empresa “A”
- Empresa “B”
- Empresa “C”
- Empresa “D”

As empresas “B”, “C” e “D” são de pequeno porte e a empresa “A” é multinacional. Para as empresas pesquisadas, os seguintes aspectos foram semelhantes:

- i. Possuem uma Central de Atendimento ou número dedicado à entrada de demanda de serviços;

- ii. Todas as empresas listadas prestam serviços de manutenção preventiva e também correções de contingência (manutenções corretivas);
- iii. O agrupamento de áreas segue o padrão geográfico;
- iv. Não há um processo otimizado de determinação da melhor sequência de manutenção preventiva;
- v. As ocorrências de demanda por serviço de manutenção corretiva são priorizadas e, quando há o mesmo o mesmo grau de urgência, a ordem é determinada pela entrada da ocorrência (sistema FCFS – *first come first served*);
- vi. Em algumas circunstâncias, porém sem precisão específica, o procedimento ‘vizinho mais próximo’ é utilizado; esse sistema é conhecido como NN – *Nearest neighbor*.

Entre os aspectos mais relevantes, destaca-se a não utilização de processos otimizados na gestão dos serviços de manutenção, mesmo por parte da empresa multinacional, o que sinaliza uma grande oportunidade de ganho de eficiência, seja em tempo, custo e qualidade dos serviços prestados.

5.3.4 Diferenças no atendimento de manutenções

Conforme explicitado na Tabela 5.1, existem diferenças significativas entre os serviços de atendimento de manutenções preventivas programadas e as ocorrências de falha (corretivas).

Basicamente, as diferenças são as seguintes:

- i. As manutenções preventivas, por serem programadas, podem ser devidamente planejadas no início de cada mês; porém, a roteirização é bastante complexa, exigindo conhecimento logístico e investimento por parte das empresas, o que não foi verificado;
- ii. As manutenções corretivas não possuem uma demanda pré-definida, sendo difícil e complexa a estimativa do número provável de ocorrências, agravado ainda pela dependência de fatores externos, tais como oscilações ou falta de energia elétrica, danos ocasionados por terceiros etc.

- iii. O efetivo de pessoal destinado à manutenção preventiva é muito superior àquele dedicado às manutenções corretivas;
- iv. Dependendo do grau de agrupamento dos setores de manutenção, a preventiva pode ser realizada sem o uso de veículos (a pé), enquanto que os atendimentos de correção são normalmente atendidos por técnicos usando veículos (carros ou motos).

5.3.5 A delimitação do estudo

A otimização dos serviços de manutenção preventiva periódica permite um vasto estudo em diversos campos da logística e da pesquisa operacional. A experiência vivida em mais de dez anos de atuação na área de manutenção permite destacar os gargalos mais significativos deste tipo de atendimento.

O estudo proposto, conforme explicitado na introdução deste trabalho, é focar a otimização das rotas de manutenção preventiva através do uso de técnicas científicas de roteirização. O objetivo real é, com isso, possibilitar que as empresas reduzam seus custos e ganhem maior eficiência logística, tornando-se mais competitivas no mercado.

5.4 Estratégia de solução para o problema

Esta seção tem por objetivo apresentar o delineamento da estratégia para a solução do problema proposto pela pesquisa.

5.4.1 Visão geral da solução proposta

A estratégia de solução do problema foi estruturada seguindo os seguintes passos:

- i. caracterização precisa e delineada do problema;
- ii. estudo da literatura clássica e recente para escolha de técnicas adequadas à solução;
- iii. definição da técnica computacional a ser utilizada e da linguagem de programação;
- iv. desenvolvimento do modelo computacional que contém algoritmo básico;

- v. melhoria do algoritmo e testes na biblioteca TSPLIB para validação dos resultados gerados;
- vi. escolha de uma aplicação prática de uma empresa real;
- vii. testes diversos e análise comparativa do ganho obtido com o uso do modelo otimizado.

Para a solução proposta, foi desenvolvido um modelo computacional que utiliza a metaheurística *Tabu Search* por tratar-se de um potente mecanismo de solução para problemas complexos de otimização, entre os quais, o de roteirização.

O algoritmo foi escrito na linguagem Delphi e preparado para aplicações distintas, conforme definido nos objetivos específicos. Para efeitos de nomenclatura, intitulamos este método de r-TS, e as aplicações foram:

- i. ***Aplicação didática-demonstrativa***: nesta aplicação, é possível gerar ou importar diferentes matrizes distância D_{ij} , a partir de um dado de entrada “ n ”, referente à quantidade de pontos a serem visitados. Os dados de saída são soluções otimizadas de sequenciamento com o objetivo de reduzir a distância total. Trata-se de uma aplicação para fins didáticos;
- ii. ***Aplicação euclidiana 2D***: esta aplicação foi desenvolvida para validar o r-TS através da execução de problemas clássicos publicados na literatura e que utilizam coordenadas (x, y) como dados de entrada. É possível utilizar essa aplicação em casos práticos; e
- iii. ***Aplicação prática com coordenadas georeferenciadas***: esta aplicação visa a solução de problemas práticos, visto que os dados de entrada são coordenadas georeferenciadas (latitude e longitude) e possibilitam maior acuracidade nos resultados obtidos.

Para aplicações práticas de casos reais, foi escolhida uma empresa de manutenção de elevadores para teste e verificação dos resultados com a aplicação do r-TS.

Algumas premissas e dificuldades tiveram de ser tratadas durante o desenvolvimento da estratégia de solução, entre as quais destacamos: o cálculo das distâncias para composição da matriz D_{ij} .

5.4.2 O cálculo das distâncias

O ponto de partida, dada uma lista de pontos ou de clientes reais, está relacionada a composição da Matriz D_{ij} . Para efeitos acadêmicos, a maioria dos artigos pesquisados considera a distância entre dois pontos quaisquer uma reta, tal como a definição do Teorema de Pitágoras.

$$d_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2},$$

sendo i o ponto de origem e j, o destino.

O seguinte código foi utilizado para carregamento da matriz distância com coordenadas euclidianas:

Quadro 5.1 – *Procedure ‘cálc.dist.euclidiana’*

```
procedure TForm1.CarregaMatrizEuclidiana;
var
  I, J, X1, X2, Y1, Y2: Integer;
  Distancia: Double;
begin
  LabelCoordenadas.Visible := True;
  Matriz.DefaultColWidth := StrToInt(EditTamanhoColunaMatriz.Text);
  PontosXY.DefaultColWidth := 32;
  PontosXY.Width := 120;
  LabelMatriz.Left := 148;
  Matriz.RowCount := PontosXY.RowCount;
  Matriz.ColCount := PontosXY.RowCount;
  Matriz.Cells[1, 1] := '0';
  For I := 1 to PontosXY.RowCount - 1 do
    begin
      Matriz.Cells[I, 0] := PontosXY.Cells[0, I];
      Matriz.Cells[0, I] := PontosXY.Cells[0, I];
      X1 := StrToInt(PontosXY.Cells[1, I]);
      Y1 := StrToInt(PontosXY.Cells[2, I]);
      For J := 2 to PontosXY.RowCount - 1 do
        begin
          X2 := StrToInt(PontosXY.Cells[1, J]);
          Y2 := StrToInt(PontosXY.Cells[2, J]);
          Distancia := SQRT(SQR(X2 - X1) + SQR(Y2 - Y1));
          Matriz.Cells[J, I] := FloatToStrF(Distancia, ffNumber, 15, 2);
          Matriz.Cells[I, J] := FloatToStrF(Distancia, ffNumber, 15, 2);
        end;
      end;
    end;
end;
```

Tal aproximação merece ser discutida e melhor analisada, pois, dependendo da aplicação e das distâncias entre os pontos analisados, o erro poderá ser significativo. Para maiores distâncias ou mesmo para se reduzir o erro estimado, é recomendável que

seja considerada a distância linear obtida da trigonometria esférica (lei dos cossenos), conforme abaixo:

$$d_{ij} = 3.965(\arccos(\sin(Lat_i) \cdot \sin(Lat_j) + \cos(Lat_i) \cdot \cos(Lat_j) \cdot \cos(Long_j - Long_i)))$$

O seguinte procedimento foi utilizado no algoritmo para carregamento da Matriz D_{ij} a partir dos pontos com coordenadas georeferenciadas:

Quadro 5.2 – *Procedure ‘cálc.dist.georreferenciada’*

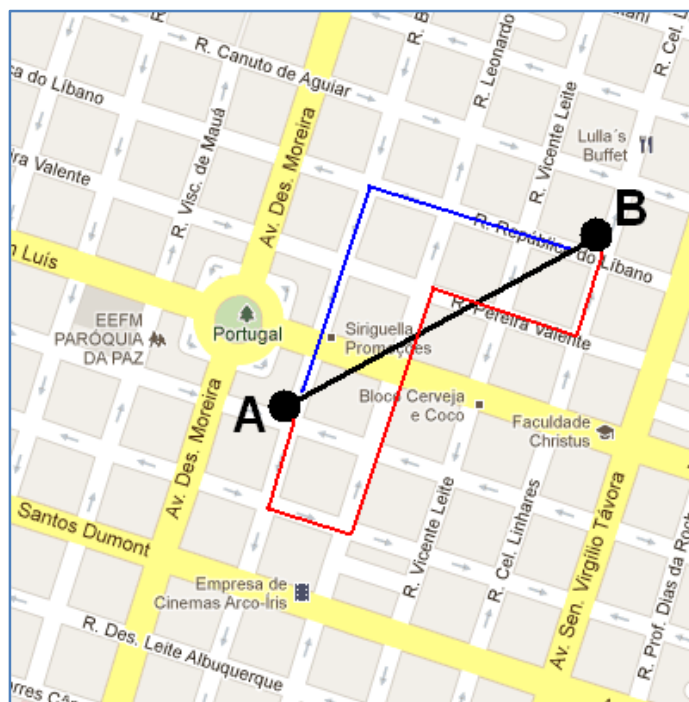
```
procedure TForm1.CarregaMatrizGeoReoreferenciada;
var
  I, J: Integer;
  X1, X2, Y1, Y2, Distancia, P1, P2, P3, P4, P5, ValorP: Double;
begin
  LabelCoordenadas.Visible := True;
  Matriz.DefaultColWidth := StrToInt(EditTamanhoColunaMatriz.Text);
  PontosXY.DefaultColWidth := 68;
  PontosXY.Width := 210;
  LabelMatriz.Left := 238;
  Matriz.RowCount := PontosXY.RowCount;
  Matriz.ColCount := PontosXY.RowCount;
  Matriz.Cells[1, 1] := '0';
  For I := 1 to PontosXY.RowCount - 1 do
    begin
      Matriz.Cells[I, 0] := PontosXY.Cells[0, I];
      Matriz.Cells[0, I] := PontosXY.Cells[0, I];
      X1 := ConverteFloat(PontosXY.Cells[1, I]);
      Y1 := ConverteFloat(PontosXY.Cells[2, I]);
      For J := 2 to PontosXY.RowCount - 1 do
        begin
          X2 := ConverteFloat(PontosXY.Cells[1, J]);
          Y2 := ConverteFloat(PontosXY.Cells[2, J]);
          P1 := Cos((90-X2) * pi/180);
          P2 := Cos((90-X1) * pi/180);
          P3 := Sin((90-X2) * pi/180);
          P4 := Sin((90-X1) * pi/180);
          P5 := Cos((Y1-Y2) * pi/180);
          ValorP := (P1 * P2) + (P3 * P4 * P5);
          Distancia := ArcCos(ValorP) * 6371; // 6371 = Raio da Terra
          Matriz.Cells[J, I] := FloatToStrF(Distancia, ffNumber, 15, 4);
          Matriz.Cells[I, J] := FloatToStrF(Distancia, ffNumber, 15, 4);
        end;
      end;
    end;
end;
```

5.4.2.1 Distâncias euclidianas (lineares) e reais

Raduan (2009) destaca que as distâncias lineares, na prática, apresentam certa imprecisão, porém, para efeito relativo de comparação de estratégias ou de desempenho com outros algoritmos, este fato não prejudica a análise. Para grandes distâncias, porém,

essa imprecisão pode se tornar mais evidente e inviabilizar a análise. A Figura 5.3, dada a seguir, ilustra este fato.

Figura 5.3 – Distância linear x Distância real



As distâncias reais são aquelas que consideram a rede de transportes, que é basicamente afetada pela malha viária disponível e sentido das ruas e avenidas. Para que o desenvolvimento do trabalho fosse realizado com essa premissa, seria necessária a disponibilização de um SIG – Sistema de Informações Georeferenciadas, o que tornaria a pesquisa mais complexa.

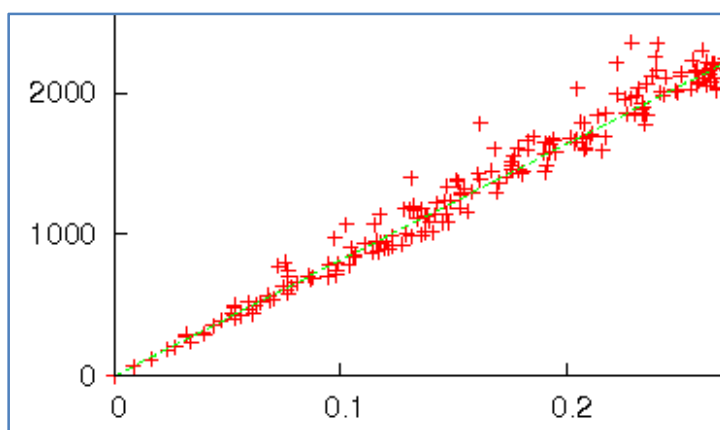
Vale destacar que existe uma consideração adicional a ser feita para o caso onde o deslocamento entre os pontos não é realizado através de um veículo e, portanto, não necessita atender às restrições impostas pelos sentidos das malhas viárias. Na Figura 5.3 estão ilustradas as distâncias linear e real. Na cor vermelha, está destacada a distância real com restrição de sentido na malha viária e, na cor azul, a distância real sem restrição de sentido.

É possível constatar que, para centros urbanos de alta densidade, a simplificação do problema real através da utilização de distâncias lineares é uma aproximação razoável,

ainda mais quando o deslocamento não é realizado por veículos, tal como o caso prático da manutenção preventiva de elevadores.

O gráfico abaixo, dado na Figura 5.4, foi obtido na *internet* em <http://obsn3.on.br/~jlkm/geopath/>. Trata-se de um estudo do professor João Luiz Khol Moreira acerca dos desvios existentes entre o cálculo da distância linear e a distância real entre todas as capitais brasileiras. É possível verificar que, para distância de até 2.000km, existe pouca discrepância entre as medidas. Para a aplicação proposta neste trabalho, que refere-se a uma região urbana, as distâncias entre pontos são inferiores a 15km, o que valida o uso da distância linear ajustada como uma distância real.

Figura 5.4 – Variações entre distância linear e real para capitais brasileiras



Fonte: <http://obsn3.on.br/~jlkm/geopath/>.

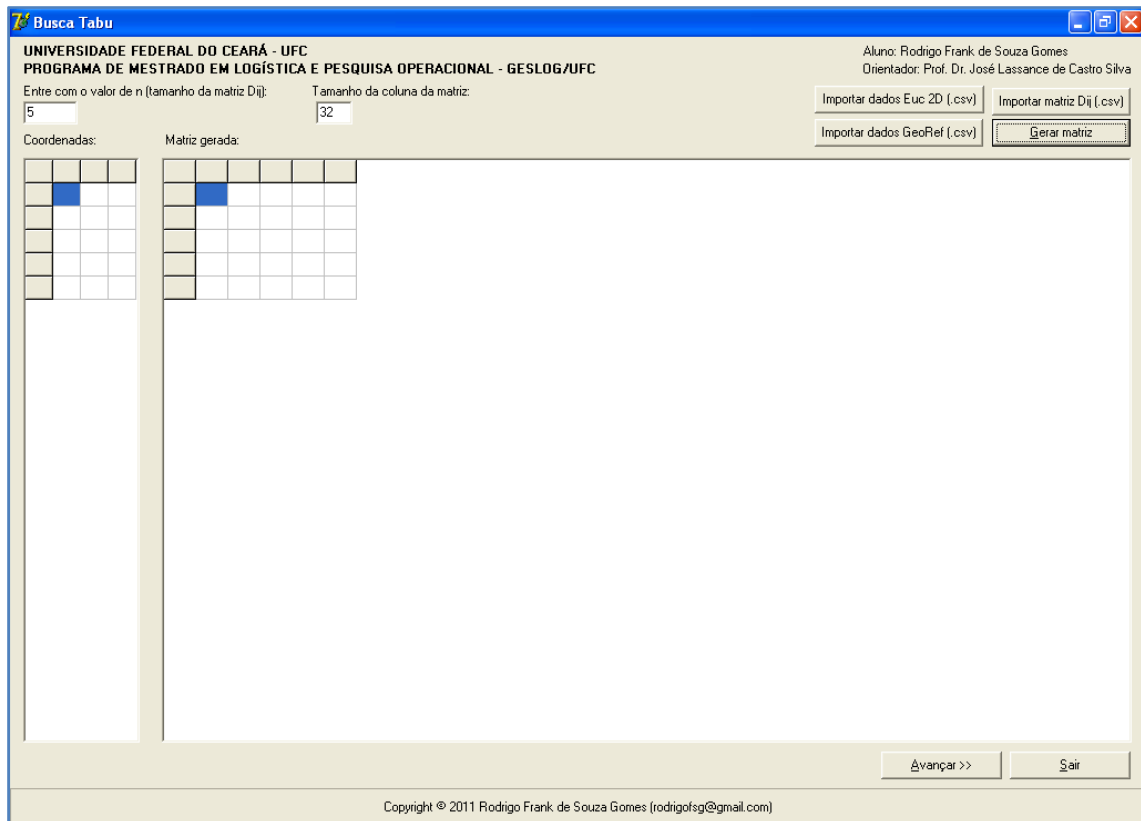
5.4.3 O algoritmo r-TS

O algoritmo utilizado foi desenvolvido com base nos objetivos específicos do presente trabalho. Desta forma, o aplicativo final permite ao usuário solucionar problemas de otimização de rotas de manutenção utilizando-se das seguintes formas de entrada:

- i. Importação de uma matriz D_{ij} que represente as distâncias entre cada um dos pontos ou clientes;
- ii. Importação de coordenadas 2D de clientes dispostos em um plano cartesiano.
- iii. Importação de coordenadas GPS (latitude e longitude) de clientes.
- iv. Geração aleatória de matrizes para testes básicos.

Todas as importações dos dados de entrada estão programadas de forma a receber exclusivamente arquivos com extensão *.csv*, do Microsoft Excel. Conforme a Figura 5.5, dada a seguir, a partir do código fonte foi elaborado um aplicativo com *interface* amigável para os experimentos a serem realizados.

Figura 5.5 Tela principal do aplicativo r-TS



As variáveis utilizadas no r-TS estão listadas no Quadro 5.3 e as funções e procedimentos desenvolvidos para busca de soluções viáveis constam no Quadro 5.4. Apesar da complexidade da metaheurística *Tabu Search* e de sua eficiência na solução de diversos problemas práticos de grande vulto, o número de variáveis e funções necessárias para implementação do método é relativamente pequeno. O código foi escrito com o rigor da linguagem, de forma a destacar que a *expertise* de programação elegante ainda não era parte dos objetivos do presente trabalho.

Quadro 5.3 – Variáveis utilizadas no r-TS

```
var
  Form1: TForm1;
  Matriz_Distancia: Array[0..4] of Array[0..4] of integer;
  Sequencia_Atual,          Sequencia_Atual_Matriz,          Proxima_Iteracao:
  Array[0..9999] of integer;
  Fila_Tabu: Array[0..99] of string;
  TamanhoFilaTabu, QtdIteracoes, QuantidadeTrocas: Integer;
  hI, mI, sI: Integer;
  MenorDistanciaEncontradaIteracao,          MenorDistanciaEncontradaTotal:
  Double;
  MelhorSequenciaGeral: String;
```

Quadro 5.4 – Funções e procedimentos do código r-TS

```
function DistanciaTotalSequenciaAtual: Double;
procedure InicializaAmbiente;
procedure InicializaIteracao;
function InformaSequenciaAtual: String;
function InformaListaTabu: String;
function CalculaHoraInterna: String;
Procedure CarregaMatrizEuclidiana;
procedure CarregaMatrizGeoReoreferenciada;
procedure TrocaEspelho;
procedure TrocaEspelhoInversa;
procedure GerarPermutacoes(Sender: TObject);
{ Private declarations }
```

5.4.3.1 O algoritmo básico do r-TS

Tendo em vista que um dos objetivos do trabalho é a aplicação prática da metaheurística TS em otimização de rotas (setores) de manutenção em campo como forma de ganho de eficiência logística, o desenvolvimento de um aplicativo computacional foi o “meio” para alcance do fim, e não o objetivo principal do trabalho.

O r-TS é um método desenvolvido com critérios de calibragem pré-definidos. Dessa forma, as variáveis de entrada do problema são fixas e fazem parte do procedimento de inicialização do algoritmo. Utilizou-se o pseudo-código básico apresentado por Viana (1998), mostrado no Quadro 4.1 do capítulo anterior, como base para o desenvolvimento do r-TS.

Nos Quadros 5.5 a 5.9, dados a seguir, são apresentados alguns dos procedimentos e funções escritas no código fonte do r-TS.

O Quadro 5.5, a seguir, ilustra a parte do código fonte dedicada a definição das variáveis iniciais, que são: a quantidade de iterações (MÁX.ITER), que representa o critério de parada do r-TS, o tamanho da Fila *Tabu* (F), a quantidade de trocas (que está diretamente ligada à diversificação de cada iteração, intitulada no aplicativo de NV-número de vizinhos) e a sequência ou permutação inicial do problema, que trata-se do ponto de partida da busca otimizada da solução do problema.

Quadro 5.5 – Procedure “TForm1.InicializaIteracao”

```

procedure TForm1.InicializaIteracao;
var
  I: Integer;
begin
  try
    QtdIteracoes := StrToInt(EditQtdIteracoes.Text);
  except
    ShowMessage('Quantidade de Iterações inválida.');
```

```

    QtdIteracoes := 1;
  end;
  //
  try
    TamanhoFilaTabu := StrToInt(EditTamanhoFilaTabu.Text);
  except
    ShowMessage('Tamanho da Fila Tabu inválido.');
```

```

    TamanhoFilaTabu := 1;
  end;
  //
  try
    QuantidadeTrocas := StrToInt(EditQtdTrocas.Text);
  except
    ShowMessage('Quantidade de Trocas da Iteração inválida.');
```

```

    QuantidadeTrocas := 1;
  end;
  // Inicia com a primeira sequência de frente para trás (exemplo: 0,
  1, 2, 3, 4);
  for I := 0 to Matriz.RowCount - 2 do
  begin
    Sequencia_Atual_Matriz[I] := I;
    Sequencia_Atual[I] := I;
    Proxima_Iteracao[I] := I;
  end;
  //
  // for I := 0 to TamanhoFilaTabu - 1 do
  //   Fila_Tabu[I] := '';

  // for I := 0 to 99 do
  //   Fila_Tabu[I] := '';
  //
end;
```

O Quadro 5.6, apresentado abaixo, descreve a função de cálculo da distância total de uma determinada permutação. É através da avaliação deste resultado que o algoritmo r-

TS compara as melhores soluções de uma determinada iteração e inclui na Fila *Tabu* o movimento *swap* que gerou a melhor solução. Conforme observa-se no código fonte do Quadro 5.6, no r-TS é possível calibrar, através da marcação de um *checkbox*, se há ou não a necessidade de incluir o retorno ao ponto inicial na avaliação do resultado de uma permutação. Conforme será abordado adiante, esse procedimento visa atender aplicações práticas onde o retorno não se faz necessário. No Quadro 5.7, a seguir, é mostrada a função e procedimento da Fila *Tabu*, onde é destacada a técnica de adição de um movimento *swap* à lista *tabu* ocupando um campo da variável vetorial que representa $F[i]$. De acordo com a calibração inicial do tamanho da Fila *Tabu*, o movimento permanece na lista por um número de iterações igual a F .

Os Quadros 5.8 e 5.9 abordam procedimentos simples que objetivam informar ao r-TS o *status* do tamanho da Fila *Tabu* e da sequência (permutação) atual a cada iteração. Tais informações são essenciais para o funcionamento do código, visto que se tratam de informações básicas e essenciais para a evolução da busca de boas soluções viáveis para cada problema.

Quadro 5.6 – Função “**TForm1.DistanciaTotalSequenciaAtual**”

```
function TForm1.DistanciaTotalSequenciaAtual: Double;
var
  I, Origem, Destino: Integer;
  DistanciaTotal: Double;
begin
  DistanciaTotal := 0;
  for I := 0 to Matriz.RowCount - 3 do
  begin
    Origem := Sequencia_Atual[I + 1] + 1;
    Destino := Sequencia_Atual[I] + 1;
    DistanciaTotal := DistanciaTotal +
    ConverteFloat(Matriz.Cells[Origem, Destino]);
  end;
  if CheckRetorno.Checked then
  begin
    Origem := Sequencia_Atual[0] + 1;
    Destino := Sequencia_Atual[Matriz.RowCount - 2] + 1;
    DistanciaTotal := DistanciaTotal +
    ConverteFloat(Matriz.Cells[Origem, Destino]);
  end;
  Result := DistanciaTotal;
end;
```

Quadro 5.7 – Função e Procedimento “FilaTabu”

```
function PertenceFilaTabu(Posicao1, Posicao2: Integer): Boolean;
var
  I: Integer;
begin
  Result := False;
  for I := Low(Fila_Tabu) to High(Fila_Tabu) do
  begin
    if (Fila_Tabu[I] = IntToStr(Posicao1) + ',' + IntToStr(Posicao2))
or (Fila_Tabu[I] = IntToStr(Posicao2) + ',' + IntToStr(Posicao1)) then
      Result := True
    end;
  end;
end;

procedure AdicionaFila_Tabu(Posicao1, Posicao2: Integer);
var
  I: Integer;
begin
  for I := (TamanhoFilaTabu - 1) downto 1 do
    Fila_Tabu[I] := Fila_Tabu[I - 1];
  Fila_Tabu[0] := IntToStr(Posicao1) + ',' + IntToStr(Posicao2);
end;
```

Quadro 5.8 – Função “TForm1.InformaSequenciaAtual”

```
function TForm1.InformaSequenciaAtual: String;
var
  I: Integer;
begin
  Result := '';
  for I := 0 to (Matriz.RowCount - 2) do
    Result := Result + Matriz.Cells[0,Sequencia_Atual[I]+1] + ' ';
  // Result := Result + IntToStr(Sequencia_Atual[I] + 1) + ' ';

  if CheckRetorno.Checked then
    Result := Result + Matriz.Cells[0,Sequencia_Atual[0]+1] + ' ';
  // Result := Result + IntToStr(Sequencia_Atual[0] + 1) + ' ';
end;
```

Quadro 5.9 – Função “TForm1.InformaListaTabu”

```
function TForm1.InformaListaTabu: String;
var
  I: Integer;
begin
  Result := '';
  for I := 0 to (StrToInt(EditTamanhoFilaTabu.Text) - 1) do
    Result := Result + '[' + Fila_Tabu[I] + ' ';
  end;
```


5.4.3.2 Calibragem inicial do r-TS

Conforme já exposto no Capítulo 4, a metaheurística Tabu Search possui aplicabilidade extensa e permite o desenvolvimento de técnicas de permutação flexíveis. O TS não estabelece parâmetros exatos quanto à calibragem dos algoritmos desenvolvidos e não há um critério de parada específico, ou mesmo uma regra pré-estabelecida. Dessa forma, o r-TS usou os seguintes parâmetros a serem calibrados antes do processamento dos dados:

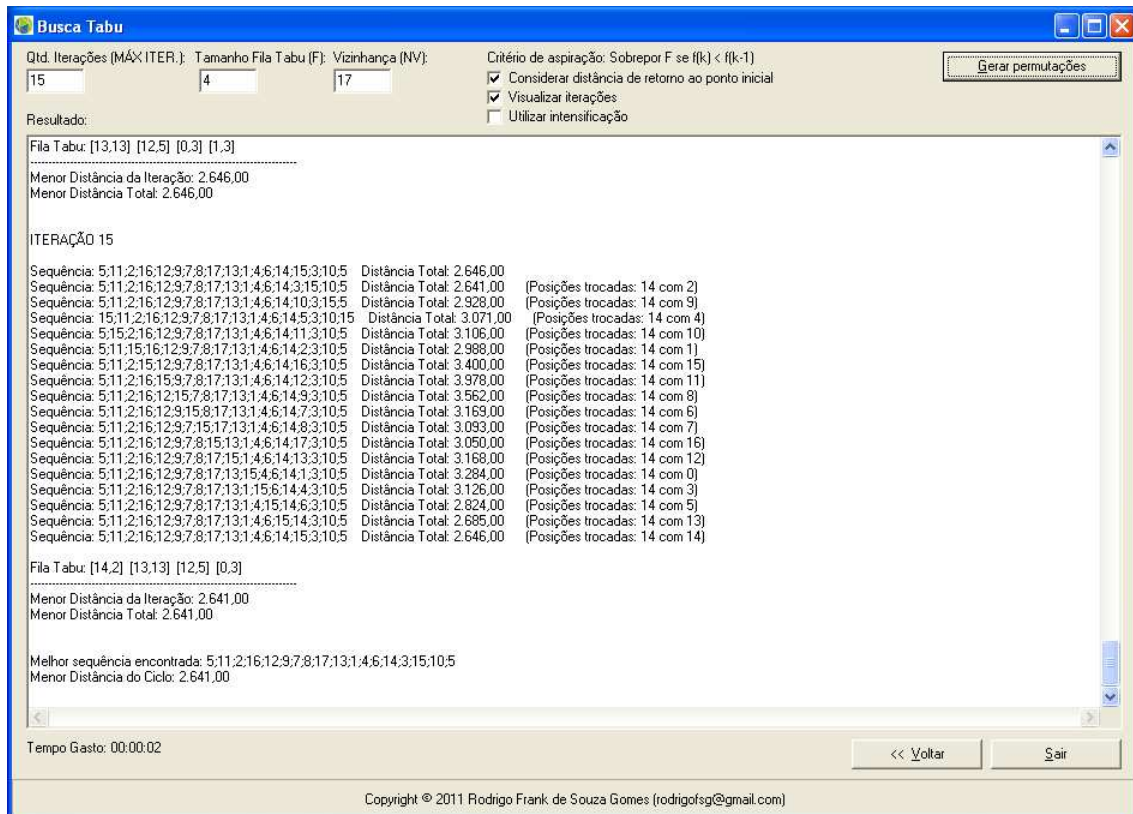
- i. MÁX_ITER: número máximo de iterações;
- ii. F: tamanho da fila tabu;
- iii. NV: número de vizinhos a serem analisados para uma dada iteração (perturbação) para diversificação.
- iv. Retorno ou não ao ponto inicial. No caso de algumas aplicações práticas, o retorno ao ponto de partida não se faz necessário.
- v. Uso ou não de intensificação.

Além dos itens apresentados acima, ainda é possível optar ou não pela visualização do processo iterativo, visto que para soluções mais complexas, torna-se inviável a visualização do passo-a-passo da solução, em virtude do “consumo” de memória e tempo de processamento, sendo necessário desabilitar essa funcionalidade.

A Figura 5.6, dada a seguir, apresenta a tela do r-TS para um problema de 17 pontos, que utiliza MÁX_ITER.=15, F=4 e NV=17, com retorno ao ponto inicial e visualização do processo iterativo.

Nos testes realizados para validação do r-TS foram utilizadas as referências de calibração encontradas na literatura (apontadas na seção 4.2.2). Porém, tais referências foram apenas para um ponto de partida, visto que dependendo da complexidade do problema, alterações na calibração são fundamentais para o alcance de soluções melhores.

Figura 5.6 – Parâmetros de Calibragem do r-TS



5.4.3.3 Critério de aspiração e a Intensificação no r-TS

Glover (1990) destaca que um critério de aspiração pode ser aplicado a um atributo *tabu*, durante o período em que ele permanece na Fila *Tabu* e sobrepondo seu *status*, se o resultado obtido por este movimento melhora o resultado anteriormente obtido no processo.

O funcionamento deste critério no algoritmo r-TS foi definido como: se *S* está na Fila *Tabu* (estado proibido) é possível mover-se para o estado corrente *S*, caso:

$$f(S) < f(S^*)$$

onde *S** é a melhor solução encontrada até o momento.

A Figura 5.7, apresentada a seguir, ilustra o funcionamento do critério de aspiração no r-TS. A calibragem foi definida para um problema de 17 pontos, que utiliza

MÁX.ITER.=50, F=4 e NV=17, com retorno ao ponto inicial e visualização do processo iterativo, conforme se observar na barra superior da tela.

A cada iteração são apresentados todos os resultados provenientes da função perturbação, a fim de que novas soluções sejam avaliadas. O movimento *swap* que gerou a melhor solução da iteração (*Menor Distância da Iteração*) é adicionado a Fila *Tabu*. Caso o melhor movimento *swap* seja *tabu*, o r-TS analisa o critério de aspiração a fim de avaliar se a solução gerada é melhor que àquela encontrada até o momento em todas as iterações anteriores (*Menor Distância Total*). Em caso afirmativo, o r-TS sobrepõe o status *tabu* e o movimento ocorre. Caso contrário, conforme Iteração 18 apresentada na Figura 5.7, dada abaixo, o movimento *swap* (1,9) gerou a melhor solução da iteração (2.518,00), porém, não suficientemente melhor que resultados obtidos em iterações anteriores (2.390,0). Neste caso, o movimento não ocorre e segue-se a ordem do segundo melhor, que na ilustração é o movimento *swap* (1,2).

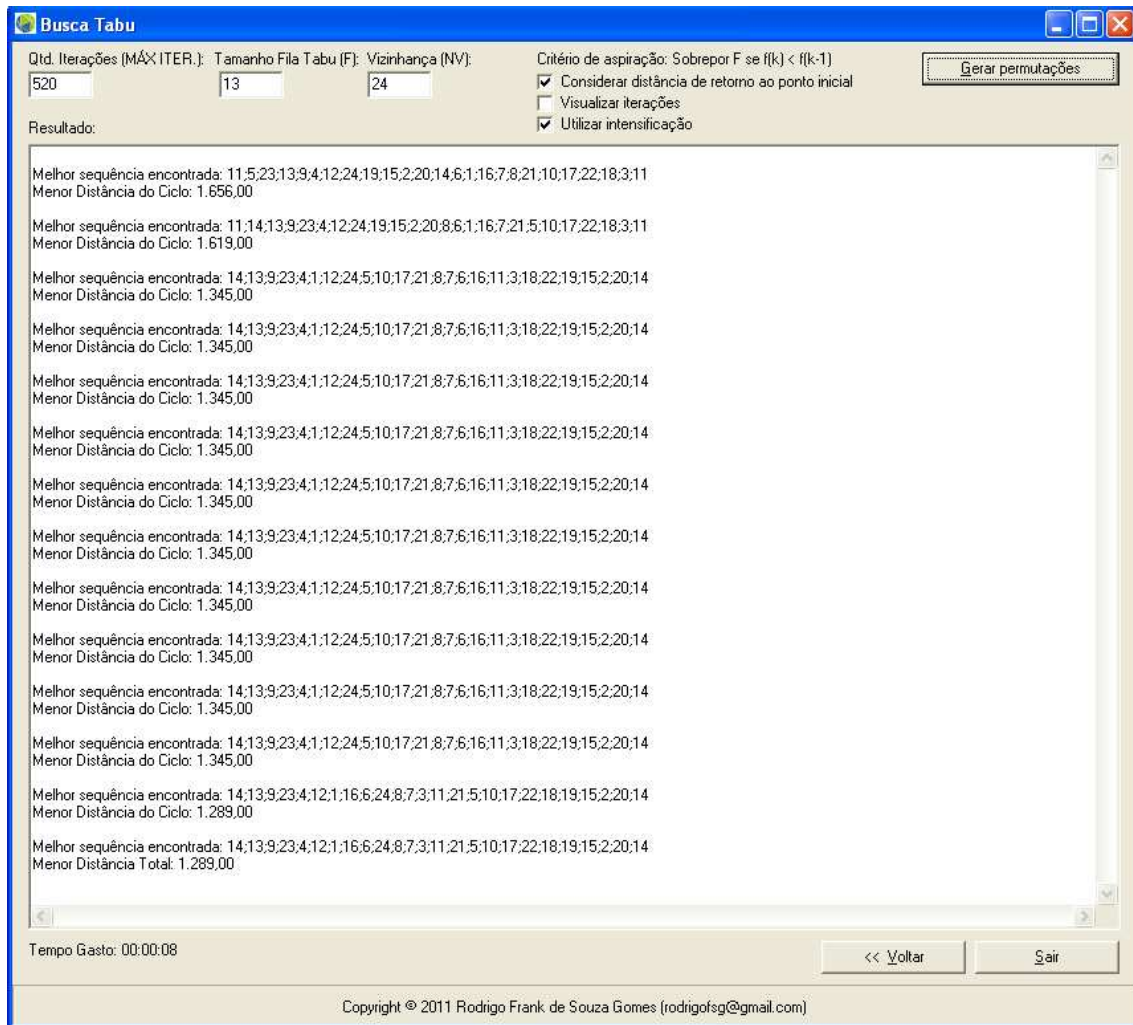
Figura 5.7 Exemplo de aplicação do critério de aspiração no r-TS

Qtd. Iterações (MÁX.ITER.):		Tamanho Fila Tabu (F):		Vizinhança (NV):		Critério de aspiração: Sobrepor F se $f(k) < f(k-1)$	
50		4		16		<input checked="" type="checkbox"/> Considerar distância de retorno ao ponto inicial <input checked="" type="checkbox"/> Visualizar iterações <input type="checkbox"/> Utilizar intensificação	
Resultado:							
Sequência: 5,11,4,16,12,2,13,1,17,6,8,7,14,15,3,10,9,5		Distância Total: 3.628,00		(Posições trocadas: 1 com 8)			
Sequência: 5,11,4,16,12,9,2,1,17,6,8,7,14,15,3,10,13,5		Distância Total: 3.588,00		(Posições trocadas: 1 com 12)			
Sequência: 5,11,4,16,12,9,13,2,17,6,8,7,14,15,3,10,1,5		Distância Total: 3.685,00		(Posições trocadas: 1 com 0)			
Sequência: 5,11,4,16,12,9,13,1,2,6,8,7,14,15,3,10,17,5		Distância Total: 3.526,00		(Posições trocadas: 1 com 16)			
Sequência: 5,11,4,16,12,9,13,1,17,2,8,7,14,15,3,10,6,5		Distância Total: 3.480,00		(Posições trocadas: 1 com 5)			
Sequência: 5,11,4,16,12,9,13,1,17,6,2,7,14,15,3,10,8,5		Distância Total: 3.582,00		(Posições trocadas: 1 com 7)			
Sequência: 5,11,4,16,12,9,13,1,17,6,8,2,14,15,3,10,7,5		Distância Total: 3.486,00		(Posições trocadas: 1 com 6)			
Sequência: 5,11,4,16,12,9,13,1,17,6,8,7,2,15,3,10,14,5		Distância Total: 3.238,00		(Posições trocadas: 1 com 13)			
Sequência: 5,11,4,16,12,9,13,1,17,6,8,7,14,2,3,10,15,5		Distância Total: 3.059,00		(Posições trocadas: 1 com 14)			
Sequência: 5,11,4,16,12,9,13,1,17,6,8,7,14,15,2,10,3,5		Distância Total: 2.699,00		(Posições trocadas: 1 com 2)			
Sequência: 5,11,4,16,12,9,13,1,17,6,8,7,14,15,3,2,10,5		Distância Total: 2.487,00		(Posições trocadas: 1 com 9)			
Fila Tabu: [1,4] [1,9] [14,2] [13,2]							
Menor Distância da Iteração: 2.518,00							
Menor Distância Total: 2.390,00							
ITERAÇÃO 18							
Sequência: 2,11,4,16,12,9,13,1,17,6,8,7,14,15,3,10,5,2		Distância Total: 2.518,00		(Posições trocadas: 1 com 10)			
Sequência: 11,2,4,16,12,9,13,1,17,6,8,7,14,15,3,10,5,11		Distância Total: 2.689,00		(Posições trocadas: 1 com 3)			
Sequência: 4,11,2,16,12,9,13,1,17,6,8,7,14,15,3,10,5,4		Distância Total: 3.182,00		(Posições trocadas: 1 com 15)			
Sequência: 16,11,4,2,12,9,13,1,17,6,8,7,14,15,3,10,5,16		Distância Total: 3.984,00		(Posições trocadas: 1 com 11)			
Sequência: 12,11,4,16,12,9,13,1,17,6,8,7,14,15,3,10,5,12		Distância Total: 3.913,00		(Posições trocadas: 1 com 8)			
Sequência: 9,11,4,16,12,2,13,1,17,6,8,7,14,15,3,10,5,9		Distância Total: 3.620,00		(Posições trocadas: 1 com 12)			
Sequência: 13,11,4,16,12,9,2,1,17,6,8,7,14,15,3,10,5,13		Distância Total: 3.571,00		(Posições trocadas: 1 com 0)			
Sequência: 1,11,4,16,12,9,13,2,17,6,8,7,14,15,3,10,5,1		Distância Total: 3.668,00		(Posições trocadas: 1 com 16)			
Sequência: 17,11,4,16,12,9,13,1,2,6,8,7,14,15,3,10,5,17		Distância Total: 3.509,00		(Posições trocadas: 1 com 5)			
Sequência: 6,11,4,16,12,9,13,1,17,2,8,7,14,15,3,10,5,6		Distância Total: 3.463,00		(Posições trocadas: 1 com 7)			
Sequência: 8,11,4,16,12,9,13,1,17,6,2,7,14,15,3,10,5,8		Distância Total: 3.565,00		(Posições trocadas: 1 com 6)			
Sequência: 7,11,4,16,12,9,13,1,17,6,8,2,14,15,3,10,5,7		Distância Total: 3.469,00		(Posições trocadas: 1 com 13)			
Sequência: 14,11,4,16,12,9,13,1,17,6,8,7,2,15,3,10,5,14		Distância Total: 3.221,00		(Posições trocadas: 1 com 14)			
Sequência: 15,11,4,16,12,9,13,1,17,6,8,7,14,2,3,10,5,15		Distância Total: 3.094,00		(Posições trocadas: 1 com 2)			
Sequência: 3,11,4,16,12,9,13,1,17,6,8,7,14,15,2,10,5,3		Distância Total: 2.682,00		(Posições trocadas: 1 com 9)			
Sequência: 10,11,4,16,12,9,13,1,17,6,8,7,14,15,3,2,5,10		Distância Total: 2.518,00		(Posições trocadas: 1 com 4)			
Sequência: 5,11,4,16,12,9,13,1,17,6,8,7,14,15,3,10,2,5		Distância Total: 2.390,00					
Fila Tabu: [1,2] [1,4] [1,9] [14,2]							
Menor Distância da Iteração: 2.682,00							
Menor Distância Total: 2.390,00							

Devido ao CRITÉRIO DE ASPIRAÇÃO a permutação (1,2) foi utilizada em virtude de F.

OBSERVE QUE HAVIA SOLUÇÕES MELHORES NESTA ITERAÇÃO

Figura 5.8 Intensificação no r-TS para o problema gr24 (TSPLIB)



Este processo iterativo de intensificação ocorre da seguinte forma no r-TS:

Seja $F[i]$ um vetor *Fila Tabu*: $F[i] = [_ , _ , _ , \dots , _] \forall i = 1, \dots, n$

Seja h = número de ciclos; $h = n$;

Para $h = 0, 1, \dots, n$, temos que:

Ciclo 0: processo normal de $F[i]$ (mesmo que sem intensificação)

Ciclo 1: $F[i] = [_ , _ , _ , \dots , \#]$, $\forall i = 1, \dots, n - 1$

// observe que após o ciclo original, $F[i]$ ‘perde’ a última posição (‘#’ representa uma posição ocupada).

Ciclo 2: $F[i] = [_ , _ , \dots , \# , \#]$, $\forall i = 1, \dots, n - 2$

// observe que agora $F[i]$ ‘perde’ as duas últimas posições.

...

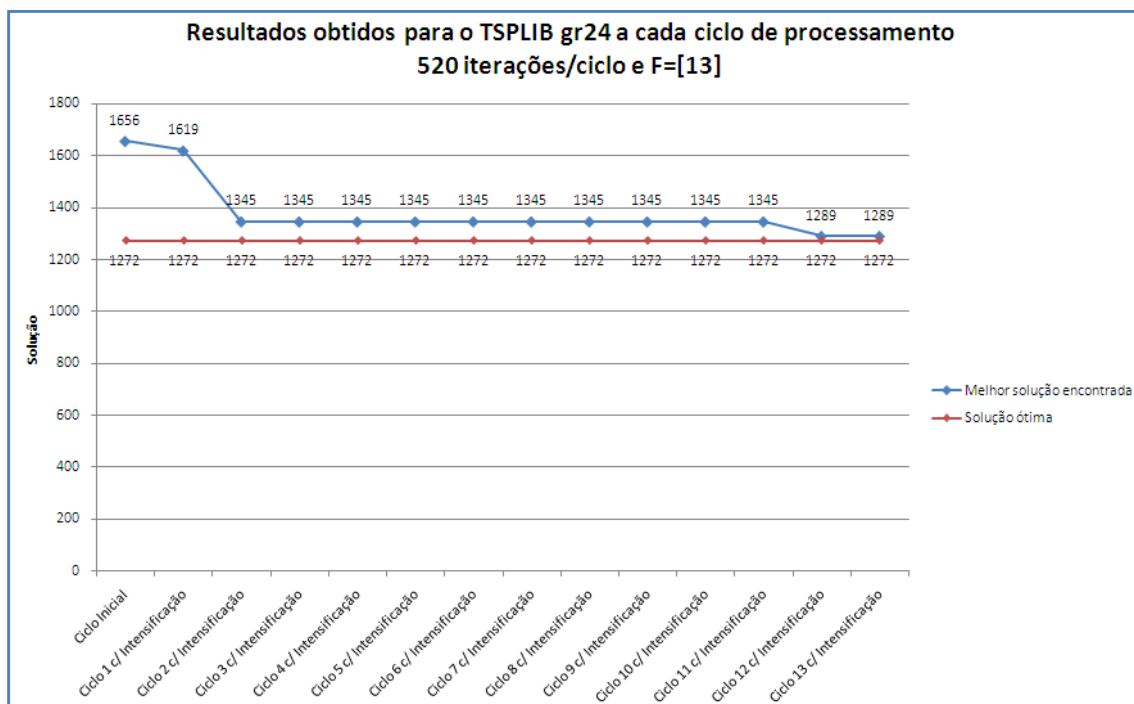
Ciclo h : $F[i] = [_ , \# , \dots , \# , \#]$, $\forall i = 1$

Onde # representa uma troca (*swap*) que encontrava-se na posição “n” da fila tabu ao final do ciclo anterior. O elemento “#” permanecerá fixo na Fila *Tabu* durante todos os ciclos seguintes.

Conforme ilustrado pela Figura 5.8 e de acordo com os experimentos computacionais realizados e apresentados no Capítulo 6, o processo de intensificação foi utilizado em praticamente todos os problemas pesquisados na TSPLIB de forma a obter a solução ótima ou o mais próximo possível de tal solução.

A Figura 5.9, dada a seguir, mostra um gráfico onde destaca-se a eficiência do r-TS em “fugir” de ótimos locais para o problema gr24 da TSPLIB..

Figura 5.9 – Evolução gráfica do resultado do problema gr-24 (TSPLIB)



CAPÍTULO 6 – EXPERIMENTOS COMPUTACIONAIS

Neste capítulo são apresentados os experimentos computacionais realizados com o algoritmo desenvolvido. A primeira seção (6.1) trata da execução do r-TS para diversas instâncias do TSP disponíveis na TSPLIB. A segunda seção (6.2) é dedicada a uma aplicação prática real em uma empresa de serviços de manutenção em elevadores, escadas rolantes e plataformas elevatórias.

6.1 Aplicação aos problemas clássicos da literatura TSPLIB

Com o objetivo de validar o algoritmo r-TS foram realizados diversos experimentos computacionais para as instâncias disponibilizadas na TSPLIB. Os experimentos foram realizados em um microcomputador INTEL CORE i7 com memória de 3GB RAM.

Foram elencados dez problemas com dados de entrada do tipo matriz D_{ij} ou do tipo coordenadas 2D, de forma a observar a eficácia dos resultados em um horizonte de 0 a 280 pontos. Além dos problemas clássicos, outras simulações foram realizadas como forma de validar as soluções apresentadas. Problemas práticos também foram abordados a fim de verificar a consistência das respostas.

Os resultados iniciais estão apresentados na Tabela 6.1, dada a seguir, e os desvios percentuais foram calculados da seguinte forma:

$$Desvio = \frac{[(Solução\ r - TS) - (Solução\ ótima\ TSPLIB)] \times 100}{Solução\ ótima\ TSPLIB}$$

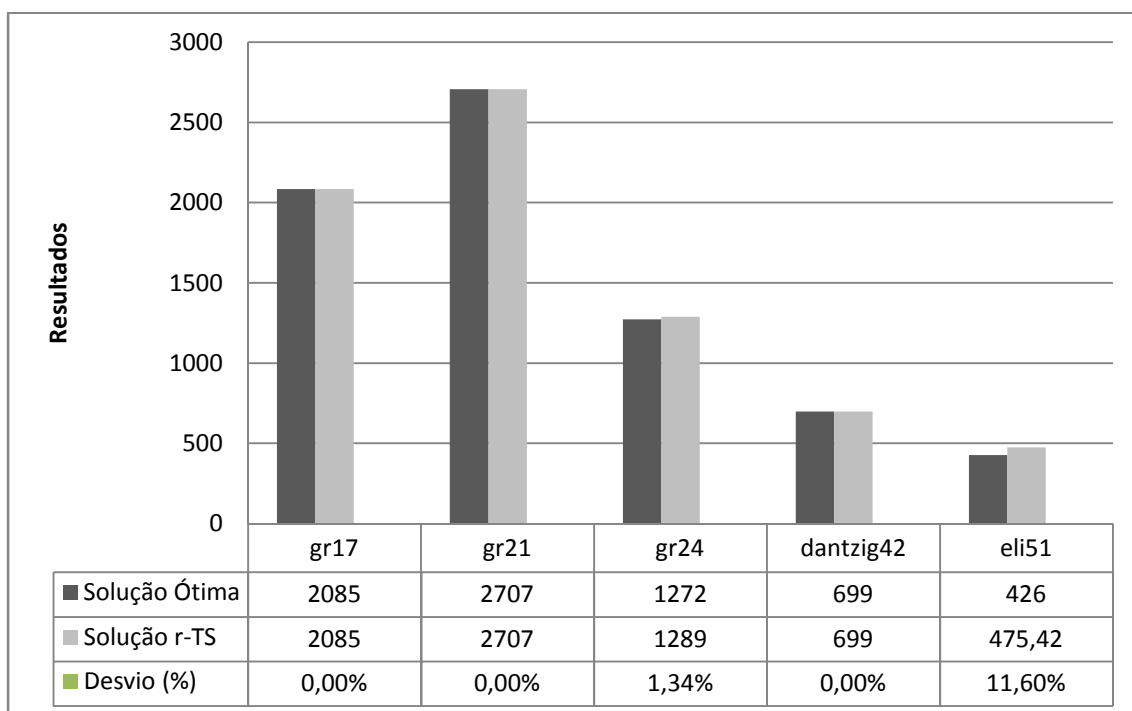
Podemos observar que o algoritmo r-TS obteve bons resultados para os problemas constantes da TSPLIB com número de pontos variando de 17 a 51. Destes problemas, apenas no gr24 e no eil51 a solução ótima não foi alcançada, apesar de o desvio registrado no gr24 ser muito pequeno, igual a 1.34%, conforme demonstrado no gráfico da Figura 6.1. Isto se deve principalmente à técnica de diversificação utilizada na função de perturbação da vizinhança a cada iteração realizada no TS, visto que o uso da

técnica *swap* é muito ‘fraca’ para diversificar soluções viáveis em um universo de pontos relativamente grande.

Tabela 6.1 – Resultados obtidos por r-TS.

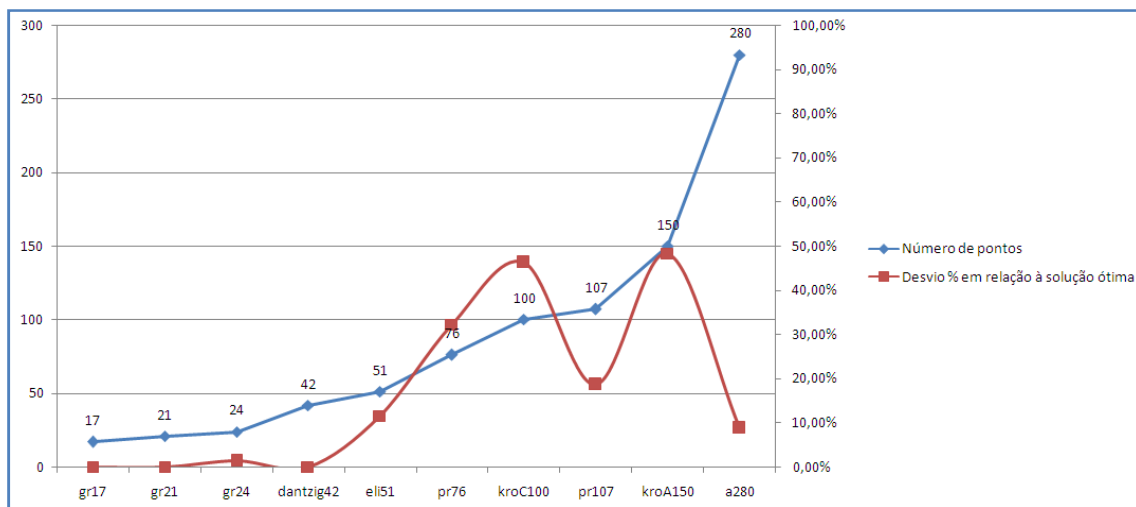
Arquivo	n	Tipo de dado de entrada	Solução ótima (TSPLIB)	Solução encontrada (r-TS)	Desvio (%)	Tempo
gr17	17	MATRIZ D_{ij}	2085	2085	0.00	1s
gr21	21	MATRIZ D_{ij}	2707	2707	0.00	0s
gr24	24	MATRIZ D_{ij}	1272	1289	1.34%	16s
dantzig42	42	COORD_2D	699	699	0.00	0s
eil51	51	COORD_2D	426	475,42	11.6%	30s
pr76	76	COORD_2D	108159	143012,84	32.22%	75s
kroC	100	COORD_2D	20749	30352,89	46,29%	184s
pr107	107	COORD_2D	44303	52555,66	18.62%	473s
kroA	150	COORD_2D	26524	39350,03	48,36%	437s
a280	280	COORD_2D	2579	2794,76	8.76%	55s

Figura 6.1 – Gráfico comparativo entre soluções obtidas para problemas simétricos do TSPLIB até 51 pontos.



A Figura 6.2 apresenta um gráfico com os desvios ilustrados de cada problema testado comparativamente ao número de pontos. Nele é possível observar a fragilidade do r-TS quando a instância do problema se torna mais complexa. Considerando desvios ‘aceitáveis’ àqueles entre 0 e aproximadamente 10%, observa-se que seis dos dez problemas testados encontram-se nesta faixa, sendo que destes seis, a maioria (5) está entre 0 e 51 pontos.

Figura 6.2 – Desvios % x Pontos para diversas instâncias do TSPLIB.



Com relação ao tempo de processamento, através dos experimentos, observamos que existe uma dependência direta do r-TS em relação à calibração antes do início do processamento. Os fatores relativos ao número máximo de iterações (MÁX.ITER. - que define o critério de parada), a função perturbação (NV - definida pelo número de vizinhos) e a Fila Tabu (F) afetam diretamente os resultados. Observando a Figura 6.3, dada a seguir, constata-se que o tempo de processamento não é proporcional ao número de pontos, devido à calibração usada no r-TS, é possível encontrar boas soluções em um tempo computacional abaixo da expectativa linear quando se compara vários problemas em sequência. No caso analisado, o problema clássico dantzig42 foi resolvido em um tempo computacional atípico, possivelmente em decorrência de a solução inicial estar bem próxima da solução ótima.

Figura 6.3 – Tempo de processamento.

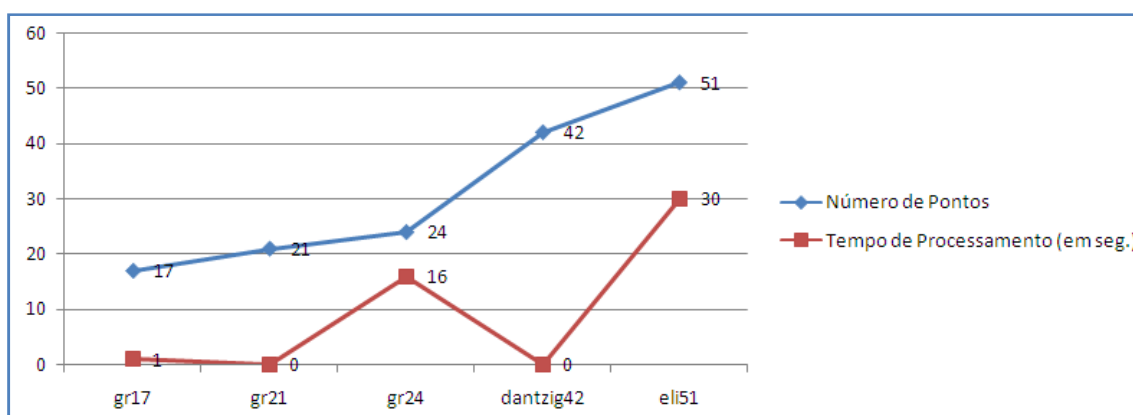


Tabela 6.2 – Calibrações utilizadas no r-TS.

Arquivo	n	MÁX.ITER.	F	NV	n/F	Tempo
gr17	17	300	4	17	4,25	1s
gr21	21	100	4	21	5,25	0s
gr24	24	520	13	24	1,85	16s
dantzig42	42	300	8	41	5,25	0s
eli51	51	305	35	51	1,46	30s
pr76	76	850	13	76	5,85	75s
kroC	100	20.000	12	99	8,33	184s
pr107	107	850	25	107	4,28	473s
kroA	150	20.000	18	149	8,33	437s
a280	280	300	2	280	140,00	55s

A Tabela 6.2 apresenta os dados das calibrações utilizadas nas resoluções dos problemas analisados. Considerando que o r-TS é um algoritmo simples, a calibragem é fixa, ou seja, não há qualquer alteração dos valores pré-definidos no início do processamento até o seu final; o número máximo de iterações, o tamanho da Fila Tabu e o número de vizinhos inerentes à função perturbação a cada iteração são valores constantes. Observa-se pela calibração, que a relação número de pontos divididos pelo tamanho da Fila Tabu (n/F) variou com bons resultados em torno de 1,46 a 5,25 de para os problemas até 51 pontos.

6.1.1 Comparativo com resultados obtidos com outra metaheurística

A Tabela 6.3 apresenta um resultado comparativo entre os resultados obtidos pelo r-TS e pelo AGHC desenvolvido por Silva, Soma e Viana (2005). Os dados utilizados para efeito comparativo são os mesmos dez problemas escolhidos na *OR-Library* (TSPLIB 2004).

Verifica-se que o AGHC de Silva *et. al.*, apresentou resultados melhores para instâncias mais complexas (valores de n acima de 50). Isso se deve, possivelmente, ao uso de uma técnica mais sofisticada de diversificação, denominada Heurística Permutacional (HP*), além do uso da conhecida técnica 3-OPT de Lin e Kernighan agindo como operador mutação.

Tabela 6.3 – Resultados comparativos entre r-TS x AGHC.

Arquivo	n	Solução ótima (TSPLIB)	Solução encontrada (r-TS)	Tempo (r-TS)	Solução encontrada (AGHC)	Tempo (AGHC)
gr17	17	2085	2085	1s	2090	2s
gr21	21	2707	2707	0s	2707	3s
gr24	24	1272	1289	16s	1272	3s
dantzig42	42	699	699	0s	699	5s
eil51	51	426	475,42	30s	428,98	10s
pr76	76	108159	143012,84	75s	110055	32s
kroC	100	20749	30352,89	184s	21411	132s
pr107	107	44303	52555,66	473s	44429,75	197s
kroA	150	26524	39350,03	437s	26933	496s
a280	280	2579	2794,76	55s	2657	2145s

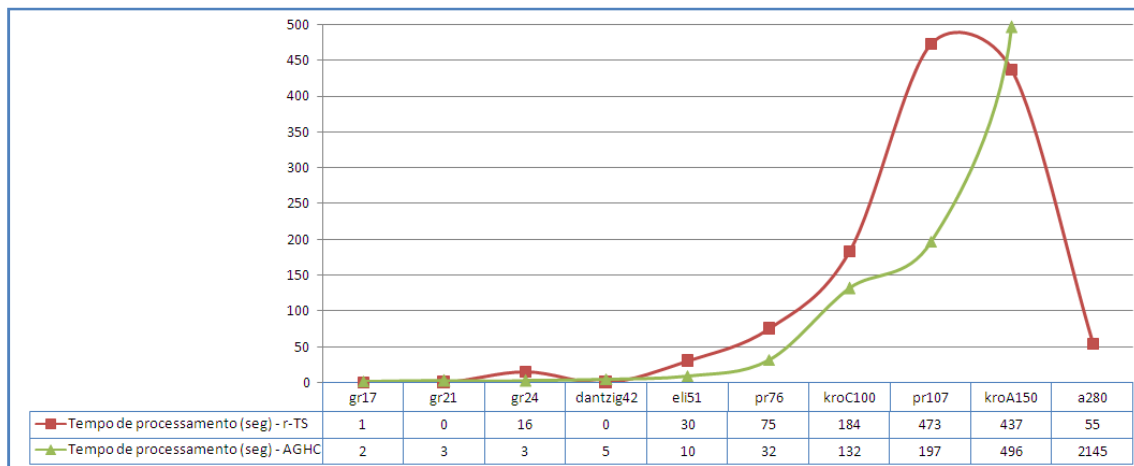
Os resultados mais eficientes do AGHC quando comparados com o r-TS não o invalidam, visto que sugerem diferentes alcances de diversificação para instâncias mais complexas. É também sugestivo que, para trabalhos futuros, seja utilizada uma técnica mais robusta de diversificação no r-TS, a fim de que obtenha resultados tão bons quanto o AGHC para valores de n acima de 100.

Outro fator relevante sobre o AGHC é a linearidade do tempo de processamento em função de “n”, conforme mostra a figura 6.4. Neste aspecto, observa-se que para

algumas instâncias o r-TS apresentou melhores resultados, porém, em uma minoria de problemas.

Vale ressaltar que, para problemas práticos tal qual o abordado no presente trabalho, o número de clientes a serem visitados em sequência não ultrapassa a 50, o que garante a eficácia do r-TS para este horizonte de aplicação.

Figura 6.4 – Tempo de processamento comparativo entre r-TS e AGHC.



6.2 Aplicação a um caso prático: manutenção de elevadores

Empresas de manutenção de elevadores são responsáveis por garantir aos seus usuários, segurança e confiabilidade no transporte. Em geral, a maioria das empresas subdivide os serviços de manutenção em campo em dois tipos: preventivas e corretivas. A caracterização do problema em questão já fora realizada no Capítulo 5. Dessa forma, apresentaremos concisamente o *case study* em análise na empresa hipotética “LIFTECH”, devido ao sigilo das informações.

6.2.1 O problema da empresa Elevadores “LIFTECH”

A “LIFTECH” é uma empresa multinacional que opera nos segmentos de siderurgia, tecnologia e materiais. No Brasil está presente há 65 anos, operando nas atividades de projeto, fabricação, instalação e manutenção de transportes verticais. Desde o início de sua operação no Brasil, já foram fabricados mais de cem mil equipamentos.

A empresa está subdividida em Unidades de Negócios, com medidas de desempenho e resultados individuais, a fim de que seja possível mensurar a eficiência regional de sua operação.

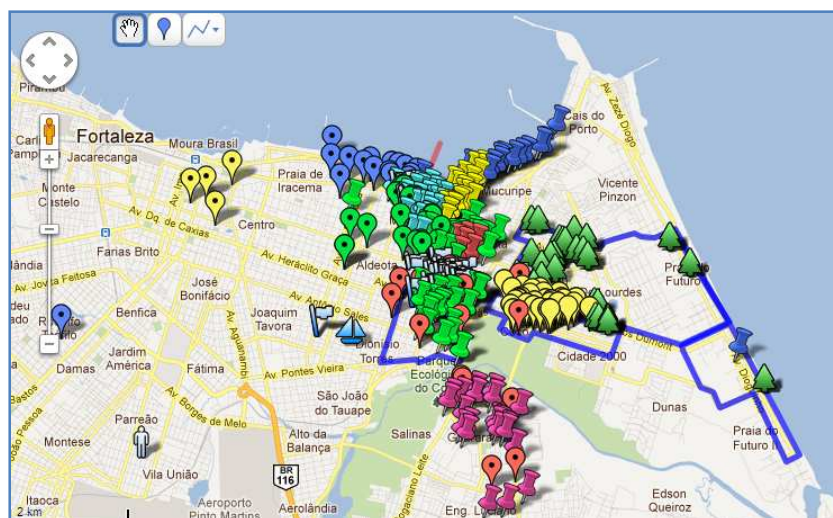
O ciclo de negócios de serviços de manutenção, assim como ocorre em outras empresas multinacionais do segmento, é o que retorna mais resultado da operação, tornando-o o principal negócio (*core business*) da empresa. As unidades de negócios possuem diferentes quantidades de equipamentos sob manutenção, que de forma dinâmica, são alterados mensalmente em virtude dos novos contratos de manutenção assinados e das perdas inerentes ao negócio.

A gestão operacional da empresa utiliza um sistema desenvolvido internamente na empresa. É de responsabilidade dos gestores de cada unidade de negócios a subdivisão da carteira de manutenção em zonas e setores, de forma a obter melhores resultados operacionais. A Figura 6.5, dada a seguir, apresenta uma ilustração da divisão da carteira de manutenção em zonas e setores. O processo é manual e realizado sem qualquer técnica científica, visto que o conhecimento da posição geográfica (endereço) é o único parâmetro levado em consideração. A ilustração é de uma divisão realizada em 2009 e não mais atualizada. O sistema não possui um SIG integrado, e o sequenciamento da manutenção é realizado pelo supervisor de manutenção, ou, às vezes, pelo próprio técnico do setor.

Devido à altíssima competitividade nesse mercado, a busca por processos mais eficientes é constante, visto que a competição por preço com empresas conservadoras de elevadores é muito agressiva. As margens de contribuição, que representam o resultado operacional (receitas brutas menos impostos e custos operacionais) de serviços de manutenção variam em média de 10 a 45%, e dependem diretamente, entre outros fatores, da produtividade dos técnicos.

O ganho de produtividade para serviços de manutenção em campo, tal como os necessários aos elevadores, é um fator primordial para se estabelecer no mercado, e estudos na área de logística são fundamentais para se garantir eliminação de perdas com deslocamentos.

Figura 6.5 – Ilustração da alocação de clientes (edifícios) usando o *Google Maps*.



É importante ressaltar que em rotas de manutenção de elevadores, não se faz necessário o retorno ao ponto inicial de partida, o que exige uma adaptação no r-TS para que desconsidere a distância de retorno como um dado relevante. Outra informação relevante é que o deslocamento em geral é à pé, o que nos permitiu considerar a distância do arco calculado entre dois pontos coordenados (latitude e longitude) como muito próximos de uma situação real.

6.2.2 “LIFTECH” em Fortaleza-CE

Em Fortaleza-CE, a LIFTECH está organizada em três zonas cada uma com determinado número de setores conforme Tabela 6.4 abaixo. Cada setor conta com um técnico específico de manutenção em rota de campo para execução de manutenções preventivas. Paralelamente à equipe de manutenção preventiva, outro grupo de técnicos é organizado para os atendimentos aos chamados de manutenção corretiva (defeitos e paralisações indesejadas) registrados na Central de Atendimento. O foco do presente trabalho é exclusivamente em relação aos setores de manutenção preventiva, dispostos conforme Tabela abaixo.

Tabela 6.4 – Organização da LIFTECH em ‘zonas’ e ‘setores’ de manutenção.

ZONA	SETOR	NÚMERO DE ELEVADORES	NÚMERO DE CLIENTES
1	1	48	30
	2	75	35
	3	77	36
	4	68	31
	5	72	37
	6	71	36
	7	74	37
	8	67	21
	9	16	7
2	1	57	17
	2	68	9
	3	47	14
	4	62	23
	5	55	20
	6	7	1
	7	43	10
3	1	60	31
	2	91	36
	3	76	39
	4	99	35
	5	68	31
	6	64	31
	7	75	36
	8	76	26

O agrupamento dos equipamentos em setores é realizado considerando alguns critérios pré-estabelecidos e não necessariamente exclusivamente geográficos, tais como:

- i. Margem do contrato;
- ii. Localização;
- iii. Tipo de equipamento;
- iv. Capacidade máxima de cumprimento de uma rota de manutenção.

Desta forma, o ganho logístico nesse problema de pesquisa está relacionado à melhor forma de sequenciamento de cada uma das 24 (vinte e quatro) rotas de manutenção existentes, tornando o problema facilmente solucionável e passível de revalidação mensal à medida que ocorram alterações nas rotas em virtude do acréscimo de novos clientes e da possibilidade de perda de alguns.

6.2.3 Procedimentos realizados

Para esta aplicação, foram utilizados os dados coletados no levantamento geográfico da base de clientes, onde constam as coordenadas de latitude e longitude obtidas através da coleta em campo via PDA (*smartphones*) com funcionalidades de registro de coordenadas, conforme ilustra a Figura 6.6.

Algumas falhas foram encontradas nos registros, o que tornou necessária a conferência amostral de alguns dados utilizando-se o aplicativo móvel GPS TOOLS utilizado no aparelho *BlackBerry v.9300*. O software possui suporte disponível em <http://www.tworoads.net/~srp/software/index.html>.

Figura 6.6 – Tela do aplicativo GPS Tool.

GPS Tool *		Jul 6, 2010 9:34 AM
Satellites:		6
Average Signal:		17dB
GPS Mode:	Autonomous, Complete	
GPS Method:	Satellite, Device Based	
Last Fix:		9:34:53 AM
Time to Complete Fix:		335s
Latitude:		47°16:26.58
Longitude:		-122°28:58.44
Altitude:		190ft
Heading:		0°
Speed:		0
Accuracy (h/v):		88ft/104ft

Fonte: <http://www.tworoads.net/~srp/software/gpstool/index.html>.

A base de dados da empresa permitiu a análise de todos os setores de manutenção, de forma que o percentual de redução do deslocamento da rota fosse obtido pela equação:

$$Redução = \left[1 - \frac{Rota\ r - TS}{Rota\ inicial} \right] . 100$$

6.2.3.1 O r-TS adaptado ao problema prático

Nesta subseção é apresentado o passo-a-passo das 5 etapas de como utilizar o r-TS no problema específico da *LIFTECH*:

Passo 1: Com base nos dados fornecidos pela empresa, conforme exemplificado na Tabela 6.5 abaixo para o setor 3 da zona 2, gerar arquivo .csv através do Microsoft Excel.

Tabela 6.5 – Sequência estabelecida no sistema da empresa para o setor 3 / zona 2.

SEQ.	CLIENTE	Latitude	Longitude
1	JURIDICAL CENTER CD ED	-3,7697	-38,4824
2	EDUCADORA SETE DE SETEMBRO	-3,76891	-38,4945
3	EMPRESARIAL WASHINGTON SOARES	-3,76067	-38,4829
4	HOSPITAL SAO MATEUS	-3,7416	-38,4816
5	COMPLEXO SAO MATEUS - TORRE 1	-3,74177	-38,48
6	FACULDADES CHRISTUS	-3,7473	-38,4819
7	CHRISTUS	-3,74933	-38,4969
8	SHOPPING IGUATEMI - Deck Park	-3,75712	-38,4906
9	SHOPPING IGUATEMI - V Etapa	-3,75712	-38,4906
10	SHOPPING IGUATEMI - Escadas Rolantes	-3,75712	-38,4906
11	C & A MODAS LTDA	-3,75712	-38,4906
12	IGUATEMI EMPRESARIAL CD ED	-3,7372	-38,4922
13	IGUATEMI EMPRESARIAL CD ED (Tasso Jereissati)	-3,7372	-38,4922
14	PATIO DOM LUIS - Escadas Rolantes	-3,73558	-38,4905

Passo 2: Gerar matriz D_{ij} de dimensão 14x14 com base nas coordenadas geográficas contidas nas colunas 3 e 4 da Tabela 6.5. A Figura 6.7 ilustra a matriz do problema específico (“*LIFTECH*”, setor 3 da zona 2).

Figura 6.7 – Matriz D_{ij} para o setor 3 / zona 2

Busca Tabu

UNIVERSIDADE FEDERAL DO CEARÁ - UFC
PROGRAMA DE MESTRADO EM LOGÍSTICA E PESQUISA OPERACIONAL - GESLOG/UFC

Aluno: Rodrigo Frank de Souza Gomes
Orientador: Prof. Dr. José Lassance de Castro Silva

Entre com o valor de n (tamanho da matriz D_{ij}): 5
Tamanho da coluna da matriz: 32

Importar dados Euc 2D (.csv) Importar matriz D_{ij} (.csv)
Importar dados GeoRef (.csv) Gerar matriz

Coordenadas:	Latitude	Longitude
JURIDICAL C	-3,7697006	-38,4823983
EDUCADOR4	-3,7689136	-38,4945135
EMPRESARI	-3,7606654	-38,4829257
HOSPITAL S	-3,7415957	-38,4816193
COMPLEXO	-3,7417656	-38,4800005
FACULDADE	-3,7472962	-38,4819275
CHRISTUS	-3,7493347	-38,496878
SHOPPING I	-3,7571201	-38,4905866
SHOPPING II	-3,7571201	-38,4905866
SHOPPING III	-3,7571201	-38,4905866
C & A MODAS	-3,7571201	-38,4905866
IGUATEMI E	-3,7372027	-38,4922478
IGUATEMI F	-3,7372027	-38,4922478
PATIO DOM	-3,7355771	-38,4904613

Matriz gerada:	JURID	EDUC	EMPF	HOSF	COMF	FACU	CHRI	SHOF	SHOF	SHOF	C & A	IGUA	IGUA	PATIO
JURID	0	1,109	1,111	3,335	3,335	2,223	3,141	1,570	1,570	1,570	1,570	3,515	3,515	3,515
EDUC	1,109	0,000	1,570	3,515	3,515	2,485	2,485	1,111	1,111	1,111	1,111	3,335	3,335	3,335
EMPF	1,111	1,570	0,000	2,223	2,223	1,111	2,482	1,109	1,109	1,109	1,109	2,485	2,485	2,485
HOSF	3,335	3,515	2,223	0,000	0,000	1,111	2,482	2,485	2,485	2,485	2,485	1,109	1,109	1,109
COMF	3,335	3,515	2,223	0,000	0,000	1,111	2,482	2,485	2,485	2,485	2,485	1,109	1,109	1,109
FACU	2,223	2,485	1,111	1,111	1,111	0,000	2,219	1,570	1,570	1,570	1,570	1,570	1,570	1,570
CHRI	3,141	2,485	2,482	2,482	2,482	2,219	0,000	1,570	1,570	1,570	1,570	1,570	1,570	1,570
SHOF	1,570	1,111	1,109	2,485	2,485	1,570	1,570	0,000	0,000	0,000	0,000	2,223	2,223	2,223
SHOF	1,570	1,111	1,109	2,485	2,485	1,570	1,570	0,000	0,000	0,000	0,000	2,223	2,223	2,223
SHOF	1,570	1,111	1,109	2,485	2,485	1,570	1,570	0,000	0,000	0,000	0,000	2,223	2,223	2,223
C & A	1,570	1,111	1,109	2,485	2,485	1,570	1,570	0,000	0,000	0,000	0,000	2,223	2,223	2,223
IGUA	3,515	3,335	2,485	1,109	1,109	1,570	1,570	2,223	2,223	2,223	2,223	0,000	0,000	0,000
IGUA	3,515	3,335	2,485	1,109	1,109	1,570	1,570	2,223	2,223	2,223	2,223	0,000	0,000	0,000
PATIO	3,515	3,335	2,485	1,109	1,109	1,570	1,570	2,223	2,223	2,223	2,223	0,000	0,000	0,000

Avançar >> Sair

Copyright © 2011 Rodrigo Frank de Souza Gomes (rodrigofsg@gmail.com)

Passo 3: Definição da Solução inicial. Está pré-definido no r-TS que a solução inicial é sempre a sequência dos dados do arquivo matricial .csv importado para o aplicativo. No exemplo citado, $S_0 = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14]$ com distância total igual a 12,02 km.

Passo 4: Calibração conforme Figura 6.8 dada abaixo, onde devem ser inseridos os parâmetros relativos ao número máximo de iterações (critério de parada), tamanho da Fila *Tabu*, Número de Vizinhos (função perturbação) e a marcação ou não do *checkbox* de retorno ao ponto inicial, visualização de iterações e uso ou não da intensificação.

Figura 6.8 – Calibração do r-TS para problema prático (setor 3 / zona 2).

Qtd. iterações (MÁX ITER.): 500
Tamanho Fila Tabu (F): 3
Vizinhaça (NV): 13

Critério de aspiração: Sobrepor F se $f(k) < f(k-1)$

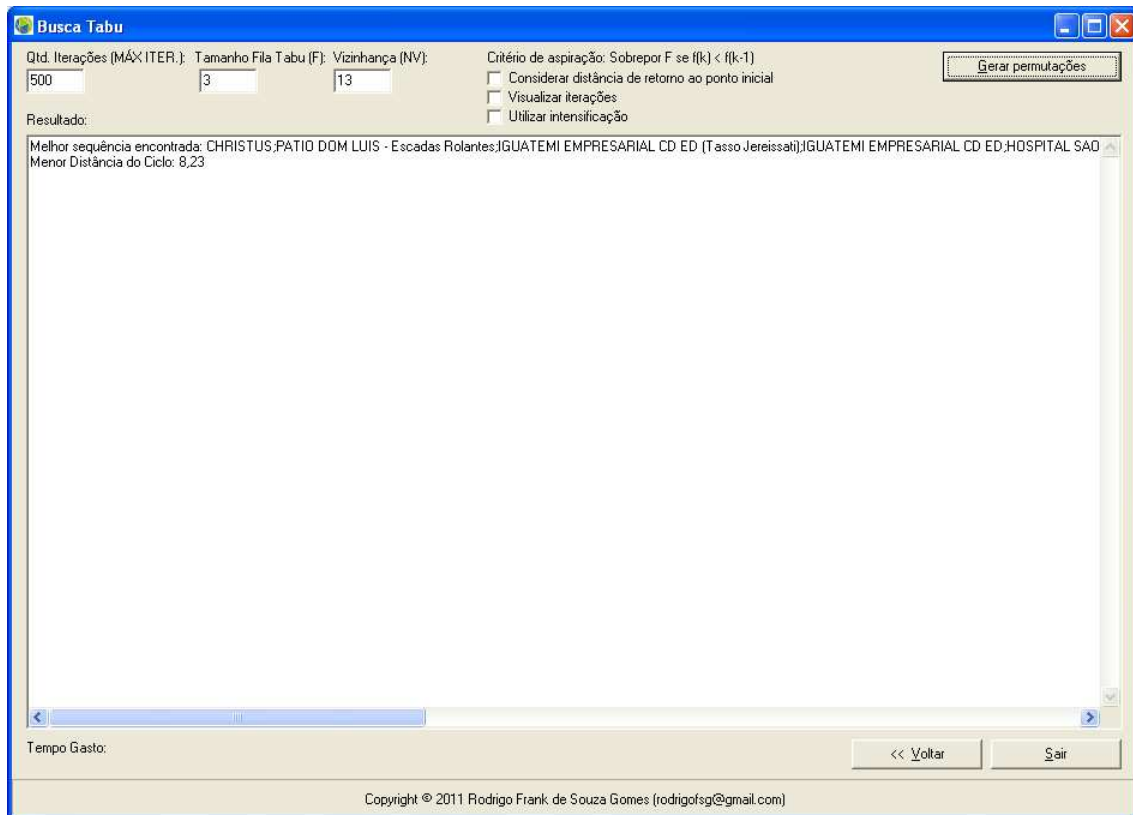
☐ Considerar distância de retorno ao ponto inicial
☐ Visualizar iterações
☐ Utilizar intensificação

Gerar permutações

Resultado:

Passo 5: Processamento e Resultado Final: ilustrado na Figura 6.9, dada a seguir.

Figura 6.9 – Resultado do r-TS para problema prático (setor 3 / zona 2).



Comparando-se o resultado da sequência inicial com a ‘melhor sequência encontrada pelo r-TS’, percebe-se uma melhoria de 31.53%, somente neste setor.

O mesmo procedimento foi executado para os demais 23 setores de manutenção da empresa, e o resultado está apresentado na Tabela 6.6, dada a seguir.

6.2.4 Resultados obtidos

A Tabela 6.6, dada ao final desta seção, apresenta os resultados obtidos com a aplicação do r-TS às rotas originais de cada um dos setores de manutenção anteriormente listados na Tabela 6.4. A rota inicial do r-TS é a rota atual de cada setor utilizada atualmente na empresa.

Tabela 6.6 – Resultados gerais obtidos na “LIFTECH”

ZONA	SETOR	Número de Elevadores	Número de Clientes	Rota Inicial (km)	Rota r-TS (km)	Redução da rota (%)
1	1	48	30	95,26	36,74	61,43
1	2	75	35	61,70	30,61	50,39
1	3	77	36	45,39	14,14	68,85
1	4	68	31	52,77	21,89	41,48
1	5	72	37	48,49	15,49	68,06
1	6	71	36	43,17	13,85	67,92
1	7	74	37	38,33	12,74	66,76
1	8	67	21	18,24	12,75	30,10
1	9	16	7	-----	-----	-----
2	1	57	17	8,04	8,04	0,00
2	2	68	9	14,82	6,84	53,85
2	3	47	14	12,02	8,23	31,53
2	4	62	23	8,69	5,36	38,32
2	5	55	20	25,17	11,55	54,11
2	6	7	1	-----	-----	-----
2	7	43	10	14,35	8,83	38,47
3	1	60	31	11,43	7,12	37,31
3	2	91	36	68,47	34,64	49,41
3	3	76	39	44,40	12,92	70,90
3	4	99	35	79,69	18,33	77,00
3	5	68	31	-----	-----	-----
3	6	64	31	34,73	10,59	69,51
3	7	75	36	24,89	6,77	72,80
3	8	76	26	41,83	16,32	60,98
TOTAL	-----	1.516	629	791,88	313,75	52,81 (média)

Os dados da Tabela 6.6 demonstram o ganho obtido com a aplicação do r-TS ao banco de dados de seqüenciamento da empresa. Observou-se que a significativa melhora obtida com o r-TS também se deve à uma parcela de desatualização de dados, visto que por se tratar de um procedimento manual, todos os meses o sistema deveria ser atualizado pelos supervisores de cada zona. A não atualização dos dados distorce a sequência aplicada, visto que novos contratos são alocados ao final da fila. Os setores 9 (zona 1), 6 (zona 2) e 5 (zona 3) não foram analisados em virtude de tratarem de contratos no interior, contrato exclusivo e dados inconsistentes, respectivamente. A consistência da base de informações disponibilizada pela empresa (coordenadas geográficas de seus clientes e a sequência cadastrada em seu sistema para cada setor de manutenção) é um fator primordial para a qualidade da análise.

A economia total com a aplicação do método é de 52,81%, o que mostra que o r-TS é uma boa técnica a ser usada na resolução de problemas práticos. Com base nos resultados obtidos, percebe-se ainda que somente em um dos setores não houve ganho,

enquanto que outros apresentaram resultados extremamente relevantes em termos de melhoria logística. As economias, quando analisadas separadamente por zonas, geraram, respectivamente, 56,87%, 35,56% e 59,55% para as zonas 1, 2 e 3. As Figuras 6.10 e 6.11, dadas a seguir, mostram os ganhos obtidos por cada setor analisado em termos de distância real e percentualmente.

O ganho de eficiência na produtividade no uso da técnica pode possibilitar à empresa, uma reestruturação organizacional, de forma a obter melhores resultados (inclusive financeiros). Outros fatores de ordem qualitativa também são influenciados pela melhoria logística do seqüenciamento de rotas de manutenção, dos quais destacamos o tempo de atendimento e uma maior qualidade percebida por parte dos clientes em virtude da precisão no agendamento de serviços.

Ressalta-se oportunamente que o número médio de clientes por setor de manutenção foi de 26, o que corrobora com a expectativa de bons resultados gerados pelo r-TS para uma quantidade de até 51 pontos (clientes), conforme abordado nos experimentos com a TSPLIB.

Figura 6.10 – Comparação entre rota inicial e rota gerada pelo r-TS

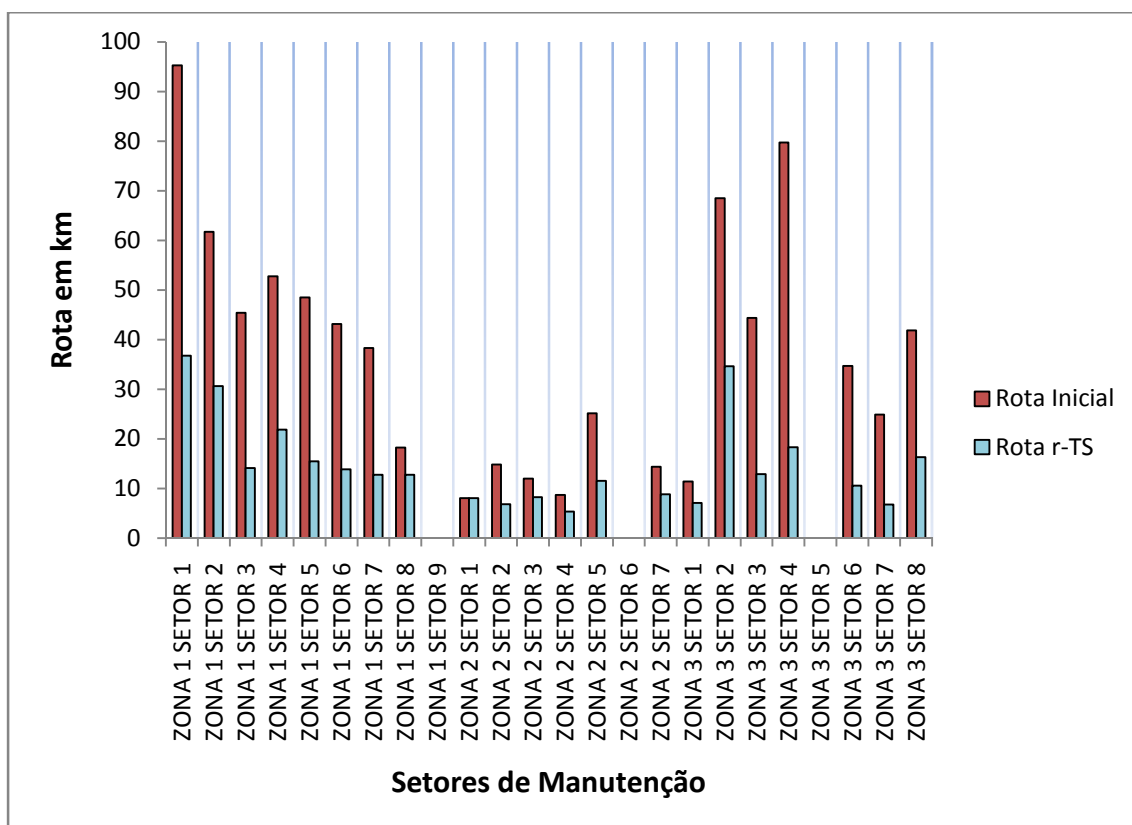
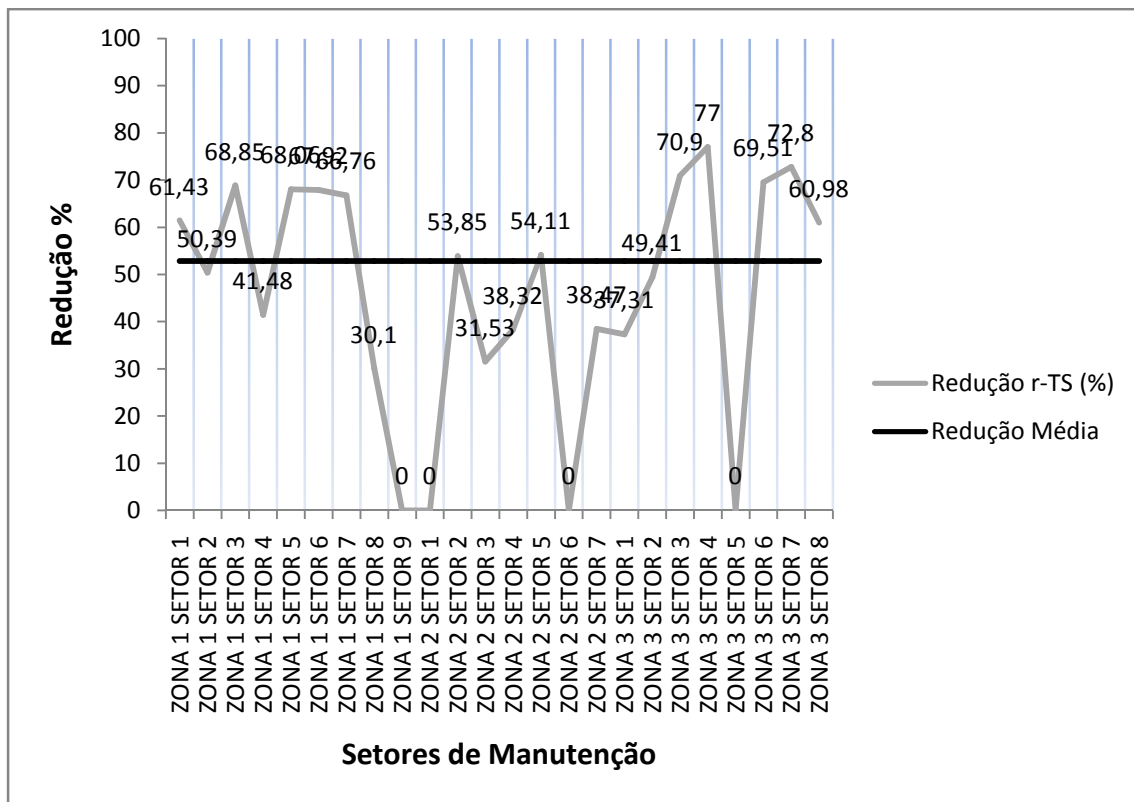


Figura 6.11 – Redução percentual da distância total das rotas pelo r-TS



Os resultados obtidos demonstram o potencial de contribuição que o uso de técnicas científicas de sequenciamento ou de gestão logística podem trazer a operação de uma empresa, determinando seu grau de competitividade no mercado.

CAPÍTULO 7 – CONSIDERAÇÕES FINAIS

7.1 Conclusões

O objetivo geral deste trabalho foi desenvolver uma técnica com base na metaheurística *Tabu Search* (TS) para obtenção de boas soluções para problemas logísticos de manutenção preventiva em campo. A pesquisa foi fundamentada no Problema clássico do Caixeiro Viajante que possui larga aplicação científica em diversos problemas práticos, notadamente na área de logística. Para desenvolvimento e evolução do algoritmo, intitulado de r-TS, foram considerados procedimentos simples e básicos dos conceitos da metaheurística TS. A técnica *swap* foi utilizada na diversificação e estruturas de memória foram incorporadas para a intensificação das soluções, trazendo bons resultados para o r-TS.

Nos testes de validação, foram escolhidos dez problemas clássicos simétricos da *OR-Library* de Beasley (1990) - TSPLIB com 17, 21, 24, 42, 51, 76, 100, 107k8, 150 e 280 pontos. Os experimentos demonstraram bons resultados para instâncias com até 51 pontos, onde várias soluções ótimas foram alcançadas. O uso de técnica de diversificação simples, como o método *swap*, não favorece que o espaço de soluções viáveis para o problema seja bem percorrido, prejudicando a eficiência do método na resolução de problemas em instâncias maiores. Outras técnicas como HP, HP* e r-OPT, entre outras abordadas na literatura, mostram-se mais eficientes quando o objetivo é o aprimoramento da diversificação.

Na aplicação prática, foi escolhida uma empresa de manutenção de elevadores com o objetivo de medir a contribuição do r-TS para o ganho de eficiência logística em termos de redução de deslocamento das rotas de manutenção. O número de clientes por setor de manutenção variou entre 1 e 39 pontos de localização. Os dados relativos às coordenadas geográficas, em geral, foram consistentes. Os resultados atingidos nesta aplicação prática em empresa de manutenção em campo demonstraram a eficiência da técnica e sugerem aprofundamento da pesquisa para desenvolvimento de aplicativos mais robustos para instâncias mais complexas que àquelas abordadas no presente trabalho. Os ganhos obtidos na empresa, objeto de estudo, também demonstraram a ausência de métodos científicos aplicados à gestão logística operacional, o que nos mostra grandes oportunidades de melhorias nos processos logísticos empresariais.

7.2 Sugestões para trabalhos futuros

O presente trabalho teve como foco a otimização de rotas de manutenção preventiva. Considerando a grande relevância deste tema apontada na Introdução deste texto, seguem abaixo algumas sugestões para ampliação do estudo ora realizado acerca da otimização de rotas de manutenção periódica.

- i. Calibrar o r-TS e incluir a rotina de solução gráfica visual para melhorar a análise da solução proposta pelo algoritmo;
- ii. Adaptar o r-TS para o GKRT (*Generalized k-Travelling Repairman Problem*) considerando os tempos de reparação das máquinas diferentes de zero, a fim de tornar o modelo mais aplicável a situações empresariais;
- iii. Definir um período de tempo de espera limite L e adaptar o modelo para o *Bounded Latency Problem*, ampliando a análise realizada para o dimensionamento da equipe técnica de reparadores necessária;
- iv. Utilizar de técnicas de agrupamento ou *clusterização* para melhorar a definição das sub-rotas já existentes, tendo em vista que o r-TS não considera se o agrupamento foi adequado ou não;
- v. Avaliar o problema sem o agrupamento de zonas e setores.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANDRADE, E. L. **Introdução à Pesquisa Operacional: métodos e modelos para a análise de decisão**. Rio de Janeiro: LTC, 2000, 277p.
- ARCHETTI, C.; BERTAZZI, L.; SPERANZA, M.G. **Reoptimizing the Travelling Salesman Problem**. In: *Networks*. Vol. 42(3), pp. 154-169. 2003.
- BALDACCI, R.; HADJICONSTANTINOU, E.; MINGOZZI, A. **An exact algorithm for the Travelling Salesman Problem with delivery and collections**. In: *Networks*. Vol. 42(1), 2003, pp. 26-41.
- BALOU, Ronal H. **Logística empresarial**. São Paulo: Atlas, 2010, 392p.
- BEASLEY, J. E. **OR-Library. Distributing test problems by electronic mail**. In: *Journal of Operations Research Society*. Vol.41, 1990, pp. 1069-1072.
- BERKHOLZ, D. **A capacity forecast model for volatile data in maintenance logistics**. In: *AIP Conference Proceedings*. Vol.1127, 2009, pp. 206-216.
- BERTSIMAS, D. J.; VAN RYZIN, G. A. **A stochastic and dynamic vehicle routing problem in the Euclidean plane**. In: *Operations Research*. Vol. 39, 1991, pp. 601-615.
- BIGGS, M.B.; ZUO, M. **Optmizing preventive maintenance models**. In: *Computational Optimization and Applications*. Vol.35, 2006, pp. 261-279.
- BLAKELEY, Fred; KNOLMAJER, Joseph. **Optmizing periodic maintenance operations for Schindler Elevator Corporation**. In: *Interfaces*. Vol. 33, n.1, 2003, pp.67-79.
- BONIFACI, V.; STOUGIE, L.. **Online k-Servers Routing Problems**. In: *Theory of Computing Systems*. Vol. 45, No. 3, 2009.
- BOWESOX, D.J., CLOSS, D.J. **Logística empresarial: o processo de integração da cadeia de suprimento**. São Paulo: Atlas, 2001. 594p.
- CALDAS, Marco A.F; SANTOS, Davidson A. **Uma heurística para roteamento de frota de veículos para a distribuição de bebidas em uma região urbana**. In: *XLSPBO - Simpósio Brasileiro de Pesquisa Operacional*, 2008.
- CHRISTOPHER, Martin. **Logística e gerenciamento da cadeia de suprimento: estratégias para a redução de custos e melhoria dos serviços**. São Paulo: Pioneira Thompson Learning, 2002. 240p.
- CLARKE, G.; WRIGHT, J.W. **Scheduling of vehicles from a central depot to a number of delivery points**. In: *Operations Research*. Vol. 12, n.4, July-August 1964, pp. 568-581.
- COOK, S. A. **The complexity of theorem providing procedures**. In: *Third Annual ACM Symposium on the Theory of Computing*, 1971, pp. 151-158.

CORBERÁN, A. *et al.* **Heuristic solutions to the problem of routing school buses with multiple objectives.** Departament d'Estadística i Investigació Operativa, Universitat de València, Burjassot 46100, Spain. August 7, 2000.

CORRÊA, Henrique L. **Gestão de serviços: lucratividade por meio de operações e de satisfação dos clientes.** São Paulo: Atlas, 2002, 479p.

COSTA, D.M.B. *et al.* **Técnicas de Pesquisa Operacional aplicadas na otimização dos serviços postais.** In: *Gestão & Produção*. Vol. 8, n.1, 2001, pp.37-55.

CROES, G.A. **A method for solving travelling salesman problems.** In: *Operations Research*. Vol. 6, n.6, November-December, 1958, pp. 791-812.

CUNHA, C.B., *et al.* **Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante.** In: XVI Congresso da Anpet. Anais, 2002.

DALI WAL, D.S. **The use of AI in maintaining and operating complex engineering systems.** In: *Expert systems and optimisation in process control*. Ed.28-33, Gower Technical Press, 1986.

DAMASO, V.C.; GARCIA, P.A.A. **Testing and preventive maintenance scheduling optimisation for aging systems modeled by generalized renewal process.** In: *Pesquisa Operacional*. Vol.29. n.3, Setembro-Dezembro 2009, pp. 563-576.

DORNIER, P.P. *et al.* **Logística e operações globais: texto e casos.** São Paulo: Atlas, 2009. 721p.

GIL, A.C. **Como elaborar projetos de pesquisa.** 4ª ed. São Paulo: Atlas, 2008. 175pp.

GLOVER, F. **Tabu Search, Part I.** In: *ORSA Journal of Computing*. Vol. 1. n.3, 1989, pp. 190-206.

GLOVER, F. **Tabu Search, Part II.** In: *ORSA Journal of Computing*. Vol. 2. n.1, 1990, pp. 4-32.

GLOVER, F. **Tabu Search: a tutorial.** In: *Interfaces*. Vol. 20. n.4. July-August, 1990, pp. 74-94.

GLOVER, F. **Tabu Search – Wellsprings and Challenges.** In: *European Journal of Operational Research*. 106, 1998, pp. 221-225.

GLOVER, F.; LAGUNA, M. **Tabu Search.** Kluwer Academic Publisher, Boston, 1997.

GLOVER, F.; LAGUNA, M. KELLY, J. **Genetic Algorithms and Tabu Search: hybrids for optimization.** In: *Computers Ops. Res.* Vol. 22, n.1, 1995, pp. 111-134.

GOLDBARG, M.C. e LUNA, H.P.L. **Otimização combinatória e programação linear: modelos e algoritmos.** Rio de Janeiro. Editora Campus, 2000.

GOMES, F.R.A. **Algoritmo genético aplicado ao problema de seqüenciamento permutacional *flowshop* sem e com restrição de espera.** Fortaleza. Universidade Federal do Ceará. Dissertação (Mestrado em Logística e Pesquisa Operacional), 141pp. 2008.

HILLIER, F. S.; LIEBERMAN, G.J. **Introdução à pesquisa operacional**. 8ª ed. Porto Alegre: AMGH, 2010, 852 p.

HUANG, J.Y.; YAO, M.J. **On the coordination of maintenance scheduling for transportation fleets of many branches of a logistic service provider**. In: *Computers and Mathematics with applications*. Vol.56, 2008, pp. 1303-1313.

HUISKONEN, J. **Maintenance spare parts logistics: special characteristics and strategic choices**. In: *International journal of production economics*. Vol.71, 2001, pp. 125-133.

HOLLAND, J.H. **Adaptation in natural artificial systems**. University of Michigan Press, Michigan, 1975, 211 pp.

JOTHI, Raja; RAGHAVACHARI, Balaji. **Approximating the k-travelling repairman problem with repair times**. In: *Journal of Discrete Algorithms*. Vol.5 (2007) 293–303.

KOBACCY, K. A. H. **Artificial intelligence in maintenance**. In: *Complex System Maintenance Handbook – Part C*, 2008.

LAPORTE, G. *et.al.* **The travelling salesman problem with pickups, deliveries and handling costs**. In: *Informatics*. Vol.4. n.3, 2010, pp.383-399.

LAPORTE, G. *et.al.* **A tabu search heuristic using genetic diversification for the clustered travelling salesman problem**. In: *Journal of Heuristics*. Vol.2, 1996, pp.187-200.

LARRAÑAGA, F. A. **A gestão logística global**. 2ª. Ed. São Paulo. Aduaneiras, 2008.

LIN, S.; KERNIGHAN, B.W. **An effective heuristic algorithm for the Travelling Salesman Problem**. In: *Operations Research*. Vol. 21. n.2, March-April 1973, pp. 498-516.

MOREIRA, M.C.O. **Balanceamento de linhas de produção com trabalhadores deficientes**. São Paulo. Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo, Dissertação (Mestrado em Ciências Matemáticas e de Computação), 90pp. 2011.

PERES, Cassiano Ricardo Gomes. **Análise do método de custeio baseado atividades aplicado à logística de distribuição**. São Paulo. Escola Politécnica da Universidade de São Paulo, Dissertação (Mestrado em Engenharia), 202pp. 2006.

PINTENLON, L.; PARODI-HERZ, A. **Maintenance: an evolutionary perspective**. In: *Complex System Maintenance Handbook – Part C*, 2008.

PIRES, T. **Configuração de uma rede de distribuição capacitada com restrições de cobertura**. São Paulo. Escola Politécnica da Universidade de São Paulo. Dissertação (Mestrado em Engenharia de Sistemas Logísticos), 111pp. 2006.

RABAK, C.S. **Otimização do processo de inserção automática de componentes eletrônicos empregando a técnica de times assíncronos**. São Paulo. Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Transportes. Dissertação (Mestrado em Engenharia), 114pp. 1999.

RADUAN, Auro C. **Roteirização parcialmente dinâmica aplicada a serviços de campo**. São Paulo. Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Transportes. Dissertação (Mestrado em Engenharia), 121pp. 2009.

REEVES, C. R. **Modern Heuristic Techniques for Combinatorial Problems**, McGraw-Hill, London, 1995.

RUIZ-VANOYE, J. A.; DÍAZ-PARRA, O. **Similarities between metaheuristics algorithms and the science of life**. In: CEJOR. DOI.10.1007/s10100-010-0135-x. 2010.

SCHEPKE, Claudio, SCHOPF, Eliseu C., SILVA, Marcus L., SILVA, Pablo F. **Um Estudo da Aplicação de Heurísticas Construtivas e de Melhoramento para um Problema de PRV**. In: III Simpósio de Informática da Região Centro do RS, 2004, Santa Maria. Anais - SIRC/RS 2004 - III Simpósio de Informática da Região Centro do RS, 2004.

SHEO, D.D.; KUO, J.Y. **A model for preventive maintenance operations and forecasting**. In: *Journal of Intelligence Manufacturing*. Vol.17, 2006. pp.441-451.

SILVA, J.L.C., SOMA, N.Y. **Uma heurística para problemas de otimização combinatória permutacional**. In: *XXIII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 2001, Campos do Jordão-SP, pp.1298-1306.

SILVA, J.L.C., SOMA, N.Y, VIANA, G.V.R. **Um algoritmo genético híbrido construtivo para problemas de otimização combinatória permutacional**. In: *XII CLAIO. Congreso Latinoamericano de Investigación de Operaciones*, 2004, Havana, anales del CLAIO, 2004.

SILVA, J.L.C. et. al. **Uma heurística aplicada ao problema do caixeiro viajante**. In: *XXVII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 2005, Gramado-RS, pp.1298-1306, 2005.

SILVA, J.L.C; XAVIER, S.L.P.; SILVA, B.C.H. **Problema de roteamento de veículos aplicado à coleta de gêneros alimentícios: um estudo de caso**. In: *XL Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 2008, João Pessoa-PB. Simpósio Brasileiro de Pesquisa Operacional, 2008.

SILVA, J.L.C. **Notas de aula**. 2010.

SOLOMON, M. M. **Algorithms for the vehicle routing and scheduling problems with time windows constraints**. In: *Operations Research*. Vol.35, n.2, 1987, pp. 254-265.

SÖRENSEN, K. **A framework for robust and flexible optimisation using metaheuristics with applications in supply chain design**. Universiteit Antwerpen. Faculteit Toegepaste Economische Wetenschappen. Tese de Doutorado. 2003.

SOUSA, Aldir S. **Tomada de decisão fuzzy e Busca Tabu aplicada ao planejamento da expansão de sistemas de transmissão**. São Paulo. Escola de Engenharia de São Carlos. Dissertação (Mestrado em Engenharia Elétrica), 128 pp., 2009.

TSPLIB–In: <http://comopt.ifl.uni-heidelberg.de/software/TSPLIB95/>.

VIANA, Gerardo V. R. **Meta-heurísticas e programação paralela em otimização combinatória**. Fortaleza: EUFC, 1998, 250 p.