



**UNIVERSIDADE FEDERAL DO OESTE DO PARÁ
INSTITUTO DE ENGENHARIA E GEOCIÊNCIAS
PROGRAMA DE CIÊNCIA E TECNOLOGIA**

**RESOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE ATRAVÉS DE
TÉCNICAS HEURÍSTICAS**

Eliaquim Monteiro Ramos

SANTARÉM - PA

2015

Eliaquim Monteiro Ramos

**RESOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE ATRAVÉS DE
TÉCNICAS HEURÍSTICAS**

Projeto de Trabalho de Conclusão de Curso – TCC
apresentado ao Curso de Bacharelado
Interdisciplinar em Ciência e Tecnologia da
Universidade Federal do Oeste do Pará-UFOPA
como requisito parcial para obtenção do grau de
Bacharel em Ciência e Tecnologia.

Orientador: Prof. Dr. Anderson Alvarenga de Moura Meneses

Co-Orientador: Prof. MSc. Marcel Antonionni de Andrade Romano

SANTARÉM - PA

2015

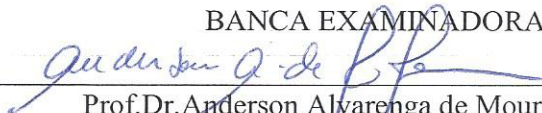
TERMO DE APROVAÇÃO

Este Trabalho de Conclusão de Curso foi analisado pelos membros da Banca Examinadora, abaixo assinados:
Eliaquim Monteiro Ramos

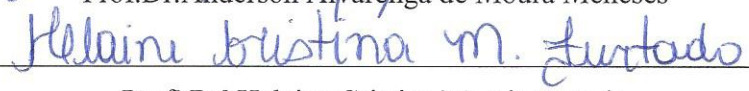
APROVADO EM: 30 / 12 / 2015

RESOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE ATRAVÉS DE TÉCNICAS HEURÍSTICAS

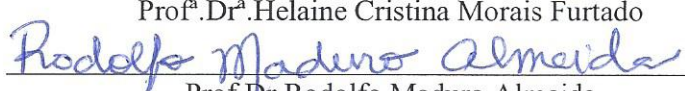
BANCA EXAMINADORA



Prof. Dr. Anderson Alvarenga de Moura Meneses



Prof.ª Dr.ª Helaine Cristina Morais Furtado



Prof. Dr. Rodolfo Maduro Almeida

SANTARÉM – PARÁ

2015

**Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema Integrado de Bibliotecas - SIBI/UFOPA**

R175r Ramos, Eliaquim Monteiro
Resolução do problema do caixeiro viajante através de técnicas heurísticas / Eliaquim Monteiro Ramos. – Santarém, 2015.
52 f. : il.
Inclui referências bibliográficas.
Orientador: Anderson Alvarenga de Moura Meneses; co-orientador: Marcel Antonionni de Andrade Romano.
Trabalho de Conclusão de Curso (Graduação) – Universidade Federal do Oeste do Pará. Curso de Bacharelado Interdisciplinar em Ciência e Tecnologia. Santarém, 2015.

1. Otimização combinatória. 2. Programação heurística. I. Meneses, Anderson Alvarenga de Moura, *orient.* II. Romano, Marcel Antonionni de Andrade, *co-orient.* III. Título.

CDD: 23.ed. 519.64

Bibliotecário-documentalista: Rogério Aoyama CRB 2/1506

AGRADECIMENTOS

Gostaria de agradecer primeiramente ao senhor criador de todas as coisas pela concretização desse sonho.

A minha família pelo apoio e compressão durante os meses que tive realizando esse trabalho. Agradeço ao meu orientador, Prof. Dr. Anderson Meneses para realização deste trabalho e depositar confiança em mim e pela paciência em retirar minhas dúvidas.

Ao meu coorientador, Prof. MSc. Marcel Antonionni pelo grande apoio na implementação das técnicas computacionais.

Aos meus amigos que contribuíram de forma direta ou indireta para o desenvolvimento do trabalho.

A todos, Muito obrigado!

RESUMO

O Problema do Caixeiro Viajante (PCV) é um problema clássico de otimização combinatória que tem sido trabalhado por diversos pesquisadores devido sua larga aplicabilidade em problemas reais, e sua resolução podem trazer inúmeros benefícios para a sociedade. O presente trabalho apresenta técnicas heurísticas de construção de rotas (vizinho mais próximo, inserção do mais próximo e inserção do mais distante) e as heurísticas de melhorias de rotas (2-opt e 3-opt) que foram implementadas no ambiente MATLAB para otimizar o PCV. Para verificar suas performances, as abordagens foram testadas em instâncias, tais como: Att48, Berlin52 e Ulysses22 que estão disponíveis na TSPLIB. Verificou-se que a combinação entre as heurísticas de construção de rotas com o mecanismo 3-opt apresentaram as melhores soluções.

Palavras-Chave: Problema do Caixeiro Viajante, Heurísticas, Otimização.

ABSTRACT

The Traveling Salesman Problem (TSP) is a classic problem of combinatorial optimization that has been worked for several researchers due to its broad applicability in real problems, and its resolution can bring many benefits to society. This work presents the construction heuristics (nearest neighbour, insertion of the nearest and insertion of the farthest) and heuristics improvements routes (2-opt and 3-opt) that were implemented in the MATLAB environment to optimize the PCV. To check your performances, the approaches were tested on instances, such as: Att48, Berlin52 and Ulysses22 that are available in the TSPLIB. It was found that the combination between heuristics of construction of routes with the mechanism 3-opt presented the best solutions.

Keywords: Traveling Salesman Problem, Heuristics, Optimization

LISTA DE FIGURAS

FIGURA 1- Representação das classes P, NP e NP-completo	17
FIGURA 2- Exemplo de grafo	18
FIGURA 3- A imagem (a) representa o problema proposto por Willian Rowan Hamilton. Imagem (b) representa uma solução para o problema.....	19
FIGURA 4- Exemplo de percurso.....	20
FIGURA 5- Problema do Caixeiro Viajante com 4 nós.....	23
FIGURA 6- Representa uma ilustração da aplicação do vizinho mais próximo.....	28
FIGURA 7- Representa uma ilustração da aplicação da inserção do mais próximo.....	30
FIGURA 8- Representa uma ilustração da aplicação da inserção do mais distante.....	31-32
FIGURA 9- A imagem (a) representa a solução original e a imagem (b) representa o resultado obtido através da heurística de busca local 2-opt com uma distância	33
FIGURA 10- A imagem (a) representa a solução original e a imagem (b) representa o resultado obtido através da heurística de busca local 3-opt.....	34

LISTA DE TABELAS

TABELA 1- Representa algumas ordens de grandezas	16
TABELA 2- Tabela que representa a complexidade de resolução de um PCV assimétrico...	21
TABELA 3- Solução de instâncias do PCV	21-22
TABELA 4- Matriz de custos referente ao PCV representado na Figura 5	24
TABELA 5- Matriz de distâncias simétricas	26
TABELA 6- Resultados obtidos através das heurísticas de construção para o problema Att48	35
TABELA 7- Desvios em porcentagem das soluções obtidas através das heurísticas de construção em relação ao valor ótimo do problema Att48.....	35
TABELA 8- Resultados obtidos através das heurísticas de construção com 2-opt para o problema Att48	36
TABELA 9- Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o algoritmo 2-opt em relação ao valor ótimo do problema Att48.....	37
TABELA 10- Resultados obtidos através das heurísticas de construção com 3-opt para o problema Att48.....	37
TABELA 11- Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o mecanismo 3-opt em relação ao valor ótimo do problema Att48.....	38
TABELA 12- Resultados obtidos através das heurísticas de construção para o problema Berlin52.....	38
TABELA 13- Desvios em porcentagem das soluções obtidas através das heurísticas de construção em relação ao valor ótimo do problema Berlin52	39
TABELA 14- Resultados obtidos através das heurísticas de construção com 2-opt para o problema Berlin52	39

TABELA 15- Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o mecanismo 2-opt em relação ao valor ótimo do problema do Berlin52.....	40
TABELA 16- Resultados obtidos através das heurísticas de construção com 3-opt para o problema Berlin52.....	40
TABELA 17- Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o algoritmo 3-opt em relação ao valor ótimo do problema Berlin52.....	41
TABELA 18- Resultados obtidos através das heurísticas de construção para o problema Ulysses22	41
TABELA 19- Desvios em porcentagem das soluções obtidas através das heurísticas de construção em relação ao valor ótimo do problema Ulysses22.....	42
TABELA 20- Resultados obtidos através das heurísticas de construção com 2-opt para o problema Ulysses22.....	42
TABELA 21- Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o mecanismo 2-opt em relação ao valor ótimo do problema Ulysses22.....	43
TABELA 22- Resultados obtidos através das heurísticas de construção com 3-opt para o problema Ulysses22.....	43-44
TABELA 23- Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o algoritmo 3-opt em relação ao valor ótimo do problema Ulysses22.....	44
TABELA 24- Representa os tempos médios obtidos nos testes computacionais.....	44
TABELA 25- Os melhores resultados obtidos.....	45
TABELA 26- Resultados obtidos através do VNS com os mecanismos k -opt.....	45

SUMÁRIO

Resumo.....	v
<i>Abstract</i>	vi
Lista de Figuras.....	vii
Lista de Tabelas.....	vii-ix
1.INTRODUÇÃO	12
1.1 Justificativa.....	13
1.2 Objetivos.....	14
1.2.1.Objetivo geral.....	14
1.2.2.Objetivos específicos.....	14
1.3 Metodologia da pesquisa.....	14-15
1.4 Estrutura do trabalho.....	15
2.FUNDAMENTAÇÃO TEÓRICA	16
2.1 Problemas de otimização combinatória.....	16-17
2.2 Algoritmos de otimização.....	17-18
2.3 Grafos.....	18
2.3.1 Grafos Hamiltonianos.....	19
2.4 Descrição do problema.....	19-25
3. ABORDAGENS DE RESOLUÇÃO	26
3.1 Heurísticas.....	26
3.1.1 Heurísticas de construção de rotas.....	26-27
3.1.1.1 Vizinho mais próximo (VMP).....	27-28
3.1.1.2 Inserção do mais próximo (IMP).....	28-30
3.1.1.3 Inserção do mais distante (IMD).....	30-32
3.1.2 Heurísticas de melhorias de rotas.....	32
3.1.2.1 Heurística de melhorias de rota 2-opt	32-33

2.3.2.2 Heurística de melhorias de rota 3-opt	33-34
4.RESULTADOS	34-45
5.CONCLUSÕES	46
REFERÊNCIAS	47-51
APÊNDICE	52

1 INTRODUÇÃO

O Problema do Caixeiro Viajante (PCV), termo em inglês denominado de *Traveling Salesman Problem* (TSP), tem sido bastante abordado por pesquisadores de diferentes áreas, como: matemática, engenharia e ciência da computação, devido sua larga aplicabilidade em problemas práticos. Pode-se citar aplicação na otimização de palhetas em turbinas a gás, no problema de roteamento de veículos, no sequenciamento de DNA, em trabalhos administrativos, na análise de estruturas cristalinas etc. Exemplos de aplicações do PCV podem ser encontrados em (GOLDBARG & LUNA, 2005).

O objetivo do caixeiro viajante é determinar a menor distância ou menor custo iniciando-se seu percurso em uma cidade origem, percorrendo todas as cidades restantes uma única vez e retornando ao local de partida. Esse percurso é denominado de ciclo de Hamilton. Esse nome foi em homenagem a Willian Rowan Hamilton que, em 1857 propôs um jogo ao qual denominou de *Around the World*. O objetivo do jogo era encontrar o melhor caminho percorrendo todos os vértices de um dodecaedro uma única vez, e retornando ao local de partida, onde cada vértice estava associado a uma cidade importante da época.

O problema consiste de um conjunto de n cidades e uma matriz de distâncias ou custos que pode ser modelado através de um grafo $G = (N, A)$ não orientado, onde N é o conjunto de vértices que representam as cidades ou nós, e A é o conjunto de arestas ou arcos que representam as distâncias ou custos para percorrer duas cidades quaisquer.

O PCV é um problema NP-Difícil de acordo com (GAREY & JOHNSON, 1979). Caso seja considerado como um problema de decisão, cuja resposta é sim ou não, passa ser do tipo NP- completo. Portanto, é um problema que não é possível resolvê-lo em tempo polinomial através de métodos exatos, devido à quantidade de soluções que cresce fatorialmente à medida que aumenta o número de cidades visitadas o que exige intenso esforço computacional para executar todas as operações. Se tivermos um computador hipotético, capaz de processar um caso a cada ciclo de relógio, e que possua uma frequência nominal de 53Mhz, seriam necessários 2.3×10^9 s ou cerca de 73 anos para encontrar o valor ótimo de um PCV composto de 20 cidades (SILVEIRA, 2000). Uma revisão histórica mais detalhada sobre o PCV pode ser encontrada em (LAWLER *et.al*, 1985). No presente trabalho vamos utilizar as heurísticas de construção e de melhorias de rotas para resolução do PCV.

1.1 Justificativa

A globalização e o sistema econômico capitalista impulsionaram a competitividade entre empresas do mesmo ramo. Cada empresa utiliza estratégias para satisfazer as necessidades dos clientes que exigem uma alta qualidade nos serviços. Em alguns casos os clientes estão distantes do local de produção e à medida que os produtos são transportados para outros lugares às empresas estão sujeitas aos custos ocasionados pelo serviço de entrega dos produtos. Sendo que um dos fatores que aumenta o custo das mercadorias é o transporte. Portanto, é uma questão bastante analisada tanto no meio profissional, como no acadêmico. Neste contexto o PCV pode ser empregado com objetivo de encontrar as melhores estratégias para diminuir os custos decorrentes dos atendimentos aos clientes. Também pode ser empregado na confecção de placas de circuito impresso (VITTES, 1999), na análise da estrutura de cristais (BLAND & SHALLCROSS, 1987), no mapeamento de genoma (AVNER *et.al*, 2001), no sequenciamento de DNA (GONNET *et.al*, 2000), nos planos de rotas (HOFFMAN & PADBERG, 2015), no problema quadrático de alocação (BOAVENTURA NETO, 2006), dentre outras.

O PCV é bastante analisado por muitos pesquisadores pela sua simplicidade na formulação, complexidade na resolução e sua larga aplicabilidade no mundo real e a descoberta de métodos eficientes para sua resolução, implica na resolução de muitos problemas complexos da área de otimização (SILVA, 2013).

1.2 Objetivos

1.2.1 Objetivo geral

Propõe-se neste trabalho a resolução do Problema do Caixeiro Viajante através de heurísticas de construção e de melhorias de rotas através de rotinas que foram implementadas no MATLAB.

1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Entender como cada heurística busca por soluções aproximadas.
- Implementar computacionalmente tanto as heurísticas de construção de rotas (vizinho mais próximo, inserção do mais próximo e inserção do mais distante) como as heurísticas de melhorias de rotas (2-opt e 3-opt).
- Aplicar as heurísticas em instâncias consolidadas disponíveis na biblioteca TSPLIB (*Traveling Salesman Problem Library*), onde contém mais de 100 exemplos de instâncias com soluções ótimas (PRESTES 2006).
- Comparar os resultados obtidos através das implementações das abordagens com o valor ótimo de cada instância apresentado em literatura.

1.3 Metodologia da pesquisa

Primeiramente foi feito um levantamento bibliográfico visando o entendimento do PCV e constatando a inviabilidade prática de resolvê-lo através de algoritmos determinísticos¹, definindo-se as heurísticas de construção e de melhorias de rotas para serem abordadas no desenvolvimento do trabalho, assim como a plataforma computacional (Matlab) a ser utilizada na implementação das abordagens.

¹Algoritmos determinísticos: Um algoritmo é dito determinístico se o resultado de cada operação é definido de forma única (ORNELAS *et.al*, 2006).

No Matlab foram implementadas rotinas computacionais em relação às heurísticas abordadas com finalidade de realizar os procedimentos de cada método. As técnicas de construção de rotas foram aplicadas de forma isoladas em instâncias consolidadas na biblioteca TSPLIB.

Cada solução formada por meio das técnicas de construção foi aprimorada através dos mecanismos de busca local (2-opt e 3-opt) com o objetivo de aumentar a eficiência em relação ao valor ótimo, e para avaliar cada solução foi determinado um desvio. As técnicas foram comparadas em desempenho e complexidade.

1.4 Estrutura do trabalho

O presente trabalho de conclusão de curso está organizado da seguinte forma:

No Capítulo 1 tratamos da descrição e histórico do PCV, onde constam a justificativa, os objetivos e a metodologia utilizada para o desenvolvimento do estudo realizado.

Já o Capítulo 2 denominado de Fundamentação Teórica, apresenta a descrição de problemas de otimização combinatória, a definição de algoritmos de otimização, o conceito de grafos, a descrição de grafos Hamiltonianos e do PCV. Já o Capítulo 3 aborda os métodos heurísticos.

O Capítulo 4 apresenta os resultados obtidos através da implementação de cada heurística utilizada.

Finalmente, o Capítulo 5 apresenta a conclusão do trabalho seguida das referências.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Problemas de otimização combinatória

A otimização combinatória é uma área da matemática e da ciência da computação que analisa os problemas de otimização com objetivo de encontrar a melhor solução no espaço amostral (conjunto de itens) que minimize ou maximize uma função objetivo atendendo certas regras (restrições do problema). Se a solução ótima encontrada satisfaz o conjunto de restrições impostas pelo problema, a solução é denominada factível.

Uma forma direta de resolver essa categoria de problemas de otimização é por meio da avaliação de todas as soluções possíveis do espaço amostral com objetivo de encontrar a melhor solução. Quando o número de soluções cresce fatorialmente à medida que aumenta a dimensionalidade do problema é preciso utilizar abordagens factíveis para resolver esses problemas.

A complexidade de uma abordagem é dada pela ordem de grandeza de uma função, que relaciona o tempo de execução de um algoritmo de acordo com o tamanho de um problema, que segundo a teoria de complexidade podem ser classificados em P (*Polynomial time*) e NP (*Non-Deterministic Polynomial time*). Para representar a ordem de grandeza utiliza-se a letra O , seguida por uma função. Na Tabela 1 podem ser vistos algumas ordens de grandezas.

Tabela 1. Representa algumas ordens de grandezas. Fonte: MOLE, 2012.

Função	Ordem de Grandeza
Constante	$O(1)$
Logarítmica	$O(\log n)$
Linear	$O(n)$
Quadrática	$O(n^2)$
Cúbica	$O(n^3)$
Polinomial	$O(n^c)$, c real
Exponencial	$O(n^c)$, c real e >1
Fatorial	$O(n!)$

Os problemas que pertencem à classe P são ditos tratáveis para os quais existem algoritmos determinísticos capaz de resolvê-los em tempo polinomial cuja função de complexidade destes problemas é dada por: $O(p(n))$, onde $p(n)$ é um polinômio. Já a classe NP engloba os problemas resolvidos por meio de algoritmos polinomiais não determinísticos. Tais problemas são denominados de intratáveis, onde a complexidade em relação a esta classe é $O(c^n)$, com $c > 1$. De acordo com (GAREY & JOHNSON, 1979) o PCV é considerado intratável devido sua característica e sua função de complexidade é dada por: $O(n!)$.

Pode-se afirmar que caso seja determinado um algoritmo capaz de resolver qualquer problema NP -completo², estará provado a igualdade entre as classes P e NP (MORAIS, 2010). A dificuldade para se provar isto é pelo fato de não existir algoritmos, e isto continua sendo uma grande incógnita para os pesquisadores. Uma ilustração sobre as classes P , NP e NP - completo pode ser visualizada na Figura 1.

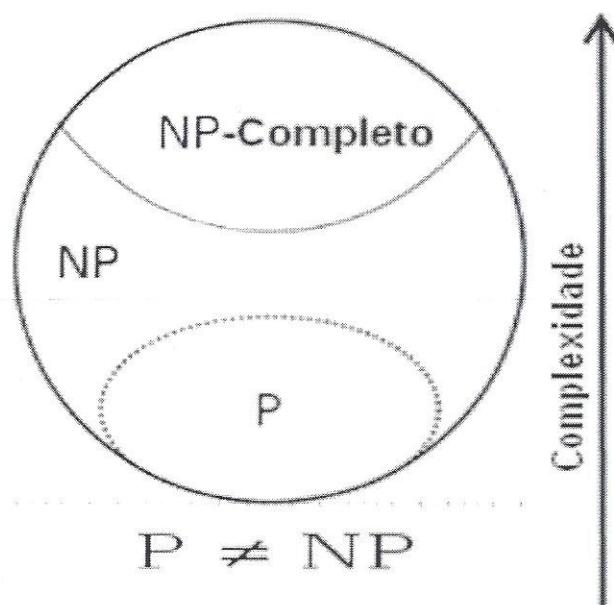


Figura 1 Representação das classes P , NP e NP -completo. Fonte: SILVA, 2013.

2.2 Algoritmos de otimização

As abordagens de resolução de problemas otimização são classificadas em três categorias: métodos exatos, as heurísticas e as metaheurísticas.

Métodos exatos são algoritmos que determinam o valor ótimo em tempo não

² Um problema X é NP -completo, se X está em NP (GERSTING, 1993)

polinomial. A maioria dos problemas reais apresenta um elevado número de soluções possíveis, o que torna inviável resolvê-los nestes casos. Já as heurísticas são procedimentos que visam encontrar soluções de boa qualidade em tempo computacional razoável. Na maioria dos casos são desenvolvidas especificamente para um determinado problema.

As metaheurísticas são procedimentos heurísticos que podem ser aplicados a vários tipos de problemas. Sua principal característica é escapar de um ótimo local através de mecanismos de fuga (GOMES, 2001). São exemplos de metaheurísticas: *Simulated Annealing* (AS), Algoritmos Genéticos (AG).

2.3 Grafos

Um grafo G é uma estrutura matemática que consiste de elementos denominados de vértices e arestas que pode ser modelado por $G = (N, A)$, onde N é o conjunto de vértices da estrutura e A é o conjunto de arestas (i, j) , onde $i, j \in N$. Se arestas não possuem orientações o grafo é dito não orientado, caso contrário; o grafo é denominado orientado ou dígrafo.

Figura 2 ilustra um exemplo de grafo que possui quatro vértices

$N = \{1,2,3,4\}$ e seis arestas $A = \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$.

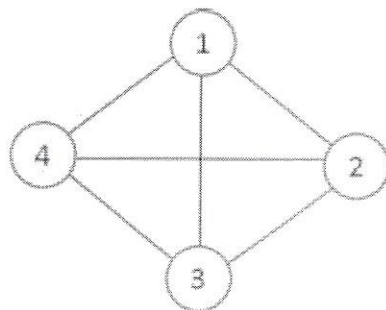


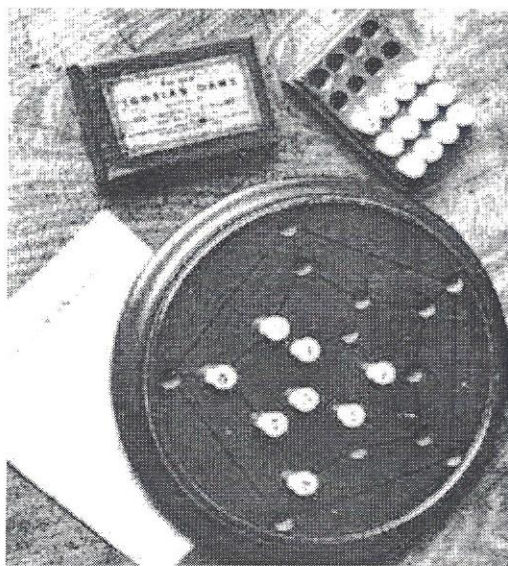
Figura 2. Exemplo de grafo. Fonte: KANDA, 2012.

Algumas definições sobre percurso de um grafo são descritas a seguir:

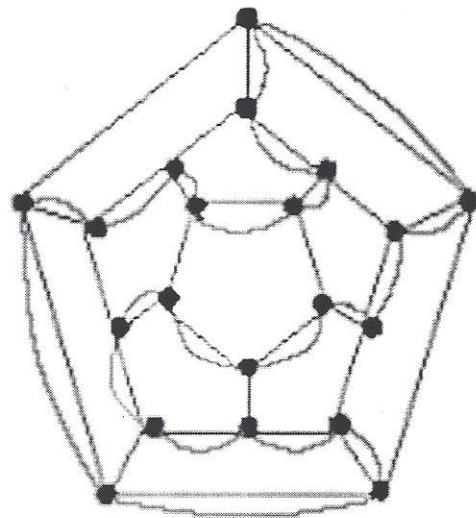
- Caminho é uma sequência de arestas em que todos os vértices visitados são diferentes.
- Circuito representa um trajeto fechado, onde o vértice inicial coincide com o vértice final.
- Cadeia é uma sequência de arcos com extremidade em comum.
- Ciclo é uma cadeia fechada, onde os vértices inicial e final coincidem.

2.3.1 Grafos Hamiltonianos

Um ciclo que passa por todos os vértices de um grafo $G = (N, A)$ uma única vez é denominado de grafo Hamiltoniano. Sendo que o ciclo tem que iniciar e terminar no mesmo vértice. Esse nome foi em homenagem ao matemático irlandês Willian Rowan Hamilton que propôs em 1857 um problema denominado *viagem pelo mundo*. A Figura 3 ilustra o problema proposto por Hamilton e uma possível solução.



(a)



(b)

Figura 3. A imagem (a) representa o problema proposto por Willian Rowan Hamilton. Fonte: <http://www.math.uwaterloo.ca/tsp/history/>, 2015. Imagem (b) representa uma solução para o problema. Fonte: http://www.siscorp.com.br/siscorpnews/sexta_edicao/assunto_marco.html, 2015.

2.4 Descrição do problema

O PCV é um clássico exemplo de problema de otimização combinatória (CAIXEIRO, 2015) e (SILVA & OLIVEIRA, 2006) que consiste em encontrar o menor ciclo de Hamilton entre todas as possibilidades de solução possíveis para um conjunto de n cidades ou nós, onde $n \geq 3$. A Figura 4 ilustra uma possível solução para um helicóptero distribuir vacinas em 50 pontos no Estado de São Paulo.

Tabela 2. Tabela que representa a complexidade de resolução de um PCV assimétrico. Fonte: CAIXEIRO, 2015.

n	Rotas por segundo	$(n - 1)!$	Cálculo total
5	250×10^6	24	Insignificante
10	100×10^6	362 880	0.003s
15	71×10^6	87×10^9	1200s
20	53×10^6	1.2×10^{17}	2.3×10^9 s
25	42×10^6	6.2×10^{23}	1.4×10^{16} s

O avanço na resolução de instâncias do PCV começou a partir da década 1980 através de investigação de Grötschel, Crowder e Padberg, como pode ser visto na Tabela 3 que apresenta tanto os problemas como os pesquisadores que os resolveram a partir de 1954 até 2006.

Em 1954, Dantzig, Fulkerson adotaram o PCV como sendo um problema de programação linear inteira, e resolveram uma instância de 49 cidades. Já em 2004, Applegate, Bixby, Chvátal, Cook e Helsgaun resolveram um problema de 24978 pontos que representam localidades da Suécia, e comprovaram que não existe caminho mais curto que a solução encontrada (APPLEGATE *et.al*, 2004). O mesmo grupo de pesquisadores resolveu por meio do Concorde TSP uma instância composta de 85900 cidades (APPLEGATE *et.al*, 2006).

Tabela 3. Solução de instâncias do PCV. Fonte: Adaptado de PRESTES, 2006.

Ano	Instância	Tamanho	Pesquisadores
1954	dantzig49	49	Dantzig, Fulkerson e Johnson (1954)
1962	aleatória	64	Held e Karp (1962)
1975	aleatória	67	Camerini, Fratta e Maffioli (1975)
1980	gr120	120	Grötschel (1980)
1980	lin138	318	Crowder e Padberg (1980)
1987	att532	532	Padberg e Rinaldi (1987)

Tabela 3. Continuação.

1991	gr666	666	Grötschel e Holland (1991)
1991	pr2392	2392	Padberg e Rinaldi (1991)
1995	pla7397	7397	Applegate, Bixby, Chvátal e Cook (1995)
1998	usa13509	13509	Applegate, Bixby, Chvátal e Cook (1998)
2001	d15112	15112	Applegate, Bixby, Chvátal e Cook (2001)
2004	sw24798	24978	Applegate, Bixby, Chvátal, Cook e Helsgaun (2004)
2006	pla85900	85900	Applegate, Bixby, Chvátal, Cook e Helsgaun (2006)

O PCV pode ser modelado através de um grafo $G = (N, A)$, onde $N = \{1, \dots, n\}$ é o conjunto de vértices que representa as cidades ou nós, e $A = \{1, \dots, a\}$ é o conjunto de arestas ou arcos que estão associados às distâncias ou custos para percorrer um trajeto entre duas cidades. Com o objetivo de encontrar o melhor ciclo de Hamilton que minimiza a distância total, as cidades serão todas visitadas uma única vez pelo caixeiro viajante.

De acordo com (GOLDBARG & LUNA, 2005), a formulação matemática mais abordada para o PCV foi proposta, segundo (CHRISTOFIDES *et al*, 1979) por Dantzig, Fulkerson e Johnson em 1954. As equações (3) a (7) representam este modelo de programação linear inteira formulado sobre um grafo $G = (N, A)$.

$$Z = \min \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij} \quad (3)$$

sujeito as restrições:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (4)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (5)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subset N \quad (6)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in N \quad (7)$$

onde:

c_{ij} = custo para ir de uma cidade i até uma cidade j . No problema em questão esta variável representa a distância.

$x_{ij} = 1$, se o arco (i, j) for escolhido para integrar o ciclo de Hamilton.

$x_{ij} = 0$, caso contrário;

S = é um sub-grafo de G .

$|S|$ = números de vértices do sub-grafo S .

Na formulação acima, a equação (3) representa a função objetivo que consiste em minimizar a distância ou custo originado a partir do ciclo de Hamilton percorrido pelo caixeiro viajante. As equações de 4 a 7 são as restrições do PCV.

A equação (4) representa a restrição que determina que o fluxo de chegada em cada cidade j deve ser 1 e a equação (5) garante que o fluxo de saída de cada cidade i deve ser 1. A restrição (6) elimina subciclo e a restrição (7) determina que as variáveis sejam binárias, ou seja, podem assumir apenas os valores 0 ou 1.

A formulação de Dantzig, Fulkerson e Johnson é da ordem $O(2^n)$, ou seja, o número de restrições cresce fatorialmente de acordo com o número de vértices do grafo, tornando o PCV intratável computacionalmente e isto implica que a resolução do problema por esse modelo passa-se a ser inviável para problemas de grande porte. Para exemplificar essa formulação vamos utilizar o PCV com 4 nós ilustrado na Figura 5. As matrizes de custos deste problema podem ser vistos na Tabela 4.

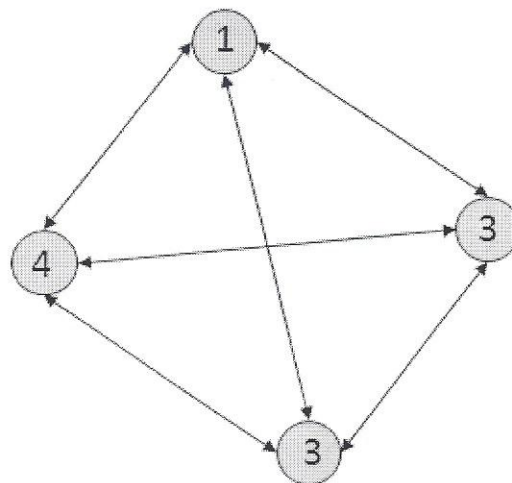


Figura 5. Problema do Caixeiro Viajante com 4 nós.

Tabela 4. Matriz de custos referente ao PCV representado na Figura 5.

Matriz C	c_1	c_2	c_3	c_4
c_1	0	x_{12}	x_{13}	x_{14}
c_2	x_{21}	0	x_{23}	x_{24}
c_3	x_{31}	x_{32}	0	x_{34}
c_4	x_{41}	x_{42}	x_{43}	0

A função objetivo para o problema ilustrado na Figura 5 é dada pela equação (8).

$$Z = \min \sum_{j=1}^4 \sum_{i=1}^4 c_{ij}x_{ij} \quad (8)$$

onde:

$$Z = c_{11}x_{11} + c_{21}x_{21} + c_{31}x_{31} + c_{41}x_{41} + c_{12}x_{12} + c_{13}x_{13} + c_{14}x_{14} + c_{22}x_{22} + c_{32}x_{32} + c_{42}x_{42} + c_{23}x_{23} + c_{24}x_{24} + c_{33}x_{33} + c_{43}x_{43} + c_{34}x_{34} + c_{44}x_{44}$$

O conjunto de restrições expressa em (9) representa o fluxo de chegada em cada cidade j para o PCV representado pela Figura 5. De acordo com a restrição (4) este fluxo deve ser igual a 1.

$$\sum_{i=1}^4 x_{ij} = 1 \quad \forall j \in \{1, 2, 3, 4\} \quad (9)$$

ou seja:

$$\begin{aligned} x_{11} + x_{21} + x_{31} + x_{41} &= 1 \\ x_{12} + x_{22} + x_{32} + x_{42} &= 1 \\ x_{13} + x_{23} + x_{33} + x_{43} &= 1 \\ x_{14} + x_{24} + x_{34} + x_{44} &= 1 \end{aligned}$$

Mesmo exemplifica a restrição (5) que garante que o fluxo de saída em cada cidade i deve ser igual a 1.

$$\sum_{j=1}^4 x_{ij} = 1 \quad \forall i \in \{1, 2, 3, 4\} \quad (10)$$

ou seja:

$$\begin{aligned} x_{11} + x_{12} + x_{13} + x_{14} &= 1 \\ x_{21} + x_{22} + x_{23} + x_{24} &= 1 \\ x_{31} + x_{32} + x_{33} + x_{34} &= 1 \\ x_{41} + x_{42} + x_{43} + x_{44} &= 1 \end{aligned}$$

Expressões (11) a (12) exemplificam a restrição (6). Estas equações impedem respectivamente que haja formação de sub-rotas com 2 e 3 nós. A seguir podem ser visualizadas tanto as equações como as inequações que representam sub-rotas que podem ser geradas à medida em que vão sendo necessárias, até que seja encontrado um ciclo de Hamilton.

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \leq 2 - 1 \leq 1 \quad (11)$$

ou seja:

$$x_{12} + x_{21} \leq 1 \quad S = \{1,2\}$$

$$x_{13} + x_{31} \leq 1 \quad S = \{1,3\}$$

$$x_{14} + x_{41} \leq 1 \quad S = \{1,4\}$$

$$x_{23} + x_{32} \leq 1 \quad S = \{2,3\}$$

$$x_{24} + x_{42} \leq 1 \quad S = \{2,4\}$$

$$x_{34} + x_{43} \leq 1 \quad S = \{3,4\}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1 \leq 3 - 1 \leq 2 \quad (12)$$

ou seja:

$$x_{12} + x_{21} + x_{13} + x_{31} + x_{23} + x_{32} \leq 2 \quad S = \{1,2,3\}$$

$$x_{12} + x_{21} + x_{14} + x_{41} + x_{24} + x_{42} \leq 2 \quad S = \{1,2,4\}$$

$$x_{13} + x_{31} + x_{14} + x_{41} + x_{34} + x_{43} \leq 2 \quad S = \{1,3,4\}$$

$$x_{23} + x_{32} + x_{24} + x_{42} + x_{34} + x_{43} \leq 2 \quad S = \{2,3,4\}$$

A última restrição representada pela equação (7) garante que a variável x_{ij} deve ser binária, e isto implica que $x_{11}, x_{21}, x_{31}, x_{41}, x_{12}, x_{13}, x_{14}, x_{22}, x_{32}, x_{42}, x_{23}, x_{24}, x_{33}, x_{43}, x_{34}, x_{44} \in \{0,1\}$.

3 ABORDAGENS DE RESOLUÇÃO

3.1 Heurísticas

A palavra heurística surgiu a partir da expressão grega *heuriskein* que significa descobrir (REEVES, 1995). São métodos que visam determinar a melhor solução no espaço amostral de soluções em tempo computacional razoável.

Os métodos heurísticos são preferidos para solucionar problemas de grande porte que apresentam várias possibilidades de soluções que não podem ser resolvidos computacionalmente em tempo polinomial através de metodologias exatas.

Os procedimentos heurísticos para resolução do PCV tratado neste trabalho podem ser divididos em dois grupos: métodos de construção de rotas e os métodos de melhoria de rotas. Um terceiro grupo pode ser considerado, devido às técnicas de construção e de melhorias de rotas que são utilizadas de forma única. Para exemplificação das heurísticas de construção são utilizados os elementos da Tabela 5 que representam as distâncias entre os vértices de um PCV constituído com 5 nós.

Tabela 5. Matriz de distâncias simétricas.

Matriz D	d_1	d_2	d_3	d_4	d_5
d_1	0	2	3	9	8
d_2	2	0	4	7	8
d_3	3	4	0	6	3
d_4	9	7	6	0	10
d_5	8	8	3	10	0

3.1.1 Heurísticas de construção de rotas

São algoritmos que constroem uma solução através de adição de elementos no ciclo (DORIGO & STÜTZLE, 2004). Na heurística clássica os elementos são inseridos segundo uma função gulosa, ou seja, é uma função que insere passo a passo o melhor elemento. O procedimento de adição de cidades na rota é forma gradativa e contínua, e termina quando não for mais possível encontrar boas soluções.

A solução encontrada na maioria dos casos não tem boa qualidade. Em cada iteração a heurística determina o melhor elemento a ser inserido na solução através das restrições que

são adotadas em cada problema, o critério de escolha é local (CAMPELLO & MACULAN, 1994).

3.1.1.1 Vizinho mais próximo (VMP)

O método define uma cidade qualquer para iniciar o ciclo, e verifica a cidade mais próxima da última cidade visitada e a adiciona. Este processo iterativo termina quando todas as cidades sejam visitadas e a rota seja concluída. O método não garante resultados exatos, mas os resultados podem ser utilizados para a aplicação dos métodos de melhorias de rotas, sendo este algoritmo de ordem $O(n^2)$. Exemplos de aplicações do vizinho mais próximo ao PCV podem encontrados em (MENGER, 1932) e (SOLOMON, 1987). O pseudocódigo que descreve a implementação do vizinho mais próximo é ilustrado no algoritmo 1.

Algoritmo 1 Heurística do Vizinho mais Próximo

Passo 1 - Escolher uma cidade aleatória para iniciar o ciclo.

Passo 2 - Adicionar a cidade em que ainda não foi visitada, mais próxima da última cidade inserida no ciclo.

Passo 3 - Retornar para o *passo 2* até que todas as cidades sejam visitadas e retorne ao nó inicial.

Aplicando a heurística do vizinho mais próximo aos elementos da Tabela 3 obtém-se um ciclo de Hamilton de ordem 1-2-3-5-4-1, com uma distância $L=28$, como pode ser visualizado na Figura 6. A seguir pode-se observar os passos realizados para determinar o tour.

Passo 1- O ciclo inicia com o vértice 1;

Passo 2- Adiciona-se a solução o vértice 2, devido está mais próximo do vértice escolhido para iniciar o ciclo;

Passo 3- Adiciona-se a solução o melhor vizinho do vértice 2 que corresponde o vértice 3;

Passo 4- Adiciona-se a solução o vértice 5, devido está mais próximo do vértice 3.

Passo 5- Adiciona-se a solução melhor vizinho do vértice 5 que corresponde o vértice 4.

Passo 6- Conclui-se o ciclo conectando o último vértice inserido com o vértice inicial.

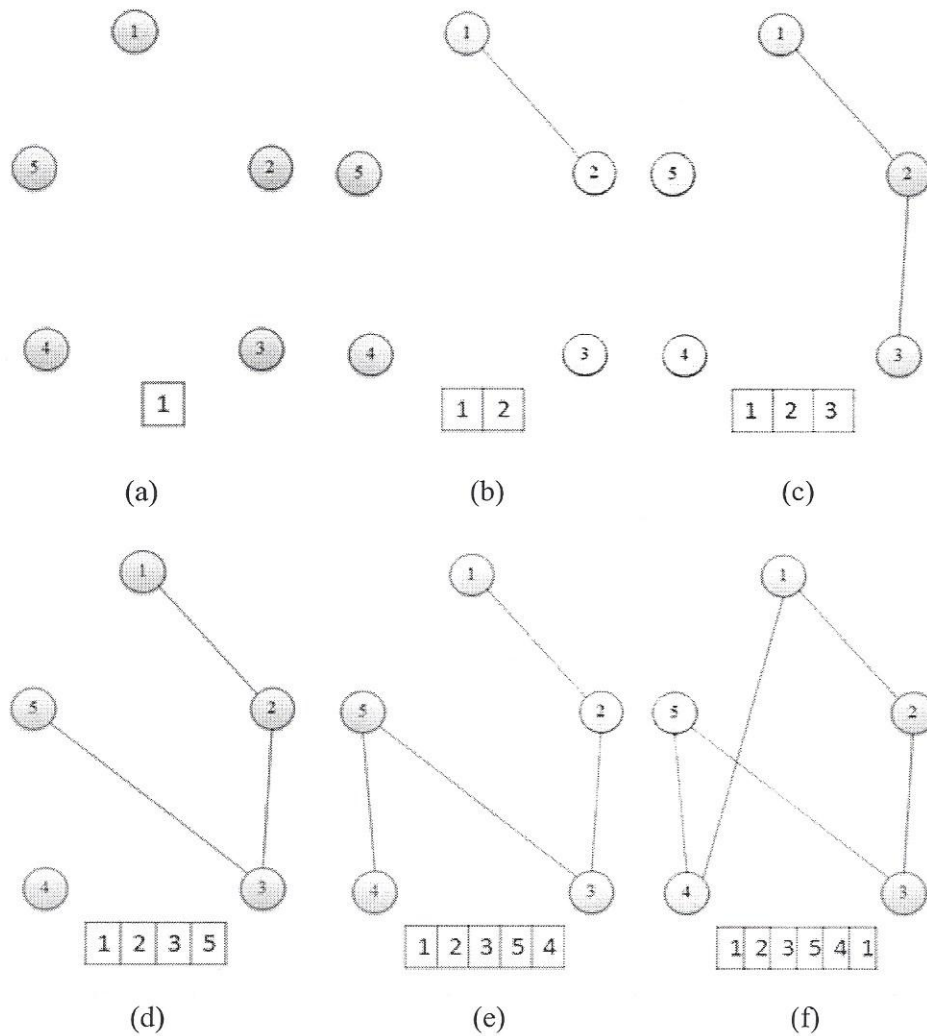


Figura 6. Representa uma ilustração da aplicação do vizinho mais próximo.

3.1.1.2 Inserção do mais próximo (IMP)

No algoritmo da inserção do mais próximo são envolvidos três níveis de decisão: inicialização, seleção e inserção (GOLDBARG & LUNA, 2005). No processo de inicialização, é escolhida aleatoriamente uma cidade i qualquer para iniciar o ciclo. Neste processo é determinada uma cidade j que representa a menor distância ou menor custo em relação à cidade escolhida inicialmente. As duas cidades definidas representam uma solução parcial, onde uma cidade k mais próxima de qualquer cidade da solução é encontrada através da distância euclidiana e posicionada na solução. A complexidade desta heurística é $O(n^2)$. O algoritmo 2 ilustra o pseudocódigo da inserção do mais distante.

Algoritmo 2 Heurística da inserção mais próximo

Passo 1 - Inicialização. Escolha uma cidade i ou nó aleatório para iniciar o ciclo; Encontrar uma cidade j tal que a distância seja mínima em relação à cidade escolhida inicial para formar a solução parcial.

Passo 2 - Seleção. Selecionar a cidade k mais próxima em relação a qualquer cidade que pertence à solução parcial.

Passo 3 - Inserção. Inserir a cidade k entre as cidades i e j escolhidas que minimize o delta apresentado pela expressão (13).

$$\Delta_{ij}(k) = c_{ik} + c_{kj} - c_{ij} \quad (13)$$

Passo 4 - Retornar para o **passo 2** até formar o ciclo de Hamilton.

O resultado da aplicação da inserção do mais próximo aos elementos da Tabela 6 conduz a um caminho 1-3-5-4-2-1 de distância $L=25$. Os passos realizados para determinar o caminho ilustrado na Figura 7 são apresentados abaixo.

Passo 1 - O caminho inicia com o vértice 1;

Passo 2- Adiciona-se ao caminho o vértice 2, devido estar mais próximo em relação ao vértice 1. Em seguida, verificou-se em relação aos elementos que compõem o caminho, que o vértice 3 é a cidade mais próxima do vértice 1, sendo inserido entre os vértices 1 e 2, resultando no sub-grafo da ordem 1-3-2-1;

Passo 3- Verificou-se que o vértice 5 está mais próximo de um vértice do caminho;

Passo 4- Constatou-se que os vértices 2, 3 e 5 que representam respectivamente as cidades i , j e k proporcionaram o menor valor para o delta, o que resultou na adição do vértice 5 entre os pares de vértices formando o caminho 1-3-5-2-1.

Passo 5- Verificou-se que o único vértice fora do caminho é o vértice 4.

Passo 6- Constatou-se que os vértices 2, 4 e 5 minimizaram o delta, o que implicou na inserção do vértice 4 entre os vértices 2 e 5 formando um tour de ordem 1-3-5-4-2-1.

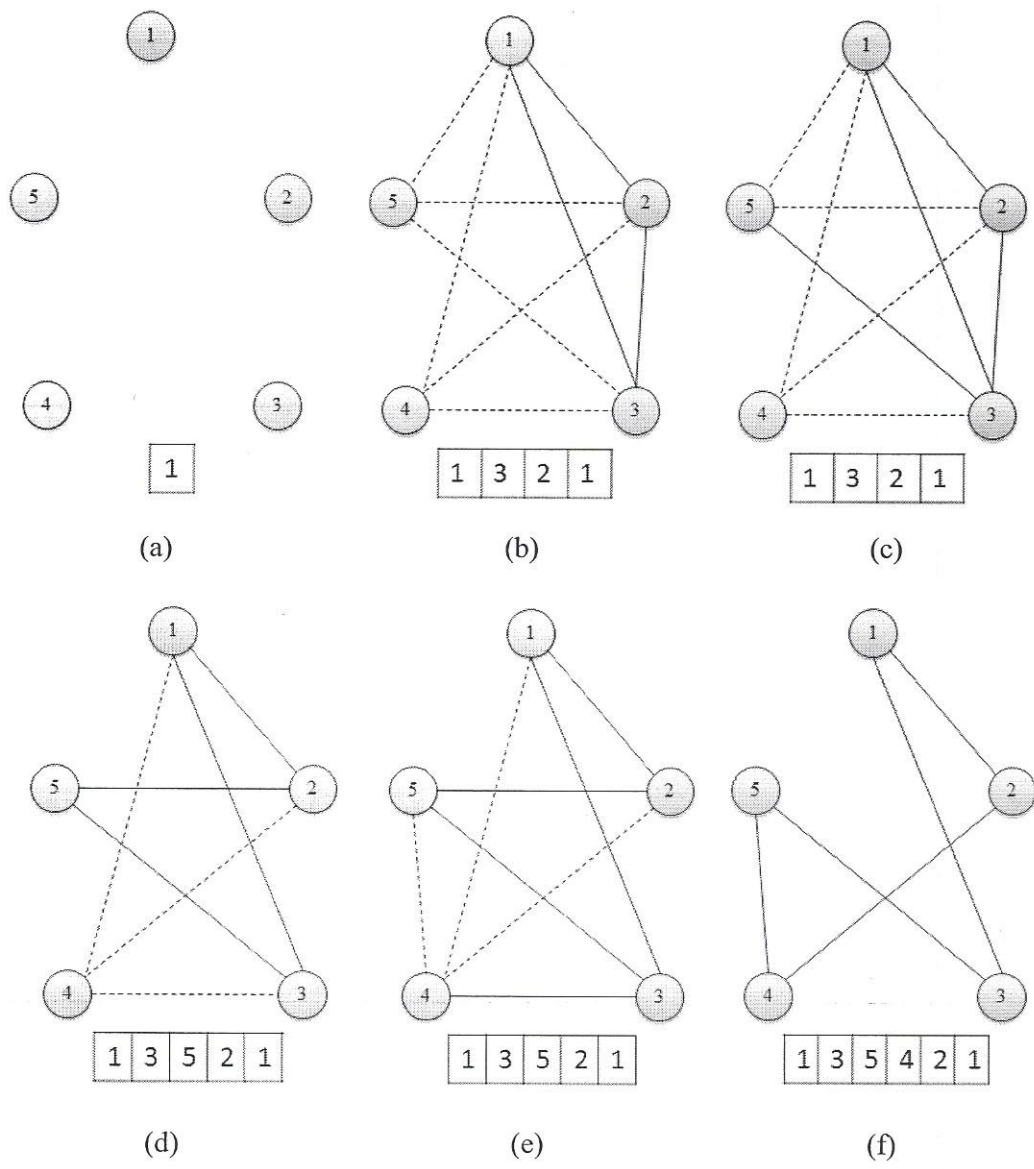


Figura 7. Representa uma ilustração da aplicação da inserção do mais próximo.

3.1.1.3 Inserção do mais distante (IMD)

Esta heurística de ordem $O(n^2)$, tem características semelhantes à inserção do mais próximo, a diferença está na determinação da cidade j e na seleção da cidade k . O pseudocódigo da inserção do mais distante está descrito no algoritmo 3.

Algoritmo 3 Heurística da inserção do mais distante

Passo 1 - Inicialização. Escolha uma cidade i ou nó aleatório para iniciar o ciclo; Encontrar uma cidade j tal que a distância seja máxima em relação à cidade escolhida inicial para formar a solução parcial.

Passo 2 - Seleção. Selecionar a cidade k mais distante em relação a qualquer cidade que pertence à solução parcial.

Passo 3 - Inserção. Inserir a cidade k entre as cidades i e j escolhidas que minimize o delta representado na equação (13).

Passo 4 - Retornar para o passo 3 até formar o ciclo de Hamilton.

A Figura 8 ilustra o resultado obtido através da heurística de inserção do mais distante. Nesta Figura, pode-se notar que a rota encontrada foi igual a 1-3-5-4-2-1 de distância $L=25$. A seguir, são apresentados os passos realizados para determinar este ciclo.

Passo 1- A rota inicia com o vértice 1;

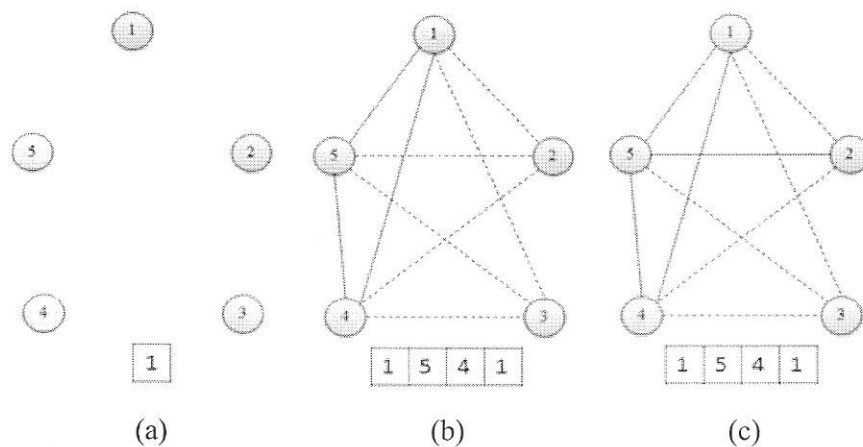
Passo 2- Adiciona-se na rota o vértice 4, devido está mais distante em relação ao vértice 1. Em seguida, verificou-se em relação aos elementos que compõem a rota, que o vértice 5 é a cidade mais distante do vértice 4, sendo inserido entre os vértices 1 e 4, resultando no sub-grafo da ordem 1-5-4-1;

Passo 3- Verificou-se que o vértice 2 está mais distante de um vértice do caminho;

Passo 4- Constatou-se que os vértices 1, 4 e 2 que representam respectivamente as cidades i , j e k proporcionaram o menor valor para o delta, o que resultou na adição do vértice 2 entre os pares de vértices formando a rota 1-5-4-2-1;

Passo 5- Verificou-se que o único vértice fora do caminho é o vértice 3;

Passo 6- Constatou-se que os vértices 1, 5 e 3 minimizaram o delta, o que implicou na inserção do vértice 3 que não pertence ao caminho entre os vértices 1 e 5 formando um tour de ordem 1-3-5-4-2-1.



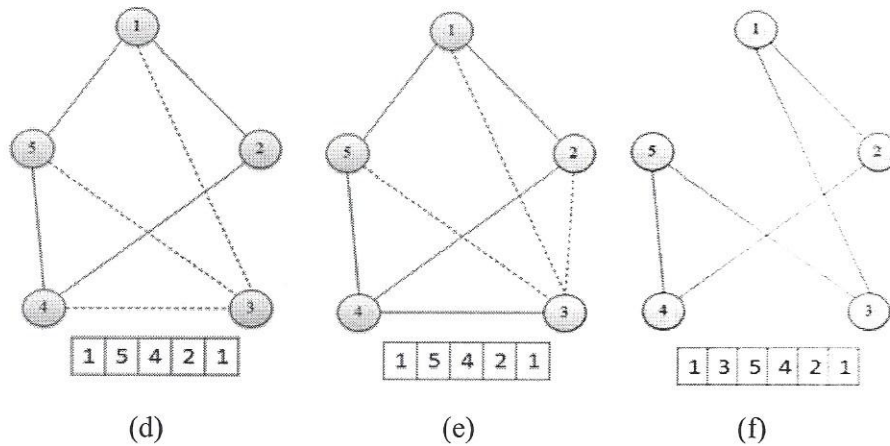


Figura 8. Representa uma ilustração da aplicação da inserção do mais distante.

3.2.1 Heurísticas de melhorias de rotas

As heurísticas de melhorias de rotas a serem apresentadas constituem uma família de heurísticas de busca local utilizadas para melhorar uma solução inicialmente determinada. Estes métodos quebra o ciclo em pontos, onde arcos são eliminados e substituídos por outros com finalidade de encontrar um novo percurso que possa substituir a solução anterior de modo que a distância seja inferior se comparado com a distância da solução que será substituída.

O procedimento de exclusão e inclusão de arcos termina quando não for mais possível melhorar a solução atual, o que é considerado um ótimo local. A seguir são descritas as heurísticas de melhorias de rotas utilizadas no trabalho.

3.1.2.1 Heurística de melhorias 2-opt

A heurística 2-opt é umas das técnicas usadas para melhorar solução do PCV, sendo proposta por (CROES, 1958). Realiza permutação de arestas, de modo que as arestas não adjacentes são desconectadas do ciclo e são conectadas com a finalidade de encontrar um ciclo que represente uma boa solução. Essa heurística elimina duas arestas e substitui por outras duas de forma não cruzada, de modo que a distância da nova solução seja menor que a distância da solução anterior. Esta heurística possui uma ordem de complexidade de $O(n^2)$. O pseudocódigo da heurística de melhorias 2-opt descrito no algoritmo 4, pode ser encontrado em (BENEVIDES, 2001).

Algoritmo 4 Heurística de melhorias 2-opt

Passo 1 - Obtenha uma rota inicial factível, utilizando algum dos procedimentos de formação de rotas.

Passo 2- Desligar 2 arcos da rota atual, reconectando os nós por meio de arcos diferentes daqueles que foram desconectados e que determine uma nova rota factível. Se o comprimento da rota nova for menor que o comprimento da rota anterior, troque a rota atual pela rota nova.

Passo 3- Prossiga no **passo 2** até que nenhuma melhoria possa ser alcançada.

A Figura 9 ilustra o processo de remoção e inserção de arcos. Pode-se observar que os arcos (3,5) e (4,1) são substituídos por (3,4) e (5,1).

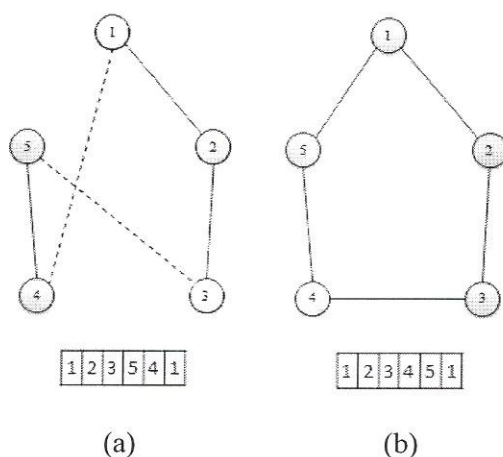


Figura 9. A imagem (a) representa a solução original e a imagem (b) representa o resultado obtido através da heurística de busca local 2-opt.

3.1.2.2 Heurística de melhorias 3-opt

A diferença dessa abordagem desenvolvida por (LIN, 1965) em relação à heurística 2-opt está no número de arcos removidos. No caso do método 3-opt são considerados três arcos ao invés de dois, e isto resulta em 7 combinações, sendo que destas combinações, apenas 4 representam trocas entre três arcos, e as outras correspondem trocas do tipo 2-opt. O pseudocódigo da heurística de busca local 3-opt que possui complexidade de ordem $O(n^3)$ é apresentado no algoritmo 5. A Figura 10 é utilizada para exemplificação da metodologia 3-opt, onde os arcos (1,4), (2,5) e (3,6) são trocados por (1,6), (2,3) e (4,5).

Algoritmo 5 Heurística de melhorias 3-opt

Passo 1- Determinar um ciclo através de outras abordagens.

Passo 2- Excluir 3 arestas do ciclo atual e inserir 3 novas arestas. Se a configuração resultante possuir um a distância menor que a distância da configuração anterior, efetue a troca.

Passo-3- Retornar para o *passo 2*, até que não seja mais possível obter melhorias.

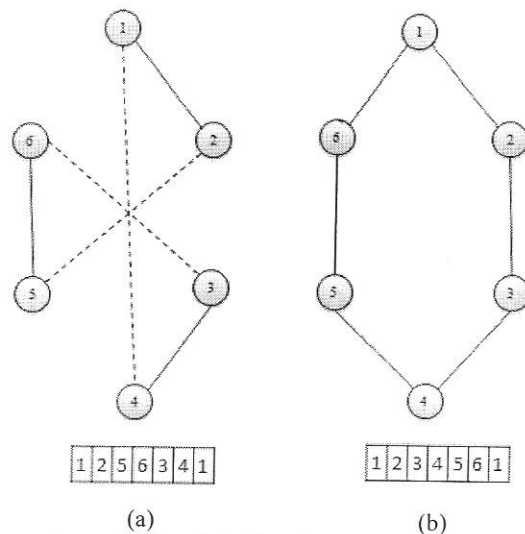


Figura 10. A imagem (a) representa a solução original e a imagem (b) representa o resultado obtido através da heurística de busca local 3-opt

4 RESULTADOS

Para verificar a eficiência de cada heurística implementada no ambiente MATLAB foram realizados testes computacionais em instâncias do PCV, mas especificamente ATT48, BERLIN52 e ULYSSES22 que estão disponibilizadas em <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95>. O número que acompanha cada instância representa as cidades ou nós a serem visitados no ciclo. As heurísticas foram executadas 10 vezes para cada um dos problemas em um computador com processador Inter(R) Core (TM)i5-2410M, 2.3GHz e 4GB de RAM. Através das implementações foram determinadas soluções a partir dos vértices adotados para iniciar o ciclo. Para cada solução foi determinado um desvio com o objetivo de avaliar cada uma em relação ao valor ótimo das instâncias abordadas. O desvio foi calculado através da expressão (14).

$$desvio = 100 \times \left(\frac{valor\ obtido - valor\ ótimo}{valor\ ótimo} \right) \quad (14)$$

Os vértices adotados encontram-se na primeira coluna das Tabelas, que apresentam também as soluções, os tempos de execução das abordagens em segundos, a média, o desvio padrão, a maior e menor solução encontrada para todos os vértices considerados. Pode-se notar em algumas a quantidade de iteração necessária para melhorar as soluções encontradas pelas heurísticas de construção de rotas. A quantidade de iteração está representada pela letra I.

A Tabela 6 mostra que a metodologia IMD apresentou a melhor solução dos testes computacionais que corresponde a 36127.6 para a instância ATT48 que representa 48 capitais dos Estados Unidos. Apesar da simplicidade da técnica, a solução ficou apenas 7.77% maior que 33523.7, que representa o valor ótimo para este problema de acordo com (MOLE, 2002), (NAGAMALAI *et al*, 2013) e (NURIYEVA. *et al*, 2012). Na Tabela 6 podem ser vistos os desvios referentes às soluções obtidas através das heurísticas de construção de rotas.

Tabela 6. Resultados obtidos através das heurísticas de construção para o problema Att48.

Instância:		Att48				
	VMP		IMP		IMD	
	T(s)	Solução	T(s)	Solução	T(s)	Solução
1	0.02	40526.4	0.25	37994.3	0.22	36687.0
2	0.02	39236.8	0.25	37519.1	0.22	36687.0
3	0.02	40963.9	0.26	38641.8	0.23	36127.6
4	0.02	41444.3	0.25	36623.1	0.22	36687.0
5	0.02	39246.0	0.25	36673.4	0.22	36342.4
6	0.02	42584.5	0.25	37668.7	0.22	36687.0
7	0.02	42982.0	0.25	37870.3	0.22	36687.0
8	0.02	40526.4	0.25	37994.3	0.22	36687.0
9	0.02	40740.8	0.27	37994.3	0.23	36687.0
10	0.02	42560.9	0.25	36766.8	0.23	36687.0
Média	0.02	41081.2	0.25	37574.6	0.22	36596.6
Desvio Padrão	0.00	1321.2	0.01	678.0	0.00	197.2
Maior	0.02	42982.0	0.27	38641.8	0.23	36687.0
Menor	0.02	39236.8	0.25	36623.1	0.22	36127.6

Tabela 7. Desvios em porcentagem das soluções obtidas através das heurísticas de construção em relação ao valor ótimo do problema Att48.

VMP	IMP	IMD
20.89%	13.34%	9.44%
17.04%	11.92%	9.44%
22.19%	15.27%	7.77%
23.63%	9.25%	9.44%
17.07%	9.40%	8.41%
27.03%	12.36%	9.44%
28.21%	12.97%	9.44%
20.89%	13.34%	9.44%
21.53%	13.34%	9.44%
26.96%	9.67%	9.44%

Na Tabela 8, pode-se observar que a combinação IMD-2opt alcançou o melhor ciclo com uma solução de 34114.4 em 14 iterações iniciando no vértice 6. Já a pior solução equivale a 36453.8, encontrada pelos métodos VMP-2opt em 27 iterações com o início no vértice 10. As de taxas de afastamentos da melhor e da pior solução foram iguais a 1.76% e 8.74% em relação ao valor ótimo existente em literatura, respectivamente. A Tabela 9 mostra os desvios das soluções obtidas através das heurísticas de construção com o mecanismo de busca local 2-opt.

Tabela 8. Resultados obtidos através das heurísticas de construção com 2-opt para o problema Att48.

Instância: Att48									
	VMP-2opt			IMP-2opt			IMD-2opt		
	T(s)	I	Solução	T(s)	I	Solução	T(s)	I	Solução
1	0.43	17	34836.5	0.28	11	35624.0	0.26	9	35090.1
2	0.58	21	35067.8	0.40	9	35763.0	0.22	7	35019.9
3	0.51	16	34734.2	0.17	12	35415.9	0.19	6	35231.0
4	0.43	24	36156.9	0.26	8	35070.5	0.29	11	34849.3
5	0.43	11	35322.3	0.19	7	35325.5	0.18	4	35141.8
6	0.74	27	34258.3	0.21	13	35293.9	0.34	14	34114.4
7	0.37	27	35515.7	0.31	13	35523.7	0.27	8	34784.0
8	0.42	19	34836.5	0.33	13	35624.0	0.25	9	35090.1
9	0.40	19	34836.5	0.33	14	35624.0	0.23	9	35090.1
10	0.62	27	36453.8	0.23	8	35087.2	0.37	9	35090.1
Média	0.49	20.8	35201.8	0.27	10.8	35435.2	0.26	8.6	34950.1
Desvio Padrão	0.12	5.4	676.6	0.07	2.6	237.7	0.06	2.7	322.3
Maior	0.74	27	36453.8	0.40	14	35763.0	0.37	14	35231.0
Menor	0.37	11	34258.3	0.17	7	35070.5	0.18	4	34114.4

Tabela 9. Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o algoritmo 2-opt em relação ao valor ótimo do problema Att48

VMP-2opt	IMP-2opt	IMD-2opt
3.92%	6.27%	4.67%
4.61%	6.68%	4.46%
3.61%	5.64%	5.09%
7.85%	4.61%	3.95%
5.37%	5.37%	4.83%
2.19%	5.28%	1.76%
5.94%	5.97%	3.76%
3.91%	6.27%	6.67%
3.91%	6.27%	4.67%
8.74%	4.66%	4.67%

A Tabela 10 mostra que a combinação entre as heurísticas de construção com a técnica de busca local 3-opt apresentaram boas soluções. Apesar de bons resultados, as metodologias implementadas não foram capazes de encontrar a solução ótima que é igual a 33523.7. A Tabela 11 reforça a eficiência superior desta metodologia em relação às heurísticas de construção com e sem o método de melhorias de rotas 2-opt.

Tabela 10. Resultados obtidos através das heurísticas de construção com 3-opt para o problema Att48.

Instância:	Att48								
	VMP-3opt			IMP-3opt			IMD-3opt		
	T(s)	I	Solução	T(s)	I	Solução	T(s)	I	Solução
1	13.58	18	33927.0	13.06	14	34182.0	12.34	19	33632.1
2	15.82	18	34794.6	13.06	15	34927.2	12.70	10	34038.1
3	14.10	30	34377.7	12.71	21	34555.0	12.49	11	35055.8
4	13.11	38	34701.3	13.81	15	33600.5	12.61	13	34038.1
5	12.94	15	34683.6	13.71	13	34567.8	12.69	9	34281.2
6	12.62	30	35222.6	12.95	15	34489.0	12.41	18	33897.1
7	12.80	32	34845.5	13.09	18	34489.0	12.74	8	35167.9
8	12.86	24	33723.7	12.83	15	34273.2	12.67	19	33632.1
9	12.72	27	33867.4	13.14	14	35316.7	12.35	19	33632.1
10	13.37	38	34064.1	13.03	12	34139.1	12.80	15	34653.6
Média	13.39	27	34420.8	13.14	15.2	34454.0	12.58	14.1	34202.8
Desvio Padrão	0.97	8.2	502.9	0.35	2.6	462.8	0.17	4.5	576.7
Maior	15.82	38	35222.6	13.81	21	35316.7	12.80	19	35167.9
Menor	12.62	15	33723.7	12.71	12	33600.5	12.34	8	33632.1

Tabela 11. Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o mecanismo 3-opt em relação ao valor ótimo do problema Att48.

VPM-3opt	IMP-3opt	IMD-3opt
1.20%	1.96%	0.32%
3.79%	4.19%	1.53%
2.55%	3.08%	4.57%
3.51%	0.23%	1.53%
3.46%	3.11%	2.26%
5.07%	2.88%	1.11%
3.94%	2.88%	4.90%
0.60%	2.24%	0.32%
1.03%	5.35%	0.32%
1.61%	1.84%	3.37%

A Tabela 12 apresenta os dados obtidos pelas as heurísticas de construção, que foram implementadas de forma isoladas utilizando a instância BERLIN52. A solução ótima para este problema que representa 52 localidades de Berlin é igual a 7544.3 de acordo com (MOSER & HENDTLASS, 2005), (LIM *et.al*, 2009) e (VERDEGAY, 2003). A Tabela 13 apresenta os desvios das soluções obtidas através das heurísticas de construção. Pode-se notar nesta Tabela que a performance do algoritmo IMD foi superior em relação às demais abordagens, considerando os desvios encontrados.

Tabela 12. Resultados obtidos através das heurísticas de construção para o problema Berlin52.

Instância: Berlin52						
	VMP		IMP		IMD	
	T(s)	Solução	T(s)	Solução	T(s)	Solução
1	0.03	8980.9	0.32	9004.9	0.28	8262.0
2	0.03	10203.5	0.32	9213.4	0.29	7782.9
3	0.03	9709.4	0.32	9004.9	0.28	7782.9
4	0.03	9458.9	0.31	9083.0	0.29	8260.5
5	0.03	9293.3	0.32	9083.0	0.28	8215.6
6	0.03	9320.2	0.33	9083.0	0.28	8260.5
7	0.03	10200.8	0.31	9213.4	0.29	7782.9
8	0.03	9506.0	0.31	9430.4	0.30	7782.9
9	0.03	9194.1	0.31	9549.3	0.29	7630.5
10	0.03	9114.2	0.31	9549.3	0.29	7630.5
Média	0.03	9498.1	0.32	9221.5	0.29	7939.1
Desvio Padrão	0.00	424.1	0.00	213.5	0.01	273.9
Maior	0.03	10203.5	0.33	9549.3	0.30	8262.0
Menor	0.03	8980.9	0.31	9004.9	0.28	7630.5

Tabela 13. Desvios em porcentagem das soluções obtidas através das heurísticas de construção em relação ao valor ótimo do problema Berlin52

VMP	IMP	IMD
19.04%	19.36%	9.51%
35.25%	22.12%	3.16%
28.70%	19.36%	3.16%
25.38%	20.40%	9.49%
23.18%	20.40%	8.90%
23.54%	20.40%	9.49%
35.21%	22.12%	3.16%
26.00%	25.00%	3.16%
21.87%	26.58%	1.14%
20.81%	26.68%	1.14%

Verifica-se na Tabela 14 que as abordagens VMP-2opt e IMD-2opt apresentaram o valor ótimo existente em literatura para a instância BERLIN52. Sendo que VMP-2opt apresentou em 22 e 24 iterações iniciando nos vértices 10 e 9, respectivamente. Enquanto, que IMD-2opt alcançou a solução com os mesmos vértices, mas em 1 e 2 iterações. Os tempos de processamentos apresentados pelo algoritmo IMD-2opt foram inferiores se comparados com os da metodologia VMP-2opt para determinar o mesmo resultado. Desta forma, pode-se afirmar que IMD-2opt teve uma performance melhor que VMP-2opt. Os desvios obtidos podem ser vistos na Tabela 15.

Tabela 14. Resultados obtidos através das heurísticas de construção com 2-opt para o problema Berlin52.

Instância: Berlin52									
	VMP-2opt			IMP-2opt			IMD-2opt		
	T(s)	I	Solução	T(s)	I	Solução	T(s)	I	Solução
1	0.46	22	7990.1	0.68	21	8008.8	0.13	5	8220.8
2	0.46	20	8009.8	0.70	28	8066.7	0.06	0	7782.9
3	0.44	24	8056.8	0.58	20	8025.9	0.07	1	7782.9
4	0.44	19	7954.6	0.25	5	8886.7	0.11	2	8252.3
5	0.44	32	7887.2	0.24	5	8886.7	0.66	3	8177.5
6	0.43	28	7953.8	0.48	11	8536.0	0.15	3	8176.5
7	0.63	31	8065.5	0.47	21	8160.4	0.06	1	7782.9
8	0.31	19	8158.3	0.80	28	7969.8	0.06	2	7782.9
9	0.81	24	7544.3	0.58	19	8226.7	0.12	1	7544.3
10	0.77	22	7544.3	0.58	18	8226.7	0.12	2	7544.3
Média	0.52	24.1	7916.5	0.54	17.6	8299.4	0.15	2	7904.7
Desvio Padrão	0.16	4.7	209.6	0.18	8.2	349.3	0.18	1.4	276.5
Maior	0.81	32	8158.3	0.80	28	8886.7	0.66	5	8252.3
Menor	0.31	19	7544.3	0.24	5	7969.8	0.06	0	7544.3

Tabela 15. Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o mecanismo 2-opt em relação ao valor ótimo do problema do Berlin52.

VMP-2opt	IMP-2opt	IMD-2opt
5.91%	6.16%	8.97%
6.17%	6.92%	3.16%
6.79%	6.38%	3.16%
5.44%	17.79%	9.38%
4.55%	17.79%	8.39%
5.43%	13.15%	8.37%
6.91%	8.17%	3.16%
8.14%	5.64%	3.16%
0.00%	9.03%	0.00%
0.00%	9.03%	0.00%

As metodologias VMP-3opt e IMD-3opt apresentaram a solução ótima para a instância BERLIN52 em 20% das execuções realizadas. A pior solução dos testes computacionais corresponde a 8241.5 que ficou 9.24% maior que o valor ótimo existente em literaturas, sendo obtida através do IMP-3opt em 27 iterações com o vértice 6.

Os resultados alcançados pelos algoritmos VMP-3opt e IMD-3opt são apresentados na Tabela 16. Já a Tabela 17 mostra os desvios das soluções obtidas através heurísticas de construção com o algoritmo 3-opt.

Tabela 16. Resultados obtidos através das heurísticas de construção com 3-opt para o problema Berlin52.

Instância:	Berlin52								
	VMP-3opt			IMP-3opt			IMD-3opt		
	T(s)	I	Solução	T(s)	I	Solução	T(s)	I	Solução
1	17.42	21	7982.1	16.59	30	7977.2	16.29	9	8114.3
2	16.63	37	7919.4	16.54	24	7979.2	16.86	0	7782.9
3	16.75	23	8053.1	16.56	20	8009.4	16.87	1	7782.9
4	16.38	26	7544.3	16.69	30	8168.4	16.67	8	8114.9
5	16.36	34	8132.0	17.22	30	8179.4	16.80	6	8078.7
6	17.24	32	8041.4	16.40	27	8241.5	16.93	6	8112.0
7	17.24	36	7966.9	16.59	25	7979.2	17.37	1	7782.9
8	16.71	17	8103.5	17.08	30	7860.1	16.90	1	7782.9
9	16.71	25	7661.0	16.47	34	7623.4	17.05	2	7544.3
10	17.48	19	7544.3	16.62	36	7718.3	16.83	2	7544.3
Média	16.89	27	7894.8	16.68	28.6	7973.6	16.86	3.6	7864.0
Desvio Padrão	0.42	7.3	226.1	0.26	4.7	198.4	0.27	3.3	227.0
Maior	17.48	37	8132.0	17.22	36	8241.5	17.37	9	8114.9
Menor	16.36	17	7544.3	16.40	20	7623.4	16.29	0	7544.3

Tabela 17. Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o algoritmo 3-opt em relação ao valor ótimo do problema Berlin52.

VMP-3opt	IMP-3opt	IMD-3opt
5.80%	5.74%	7.56%
4.97%	5.76%	3.16%
6.74%	6.16%	3.16%
0.00%	8.27%	7.56%
7.79%	8.42%	7.08%
6.58%	9.24%	7.52%
5.60%	5.76%	3.16%
7.41%	4.19%	3.16%
1.55%	1.05%	0.00%
0.00%	2.31%	0.00%

Verifica-se na Tabela 18, que a abordagem IMD apresentou a melhor solução dos testes que corresponde a 76.2 para o problema ULYSSES22. No entanto, esta solução ficou 1.20% pior que 75.3 que representa o melhor valor encontrado por (JÚNIOR, 2009) através das metaheurísticas ACO (*Ant Colony Optimization*) e ACO-DM (*Ant Colony Optimization-Data Mining*) e 8.70% maior que 70.1 que representa o valor ótimo para o problema de acordo com a biblioteca TSPLIB. Pode-se constatar por meio da Tabela 19 que a eficiência da metodologia IMD foi superior em relação às performances das demais técnicas.

Tabela 18. Resultados obtidos através das heurísticas de construção para o problema Ulysses22.

Instância:	Ulysses22					
	VMP		IMP		IMD	
	T(s)	Solução	T(s)	Solução	T(s)	Solução
1	0.01	89.6	0.04	81.2	0.03	77.2
2	0.01	88.1	0.04	81.2	0.03	76.9
3	0.01	87.1	0.04	81.2	0.03	76.9
4	0.01	87.8	0.04	81.2	0.03	76.9
5	0.01	89.4	0.04	79.8	0.03	76.2
6	0.01	95.3	0.04	79.8	0.03	77.2
7	0.01	96.2	0.04	79.8	0.03	77.4
8	0.01	88.9	0.04	81.2	0.03	76.7
9	0.01	90.7	0.04	81.2	0.03	77.0
10	0.01	97.7	0.04	80.0	0.03	77.0
Média	0.01	91.1	0.04	80.1	0.03	76.9
Desvio Padrão	0.00	3.8	0.00	0.7	0.00	0.3
Maior	0.01	97.7	0.04	81.2	0.03	77.4
Menor	0.01	87.1	0.04	79.8	0.03	76.2

Tabela 19. Desvios em porcentagem das soluções obtidas através das heurísticas de construção em relação ao valor ótimo do problema Ulysses22.

VMP	IMP	IMD
27.82%	15.83%	10.13%
25.68%	15.83%	9.70%
24.25%	15.83%	9.70%
25.25%	15.83%	9.70%
27.53%	13.84%	8.70%
35.95%	13.84%	10.13%
37.23%	13.84%	10.41%
26.82%	15.83%	9.42%
29.39%	15.83%	9.84%
39.37%	14.12%	9.84%

A Tabela 20 mostra que metodologia IMD-2opt determinou em 5 iterações iniciando no vértice 9 o valor apresentado em (JÚNIOR, 2009) para a instância ULYSSES22, mas seu desempenho foi 60 % pior e 70% melhor dos casos se comparado com as performances dos algoritmos VMP-2opt e IMP-2opt, respectivamente. Na Tabela 21 podem ser vistos os desvios das soluções em relação ao valor ótimo.

Tabela 20. Resultados obtidos através das heurísticas de construção com 2-opt para o problema Ulysses22.

Instância: Ulysses22									
Vértice	VMP-2opt			IMP-2opt			IMD-2opt		
	T(s)	I	Solução	T(s)	I	Solução	T(s)	I	Solução
1	0.05	12	76.1	0.01	3	76.7	0.02	3	76.4
2	0.04	12	75.7	0.02	5	76.7	0.02	3	76.5
3	0.05	16	76.1	0.02	4	76.4	0.02	2	76.5
4	0.03	14	76.0	0.01	4	76.7	0.02	3	76.5
5	0.03	11	76.0	0.01	4	76.1	0.02	3	76.0
6	0.04	15	77.1	0.01	3	76.1	0.02	5	76.1
7	0.04	22	75.9	0.01	2	76.1	0.02	5	76.6
8	0.02	15	75.8	0.02	4	76.7	0.01	3	76.0
9	0.03	12	76.1	0.03	6	76.1	0.02	5	75.3
10	0.02	11	75.9	0.01	3	76.1	0.02	3	75.7
Média	0.04	14	76.1	0.02	3.8	76.4	0.02	3.5	76.2
Desvio Padrão	0.01	3.3	0.4	0.01	1.1	0.3	0.00	1.1	0.4
Maior	0.05	22	77.1	0.03	6	76.7	0.02	5	76.6
Menor	0.02	11	75.7	0.01	2	76.1	0.01	2	75.3

Tabela 21. Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o mecanismo 2-opt em relação ao valor ótimo do problema Ulysses22

VMP-2opt	IMP-2opt	IMD-2opt
8.56%	9.41%	8.99%
7.99%	9.41%	9.13%
8.56%	8.99%	9.13%
8.42%	9.42%	9.13%
8.42%	8.56%	8.42%
9.99%	8.56%	8.56%
8.27%	8.56%	9.27%
8.13%	9.42%	8.42%
8.56%	8.56%	7.42%
8.27%	8.56%	7.99%

Verifica-se na Tabela 22 que os algoritmos VMP-3opt, IMP-3opt e IMD-3opt apresentaram a solução encontrada por (JÚNIOR, 2009) em 40%, 30% e 30% das execuções, respectivamente. No entanto, as abordagens não foram capazes de determinar a solução ótima que corresponde a 70.1. A Tabela 23 apresenta os desvios das soluções em relação ao valor ótimo. A Tabela 24 mostra o tempo de execução médio em segundo necessário para processar cada metodologia. Pode-se notar que a metodologia VMP apresentou os menores tempos, os quais se aproximaram dos tempos determinados analiticamente pela função de complexidade dos algoritmos de construção. Os tempos encontrados de forma analítica para executar os problemas Att48, Berlin52 e Ulysses22 foram iguais a 0.023, 0.027 e 0.05 s, respectivamente. Considerando que cada operação elementar no algoritmo é executada em um decimo de microssegundo. Os melhores resultados obtidos estão resumidos na Tabela 25.

Tabela 22. Resultados obtidos através das heurísticas de construção com 3-opt para o problema Ulysses22.

Instância: Ulysses22									
	VMP-3opt			IMP-3opt			IMD-3opt		
	T(s)	I	Solução	T(s)	I	Solução	T(s)	I	Solução
1	0.89	19	75.3	0.92	11	75.3	0.89	7	75.3
2	0.92	19	75.7	0.94	4	76.1	0.91	5	76.0
3	0.90	14	75.6	0.93	10	75.3	0.90	4	76.0
4	0.92	22	75.4	0.96	5	75.7	0.90	4	75.7
5	0.89	11	75.3	0.95	6	75.9	0.88	5	75.9

Tabela 22. Continuação.

6	0.91	16	76.0	0.94	4	75.3	0.87	7	76.2
7	0.90	25	75.3	0.94	7	75.7	0.88	10	75.3
8	0.90	24	75.3	0.92	5	76.7	0.89	5	75.7
9	0.90	15	75.9	0.98	10	75.9	0.91	4	75.3
10	0.92	19	77.2	0.96	6	75.4	0.90	3	75.7
Média	0.91	18.4	75.7	0.94	6.8	75.7	0.89	5.4	75.7
Desvio Padrão	0.01	4.5	0.6	0.02	2.6	0.4	0.01	2.1	0.32
Maior	0.92	25	77.2	0.98	11	76.7	0.91	10	76.2
Menor	0.89	11	75.3	0.92	4	75.3	0.87	3	75.3

Tabela 23. Desvios em porcentagem das soluções obtidas através das heurísticas de construção com o algoritmo 3-opt em relação ao valor ótimo do problema Ulysses22.

VMP-3opt	IMP-3opt	IMD-3opt
7.42%	7.42%	7.42%
7.99%	8.56%	8.42%
7.85%	7.42%	8.42%
7.56%	7.99%	7.99%
7.42%	8.27%	8.27%
8.42%	7.42%	8.70%
7.42%	7.99%	7.42%
7.42%	9.42%	7.99%
8.27%	8.27%	7.42%
10.13%	7.56%	7.99%

Tabela 24. Representa os tempos médios obtidos nos testes computacionais.

Instâncias	VMP	IMP	IMD	VMP	IMP	IMD	VMP	VMP	VMP
	2opt	2opt	2opt	2opt	2opt	2opt	3opt	3opt	3opt
	T(s)	T(s)	T(s)	T(s)	T(s)	T(s)	T(s)	T(s)	T(s)
Att48	0.02	0.25	0.22	0.49	0.27	0.26	13.39	13.14	12.58
Berlin52	0.03	0.32	0.29	0.52	0.54	0.15	16.89	16.68	16.83
Ulysses22	0.01	0.04	0.03	0.04	0.02	0.02	0.91	0.94	0.89

Tabela 25. Os melhores resultados obtidos.

Instâncias	Heurística	Vértice	T(s)	I	Solução encontrada	Solução conhecida
Att48	IMP-3opt	4	13.81	15	33600.5	33523.7
Berlin52	IMD-3opt	9	17.05	2	7544.3	7544.3
Ulysses22	IMD-3opt	1	0.89	7	75.3	70.1

A Tabela 26 mostra soluções ainda melhores do que as apresentadas neste trabalho para as instâncias Att48, Berlin52 e Ulysses22 determinadas pela implementação do VNS (*Variable Neighborhood Search*) com os mecanismos k -opt (BINGÜLER & BULKAN, 2015).

Tabela 26. Resultados obtidos através do VNS com os mecanismos k -opt. Fonte: BINGÜLER & BULKAN, 2015.

Problema	Desempenho em relação à solução conhecida		Solução encontrada	Solução conhecida	Tempo médio da CPU (segundos)
	Min	Max			
Att48	0	0.006	10648	10648	42.266
Berlin52	0	0	7542	7542	45.250
Ulysses22	0	0	7013	1073	25.047

5 CONCLUSÕES

Neste presente trabalho foram implementadas no ambiente Matlab metodologias que buscam por soluções aproximadas para resolver o PCV. Verificou-se através dos dados apresentados que o algoritmo IMD apresentou os melhores resultados dos testes computacionais para os problemas abordados se comparados com os de outros métodos de construção analisados. As soluções encontradas foram aperfeiçoadas por meio das técnicas de busca local (2-opt e 3-opt). Os melhores resultados foram apresentados tanto através da combinação da abordagem IMP com a técnica de busca local 3-opt com taxa de afastamento de 0.23% em relação ao valor ótimo da instância Att48 como por meio da combinação do algoritmo IMD com o mecanismo 3-opt com desvios de 0% e 7.42% em relação ao valor ótimo das instâncias abordadas (Berlin52 e Ulysses22), respectivamente. A eficiência da heurística 3-opt foi superior se comparada com a do método 2-opt, devido o ligeiro aumento de execução do algoritmo 3-opt.

As heurísticas de construção (vizinho mais próximo e inserção do mais distante) com os mecanismos de busca local (2-opt e 3-opt) determinaram o valor ótimo existente em literaturas para o problema Berlin52 e isto mostra que as heurísticas de melhorias abordadas possuem grande potencialidade na solução de problemas dessa natureza, apesar da simplicidade de formulação e implementação das técnicas.

Para sugestões de trabalhos futuros, pode-se utilizar mecanismos eficientes para resolução de forma exata o PCV, tais como: ILS (*Iterated Local Search*) (STÜTZLE, 2006) e VNS com os mecanismos k -opt ou aprimorar as técnicas apresentadas neste trabalho.

REFERÊNCIAS

- APPLEGATE, D.L et al. 1995. Finding Cuts in the TSP (A preliminary report), Technical Report 95-05, DIMACS.
- APPLEGATE, D.L et al. 1998. On the Solution of Traveling Salesman Problems, Documental Mathematica, v. Extra Volume ICM III, p.645-656.
- APPLEGATE, D.L et al. 2001. TSP Cuts Which Do Not Conform to the Template Paradigm, Computational Combinatorial Optimization ,Jünger, M. e D. Naddef, eds., Springer, 261-304.
- APPLEGATE, D.L et al. 2004. Optimal tour Sweden. Disponível em: <http://www.tsp.gatech.edu/sweden>. Acesso em: 25 de maio de 2015.
- APPLEGATE, D.L et al. 2006. Concorde tsp solver. Disponível em: <http://www.tsp.gatech.edu/concorde>. Acesso em: 30 de junho de 2015.
- AVNER, P et al. 2001. A radiation hybrid transcript map of the mouse genome. Nature Genetics 29: 194–200.
- BENEVIDES, P.F. 2001. Aplicação de heurísticas e meta-heurísticas para o problema do caixeiro viajante em um problema real de roteirização de veículos. Dissertação (Mestrado). UFP. Curitiba.
- BINGÜLER, A.H.E; BULKAN, S. 2015. New variable neighborhood search structure for travelling salesman problems. In: British journal of mathematics & computer science 6(5): 422-434.
- BLAND, R. E; SHALLCROSS, D. F. 1987. Large Traveling Salesman Problem Arising from Experiments in X-ray Crystallography: a Preliminary Report on Computation, Relatório Técnico No. 730, School of OR/IE, Cornell University, Ithaca, New York.
- BOAVENTURA NETO, P.O. 2006. Grafos: Teoria, Modelo, Algoritmo. 4ª ed. Edgar. Blucher. São Paulo.
- CAIXEIRO. Disponível em: <http://www.mat.ufrgs.br/~portosil/caixeiro.html>. Acesso em 9 de maio de 2015.
- CAMERINI, P.M; FRATTA, L; MAFFIOLI, F. 1975. On improving relaxation methods by modified gradient techniques, Mathematical Programming Study, v.3, 26-34.

- CAMPELLO, R. E; MACULAN, N. 1994. Algoritmos e Heurísticas. Desenvolvimento e Avaliação de Desempenho. Editora. UFF. Rio de Janeiro.
- CHRISTOFIDES, N; MINGOZZI, A; TOTH, P. 1979. The vehicle routing problem, In: Combinatorial Optimization. John Wiley & Sons, 315-338, New Jersey.
- CROES, G. 1958. A method for solving travelling salesman problems. Operations Research, v. 6, p. 791-812.
- CROWDER, H; PADBERG, M.W. 1980. Solving Large-Scale Symmetric Traveling Salesman Problems to Optimality, Management Science, v. 26, p. 495-509.
- DANTZIG, G.B; FULKERSON, D.R; JOHNSON, S.M. 1954. Solution of a Large-Scale Traveling Salesman Problem, Operations Research, v. 2, p.393-410.
- DORIGO, M. e STÜTZLE, T. 2004. Ant Colony Optimization. MIT Press.
- GAREY, M. R; JOHNSON, D. S. 1979. Computers and Intractability. A Guide to the Theory of NP-completeness. W.H. Freeman and Company, San Francisco.
- GERSTING, J. L. 1993. Fundamentos Matemáticos para a Ciência da Computação. 2ª ed. LTC – Livros Técnicos e Científicos. S.A. Rio de Janeiro.
- GOLDBARG, M. C.; LUNA, H. P. L. 2005. Otimização combinatória e programação linear- Modelos e Algoritmos. 2ª ed. Elsevier. Rio de Janeiro.
- GOMES, R.F.S. 2001. Aplicação da metaheurística tabu search na otimização de rotas de manutenção preventiva em campo. Dissertação (Mestrado). UFC. Fortaleza.
- GONNET, G.; KOROTENSKY, C.; BENNER, S. 2000. Evaluation measures of multiple sequence alignments. In: Journal of Computational Biology 7(1-2), 261-276.
- GRÖTSCHEL, M.1980. On the Symmetric Traveling Salesman Problem: Solution of a 120-city problem, Mathematical Programming Studies, v.12, p.61-77.
- GRÖTSCHEL, M; HOLLAND, O.1991. Solution of Large- Scale Symmetric Traveling Salesman Problems, Mathematical Programming, v. 51, p.141-202.
- HELGAUN, K. 2000. An effective implementation of the Lin-Kernighan Traveling Salesman Heuristic, European Journal of Operational Research, v.126, p.106-130.
- HELD, M; KARP, R.M. 1962. A dynamic programming approach to sequencing problems. Journal of the Society of Industrial and Applied Mathematics 10, 196-210.

HOFFMAN, K.; PADBERG, M. Traveling Salesman Problem. Karla Hoffman home page, Encyclopedia Articles. Disponível em: http://iris.gmu.edu/%7Ekhoffman/papers/trav_saleman.htm. Acesso em 20 de junho de 2015.

http://www.siscorp.com.br/siscorpnews/sexta_edicao/assunto_marco.html. Acesso em 10 de junho de 2015.

<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95>. Acesso em 18 de agosto de 2015.

<http://www.math.uwaterloo.ca/tsp/history/>. Acesso em 4 de agosto de 2015.

JÚNIOR, R.R. 2009. Um método de solução baseado na metaheurística colônia de formigas e uma técnica de mineração de dados para resolução do problema do caixeiro viajante. Relatório final da disciplina Princípios e Aplicações de Mineração de Dados, INPE, São José dos Campos.

KANDA, J. Y. 2012. Uso de meta-aprendizado na recomendação de metaheurísticas para o problema do caixeiro viajante. Dissertação (Doutorado). Instituto de Ciências Matemáticas e de Computação. USP. São Paulo.

LAWLER, E.L et al.1985. The traveling salesman problem: a guided tour of combinatorial optimization. 1ªed.John Wiley & Sons, New York.

LIM, C.P; JAIN, L.C; DEHURI. S. 2009. Innovations in Swarm Intelligence. Springer Verlag Ny, p.48.

LIN, S. 1965. Computer solution of the traveling salesman problem. In: Bell System. Technical. Journal. n. 44, p. 2245-2269.

MENGER, K. 1932. Das Botenproblem. In: K. Menger, ed., Ergebnisse eines Mathematischen Kolloquiums 2, 11–12.

MOLE, V.L.D. 2002. Algoritmos Genéticos-Uma abordagem paralela baseada em populações cooperantes. Dissertação (Mestrado). UFSC. Florianópolis.

MORAIS, J.L.M.2010. Problema do Caixeiro Viajante Aplicado ao Roteamento de Veículos numa Malha Viária. Trabalho de conclusão de curso. UNIFESP. São José dos Campos – SP.

MOSER, I; HENDTLASS, T. 2005. Improving the Performance of Ant Colony Optimization Algorithms Using Biased Visibility. Complexity International, v.12. Disponível em: <http://www.ict.swin.edu.au/personal/imoser/publications/biassedvisibility.pdf>.

NAGAMALAI, D; KUMAR, A; ANNAMALAI, A. 2013. Advances in Computational Science, Engineering and Information Technology. Proceedings of the Third International Conference on Computational Science, Engineering and Information Technology (CCSEIT-2013), KTO Karatay University, June 7-9. Springer, v.1.p.117. Konya, Turkey.

NURIYEVA, F; KIZILATES, G; BERBERLER, M.E. 2012. Experimental Analysis of New Heuristics for the TSP. IV International Conference "Problems of Cybernetics and Informatics" (PCI2012), September 12-14. Disponível em: <http://www.pci2012.science.az/7/22.pdf> 4848. Acesso em 21 de setembro de 2015.

ORNELAS, C et al. 2006. Problemas *NP*-Completo e Algoritmos Aproximados. Disponível em: <http://www2.dcc.ufmg.br/livros/algoritmos-java/cap9/transp/completo4/cap9.pdf>. Acesso em 20 de agosto de 2015.

PADBERG, M; RINALDI, G.1987. Optimization of a 532-city Symmetric Traveling Salesman Problem by Branch and Cut, Operations Research Letters, v.6, p.1-7

PADBERG, M; RINALDI, G. 1991. A Branch-and- cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems, SIAM Review, v.33, p.60-100.

PRESTES, A.N. 2006. Uma análise experimental de abordagens Heurísticas aplicadas ao problema do caixeiro viajante. Dissertação (Mestrado). UFRGN. Natal/RS.

REEVES, C. R. 1995. Modern Heuristic Techniques for Combinatorial Problems, McGraw-Hill, London.48.

SILVA, B. C. H. 2013. Otimização de rotas utilizando abordagens heurísticas em um ambiente georreferenciado. Dissertação (Mestrado). UEC. Fortaleza.

SILVEIRA, J. F. Porto da. 2000. Problema do caixeiro viajante. Porto Alegre: UFRGS. Disponível em: <http://www.mat.ufrgs.br/portosil/caixeiro.html>. Acesso em: 26 de Março de 2015.

SILVA, A. F.; OLIVEIRA, A. C. de. 2006. Algoritmos genéticos: alguns experimentos com os operadores de cruzamento ("Crossover") para o problema do caixeiro viajante assimétrico. Anais do ENEGEP-Fortaleza, CE, Brasil.

SOLOMON, M. M. 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. Operations Research, v. 35, p. 254-265.

STÜTZLE, T. 2006. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, v. 174, n. 3, p. 1519-1539.

VERDEGAY, J.L. 2003. *Fuzzy Sets Based Heuristics for Optimization*. Springer Verlag, p.74.

VITTES, F. J. 1999. *Optimizing the performance of a chip shooter machine*. Faculty of the Virginia Polytechnic Institute and State University. Dissertação (Master of Science Industrial and Systems Engineering).

APÊNDICE

Tabela 1. Tabela que representa os tempos determinados pela função de complexidade do vizinho mais próximo, considerando que cada operação elementar no algoritmo é executada em um decimo de microssegundo.

Função	ATT48	BERLIN52	ULYSSES22
N^2	0.023s	0.027s	0.05s

Tabela 2. Tabela que representa os tempos determinados pela função de complexidade da inserção do mais próximo, considerando que cada operação elementar no algoritmo é executada em um decimo de microssegundo.

Função	ATT48	BERLIN52	ULYSSES22
N^2	0.023s	0.027s	0.05s

Tabela 3. Tabela que representa os tempos determinados pela função de complexidade da inserção do mais distante, considerando que cada operação elementar no algoritmo é executada em um decimo de microssegundo.

Função	ATT48	BERLIN52	ULYSSES22
N^2	0.023s	0.027s	0.05s