

GRASP PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MULTI-COMPARTIMENTOS E RESTRIÇÃO DE JANELA DE TEMPO

FABRÍCIO DA SILVEIRA LISBOA

“Dissertação apresentada ao Centro de Ciências e Tecnologia, da Universidade Estadual do Norte Fluminense, como parte das exigências para obtenção do título de Mestre em Engenharia de Produção”.

Orientador: prof. Geraldo Galdino de Paula Jr., D.Sc.

CAMPOS DOS GOYTACAZES – RJ

Fevereiro - 2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

GRASP PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MULTI-COMPARTIMENTOS E RESTRIÇÃO DE JANELA DE TEMPO

FABRÍCIO DA SILVEIRA LISBOA

“Dissertação apresentada ao Centro de Ciências e Tecnologia, da Universidade Estadual do Norte Fluminense, como parte das exigências para obtenção do título de Mestre em Engenharia de Produção”.

Aprovada em : _____

Comissão Examinadora:

Prof. Geraldo Galdino de Paula Jr. D.Sc., UENF (Presidente)

Prof. Carlos Leonardo Ramos Póvoa D.Sc., UENF

Prof. Rodrigo Tavares Nogueira D.Sc., UES

Dr. Euclides Vieira Neto D.Sc., ISECENSA

DEDICATÓRIA

A Minha Esposa Ana
Paula, a minha Filha Laura e a Meus Pais Francisco e Elena.

AGRADECIMENTOS

A Deus que me ajudou e me iluminou nos momentos de dificuldade e dúvidas, semeando sempre em meu coração o desejo de prosseguir.

A minha esposa pelo incentivo, paciência e compreensão, me dando força e estímulo nos momentos de dúvidas e dificuldades.

A minha mãe pela incansável luta de buscar sempre para mim um ensino de qualidade.

A meu orientador, professor Geraldo Galdino, pela grande amizade, incentivo, compreensão e dedicação fundamental para a conclusão deste trabalho.

Ao meu professor de graduação Carlos Leonardo Póvoa pelo grande apoio para ingresso no mestrado, na elaboração dos algoritmos e na disponibilização dos códigos que foram readaptados para esta pesquisa.

A meus amigos, colegas de mestrado e funcionários do CCT da UENF, pela amizade e apoio.

A todos os professores do Laboratório de Engenharia de Produção, especialmente a André Luís Policani e a Rodrigo Tavares, pelas palavras de incentivo.

Aos profs. Carlos Leonardo Ramos Póvoa, Euclides Vieira Neto e Rodrigo Tavares Nogueira pela presença na banca examinadora.

E a todos aqueles que, de uma forma ou de outra, contribuíram para a conclusão deste trabalho.

Sumário

Lista de Figuras	IX
Lista de Tabelas	XI
Resumo	XIII
Abstract	XIV
Capítulo 1 - Introdução	15
1.1 - Objetivo	17
1.2 - Organização da Tese	18
Capítulo 2 – Problema de Roteamento de Veículos e sua Extensões	19
2.1 – Classificação dos Problemas de Roteirização de Veículos	23
2.2 – Problema de Roteamento de Veículos com Restrição de Janela de Tempo (PRVJT)	24
Capítulo 3 – Estratégias e Métodos Heurísticos de Soluções	28
3.1 – A Heurística de Clark e Wright	29
3.2 – A Heurística de Gillet – Miller (Método da Varredura)	30
3.3 – Heurística de Pétalas	31
3.4 – Algoritmo Primeiro Agrupa, Segundo Roteiriza	31
3.5 – Método das Inserções	31
3.6 – Heurísticas de Melhoria	32
Capítulo 4 – Estratégias e Métodos Metaheurísticos de Soluções	33
4.1 – Simulated Annealing (SA)	34
4.2 – Busca Tabu (BT)	35
4.3 – Algoritmos Genéticos (AG)	35
4.4 – Ant Colonies (AC)	37

4.5 – Busca em Vizinhança Variável (BVV)	38
4.6 – Algoritmos Meméticos (AM)	38
4.7 – Greedy Randomized Adaptive Search Procedures - GRASP	39
Capítulo 5 – Fecho Convexo (Convex Hull)	41
5.1 – Algoritmo de Jarvis	41
5.2 – Algoritmo de Quickhull	42
5.3 – Algoritmo de Mergehull	42
5.4 – Algoritmo de Graham	42
Capítulo 6 – O Problema de Roteamento de Veículos com Multi-compartimentos e Restrição de Janela de Tempo (PRVMCJT)	45
6.1 – Metaheurística GRASP Aplicada no Problema de Roteamento de Veículos com Multi-compartimentos com Restrição de Janela de Tempo	48
6.1.1 – Inicialização de Rotas	48
6.1.2 – Procedimento Construir Solução	50
6.1.3 – Inserção de Clientes	50
6.1.4 – Condições de Viabilidade	52
6.1.5 – Procedimento de Busca Local	54
Capítulo 7 – Implementação Computacional e Testes	55
7.1 – Implementação Computacional	55
7.2 – Testes Computacionais	56
7.1.1 – Testes Utilizando Algoritmos Exatos	56
7.1.2 – Testes com Instâncias Aleatórias – Dois Compartimentos	57
7.1.3 – Testes com Instâncias Aleatórias – Três Compartimentos	68
7.1.4 – Testes com Instâncias Aleatórias	

– Quatro Compartimentos	78
7.1.5 – Testes com Instância Maior Aleatória – Dois Compartimentos	89
Capítulo 8 – Conclusão	92
8.1 – Conclusões	92
8.2 - Sugestões para Trabalhos Futuros	93
REFERÊNCIAS BIBLIOGRÁFICAS	94

LISTA DE FIGURAS

Figura 2.1	- Problema de Roteamento de Veículos	20
Figura 2.2	- Modelo matemático PRV	21
Figura 3.1	- Método das economias	30
Figura 4.1	- Pseudocódigo da fase de construção	40
Figura 4.2	- Pseudocódigo da fase de melhoria	40
Figura 5.1	- Algoritmo básico de Graham	43
Figura 5.2	- Funcionamento do algoritmo de Graham	44
Figura 6.1	- Modelo matemático (PRVCMJT)	47
Figura 6.2	- Procedimento de inicialização	49
Figura 6.3	- Procedimento construir solução	52
Figura 6.4	- Busca local	54
Figura 7.1	- Tela principal do programa	56
Figura 7.2	- Gráfico Comparativo – Iterações	59
Figura 7.3	- Gráfico Comparativo – Lista de Candidatos Restritos	60
Figura 7.4	- Gráfico Comparativo – Parâmetros $\delta_1, \delta_2, \delta_3$ e δ_4	61
Figura 7.5	- Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_10)	62
Figura 7.6	- Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_20)	63
Figura 7.7	- Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_50)	63
Figura 7.8	- Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_100)	64
Figura 7.9	- Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_120)	64
Figura 7.10	- Variação da função objetivo – L.C.R – Inst_1_10 Clientes	65
Figura 7.11	- Variação da função objetivo – L.C.R – Inst_1_20 Clientes	66
Figura 7.12	- Variação da função objetivo – L.C.R – Inst_1_50 Clientes	66
Figura 7.13	- Variação da função objetivo – L.C.R – Inst_1_100 Clientes	67
Figura 7.14	- Variação da função objetivo – L.C.R – Inst_1_120 Clientes	67
Figura 7.15	- Gráfico Comparativo – Iterações	70
Figura 7.16	- Gráfico Comparativo – Lista de Candidatos Restritos	71
Figura 7.17	- Gráfico Comparativo – Parâmetros $\delta_1, \delta_2, \delta_3$ e δ_4	72
Figura 7.18	- Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_10)	73
Figura 7.19	- Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_20)	73

Figura 7.20 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_50)	74
Figura 7.21 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_100)	74
Figura 7.22 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_120)	75
Figura 7.23 - Variação da função objetivo – L.C.R – Inst_2_10 Clientes	76
Figura 7.24 - Variação da função objetivo – L.C.R – Inst_2_20 Clientes	76
Figura 7.25 - Variação da função objetivo – L.C.R – Inst_2_50 Clientes	77
Figura 7.26 - Variação da função objetivo – L.C.R – Inst_2_100 Clientes	77
Figura 7.27 - Variação da função objetivo – L.C.R – Inst_2_120 Clientes	78
Figura 7.28 - Gráfico Comparativo – Iterações	81
Figura 7.29 - Gráfico Comparativo – Lista de Candidatos Restritos	82
Figura 7.30 - Gráfico Comparativo – Parâmetros $\delta_1, \delta_2, \delta_3$ e δ_4	83
Figura 7.31 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_10)	84
Figura 7.32 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_20)	84
Figura 7.33 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_50)	85
Figura 7.34 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_100)	85
Figura 7.35 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_120)	86
Figura 7.36 - Variação da função objetivo – L.C.R – Inst_3_10 Clientes	87
Figura 7.37 - Variação da função objetivo – L.C.R – Inst_3_20 Clientes	87
Figura 7.38 - Variação da função objetivo – L.C.R – Inst_3_50 Clientes	88
Figura 7.39 - Variação da função objetivo – L.C.R – Inst_3_100 Clientes	88
Figura 7.40 - Variação da função objetivo – L.C.R – Inst_3_120 Clientes	89
Figura 7.41 - Variação da função objetivo – L.C.R – Inst_4_534 Clientes	90
Figura 7.42 - Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_4_534)	91

LISTA DE TABELAS

Tabela 2.1 - Classificação do PRV	24
Tabela 3.1 - Estimativa do tempo de solução do PCV	28
Tabela 7.1 - Resultados algoritmo exato	57
Tabela 7.2 - 50 Iterações	58
Tabela 7.3 - 100 Iterações	58
Tabela 7.4 - 200 Iterações	58
Tabela 7.5 - 500 Iterações	58
Tabela 7.6 - Resultados Médios - Iterações	59
Tabela 7.7 - Resultados Médios – L.C.R	60
Tabela 7.8 - Resultados Médios	61
Tabela 7.9 - Avaliação dos parâmetros δ_1 , δ_2 , δ_3 e δ_4 (Inst_1)	61
Tabela 7.10 - Resultados Médios L.C.R – Inst_1	65
Tabela 7.11 - 50 Iterações	68
Tabela 7.12 - 100 Iterações	69
Tabela 7.13 - 200 Iterações	69
Tabela 7.14 - 500 Iterações	69
Tabela 7.15 - Resultados Médios - Iterações	70
Tabela 7.16 - Resultados Médios – L.C.R	71
Tabela 7.17 - Resultados Médios	71
Tabela 7.18 - Avaliação dos parâmetros δ_1 , δ_2 , δ_3 e δ_4 (Inst_2)	72
Tabela 7.19 - Resultados Médios L.C.R – Inst_2	75
Tabela 7.20 - 50 Iterações	79
Tabela 7.21 - 100 Iterações	79
Tabela 7.22 - 200 Iterações	79
Tabela 7.23 - 500 Iterações	80
Tabela 7.24 - Resultados Médios - Iterações	80
Tabela 7.25 - Resultados Médios – L.C.R	81
Tabela 7.26 - Resultados Médios	82
Tabela 7.27 - Avaliação dos parâmetros δ_1 , δ_2 , δ_3 e δ_4 (Inst_3)	83
Tabela 7.28 - Resultados Médios L.C.R – Inst_3	86
Tabela 7.29 - Resultados Médios L.C.R – Inst_4	90

GRASP PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MULTI-COMPARTIMENTOS E RESTRIÇÃO DE JANELA DE TEMPO

Fabrício da Silveira Lisboa

27 de Fevereiro de 2007

Resumo

A crescente complexidade do mercado e o aumento de demanda dos clientes são fatores importantes que impulsionam as empresas na procura de soluções que compatibilizem o baixo custo com a qualidade de produtos e serviços. As operações de logística, em especial a entrega e coleta de mercadorias, têm se mostrado bastante promissoras para a obtenção desta conformação. Na entrega e coleta de mercadorias aparece um problema conhecido como problema de roteamento de veículos onde os modelos matemáticos, juntamente com as soluções computacionais, têm se mostrado satisfatórios no trato com esse tipo de engenharia logística. Dentre os diversos problemas de roteamento de veículos existentes surgem, na prática, aqueles de empresas que utilizam veículos com mais de um compartimento para o serviço de entrega de mercadorias. Devem ser citadas as empresas de distribuição de combustível e as de entrega de alimentos que podem ter o seu transporte feito em veículos que possuem divisões de temperaturas de acordo com o tipo de produto. Essas empresas também podem possuir restrição temporal para recebimento de veículos. Estas duas condições associadas criam o problema de roteamento de veículos multi-compartimentados com restrição de janela de tempo. Nesta pesquisa a solução deste problema é alcançada através de um algoritmo heurístico GRASP (Greedy Randomized Adaptive Search Procedure).

Palavras Chaves: Roteamento de Veículos, GRASP, Janela de tempo, Heurística.

Orientador: Prof. Geraldo Galdino de Paula Jr.

GRASP FOR THE VEHICLE ROUTING PROBLEM WITH MULTI-COMPARTMENTS AND TIME WINDOWS CONSTRAINTS

Fabrício da Silveira Lisboa

February 27, 2007

Advisor: Prof. Geraldo Galdino de Paula Jr.

Abstract

The growing complexity of the market and the increasing level of customer's demand are important factors impeling the companies to look for solutions that conjoin low cost with quality of products and services. The logistic services, and especially, the delivery and collect of goods, have shown quite promising for obtaining this conformation. In the delivery and collect of goods appears a problem known as vehicle routing problem, where the mathematical models allied to computer solutions has been showing satisfactory in dealing with this kind of Logistic engineering. Among several vehicle routing problems that appear in practice, exist those of companies that use vehicles with more than one compartment for the delivery service. The companies of fuel distribution should be mentioned along with those of food delivery that can have its transport done in vehicles possessing temperatures divisions in agreement with the product type. Those companies can also possess a time window restriction for vehicles reception. These two conditions associated create the vehicle routing problem with multi-compartment and time window restrictions. In this research the solution for this problem is achieved through a heuristic algorithm GRASP (Greedy Randomized Adaptive Search Procedure).

Key Words: Vehicle Routing Problem, GRASP, Time Window, Heuristic.

1. INTRODUÇÃO

Os critérios de competitividade do mercado estão cada vez mais complexos e em permanente mudança. Os clientes têm se tornado cada vez mais exigentes quanto à qualidade, e quanto a prazos de entrega. Estas características de mercado impõem às empresas uma cultura dinâmica que possibilite sua sobrevivência e proporcione a diferenciação de seus serviços e produtos em paralelo à redução de custos.

Ballou (1993) analisa o problema de logística destacando sua importância numa escala global. As empresas, obrigatoriamente, passam a inserir no seu planejamento estratégico políticas e ações que contemplam melhorias e diferenciações na logística de distribuição.

Segundo Christopher (1997), “logística é o processo de gerenciar estrategicamente a aquisição, movimentação e armazenagem de materiais, peças e produtos acabados, por meio da organização e de seus canais de marketing, de modo a maximizar a lucratividade presente e futura, com o atendimento dos pedidos a baixo custo”.

Aspectos econômicos e sociais impulsionaram as pesquisas ligadas à melhoria dos processos logísticos. A estabilização econômica mudou o posicionamento das empresas com relação à obtenção de lucro. Na ausência de índices de inflação exorbitantes, os empresários viram desaparecer a oportunidade de obter a parcela de lucros com procedimentos financeiros. A eficiência operacional, competitividade e posteriormente a flexibilidade passaram a ser imprescindível para o lucro das empresas e principalmente para a sua sobrevivência.

Outra consequência relevante em relação à ausência de altas taxas de inflações é que os clientes passam a ter outra postura para formatação dos pedidos de compras. Os pedidos de compra antes em grandes lotes de produtos e com pouca frequência passaram a pequenos lotes e com elevado número de pedidos. Isso faz com que aumente também o número de visitas para entrega de mercadorias aos clientes. Soma-se a este novo perfil consumidor o aumento da exigência dos clientes quanto ao local e horário de entrega.

Esta nova conjuntura do mercado causa um impacto significativo nas operações e nos custos associados aos sistemas de distribuição, levando as empresas a incrementar suas rotinas operacionais com redução dos custos associados.

A diferenciação no mercado e redução de custos proporciona um desempenho melhor de determinada empresa em relação aos seus concorrentes. Neste contexto o bom gerenciamento da logística de distribuição assume um papel de relevância, uma vez que reverterá para o cliente melhorias no serviço com diferenciação e agilidade. Com o nível atual de concorrência, onde as empresas geralmente buscam em primeiro lugar a redução dos custos do que repassar o aumento para o produto final, o aproveitamento de todas as possibilidades de redução nos custos se torna, para muitas delas, um fator crítico para a sobrevivência.

O nível de serviço logístico passou a ser mais um diferenciador no mercado. Com isso a logística tem mais um objetivo, agregar valor ao produto através do serviço oferecido. Reduzir prazo de entrega, aumentar a disponibilidade de produtos, efetuar entregas com horas determinadas, cumprimento dos prazos, são algumas exigências do mercado que podem ser atendidas pelo serviço logístico. A importância de cada exigência varia de acordo com o perfil de cada cliente, uma vez que as suas necessidades são diferenciadas. Então cabe a cada empresa equacionar a redução de custos com o melhoramento do nível de serviços oferecidos, observando as especificidades de sua carteira de clientes. Fica evidente que a atividade logística precisa ser incorporada na estratégia de negócio da empresa.

Segundo Ballou (1993, p. 388) o problema enfrentado pela logística é “diminuir o hiato entre a produção e a demanda, de modo que os consumidores tenham bens e serviços quando e onde quiserem, e na condição física que desejarem”. As metas da logística são as de dispor o produto certo, na quantidade certa, no local certo, no período certo, nas condições adequadas para o cliente certo com o menor custo possível. Assim, é evidente a necessidade de se atingir, simultaneamente, a eficiência e a eficácia nesse processo.

Os custos envolvidos na distribuição de produtos costumam ser elevados, Lima (2006) estima em 12,6% do PIB o custo logístico brasileiro, sendo que deste montante, aproximadamente 60% são gastos com o transporte. Estes números justificam a crescente atenção dada pelas empresas para o problema de roteamento.

Para dispor de forma ótima os produtos em tempo e local desejado pelos clientes, as empresas necessitam solucionar o problema básico de distribuição. Onde tem que se definir o percurso do veículo de forma a atender o maior número

possível de clientes a um custo operacional mínimo. Este problema é conhecido como problema de roteamento de veículos. Um dos principais objetivos de tal problema é a determinação de rotas com custo mínimo para os veículos de uma frota. Este objetivo, em geral, é complexo e difícil de ser alcançado, devido às restrições operacionais que dependem de fatores como, a natureza e características dos itens a serem transportados ou da qualidade com que o serviço deve ser executado.

O Problema de Roteamento de Veículos tem recebido bastante atenção dos pesquisadores devido a sua potencialidade em obter diferenciação nos serviços logísticos em contrapartida com a redução de custos operacionais. O planejamento de rotas para frotas de veículos encontra na prática uma enorme aplicação para apoio à tomada de decisão no que se refere à distribuição física dos produtos.

Os pesquisadores procuram uma solução matemática a mais próxima possível da prática encontrada dentro das empresas. Existem empresas que utilizam mais de um depósito para armazenamento e distribuição de produtos, ou ainda que possuam frota de veículos com mais de um compartimento, ou então empresas que possuem clientes com restrição de horário para o recebimento de mercadorias. Enfim, a inserção destas restrições tem levado a soluções cada vez melhores e mais próximas da realidade empresarial.

Uma situação prática surge, também, com a junção das duas últimas variáveis citadas no parágrafo anterior. Estes casos são observados em empresas que fazem a distribuição de combustíveis, ou a distribuição de diferentes tipos de alimentos. Em geral, empresas assim necessitam de uma frota de veículos com mais de um compartimento e ainda precisam adequar sua distribuição de produtos a restrição temporal dos clientes.

1.1 OBJETIVO

O objetivo principal a que se propõe esta pesquisa é o de contribuir para o desenvolvimento de processos que otimizem os serviços ligados à logística de distribuição. Isto foi possível a partir da elaboração de uma heurística GRASP para o Problema de Roteamento de Veículos com Multi-compartimentos e Restrição de Janela de Tempo (PRVMCJT).

Objetivos intermediários:

- Desenvolver um procedimento baseado na metaheurística GRASP, que possibilite a obtenção de soluções satisfatórias ao PRVMC e Restrição de Janela de Tempo.
- Agregar novos conhecimentos aos estudos realizados sobre veículos multi-compartimentados no Laboratório de Engenharia de Produção da UENF em particular sobre o tema proposto.
- Investigar trabalhos correlatos que possam contribuir para o desenvolvimento da pesquisa e análise dos resultados.

1.2 ORGANIZAÇÃO DA TESE

Esta dissertação está organizada em 8 capítulos. Seguindo este capítulo de introdução, tem o capítulo 2 onde são apresentados os principais problemas de roteamento de veículos, suas classificações e suas extensões.

O capítulo 3 trata das estratégias e métodos heurísticos para solução do Problema de Roteamento. No capítulo 4 são tratadas as estratégias e métodos metaheurísticos e apresenta em particular a heurística GRASP desenvolvida por Feo e Resende (1995).

O capítulo 5 trata dos conceitos e algoritmos básicos do fecho convexo. O capítulo 6 trata do Problema de Roteamento de Veículos com Multi-compartimentos e Restrição de Janela de Tempo, e dos algoritmos e estratégias de solução deste problema.

O capítulo 7 apresenta os testes realizados, e por fim no capítulo 8 as conclusões, e sugestões para trabalhos futuros.

2. PROBLEMA DE ROTEAMENTO DE VEÍCULOS E SUAS EXTENSÕES

O roteamento de veículos é intrínseco ao problema de distribuição de mercadorias das empresas, onde os veículos da frota são alocados de forma a atender os clientes que geralmente estão dispersos geograficamente. Uma operação de distribuição de mercadorias ideal é aquela onde todos os clientes são atendidos com qualidade de serviço e com o menor custos possíveis. Na prática este problema surge com muita frequência e é chamado de Problema de Roteamento de Veículos (PRV), por exemplo, Brown e Graves (1981) no setor de petróleo com otimização de caminhões tanques, Evans e Norback (1985) ramo alimentício com economia de 10,7 % na distribuição de uma rede americana de *foodservice* e Golden e Wasil (1987) no setor de bebidas.

Muitas vezes este problema surge em condições em que não há entrega de mercadorias, mas em que há necessidade de roteamento de veículos, seja, na coleta de lixo hospitalar, coleta de correspondências, transporte de alunos etc. Znamensky (2001), por exemplo, tratou do problema de roteamento para transporte de deficientes físicos e idosos, utilizando veículos de pequeno porte, e respeitando as restrições operacionais e temporais.

O Problema Básico de Roteamento de Veículos pode ser descrito da seguinte forma: Dado um conjunto de clientes, cada qual com uma demanda por um produto, um depósito com uma frota de veículos de determinada capacidade, o objetivo é definir a sequência de consumidores a serem atendidos pela frota de veículos de forma que o somatório das distâncias percorridas seja mínimo e que todas as demandas sejam atendidas. O PRV básico está representado na figura 2.1, com três rotas definidas para atendimento de 17 clientes. Entre parênteses encontra-se a demanda de cada cliente.

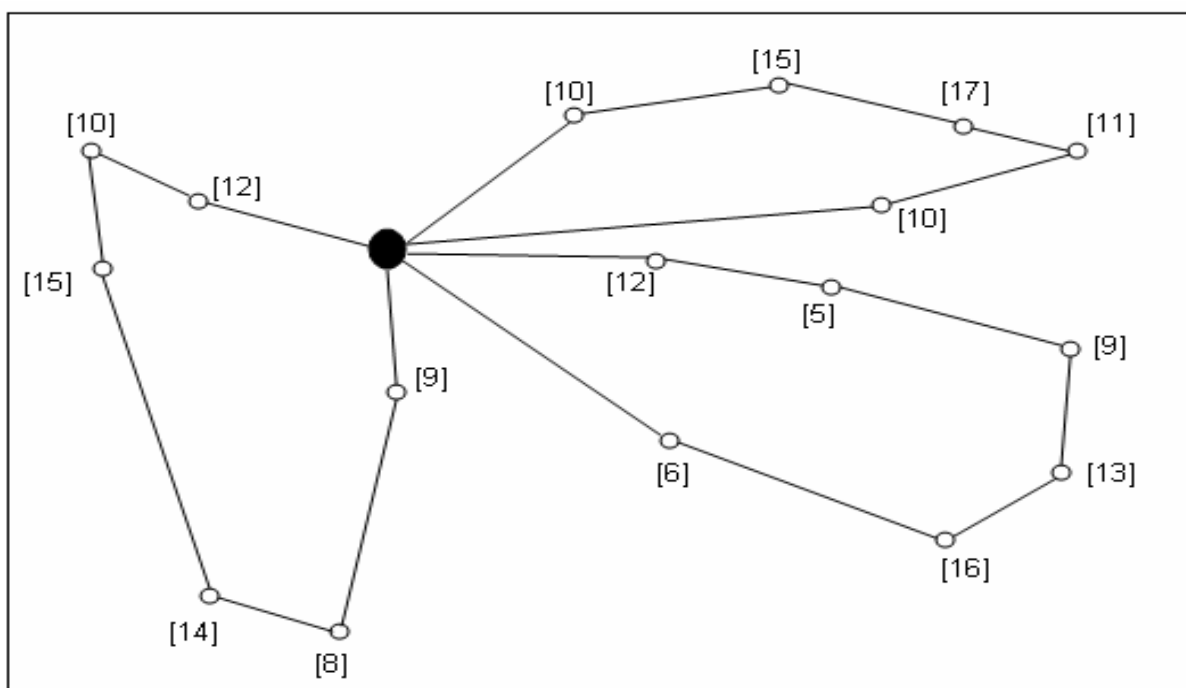


Figura 2.1 – Problema de Roteamento de Veículos

A figura 2.2 mostra o modelo matemático proposto por Bodin *et al.* (1983) para o Problema Básico de Roteamento de Veículos (PRV). O modelo considera o depósito como nó inicial (zero), com uma frota de veículos v e um conjunto de clientes a ser atendidos numerados de 1 a n .

A fórmula 2.1 é a função objetivo do problema. As restrições 2.2 e 2.3 asseguram que somente um veículo chegue a cada consumidor. A continuidade da rota é garantida pela restrição 2.4, ou seja, se um veículo chega a um cliente ele deverá sair do mesmo para o cliente seguinte. A restrição 2.5 restringe a capacidade do veículo. A restrição 2.6 limita o comprimento máximo da rota. As restrições 2.7 e 2.8 asseguram que cada veículo é usado apenas uma vez.

$$\text{Min } \sum_{i=0}^n \sum_{j=0}^n \sum_{v=0}^v c_{ij}^v x_{ij}^v \quad 2.1$$

Sujeito a

$$\sum_{i=0}^n \sum_{v=1}^v x_{ij}^v = 1 \quad j = 1, \dots, n \quad 2.2$$

$$\sum_{j=0}^n \sum_{v=1}^v x_{ij}^v = 1 \quad i = 1, \dots, n \quad 2.3$$

$$\sum_{i=0}^n x_{ie}^v - \sum_{j=0}^n x_{ej}^v = 0 \quad e = 1, \dots, n \quad 2.4$$

$$\sum_{j=0}^n p_j - \left(\sum_{i=0}^n x_{ij}^v \right) \leq C_v \quad v = 1, \dots, k \quad 2.5$$

$$\sum_{j=1}^n s_i^v \left(\sum_{v=1}^v x_{ij}^v \right) + \sum_{i=0}^n \sum_{j=0}^n t_{ij}^v x_{ij}^v \leq T_v \quad v = 1, \dots, k \quad 2.6$$

$$\sum_{j=1}^n x_{0j}^v \leq 1 \quad v = 1, \dots, k \quad 2.7$$

$$\sum_{j=1}^n x_{i0}^v \leq 1 \quad v = 1, \dots, k \quad 2.8$$

$$x_{ij}^v \in \{0, 1\} \quad \forall i, j, k \quad 2.9$$

Onde:

v = número de veículos; n = número de clientes;

c_{ij}^v = custo de viagem do cliente i ao cliente j com o veículo v ;

$x_{ij}^v = 1$ se o veículo viaja do cliente i ao cliente j , caso contrário é igual a 0;

p_j = tamanho do pedido do cliente; C_v = capacidade do veículo;

s_i^v = tempo de desembarque de mercadorias no cliente i atendido pelo veículo v ;

t_{ij}^v = tempo de viagem do cliente i ao cliente j com o veículo v ;

T_v = tempo de rota máximo para o veículo v ;

Figura 2.2 – Modelo matemático PRV

Um dos primeiros problemas de roteamento de veículos pesquisado foi o Problema do Caixeiro Viajante (“traveling salesman problem” ou TSP), que consiste

em encontrar o roteiro ou sequência de cidades a serem visitadas por um vendedor onde a distância total percorrida seja minimizada e assegure que cada cidade seja visitada exatamente uma vez. Desde então, novas restrições vêm sendo incorporadas ao problema de roteamento, de modo a melhor representar a realidade prática encontrada nas empresas. Com a inserção de novas restrições surgem novas extensões para o problema, dentre as quais podem-se destacar as seguintes:

- Problema de Roteamento de Veículos com Divisão de Entrega – Esta restrição permite tratar problemas onde há a necessidade de divisão da carga dos clientes. As empresas podem optar por este procedimento para redução de custos operacionais, ou até mesmo quando a carga do cliente ultrapassa a capacidade do veículo.
- Problema de Roteamento de Veículos com Janela de Tempo – Este problema surge quando clientes têm um período fixo para recebimento de mercadorias.
- Problema de Roteamento de Veículos com Múltiplos Depósitos – Neste caso, a empresa possui mais de um depósito na sua rede de distribuição, então deve otimizar suas entregas levando em consideração a localização, capacidade etc de seus vários depósitos.
- Problema de Roteamento de Veículos com Frota Heterogênea – Esta restrição cuida do problema de empresas que possui frota heterogênea de veículos.
- Problema de Roteamento de Veículos com Multi-compartimentos – Este problema é encontrado em sistemas de distribuição que utilizam frota de veículos com mais de um compartimento.
- Problema de Roteamento de Veículos com Coleta e Entrega – Este problema é encontrado, principalmente, no transporte de pessoas, transporte escolar etc.
- Problema Dinâmico de Roteamento de Veículos – O PRV básico não sofre mudanças de roteamento quando o processo de distribuição está sendo executado. Esta extensão do problema permite mudanças no processo durante as entregas.

No cotidiano das empresas, na maioria dos casos, estes problemas não surgem de forma isolada. Em geral, uma combinação destes problemas irá interferir nos procedimentos e nas decisões a serem tomadas para efetuar as entregas de mercadorias das empresas.

2.1 CLASSIFICAÇÃO DOS PROBLEMAS DE ROTEIRIZAÇÃO DE VEÍCULOS

O resultado esperado de todos os sistemas de roteamento é basicamente o mesmo, ou seja, uma rota atendida por um ou mais veículos e uma programação para atendimento dos clientes roteados. As rotas definem a sequência de locais que devem ser visitados e a programação define o horário em que cada atividade deve ser executada nestes locais.

Bodin *et al.* (1983) apresentam uma classificação para problemas de roteamento e programação e divide este grupo em problemas de roteamento, problemas de programação e problemas de roteamento e programação. Estes autores propuseram uma estrutura que classifica os problemas de acordo com as restrições espaciais e/ou temporais. A quadro 2.1 apresentada por Bodin *et al* (1983) demonstra a classificação proposta por estes autores.

Características	Possíveis Opções
1. Tamanho da frota	<ul style="list-style-type: none">• Um veículo• Múltiplos veículos
2. Tipo de frota disponível	<ul style="list-style-type: none">• Homogênea• Heterogênea• Tipos de veículos especiais
3. Localização dos veículos	<ul style="list-style-type: none">• Um depósito• Múltiplos depósitos
4. Natureza das demandas	<ul style="list-style-type: none">• Determinística• Estocástica
5. Localização das demandas	<ul style="list-style-type: none">• Nos nodos• Nos arcos
6. Tipo de rede	<ul style="list-style-type: none">• Não direcionada• Direcionada• Misturada• Euclidiana
7. Restrições de capacidade do	<ul style="list-style-type: none">• Mesma capacidade

veículo	<ul style="list-style-type: none"> • Diferentes capacidades • Capacidade ilimitada
8.Tempo máximo das rotas	<ul style="list-style-type: none"> • Mesmo tempo de duração • Tempos diferentes • Não imposto
9.Operação	<ul style="list-style-type: none"> • Coletas • Entregas • Entrega e Coleta • Demandas quebradas
10.Custo	<ul style="list-style-type: none"> • Custo variável • Custos fixos de operação
11Objetivos	<ul style="list-style-type: none"> • Minimizar o custo total das rotas • Minimizar a soma dos custos fixos e variáveis • Minimizar o número de veículos • Maximizar a utilização baseada no serviço ou na conveniência <ul style="list-style-type: none"> • Maximizar a utilização baseada no nas prioridades de atendimento

Quadro 2.1 – Classificação do PRV

2.2 PROBLEMA DE ROTEAMENTO DE VEÍCULOS E RESTRIÇÃO DE JANELA DE TEMPO (PRVJT)

Esta extensão do PRV ocorre em situações onde os clientes têm um horário definido para receber suas mercadorias. Geralmente esta restrição temporal é imposta pelo próprio cliente, por limitações de horários devido à localização física do cliente, ou por alguma necessidade operacional. As situações onde se pode encontrar a restrição de janela de tempo são diversas, tais como, distribuição do varejo, transporte escolar, coleta de refugo industrial, serviços de pós-venda, entrega bancária, entrega de combustíveis etc. Nesta extensão do problema há também a preocupação com a questão temporal, janela de tempo, e não mais somente a questão espacial.

A restrição temporal pode ser violada ou não. Quando a janela de tempo deve ser estritamente respeitada, ou seja, o cliente não pode ser atendido depois do horário de fechamento da janela de tempo, se diz que é “forte” (hard), e quando a janela pode ser violada se diz que é “fraca” (soft) e sua violação estará sujeita a penalidades.

Conforme Ioannou *et al.* (2003) as empresas têm forçado o uso da janela de tempo em todos os campos da distribuição de mercadorias, motivados pelos princípios da filosofia Just-in-Time e da consciência de competitividade. Ioannou *et al.* (2003) propuseram uma solução para o PRVJT com relaxamento da restrição temporal, a sua proposta é permitir que os clientes sejam atendidos antes e depois do horário da janela de tempo.

Segundo Bräysy *et al.* (2003) o objetivo do PRVJT é obter o menor custo de rotas a partir de um depósito para entrega em clientes dispersos geograficamente. As rotas devem ser projetadas de tal forma que cada cliente será visitado apenas uma vez e por um veículo dentro do intervalo de tempo determinado. As rotas começam e terminam no depósito e a demanda total dos clientes de uma rota não deve exceder a capacidade do veículo.

Lau *et al.* (2003) define o PRVJT da seguinte forma: Dado um grafo não orientado $G(V, A)$ onde $V = \{V_0, V_1, \dots, V_n\}$, V_0 é o depósito, V_i , $i \neq 0$ é um cliente com demanda d_i , janela de tempo (e_i, l_i) , e tempo de serviço S_i ; $A = \{(v_i, v_j) : i \neq j, v_i, v_j \in V\}$, cada arco (v_i, v_j) tem um tempo de viagem t_{ij} , e capacidade de veículo Q , encontre o conjunto mínimo de rotas partindo e chegando no depósito tal que cada cliente v_i é atendido por exatamente um veículo dentro da janela de tempo, $S_i \leq l_i$, para todos os clientes v_i atendidos por um veículo é menor que Q , e o tempo total de viagem deve ser minimizado.

Lau *et al.* (2003) utiliza a Busca Tabu com um número limitado de veículos para resolver o PRVJT, sendo que a restrição de janela de tempo é relaxada, mas com a inserção de penalidade pelo atraso no atendimento. Outro trabalho baseado em Busca Tabu é proposto por Ho *et al.* (2003). Nesta proposta é permitida a divisão da entrega de determinado cliente quando a capacidade do veículo é ultrapassada.

Bräysy *et al.* (2003) aplica o algoritmo de busca local multi-start na solução do PRVJT. O método para resolver o problema é dividido em fases, a primeira de construção de rotas e posteriormente redução do número de rotas, e uma outra para melhoria das distâncias das rotas. Na fase de construção a função de custo é baseada na distância dos clientes em relação ao depósito, e no tempo de espera

provocado pela chegada ao cliente antes do início da janela de tempo. Na fase seguinte um segmento composto por dois clientes é trocado entre duas rotas preservando a orientação das mesmas. Nesta pesquisa a restrição de janela de tempo é do tipo *hard*.

Modelos baseados em algoritmo genético são propostos por Berger *et al* (2002), por Oliveira *et al.* (2001), e por Alvarenga *et al* (2005) para resolução do PRVJT. A primeira pesquisa utiliza um algoritmo baseado na evolução paralela de duas populações, que estão focalizadas em objetivos diferentes. A primeira população evolui no sentido de obter redução da distância percorrida na rota, a segunda população objetiva a redução do tempo de violação da janela de tempo. A pesquisa de Oliveira *et al.* (2001) utiliza apenas uma população para inicialização do problema, o custo para inserir o cliente na rota é calculado de acordo com a sua posição geográfica, com o horário de fechamento da janela de tempo e do ângulo entre as posições geográficas do cliente e o depósito. Os autores da última pesquisa citada no início deste parágrafo buscam solucionar o problema primeiramente reduzindo o número de veículos, e posteriormente aplicam uma melhoria para reduzir a distância percorrida.

A metaheurística *Simulated Annealing* é utilizada por Souza Júnior *et al.* (2005) na solução do problema de roteamento com restrição temporal. Neste trabalho o objetivo é determinar as rotas de custo mínimo com uma frota de veículos de mesma capacidade. A solução inicial é obtida com a inserção, a cada iteração, de um cliente em uma rota qualquer. A inserção é feita respeitando o limite de capacidade do veículo e a janela de tempo de cada cliente. Os clientes são escolhidos aleatoriamente. As instâncias utilizadas para validação do algoritmo são de clientes em *clusters*, ou seja, aglomerados geograficamente.

O PRVJT é tratado por Kontoravdis (1995) e por Chaovalitwongse *et al.* (2003) de forma muito parecida. Os autores utilizam uma metaheurística GRASP na solução do problema. Os objetivos são reduzir o número de veículos utilizados e o tempo de percurso das rotas, a frota de veículos é homogênea e esses partem de um único depósito, a capacidade dos veículos é restrita, o tempo de entrega de mercadorias é igual a zero para todos os clientes, e a velocidade dos veículos é constante. Os resultados obtidos foram satisfatórios para diversos arranjos geográficos de clientes.

Como visto, encontram-se na literatura diversas pesquisas utilizando procedimentos baseados em heurísticas diferentes na busca de soluções do Problema de Roteamento de Veículos com restrição de Janela de Tempo.

2.3 PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MULTI-COMPARTIMENTOS (PRVMC).

Este problema proposto por Póvoa (2005) é definido da seguinte forma: Considera - se n clientes cada um com uma demanda de mercadorias específica para cada tipo de compartimento do veículo. As mercadorias são entregues a partir de um depósito por uma frota de veículos heterogênea. Sendo que cada veículo tem o seu próprio conjunto de compartimentos.

Como exemplo podemos imaginar uma empresa que tenha uma linha de produtos com características de volume e forma de acondicionamento diferente. Por exemplo, uma empresa que vende produtos congelados e secos e que possui uma frota de veículos dividida de acordo com a temperatura do produto. Ou uma transportadora de combustíveis que possui uma frota de veículos com divisórias para tipos distintos de combustível.

Segundo Póvoa (2005) a solução do problema consiste em encontrar as rotas dos veículos que minimizem o custo total, que consiste numa combinação de custo variável com a distância percorrida e custo fixo de cada veículo.

Chajakis e Guignard (2003) estudaram o PRVMC aplicado na distribuição de lojas de conveniência. Utilizando veículos com divisórias de tamanhos diferentes desenvolveram uma solução a partir da relaxação lagrangeana. No método de relaxação lagrangeana, o problema original com inúmeras restrições, normalmente complexo, tem algumas restrições relaxadas, permitindo sua simplificação. Essas restrições relaxadas são penalizadas através de multiplicadores de Lagrange, os quais são gradativamente ajustados ao longo das iterações. A sequência de soluções gerada, cada vez mais se aproxima da solução ótima do problema.

Na proposta de Póvoa (2005) as rotas iniciais são definidas inicialmente a partir dos clientes mais dispersos geograficamente e são compatíveis em quantidade com o número de veículos. A partir deste ponto os clientes são roteados em função de um custo que varia com a distância, a demanda e por Km rodado.

3. ESTRATÉGIAS E MÉTODOS HEURÍSTICOS DE SOLUÇÕES

Apesar da existência de métodos exatos para solução dos Problemas de Roteamento de Veículos, o esforço computacional para resolvê-los inviabiliza a utilização desta metodologia, principalmente, para problemas com grandes instâncias. Para problemas de roteamento de veículos, segundo pesquisas, os métodos exatos só fornecem boas soluções para um número reduzido de clientes, algo em torno de 50 clientes. A partir daí o tempo de descoberta da solução aumenta consideravelmente.

Tomemos o exemplo de Viana (1996) do Problema do Caixeiro Viajante (PCV), a quantidade de clientes a ser atendida é definida por C , e o número de soluções possíveis é dado por $C!$. A tabela 3.1 retirada do trabalho de Viana (1996) mostra uma estimativa do tempo gasto para solução do problema com o aumento do número de clientes. Percebe-se claramente a inviabilidade de buscar a solução através de métodos exatos. Percebe-se que devido ao crescimento exponencial que estes problemas apresentam, é evidente que a obtenção de uma solução exata é totalmente inviável.

Tabela 3.1 – Estimativa do tempo de solução do PCV		
C	C!	Tempo
5	120	0,00012 seg
10	3628800	3,62880 seg
12	479001600	8 mim
15	1307674368000	15 dias
20	2432902008176640000	77147 anos
50	3.0414093201713378043612E+0064	Infinito

Uma considerável redução do tempo e da área de busca de resposta para o problema pode ser obtida com a utilização de uma heurística. Esta redução considerável é a propulsora das pesquisas atuais.

Existem diversas propostas baseadas em heurísticas para o PRV. Laporte *et al* (2000) divide os métodos de solução em dois grupos principais: heurísticas clássicas e metaheurísticas. Os métodos heurísticos exploram uma área relativamente limitada do espaço de soluções, em geral produzem boas respostas

com tempo computacional pequeno. Os procedimentos metaheurísticos dão ênfase na busca de soluções teoricamente mais promissoras, a qualidade de solução produzida por estes métodos é melhor se comparados com os heurísticos, mas com um custo maior de tempo computacional.

A seguir são apresentados alguns métodos heurísticos para solução do PRV.

3.1 A HEURÍSTICA DE CLARK E WRIGHT

Proposto por Clarke e Wright (1964), este método é possivelmente o mais conhecido para o PRV, aplicado a problemas onde o número de veículos é uma variável de decisão. Diversos autores utilizaram esta metodologia tais como: Gaskell (1967), Paessens (1988) e Nelson *et al.* (1985). O método, também conhecido como o método das economias, esteve em evidência ao longo dos anos como sendo flexível o suficiente para manusear computacionalmente uma ampla faixa de restrições práticas (BALLOU, 2001).

O método proposto por Clarke e Wright (1964) inicialmente assume que um único veículo irá atender um único cliente, Desta forma para N clientes serão necessários N veículos. O método calcula a economia de distância entre os clientes i e j S_{ij} utilizando apenas um veículo para atendê-los. O método é calculado da seguinte forma:

$S_{ij} = S_{ji} = d_{id} + d_{dj} - d_{ij}$, onde d é o depósito.

O algoritmo básico do método das economias é mostrado abaixo.

Passo 1: calcula-se a economia S_{ij} para todos os pares de clientes i e j . A economia é dada pela seguinte fórmula $S_{ij} = S_{ji} = d_{id} + d_{dj} - d_{ij}$.

Passo 2: Ordenar as economias em ordem decrescente.

Passo 3: iniciando do topo da lista, fazer:

Versão Paralela

Passo 4: se ao incluir o segmento (i, j) na rota, resultar em uma rota viável de acordo com as restrições do PRV a ser resolvido, então anexar este segmento à solução; se não, rejeitar o segmento.

Passo 5: tentar com o próximo segmento da lista e repetir o passo 4 até que todos os segmentos já tenham sido investigados.

Versão Sequencial

Passo 4: buscar o primeiro segmento viável da lista, que pode ser usado para estender um dos extremos da rota em construção.

Passo 5: se a rota não pode ser mais estendida, terminar a rota. Escolher o primeiro segmento factível na lista para começar uma nova rota.

Passo 6: repetir os passos 4 e 5 até que nenhum segmento possa ser escolhido.

A economia é obtida pela redução do número de veículos requeridos para atender os clientes i e j , e pela redução da distância após a combinação dos percursos. As economias calculadas são ordenadas decrescentemente e os clientes i e j que têm as maiores economias são unidos enquanto a capacidade de restrição não é extrapolada. O método une clientes sem violar as restrições de capacidade e pára quando nenhuma união a mais pode ser feita.

A seguir a figura 3.1 representa a economia obtida através do link de dois clientes i e j que passam a ser atendidos por um único veículo.

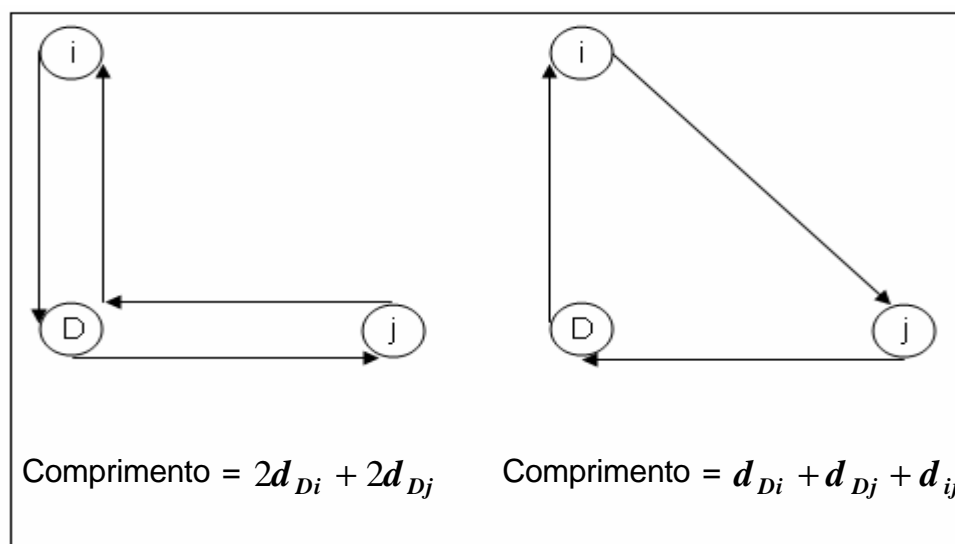


Figura 3.1 – Método das economias

3.2 A HEURÍSTICA DE GILLET- MILLER (MÉTODO DA VARREDURA)

Esta heurística procura obter a solução em duas etapas distintas: primeiro agrupar os pontos de demanda segundo algum critério de proximidade formando assim clusters, depois solucionar cada cluster independentemente criando uma rota para estes clientes.

Os clientes são agrupados a partir de uma linha imaginária centrada no depósito. Um cliente é escolhido de forma aleatória e o raio centrado no depósito percorre de forma rotativa no sentido horário ou anti-horário. Os clientes são designados aos veículos, no decorrer da varredura, até atingir a restrição de

capacidade. Depois um novo veículo é selecionado e a varredura continua até que a capacidade do novo veículo seja atingida e assim continua até que todos os clientes tenham sido definidos para um dado veículo. Este método foi bastante difundido por Gillett e Miller (1974).

Segundo Ballou (2001) a desvantagem do método tem relação com a maneira que as rotas são formadas. O processo tem dois estágios, com as paradas sendo atribuídas aos veículos primeiro. Por causa deste processo em dois estágios, as questões de tempo, como tempo total da rota e janelas de tempo, não são bem-manipuladas.

3.3 HEURÍSTICA DE PÉTALAS

Alguns autores caracterizam este procedimento como uma extensão do método da varredura. Este procedimento foi proposto inicialmente por Balinski e Quandt (1964) para solução de um problema de entrega, mas não apresentou bons resultados quando existia um número grande de rotas. Agarwal *et al.* (1989) obteve resultados ótimos para problemas com poucos clientes (< 25).

3.4 ALGORITMO PRIMEIRO AGRUPA, SEGUNDO ROTEIRIZA

Consiste no procedimento de agrupar os nós ou arcos de demanda primeiro, e depois construir rotas econômicas para cada agrupamento. Exemplos destas idéias foram aplicados nos trabalhos de Gillett e Miller (1974), Chapleau *et al.* (1981) para o problema básico de roteirização de veículos, Gillett e Johnson (1976) para o problema de roteamento de veículos com multi depósitos.

3.5 MÉTODO DAS INSERÇÕES

Este método inicia com uma rota unindo dois vértices, então adiciona os vértices restantes um a um conforme procedimento escolhido, segundo Cook *et al* (1998) este método pode assumir diversas variações:

- Inserção do mais Distante: neste caso, inicia-se uma rota com dois vértices que estão no final de uma aresta de alto custo. Para cada vértice não inserido v , é calculado o custo mínimo entre v e qualquer vértice na rota construída. Então é escolhido como o próximo a ser inserido aquele para o qual o custo for máximo.

- Inserção do mais Próximo: inicia-se o subgrafo com apenas um vértice, encontra-se um outro para formar o subrota $i-k-i$, cujo custo C_{ik} seja mínimo. Dada esta subtour seleciona-se o próximo vértice como sendo o mais próximo a qualquer vértice na subtour e o insere entre $i-k$.

- Inserção do mais Barato: semelhante à inserção do mais próximo, com exceção de que o próximo vértice a ser inserido não precisa ser o mais próximo a qualquer um do subtour já existente.

3.6 HEURÍSTICAS DE MELHORIA

Alguns autores também denominam estes métodos como “métodos de melhoramento” (Laporte, 2000), (Bodin *et al.*, 1983) e (Breedam, 2001). A heurística de melhoria pode ser aplicada em cada rota separadamente, esta forma tem bastante aplicação no Problema do Caxeiro Viajante, ou pode ser utilizada explorando diversas rotas ao mesmo tempo. Muitos procedimentos de melhoria para o Problema do Caixeiro Viajante foram implementados por Lin (1965).

Este tipo de método procura um mínimo local através de movimentos que causem uma melhora no valor da função objetivo (BREEDAM, 2001). O término da sua execução ocorre quando mais nenhum movimento de melhoramento puder ser feito. Breedam (2001) propõe movimentos para melhoria da solução, segundo o autor os movimentos são mudanças para tentar buscar diferentes soluções em vizinhanças distintas. Basicamente os movimentos são feitos com a troca ou relocação de clientes entre rotas.

4. ESTRATÉGIAS E MÉTODOS METAHEURÍSTICOS DE SOLUÇÕES

Os problemas de roteamento são classificados na literatura como NP-difíceis. São problemas para os quais não existem algoritmos exatos que os resolvam em tempo polinomial determinístico. Estes problemas são classificados como problemas de otimização combinatória. Como mencionado, é inviável a solução destes problemas através de métodos exatos.

Os procedimentos heurísticos são aplicados de forma a obter soluções próximas do ótimo com um tempo computacional razoável, mas segundo Santana *et al.* (2004) a maioria destes procedimentos é específica para um problema em particular, tendo pouca eficiência na solução de uma classe mais ampla de problemas.

A reunião de conceitos de Otimização e Inteligência Artificial, e a necessidade de procedimentos para problemas de maior amplitude viabilizou a construção das chamadas metaheurísticas, ou métodos inteligentes flexíveis.

O termo metaheurística é obtido com a inserção do prefixo *meta* (além) antes da palavra heurística. As metaheurísticas são estratégias inteligentes para projetar e melhorar o rendimento de procedimentos heurísticos.

As técnicas de metaheurísticas são princípios e idéias que conduzem ao desenvolvimento de novas classes de heurísticas que melhoram consideravelmente a eficiência de algoritmos aproximados e o tamanho de instâncias solúveis de problemas difíceis encontrados na prática (NORONHA, 2001). O objetivo da metaheurística é evitar a armadilha dos mínimos locais, possibilitando momentaneamente soluções ruins durante as iterações, de forma a buscar novas soluções vizinhas que sejam melhores que as soluções encontradas até então. Basicamente, os métodos metaheurísticos inserem perturbações ou aleatoriedade no procedimento para impedir que a solução fique restrita a um ótimo local. Esta característica faz com que os modelos metaheurísticos sejam mais eficientes no trato do problema se comparados aos modelos heurísticos.

Com o objetivo de escapar de ótimos locais, algumas delas permitem controlar a deterioração das soluções para diversificar a procura. Em geral, uma metaheurística constitui uma estrutura mais genérica baseada em princípios ou conceitos sobreposta a uma determinada heurística.

A seguir é apresentada uma classificação das metaheurísticas proposta por Santana *et al.* (2004).

- **Metaheurísticas de relaxação** – Se referem a procedimentos de resolução de problemas que utilizam relaxação do modelo original, ou seja, modificações do problema original, que facilitam a solução. O método mais conhecido desta classe é a relaxação Lagrangeana.
- **Metaheurísticas construtivas** – Procedimentos que obtêm soluções do problema por meio de um procedimento que incorpora iterativamente elementos a uma estrutura, inicialmente vazia, que representa a solução. Estas metaheurísticas são baseadas numa função gulosa, a partir da escolha dos elementos que produzem melhores resultados imediatos. Neste grupo de metaheurística destaca-se o procedimento GRASP.
- **Metaheurísticas de busca** - Estabelecem estratégias para percorrer o espaço de soluções do problema, transformando de forma iterativa a solução inicial.
- **Metaheurísticas evolutivas** – Estabelecem estratégias para conduzir a evolução no espaço de busca do conjunto de soluções. A diferença entre a classe de metaheurísticas de busca é que utiliza um conjunto de soluções simultaneamente para definir a melhor solução. Pode-se destacar os algoritmos genéticos e os algoritmos meméticos.
- **Metaheurísticas de decomposição** – estabelecem estratégias para resolver um problema determinando subproblemas a partir do problema original. Trata-se de metaheurísticas intermediárias entre as de relaxação e as construtivas. O objetivo principal é obter subproblemas significativamente mais fáceis de resolver que os originais, e cujas soluções podem ser utilizadas efetivamente.

Um resumo das principais técnicas metaheurísticas é apresentado a seguir.

4.1 SIMULATED ANNEALLING (SA)

O Simulated Annealing, ou recozimento simulado é baseado nos princípios da termodinâmica, proposto inicialmente por Kirkpatrick *et al.* (1983). É um método de busca local que requer uma estrutura de vizinhança e uma função de avaliação que associa a cada solução um valor numérico representando o custo da solução.

Seria a simulação do resfriamento de um conjunto de átomos aquecidos. Quando isto ocorre, os átomos não mais estarão nas suas posições iniciais. Fazendo uma analogia, o procedimento estará inserindo perturbações no espaço de soluções, permitindo movimentos piores na vizinhança para fugir de ótimos locais. Os átomos seriam as possíveis soluções para o problema, e a temperatura é o regulador probabilístico do sistema. Se o metal é resfriado de forma brusca, a microestrutura tende a um estado randomicamente instável. Se o metal é resfriado de forma suficientemente lenta, o sistema procurará um ponto de equilíbrio caracterizado por uma microestrutura ordenada e estável.

4.2 BUSCA TABU (BT)

Os métodos de Busca Tabu, propostos inicialmente por Glover (1989) e Glover (1990), são metaheurísticas de melhoramento local que utilizam uma lista de movimentos proibidos para avançar em direção ao ótimo. Este método permite que se obtenha soluções piores, temporariamente, com o objetivo de escapar de ótimos locais. É metodologia que procura imitar o processo de memória humana, não somente a memória local, mas também a memória de maior temporalidade.

Começando com uma solução inicial qualquer, o método explora em cada iteração uma parcela da vizinhança da solução corrente. A melhor solução encontrada é armazenada, mas, para não criar um *loop* dentro de um determinado número de soluções, inseri-se a noção de movimentos Tabu e Lista Tabu. O objetivo desta lista é evitar movimentos a regiões já exploradas do espaço de soluções. Isto pode ser obtido com a proibição dos últimos movimentos feitos. O procedimento termina geralmente segundo um critério de quantidade de iterações.

4.3 ALGORITMOS GENÉTICOS (AG)

Diversos problemas de otimização envolvem procurar uma solução num espaço vasto de candidatos sujeito a restrições de tempo e espaço. Quando não existe nenhum conhecimento a priori sobre o problema não podemos usar estratégias estáticas específicas do domínio. É neste contexto que surgem os algoritmos genéticos como uma forma de resolver problemas de procura adaptativa nos quais o conhecimento para controlar a procura é obtido dinamicamente. Proposto inicialmente por Holland (1975), com intuito de aplicar a teoria da evolução

das espécies, ou seja, utilizar os conceitos da evolução biológica, tais como, como genes, cromossomos, cruzamento, mutação e seleção em outros problemas de otimização através de algoritmos computacionais.

As espécies naturais lutam permanentemente pela sobrevivência tentando adaptar-se ao ambiente em que estão inseridas. É sabido que ao longo dos tempos espécies têm desaparecido enquanto outras novas têm surgido. O ponto de vista de que os seres vivos evoluem de acordo com o princípio da seleção natural, defendido por Charles Darwin no século XIX, é hoje consensualmente aceita. Em termos gerais este paradigma define que:

- Cada indivíduo transmite aos seus descendentes os seus traços característicos (genéticos);
- Numa população existem indivíduos com traços (características) diferentes;
- Os indivíduos com melhores características (mais adaptados) tendem a reproduzir-se mais, pelo que uma população caminha no sentido da existência em maior número dos elementos mais adaptados;
- Ao longo do tempo a acumulação de pequenas variações pode originar o aparecimento de novas espécies com melhor capacidade de sobrevivência;
- Processos de troca e aparecimento de (novo) material genético podem concorrer para uma melhor adaptabilidade.

Os algoritmos genéticos são uma versão computacional simplificada do que se passa na natureza. Qualquer problema a ser resolvido nesta abordagem passa pela identificação de um conjunto de elementos e pela definição de um conjunto de valores.

Os Algoritmos Genéticos diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos:

- Trabalham com uma codificação do conjunto de parâmetros e não com os próprios parâmetros.
- Trabalham com uma população e não com um único ponto.
- Utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar.
- Utilizam regras de transição probabilísticas e não determinísticas.

4.4 ANT COLONIES (AC)

Metaheurística de otimização de colônia de formigas proposta por Dorigo e Caro (1999), é uma metodologia que simula o comportamento de um conjunto de agentes que cooperam para resolver um problema de otimização por meio de comunicações muito simples. Sendo assim, possui características estreitas com o comportamento de uma comunidade de formigas. Formigas parecem poder achar o seu caminho com facilidade relativa. Estudos descobriram isso, e em muitos casos esta capacidade é o resultado da comunicação química entre as formigas e um fenômeno emergente causado pela presença de muitas formigas.

As formigas são insetos sociais que possuem um sistema complexo de organização e divisão de tarefas, cuja principal função é garantir a sobrevivência do formigueiro. Chama atenção a complexidade das tarefas executadas pelo formigueiro quando comparada com a simplicidade de cada formiga individualmente. Para a obtenção do alimento para o formigueiro, a colônia resolve um interessante problema de otimização. Inicialmente, as formigas percorrem de modo aleatório a região próxima ao formigueiro em busca do alimento. Cada formiga, enquanto percorre o seu caminho, deposita sobre o solo uma substância chamada feromônio, formando um caminho ou rastro de feromônio. As formigas subseqüentes detectam a presença desta substância e tendem a escolher o caminho marcado com a maior concentração de feromônio. O feromônio, portanto, além de possibilitar a formação de um caminho de volta para a formiga, também tem a função de informar as outras formigas sobre quais os melhores caminhos até o alimento. Depois de algum tempo, os caminhos mais eficientes – ou de menor distância percorrida até o alimento – acumulam uma quantidade maior de feromônio. Inversamente, os caminhos menos eficientes – ou de maior distância percorrida até o alimento – apresentam uma pequena concentração de feromônio, devido ao menor número de formigas que passaram por ele e ao processo de evaporação natural do feromônio. No problema de otimização que o formigueiro se defronta, cada formiga é capaz de construir uma solução completa do problema; contudo, a melhor solução só é obtida mediante cruzamento das diversas soluções encontradas.

4.5 BUSCA EM VIZINHANÇA VARIÁVEL (BVV)

Proposto por Mladenovic e Hansen (1997) é um método que explora o espaço de soluções através de trocas sistemáticas de estrutura de vizinhanças. Contrariamente a outras metaheurísticas baseadas em métodos de busca local, o método VNS não segue uma trajetória, mas sim explora vizinhanças gradativamente mais “distantes” da solução corrente e focaliza a busca em torno de uma nova solução se e somente se um movimento de melhora é realizado. O método inclui, também, um procedimento de busca local a ser aplicado sobre a solução corrente. Esta rotina de busca local também pode usar diferentes estruturas de vizinhança. A melhora é definida pela função de avaliação de cada solução.

Considerando um conjunto finito pré-definido de estruturas de vizinhança N^k e com N_s^k sendo o conjunto de soluções da k-ésima vizinhança da solução corrente s . O método inicializa com uma solução inicial qualquer e a cada iteração é gerado aleatoriamente um vizinho, s' , dentro da vizinhança N^k da solução corrente. Aplica-se então um método de busca local no vizinho gerado, encontrando uma solução que representa um ótimo local, s'' . Se este ótimo local for melhor que a solução corrente a busca continua a partir deste, recomeçando da primeira estrutura de vizinhança. Caso contrário, continua-se à busca a partir da próxima estrutura de vizinhança, N^{k+1} . Este procedimento é encerrado quando uma condição de parada for atingida, tal como o tempo máximo de processamento ou o número máximo de iterações consecutivas sem melhora da solução corrente.

4.6 ALGORITMOS MEMÉTICOS (AM)

Alguns estudiosos chamam este procedimento de algoritmo genético híbrido. O termo Algoritmo Memético (AM) surgiu pela primeira vez com Moscato (1989) quando ele procurou definir os limites e diferenças entre algoritmo memético e algoritmo genético.

Algoritmos Meméticos utilizam o conceito de “evolução cultural”, onde a adaptabilidade de um indivíduo pode ser modificada no decorrer de sua existência dentro da população. Um indivíduo pode ser geneticamente pouco favorecido ao nascer, mas devido às condições em que vive, por trocas de informação com outros indivíduos, experiências pessoais, entre outros aspectos, podem se tornar mais

adaptado, e mais do que isso, transmitir essa experiência aos seus descendentes (evolução cultural).

4.7 GREEDY RANDOMIZED ADAPTIVE SEARCH PROCEDURES - GRASP

A metaheurística GRASP proposta por Feo e Resende (1995) tem sido usada para obter soluções de qualidade para muitos problemas de otimização combinatória. Possui várias vantagens sobre outras metaheurísticas, pois procura conjugar características gulosas e aleatórias. Sua implementação é simples, usando-se algoritmos para construção de soluções e de busca local, desenvolvido para serem usados em outras abordagens. Além disso, o ajuste de parâmetros é mais simples em um procedimento GRASP do que em outras metaheurísticas.

O Procedimento de Busca Adaptativa Gulosa e Randomizada (GRASP), nomenclatura utilizada em português, é um método iterativo composto de duas fases:

- Uma fase de construção, onde uma solução viável é gerada elemento a elemento;
- Uma fase de busca local, onde é pesquisado um ótimo local na vizinhança da solução viável construída na primeira fase.

Após ocorrer todas as iterações GRASP, a melhor solução encontrada será o resultado obtido. Na fase de construção, a solução viável é gerada iterativamente, um elemento da solução por vez, até que esteja completa. Os elementos candidatos que formam a solução são ordenados em uma lista de Candidatos (C), e serão escolhidos segundo uma função gulosa que irá medir o benefício da inserção de cada elemento na formação da solução. A função gulosa demonstra o aumento ou a redução da função objetivo após a entrada de um destes elementos na solução em construção (FEO e RESENDE, 1995).

Um subconjunto construído com os melhores elementos é formado, este subconjunto é chamado de Lista de Candidatos Restrita – LCR. O tamanho da LCR é controlado por um parâmetro $\alpha \in [0,1]$, sendo que o comportamento do algoritmo será considerado puramente guloso se $\alpha=0$ e será considerado puramente aleatório se $\alpha=1$. Mas dependendo da formulação do problema este parâmetro α poderá ser descartado definindo-se um número fixo de candidatos.

Esta característica gulosa do algoritmo é importante, pois se podem obter soluções próximas do ótimo em cada iteração. É que, conforme mencionado, a

função gulosa do algoritmo irá aumentar ou diminuir o resultado da função objetivo ao máximo ou ao mínimo possível respectivamente. A qualidade da solução está associada ao critério guloso aplicado.

O procedimento é adaptativo, pois a cada iteração os benefícios agregados com a escolha de cada elemento são atualizados, isto reflete as mudanças ocorridas a partir da inserção do elemento anterior. A figura 4.1 apresenta o pseudocódigo da fase de construção.

As soluções geradas na fase de construção nem sempre são ótimos locais. Então, a fase de busca local se inicia para que soluções melhores sejam encontradas. A rapidez da busca local é proporcional à proximidade do ponto de partida com o ótimo local. Na figura 4.2 é apresentado o pseudocódigo da fase de busca local.

Procedimento	Construção Randomizada Gulosa
1	<i>SOLUÇÃO ? { };</i>
2	Calcule o custo com o incremento dos elementos candidatos;
3	Enquanto <i>SOLUÇÃO</i> não é uma solução completa Faça
4	Construa a Lista de Candidatos Restrita (LCR);
5	Selecione aleatoriamente um elemento s na LCR;
6	<i>SOLUÇÃO ? SOLUÇÃO U { s };</i>
7	Recalcule o custo com o incremento de s ;
8	Fim:

Figura 4.1 – Pseudocódigo da fase de construção

Procedimento	Busca Local (minimização)
1	Enquanto <i>SOLUÇÃO</i> não é a ótima local Faça
2	Encontre $s' \in N(SOLUÇÃO)$ sendo $f(s') < f(SOLUÇÃO)$;
3	Solução ? s' ;
4	Fim;
5	Retorna Solução
Fim	Busca Local

Figura 4.2 – Pseudocódigo da fase de melhoria

5. FECHO CONVEXO (CONVEX HULL)

O fecho convexo é uma das estruturas mais usadas em geometria computacional. Tem utilidade na geração de outras estruturas e serve como apoio a algoritmos mais complexos. Nesta pesquisa o fecho convexo será utilizado no procedimento de inicialização de rotas. O algoritmo de fecho convexo irá separar os clientes a serem roteados em dois grupos, clientes convexos e clientes não convexos. Este procedimento será detalhado no capítulo 7.

Exemplos do uso de fechos convexos podem ser encontrados em robótica, detecção de colisões, reconhecimento de padrões, processamento de imagens, e até mesmo em otimização. O fecho convexo de um conjunto S de pontos é o menor conjunto convexo que contém S . Na situação mais extrema o conjunto convexo terá o mesmo conjunto de vértices do próprio conjunto.

O fecho convexo pode ser descrito da seguinte forma: Dado um conjunto S com n pontos do plano, freqüentemente necessita-se encontrar um conjunto convexo que contenha todos os pontos de S . Em particular, o fecho convexo de um conjunto S de pontos é um dos conjuntos convexos que contém S e consiste na interseção de todos os conjuntos convexos que contêm S .

A seguir são listados os principais algoritmos para encontrar o fecho convexo. Em seguida há uma explicação mais detalhada do Algoritmo de Graham que foi utilizado nesta pesquisa.

5.1 ALGORITMO DE JARVIS

Este algoritmo tem complexidade de tempo dependente do conjunto de pontos de entrada. O algoritmo identifica o fecho convexo como faz a maioria dos seres humanos, ou seja, procura pelas arestas da fronteira, uma a uma, na ordem correta (JARVIS, 1973). O algoritmo primeiramente encontra o ponto de menor ordenada. A abcissa é o critério de desempate. Este ponto certamente fará parte do fecho convexo. Uma reta suporte é traçada horizontalmente sobre o ponto, e pela forma de escolha do mesmo, pode-se afirmar que nenhum outro ponto está sobre esta reta. O próximo ponto será definido de acordo com o ângulo que seu posicionamento forma em relação à reta suporte. O algoritmo continua este processo considerando como reta suporte a última aresta encontrada até o momento. O procedimento termina quando o ponto inicial é encontrado novamente.

5.2 ALGORITMO QUICKHULL

Preparata e Shamos (1985) chamaram este algoritmo desta forma devida a semelhança com o algoritmo de ordenação QuickSort. O algoritmo é recursivo, e tem como princípio básico dividir o conjunto de pontos S em dois subconjuntos S_1 e S_2 . O procedimento recursivo deve atuar sobre um conjunto de pontos e uma aresta inicial do fecho convexo. Neste caso, inicialmente, o algoritmo acha os pontos de S com maior e menor abscissa (pontos a e b). Em seguida, divide o conjunto S em dois conjuntos distintos, pontos à esquerda de ab e pontos à direita de ab . Após a subdivisão inicial, o procedimento recursivo é aplicado a cada um dos dois conjuntos encontrados no passo anterior.

5.3 ALGORITMO MERGEHULL

Este algoritmo divide o problema em diversos subproblemas similares ao problema original, mas de tamanho menor. Após a resolução dos subproblemas, as soluções são combinadas para se obter a solução do problema original. Este algoritmo proposto por Preparata e Shamos (1985) é dividido em três passos:

- **Dividir** – O problema é subdividido em problemas de tamanho menor.
- **Conquistar** – Os subproblemas serão resolvidos recursivamente até serem tão pequenos que possam ser resolvidos diretamente.
- **Combinar** - As soluções dos subproblemas são combinadas para se obter uma solução do problema original.

5.4 ALGORITMO DE GRAHAM

Algoritmo elaborado por Graham (1972) é considerado um dos pioneiros no campo da geometria computacional. A ideia básica do algoritmo de Graham é bastante simples. O algoritmo de Graham constrói o fecho convexo em duas fases, inicialmente devem-se ordenar os pontos do conjunto S com relação a sua abscissa, o desempate é feito pela ordenada. Então se defini o ponto inicial p_0 . Em seguida, calcula-se o ângulo com a horizontal, no sentido anti-horário, formado pelo segmento entre o ponto p_0 e todos os demais. Os pontos são ordenados segundo este critério. Os pontos são então examinados um a um, segundo essa ordem crescente, em uma

varredura (daí a denominação usual deste algoritmo, Graham's scan, ou varredura de Graham), e o fecho convexo é construído de maneira incremental a partir de p_0 .

A cada passo, o fecho já determinado é correto, considerando apenas o conjunto de pontos examinado até o momento. A avaliação de cada ponto adicional, no entanto, pode fazer com que pontos sejam retirados do fecho, na ordem inversa àquela que foram inseridos. Por isso, costuma-se armazenar os pontos do fecho convexo em uma pilha.

Na figura 5.1 é apresentado um algoritmo de Graham básico.

Entrada: Conjunto de pontos $S=\{P=(P.x,P.y)\}$

Passo 1 - Selecciona o ponto mais baixo p_0 em S .

Passo 2 - Ordene S ao redor de p_0 de acordo com o ângulo.

Em caso de empate favorecer o mais próximo de p_0 .

Passo 3 – Seja $P[n]$ a lista de pontos ordenados.

Passo 4 – Insira $P[0]=p_0$ e $P[1]$ numa lista O .

Passo 5 – Enquanto $i < n$ faça,

Seja p_{T1} = O ponto do topo da lista O .

Seja p_{T2} = O segundo ponto do topo da lista O .

Se $P[i]$ está estritamente à esquerda da linha formada de p_{T2} para p_{T1} Então insira $P[i]$ na lista O .

Caso contrário retire o ponto p_{T1} da lista O .

Saída : O Fecho Convexo de S

Figura 5.1 – Algoritmo básico de Graham

Na figura 5.2 os pontos estão numerados segundo a ordem crescente do ângulo ao redor de p_0 . A implementação do algoritmo deve prever recursos para resolver os casos de colinearidade de pontos. Especificamente, é necessário acrescentar um critério ao algoritmo de ordenação dos pontos ao redor de p_0 , dando prioridade em caso de colinearidade para os pontos mais próximos de p_0 . Ou seja, considera-se $p_a < p_b$ se os ângulos forem iguais mas $p_a - p_0 < p_b - p_0$. Esta regra resolve o caso em que existe uma colinearidade nos últimos pontos, fazendo com que o mais próximo seja o penúltimo e o mais distante o último, e, portanto pertencente ao fecho convexo conforme desejado. Um exemplo desta situação,

também pode ser vista na figura 5.2. No caso, o ponto p_1 tem que ficar antes de p_2 na ordenação para que seja possível eliminá-lo sem problemas, o que não ocorreria se a situação fosse invertida.

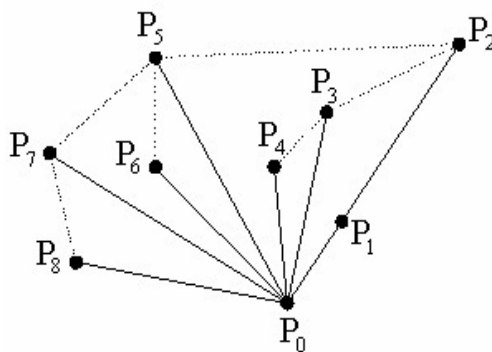


Figura 5.2 – Funcionamento do algoritmo de Graham

6. O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MULTI-COMPARTIMENTOS E RESTRIÇÃO DE JANELA DE TEMPO (PRVMCJT).

A restrição de janela de tempo está inserida em diversas extensões do Problema de Roteamento de Veículos - PRV. O escopo desta pesquisa é justamente inserir esta restrição em mais uma variação do PRV, o Problema de Roteamento de Veículos com Multi-compartimentos proposto por Póvoa (2005).

Janelas de tempo são restrições fortes para o problema de roteamento de veículos. Em geral as empresas tem que aumentar a sua frota de veículos para atender aos clientes com restrição temporal. A utilização de uma frota de veículos multi-compartimentados pode amenizar esta tendência de crescimento da frota em situações onde está presente a restrição de janela de tempo. Na prática podem-se encontrar clientes com janelas de tempo próximas, mas com demanda por produtos muito diferentes, por exemplo, congelados e refrigerados. Um veículo multi-compartimentado pode ter compartimentos “secos” e “molhados”, ou ainda compartimentos com diferentes temperaturas de acondicionamento, estas características aumentam as possibilidades de atendimento de um veículo.

A janela de tempo é definida como o intervalo $[a_i, b_i]$, e indica que a partir do instante inicial a_i é permitido o início da entrega ou coleta no consumidor i . Caso a chegada do veículo no consumidor i se dê antes do instante a_i , o veículo deverá esperar. O veículo nunca poderá chegar depois do instante b_i , pois viola a restrição de tempo do problema.

O PRVMCJT consiste em definir um número de rotas de entrega tal que:

- a) Cada rota inicia e termina no depósito central;
- b) Cada cliente é visitado exatamente uma vez por exatamente um veículo;
- c) A demanda de determinado cliente a ser alocado não pode exceder a capacidade livre do veículo que está atendendo esta rota;
- d) A capacidade livre dos compartimentos deve ser maior que a demanda dos clientes;
- e) A janela de tempo $[a_i, b_i]$ não pode ser excedida, ou seja, um cliente não pode ser atendido após o tempo b_i ;
- f) O custo total da rota seja minimizado.

Para garantir o horário de início a_i da janela de tempo, o instante de chegada de determinado veículo a um cliente, não poderá ocorrer antes do tempo de chegada ao cliente anterior somado tempo de percurso entre os dois clientes, conforme representado na equação 6.1 a seguir.

$$t_j = t_i + t_{ij} - K(1-x_{ij}) \quad (6.1)$$

Onde:

t_j = instante de chegada ao cliente j

t_i = instante de chegada ao cliente anterior i

t_{ij} = tempo de deslocamento do cliente i ao cliente j

K = constante suficientemente grande para garantir que a fórmula seja realmente efetiva quando x_{ij} for igual a 1, ou seja, quando houver um veículo atendendo os clientes i e j .

O horário de término b_i da janela de tempo é garantido quando o instante de chegada de determinado veículo a um cliente ocorre dentro do limite de horário definido para o atendimento do mesmo, representado pela equação 6.2 a seguir.

$$a_i = t_i = b_i \quad (6.2)$$

A partir das definições anteriores podem-se inserir as restrições de janela de tempo no modelo matemático proposto por Póvoa (2005) para o Problema de Roteamento de Veículos com Multi-Compartimentos (PRVMC). Este modelo está representado na figura 6.1.

As restrições 6.4 e 6.5 garantem que cada cliente será servido exatamente uma vez. A continuidade da rota é garantida pela restrição 6.6, onde um veículo que chega num ponto de entrega qualquer deverá também partir deste mesmo ponto. A restrição 6.7 é a restrição da capacidade de carga do veículo. As restrições 6.8 e 6.9 asseguram que cada veículo é usado não mais do que uma vez. A restrição 6.10 assegura que o volume máximo de cada compartimento de cada veículo não será excedido. As restrições 6.11 e 6.12 são as restrições temporais, ou seja, garantem a janela de tempo.

A função objetivo será calculada com o somatório do custo por distância percorrida (Km rodado) mais o custo fixo do veículo. Será adotado ainda que a janela de tempo não poderá ser violada (*hard*).

$$\text{Minimize } \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^k x_{ij}^v d_{ij} a_v + x_{ij}^v F_v \quad (6.3)$$

Sujeito a

$$\sum_{i=0}^n \sum_{v=1}^k x_{ij}^v = 1 \quad j = 1, \dots, n \quad (6.4)$$

$$\sum_{j=0}^n \sum_{v=1}^k x_{ij}^v = 1 \quad i = 1, \dots, n \quad (6.5)$$

$$\sum_{i=0}^n x_{ip}^v - \sum_{j=0}^n x_{pj}^v = 0 \quad p = 1, \dots, n \text{ e } v = 1, \dots, k \quad (6.6)$$

$$\sum_{j=0}^n q_j \left(\sum_{i=0}^n x_{ij}^v \right) \leq Q_v \quad v = 1, \dots, k \quad (6.7)$$

$$\sum_{j=1}^n x_{0j}^v \leq 1 \quad v = 1, \dots, k \quad (6.8)$$

$$\sum_{i=1}^n x_{i0}^v \leq 1 \quad v = 1, \dots, k \quad (6.9)$$

$$\sum_{j=0}^n v_{cj} \left(\sum_{i=0}^n x_{ij}^v \right) \leq G_v^c \quad l = 1, \dots, k; c = 1, \dots, |C| \quad (6.10)$$

$$t_j \geq t_i + t_{ij} - K(1 - x_{ij}^v) \quad i = 1, \dots, n \text{ e } j = 1, \dots, k \quad (6.11)$$

$$a_i \leq t_i \leq b_i \quad i = 1, \dots, n \quad (6.12)$$

$$x_{ij}^v \in \{0, 1\} \quad \forall i, j, v \quad (6.13)$$

Onde:

n = número de clientes; k = número de veículos

c_{ij} = Custos de transporte do cliente i ao cliente j ; C = Conjunto de diferentes compartimentos

v_n = Conjunto do volume ocupado para os diferentes compartimentos para cada consumidor n

G_k = Conjunto de capacidade de volumes para cada tipo de compartimento c , do veículo v

t_j = Instante de chegada ao cliente j , t_i = Instante de chegada ao cliente i

t_{ij} = Tempo de viagem do cliente i ao cliente j ; K = constante

$$x_{ij}^l = \begin{cases} 1, & \text{se o veículo } l \text{ viaja do nó } i \text{ ao nó } j \\ 0, & \text{caso contrário} \end{cases}$$

q_i = tamanho do pedido do cliente i ; Q_l = capacidade do veículo l

Figura 6.1 – Modelo matemático (PRVMCJT)

6.1 METAHEURÍSTICA GRASP APLICADA NO PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM MULTICOMPARTIMENTOS E RESTRIÇÃO DE JANELA DE TEMPO.

6.1.1 Inicialização de rotas

A construção de soluções viáveis começa com o procedimento de inicialização de rotas também chamado de seleção de sementes. Nesta etapa um algoritmo irá classificar os clientes de acordo com a distância em relação ao depósito, e de acordo com a sua folga de tempo. Dentre estes clientes serão escolhidos aqueles que iniciarão as rotas, em cada rota haverá um cliente semente.

Kontoravdis (1994) apresenta um procedimento onde a seleção de rotas é iniciada a partir do número de clientes. Uma alternativa, aplicada por Póvoa (2005), é iniciar o procedimento em função do número de veículos disponíveis v . O procedimento adotado nesta pesquisa, mostrado na figura 6.2, utiliza duas informações para inicialização das rotas, número de veículos e de número clientes.

No primeiro passo do procedimento são construídas duas listas, L_{CH} e \underline{L}_{CH} . Para a construção destas listas foi utilizado o algoritmo de Graham para calcular o fecho convexo. L_{CH} engloba os clientes do convex hull e \underline{L}_{CH} contém os clientes restantes. Os clientes da lista \underline{L}_{CH} são organizados em ordem crescente de acordo com a sua folga de tempo $(b_i - t_{oi})$, onde b_i é o horário de encerramento da janela de tempo e t_{oi} é o tempo de viagem do depósito até o cliente i , assim estará no início desta lista o cliente que possui menor tempo para ser atendido. Valores pequenos de folga indicam que o cliente tem baixa flexibilidade para ser inserido numa rota, uma vez que o tempo de viagem do depósito até o cliente i é próximo do tempo de término da janela de tempo.

No passo seguinte o cliente $\in L_{CH}$ mais distante do depósito é definido como primeira semente, a primeira semente é sempre o cliente que está mais distante do depósito. Enquanto houver veículo disponível e o conjunto de clientes sementes for menor que número de veículos, ocorrerá o *loop* descrito no quarto passo.

Neste *loop*, a cada iteração, é encontrado o cliente $j \in L_{CH}$ que maximiza a distância entre j e as sementes existentes $\in S$, então calcula-se a distância d_j . A distância d_j é a mínima distância entre j e a semente $s \in S$, onde k é o último cliente

semente inserido em S . Nesta definiu-se o possível cliente $j \in L_{CH}$ que poderá ser a próxima semente.

Em seguida, de forma similar, será definida a distância d_i , sendo esta a mínima distância entre o primeiro cliente $i \in L_{CH}$ e a semente $s \in S$.

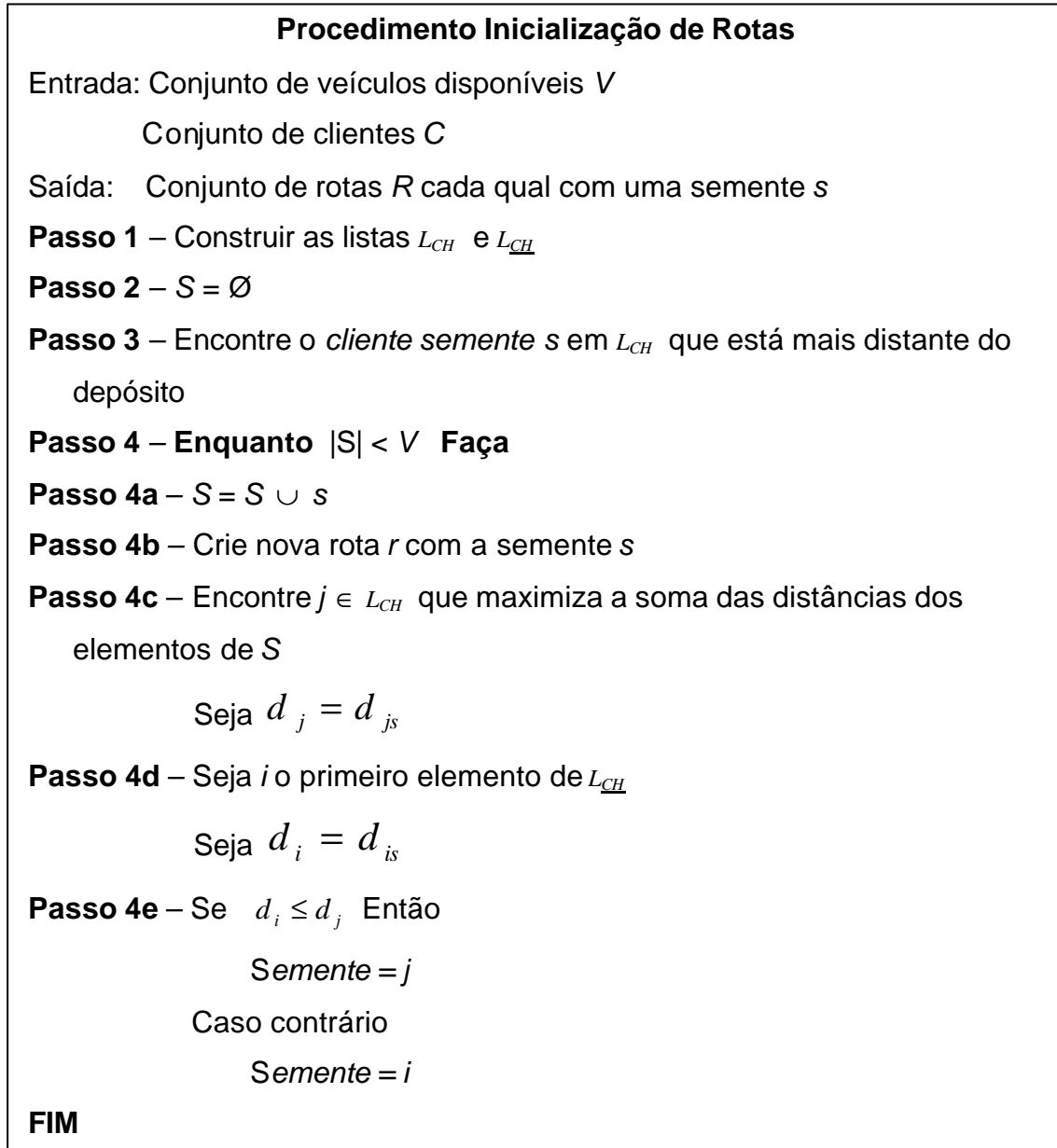


Figura 6.2 – Procedimento de inicialização

Após a definição, as distâncias d_i e d_j são comparadas. Se d_i for menor ou igual à d_j , a nova semente será o cliente j , caso contrário será o cliente i . Este procedimento permite reduzir a possibilidade de selecionar duas sementes que possam estar na mesma rota da solução ótima, ou seja, garante que não estará na mesma rota o cliente mais distante e o cliente com necessidade de atendimento mais cedo.

6.1.2 Procedimento construir solução

Após a inicialização das rotas, onde se terá n rotas e cada qual com um cliente cada, utiliza-se o procedimento principal de roteamento para avaliar e inserir os clientes nas melhores posições das rotas. Serão analisadas todas as possibilidades de inserção de um cliente entre os clientes i e j , assim na primeira iteração, quando a rota é formada apenas pelo depósito e o cliente semente, tem-se i (depósito) e j (cliente semente), será analisada a possibilidade de inserção dos clientes entre o depósito e o cliente semente, e também entre o cliente semente e o depósito. A posição viável será armazenada para posterior escolha.

No momento da análise, o algoritmo considera dois tipos de custo para rotear os clientes. O custo de inserção de determinado cliente entre os clientes i e j numa determinada rota utilizando o veículo v , e o custo de oportunidade de não se ter inserido o cliente na sua melhor posição e conseqüentemente melhor rota.

O custo de oportunidade pode ser definido como a diferença entre o custo real e o custo da melhor opção, no contexto de roteamento de veículos seria o custo de um determinado cliente não ser inserido na sua melhor posição (rota).

Na figura 6.3 é apresentado um procedimento de roteamento baseado na pesquisa de Chaovalitwongse *et al* (2003). No primeiro passo é calculado o menor custo de inserção do cliente k , na rota s , utilizando o veículo v . No passo seguinte é verificada a rota que tem o menor custo de inserção do cliente k . Então já se pode calcular o custo de oportunidade no próximo passo. Os clientes com custos de oportunidade altos são roteados primeiro, os clientes com menores custos de oportunidade devem esperar para serem inseridos, esta espera não acarreta perda significativa da melhor solução.

6.1.3 Inserção de clientes

Diversas estratégias são encontradas na literatura para o cálculo do custo de inserção de um cliente nas rotas. Oliveira *et al* (2003) utiliza uma heurística chamada Push-Forward Insertion Heuristic (PFIH), obtendo bons resultados quando os clientes estão agrupados. A PFIH foi introduzida por Solomon (1987) e basicamente calcula o custo de inserção de um cliente em determinada rota. O custo é calculado em função da localização geográfica, de acordo com o horário de fechamento da janela de tempo e o ângulo formado entre o cliente e o depósito central.

Nesta pesquisa a estratégia do cálculo de custo de inserção de clientes é baseada a partir da função apresentada na equação 6.14 abaixo:

$$c_{ij,k} = d_1 c^1_{ij,k} + d_2 c^2_{ij,k} + d_3 c^3_{ij,k} + d_4 c^4_{ij,k}; \quad (6.14)$$

onde d_1, d_2, d_3, d_4 são pesos não negativos e são calibrados conforme a especificidade da rota e/ou dos clientes, ou de acordo com a prioridade de atendimento.

A primeira parcela do custo representa a demanda, e é dada pela equação 6.15 abaixo:

$$c^1_{ij,k} = \sum_{c=1}^w MCL^{cp}_v - D^{cp}_j; \quad (6.15)$$

onde MCL^k_v é mínima capacidade livre do compartimento cp do veículo v , e D^{cp}_j é a demanda do cliente j pelos produtos do compartimento cp . Esta parcela irá mapear grandes demandas primeiramente com custos menores de atendimento, pois os clientes com maior demanda serão atendidos primeiro por veículos de menor custo.

A segunda parcela $c^2_{ij,k}$ mede o incremento da distância com a inserção do cliente k na rota, é definida a partir da equação 6.16 a seguir:

$$c^2_{ij,k} = d_{ik} + d_{kj} - d_{ij} \quad (6.16)$$

A terceira parcela contempla o deslocamento do horário quando um cliente k é inserido entre os clientes i e j . Esta parcela quando elevada irá dificultar a inserção de outros clientes na rota. Sendo assim $c^3_{ij,k}$ é definida de acordo com a equação 6.17 abaixo:

$$c^3_{ij,k} = t^k_j - t_j; \quad (6.17)$$

onde t^k_j é o novo horário de visita do cliente j com a inserção do cliente k .

Os custos $c^2_{ij,k}$ e $c^3_{ij,k}$ podem equilibrar o incremento da distância e o deslocamento do tempo de visita quando há a inserção de um novo cliente. O objetivo é compatibilizar o incremento de distância com tempo de espera para iniciar o atendimento do cliente.

Suponha-se que um cliente k está próximo aos clientes i e j , devido a proximidade, a parcela $c^2_{ij,k}$ será pequena e o incremento da distância da rota também será pequena, por outro lado, um pequeno incremento de $c^3_{ij,k}$ irá provocar um tempo de espera maior, e conseqüentemente a possibilidade de inserção de outros clientes na rota entre k e j .

A última parcela mede o custo fixo do veículo v e é definida pela equação 6.18 a seguir:

$$c_{ij,k}^4 = CF_v. \quad (6.18)$$

Procedimento Construir Solução

Entrada: Conjunto de veículos disponíveis v

Conjunto de clientes não roteados k

Conjunto de rotas inicializadas S (sementes)

Saída: Roteamento viável de todos os clientes n

Passo 1 – Para cada cliente não roteado do conjunto K e para cada rota inicializada S **Faça**

Encontre o custo mínimo de inserção do cliente k na rota s utilizando o veículo v $C_{k,s,v} = \min\{c_{ij,k}\} \forall ij \in s$

Passo 2 – Para cada cliente não roteado do conjunto K **Faça**

Encontre a rota s^* com o menor custo de inserção do cliente

$$kC_{k,s^*,v} = \min_{s \in S}^{v \in V} \{c_{k,s,v}\}$$

Passo 3 – Para cada cliente não roteado do conjunto K **Faça**

Encontre o custo de oportunidade $C = \sum_{s \in S} (C_{k,s,v} - C_{k,s^*,v}) \forall v \in V$

Passo 4 – Construa uma lista com os maiores custos de oportunidade e selecione aleatoriamente um destes clientes para ser roteado

Passo 5 – Se ainda existir clientes não roteados retorne ao passo 1

Fim

Figura 6.3 – Procedimento construir solução

6.1.4 Condições de viabilidade

Póvoa (2005) propõe duas condições de viabilidade para o PRVMC, a primeira contempla a restrição de capacidade do veículo e uma outra a cubagem dos compartimentos. Este trabalho insere uma terceira condição que contempla as restrições de janela de tempo. Esta condição irá avaliar se o cliente pode ser atendido dentro da sua restrição temporal, sendo que não será permitida a violação desta restrição (*hard*), e se o veículo chegar antes do horário de abertura da janela de tempo (*a_i*) deverá esperar.

A capacidade do veículo é definida segundo Kontoravdis (1994) e Póvoa (2005) pela equação 6.18 mostrada a seguir:

$$q_k \leq CM_v^i; \quad (6.18)$$

onde CM_v^i é capacidade mínima livre do veículo v quando parte do cliente i para o cliente k . Esta restrição direciona o atendimento para aqueles clientes que possuem demanda compatível com o restante de oferta de volume no veículo.

Póvoa (2005) define a condição de cubagem a partir da equação 6.19:

$$f_k \leq ?^{cp}_v; \quad (6.19)$$

onde f_k é a demanda (cubagem) do cliente k , e $?^{cp}_v$ é a capacidade do compartimento cp do veículo v . Assim está garantida a verificação da oferta de volume de cada compartimento.

As restrições temporais são apresentadas a seguir, e estão baseadas nas pesquisas de Kontoravdis (1994) e Chaovalitwongse (2003).

Será considerado que o depósito não possui restrição de janela de tempo, ou seja, não há limite de horário de chegada do veículo no depósito.

Como mencionado anteriormente, o intervalo da janela de tempo do cliente k é $[a_k, b_k]$, mas numa rota dinâmica este intervalo temporal pode sofrer variações por causa dos horários dos clientes anterior e posterior do cliente k . Esta variação está representada na equação 6.20 abaixo:

$$[a_n, b_n] = [\max(a_n, a_a + ts_a + t_{an}), \min(b_n, b_p - ts_p - t_{np})]; \quad (6.20)$$

Onde:

a é o cliente antecessor de n ;

p é o cliente posterior a n ;

ts_a e ts_p são os tempos de serviço dos clientes a e p ;

t_{an} é o tempo de deslocamento entre a e n ;

t_{np} é o tempo de deslocamento entre n e p .

Então seja uma rota qualquer que contenha o cliente i . Para que a inserção de um cliente na rota seja viável o veículo v deve chegar ao cliente i antes do fechamento da sua janela de tempo b_i . Seja, tl_i o tempo limite de chegada ao cliente i , e s o próximo cliente da rota depois de i . A equação 7.21 demonstra esta condição de viabilidade:

$$tl_i + t_{is} \leq tl_s \quad (6.21)$$

Também é considerada a condição de viabilidade de inserção do cliente k entre os clientes i e j que já estão inseridos em determinada rota. O veículo deve chegar ao cliente k antes do fechamento de sua janela de tempo b_k , e o horário de saída (t_k) de k mais o tempo de deslocamento de k até j (t_{kj}) deve ser menor ou igual

ao tempo limite de chegada a j (tl_j). Estas relações estão representadas nas equações 6.22 e 6.23 abaixo:

$$t_i + t_{ik} \leq b_k \quad (6.22)$$

$$\max(a_k, t_i + t_{ik}) + t_{kj} \leq tl_j; \quad (6.23)$$

onde t_i é o tempo de saída do cliente i .

Estas equações garantem se um novo cliente pode ser inserido numa determinada posição na rota sem violar a restrição de janela de tempo.

6.1.5 Procedimento de Busca Local

Como visto, o procedimento GRASP é formado por duas etapas, sendo elas a fase de construção e a fase de busca local. Na fase de construção foram definidos os custos de inserção de determinado cliente. O primeiro é o custo mínimo de inserir um cliente k em determinada rota s , e o outro é o custo de oportunidade de não inserir este mesmo cliente numa outra rota s^* onde se teria um custo final menor. O procedimento de busca local tentará eliminar as rotas $s \in S$. A eliminação da rota s ocorrerá quando todos os clientes nela alocados forem transferidos para outras rotas s^* . O procedimento deve ser iniciado pelas rotas que possuem o menor número de clientes.

A figura 6.4 apresenta um procedimento de busca local a partir da proposta de Kontoravdis (1994).

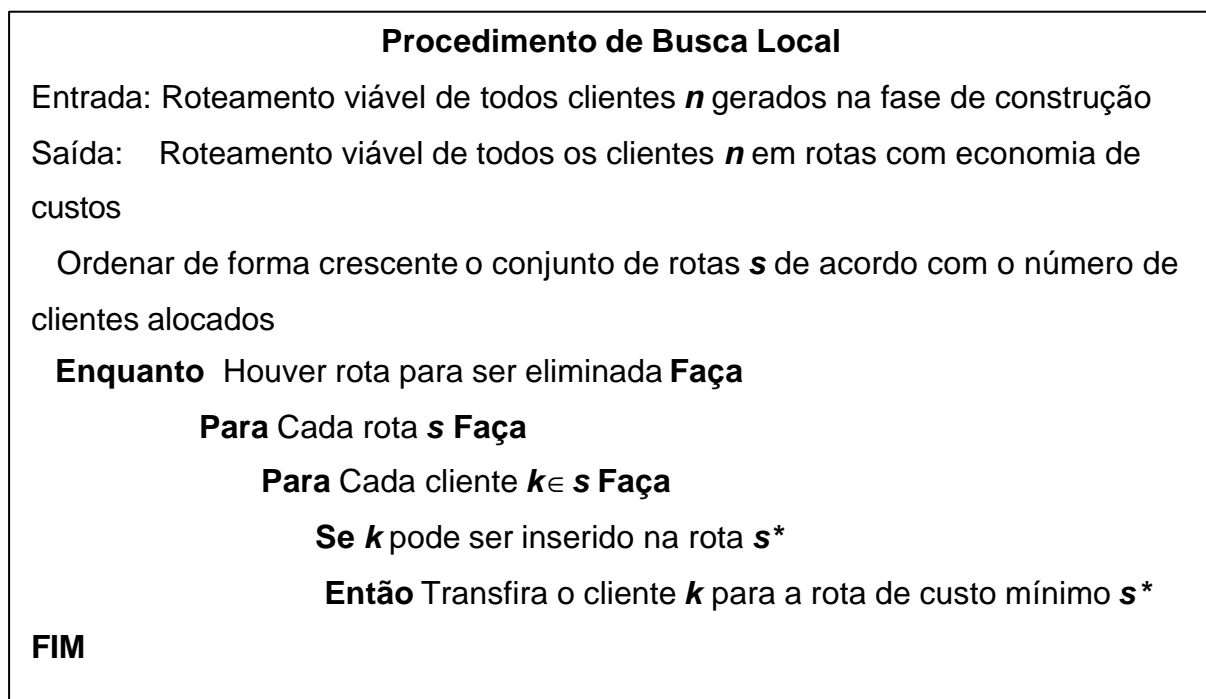


Figura 6.4 – Busca local

7. IMPLEMENTAÇÃO COMPUTACIONAL E TESTES

A partir de um modelo utilizado por Póvoa (2005), feita com modelagem em UML, algumas classes já desenvolvidas por Póvoa (2005) foram adaptadas e outras foram desenvolvidas para agregar ao modelo as restrições de janela de tempo.

Destaca-se que assim como o modelo de Póvoa (2005) serviu como base para esta heurística, o modelo desenvolvido com as restrições temporais pode ser utilizado numa variação do PRV que tenha as restrições contempladas nesta dissertação.

A implementação computacional do modelo descrito acima foi feita em *object pascal (Delphi)*. Foi desenvolvido um gerador aleatório de instâncias para avaliar o comportamento da heurística. Na figura 7.1 é mostrada a tela principal do programa desenvolvido. Nela pode-se observar três áreas principais, Veículos, Clientes e GRASP. Nos campos das áreas de Veículos e Clientes são gerados os respectivos arquivos, em GRASP são inseridos os parâmetros da heurística. O programa permite ainda a geração da lista de clientes do fecho convexo e dos clientes que não estão no fecho.

7.1 IMPLEMENTAÇÃO COMPUTACIONAL

Foi feita uma série de testes para avaliar o comportamento da heurística, sendo que a mesma pode ser dividida em testes que visam à comparação com algoritmos exatos (modelo descrito no capítulo 6) e algumas instâncias.

Todos os testes foram processados em um computador *AMD Duron (2.4Ghz)* com 512 Mb de memória *RAM*.

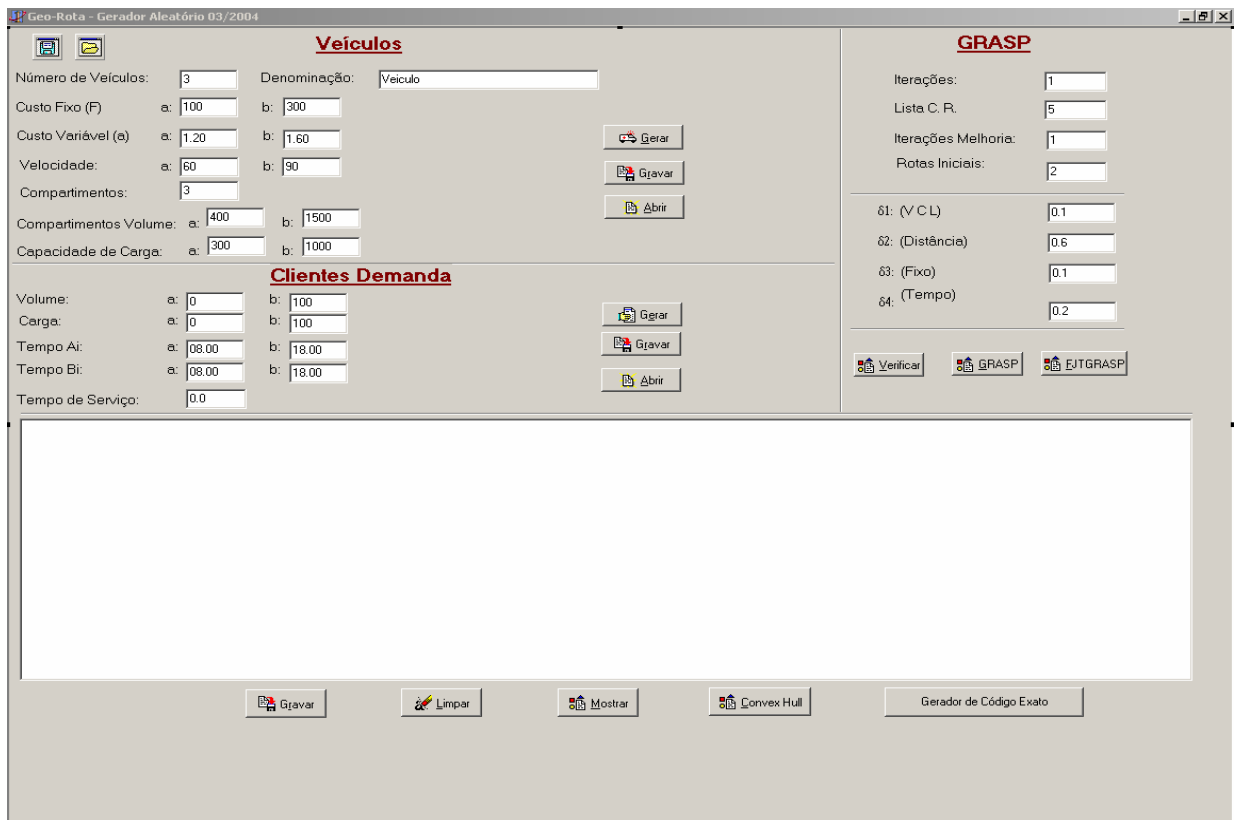


Figura 7.1 – Tela principal do programa

7.2 TESTES COMPUTACIONAIS

7.1.1 Testes Utilizando Algoritmos Exatos

Nesses testes o modelo matemático da instância é gerado pelo próprio programa de maneira automática, permitindo sua leitura no *Lindo*, que utiliza o algoritmo *branch and bound* para solução exata do problema. Sete instâncias foram geradas com 4, 5, 6, 7, 8, 9 e 10 clientes respectivamente, utilizando o gerador aleatório que utiliza a distribuição uniforme. Foram utilizados três veículos com dois compartimentos cada. A GRASP foi testada com os seguintes parâmetros: $T=200$, $\delta_1=0.1$, $\delta_2=0.6$, $\delta_3=0.1$ e $\delta_4=0.2$ e $\lambda=5$. Os resultados estão na Tabela 7.1.

A heurística desenvolvida alcançou o valor ótimo em todos os casos. Destaca-se que para a instância de 10 clientes (Inst_10) não foi possível encontrar a solução ótima, depois de 4 horas de processamento, o *Lindo* apresentou estouro de memória sem nenhuma solução inteira viável, inviabilizando a sua comparação.

Tabela 7.1 – Resultados algoritmo exato					
	Exato		GRASP		
Instância	Custo	Tempo	Custo	Tempo	Diferença %
Inst_4	238,76	00:00:01	238,76	00:00:01	0
Inst_5	191,86	00:00:01	191,86	00:00:01	0
Inst_6	232,74	00:00:02	232,74	00:00:01	0
Inst_7	294,66	00:00:15	294,66	00:00:01	0
Inst_8	217,72	00:00:20	217,72	00:00:01	0
Inst_9	194,61	02:04:30	194,61	00:00:01	0
Inst_10	-----	04:25:03	192,87	00:00:01	-----

7.1.2 Testes com Instâncias Aleatórias – Dois Compartimentos

Os testes foram feitos com problemas contendo 10, 20, 50, 100, 120 clientes. Essas instâncias foram baseadas nos problemas de Chiristofides *et al.* (1979). Foram geradas de forma aleatória as demandas de cada compartimento, a cubagem da carga e a janela de tempo de cada cliente, o custo fixo, o custo variável e a velocidade de cada veículo, bem como a capacidade de carga e a cubagem máxima de cada compartimento. Para realização dos testes não houve preocupação em definir parâmetros que caracterizassem uma situação real, este aspecto não tem nenhuma influência na validação da heurística.

Os primeiros testes objetivaram avaliar o comportamento dos diversos parâmetros da heurística. O número de iterações (**T**) foi testado em 50, 100, 200 e 500. Já a lista restrita de candidatos (**l**) foi testada com os seguintes valores: 1 (totalmente guloso), 2, 5, 10 e 10000 (totalmente aleatório). Foram gerados cinco resultados para avaliar o comportamento médio dos parâmetros e do Tempo Médio de Espera (TME), ou seja, o tempo que o veículo deve esperar no cliente para iniciar o serviço de entrega. Os valores de d_1 , d_2 , d_3 e d_4 foram fixados em 0.1, 0.6, 0.1 e 0.2. As tabelas 7.2, 7.3, 7.4 e 7.5 mostram os resultados médios dos testes. Para as instâncias de 10, 20 e 50 clientes foi utilizada frota de três veículos diferentes com dois compartimentos, e com capacidade de carga variando entre 2500 a 6000 e a cubagem de cada compartimento variando entre 2000 a 3000. Para as instâncias de 100 e 120 clientes utilizou-se frota com 5 veículos diferentes com dois compartimentos com capacidade de carga variando entre 2500 a 6000 e a cubagem de cada compartimento variando entre 2000 a 3000. Em todos os veículos a velocidade foi definida aleatoriamente variando entre 70 a 90. A geração da janela

de tempo de todos os clientes se deu da seguinte forma. Horário de abertura da janela (a) variando entre 08:00 e 14:00, e tempo de fechamento da janela variando entre 10:00 e 18:00. O tempo de serviço para todos os clientes foi considerado nulo.

Tabela 7.2 – 50 iterações

L.R.C	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
1	446.42	1007.65	1515.17	2270.9	3110.82
2	444.022	888.722	1467.62	1966.03	2840.37
5	443.13	816.482	1471.002	2057.89	2788.23
10	442.02	887.258	1523.652	2138.52	2794.00
10000	442.73	848.96	1609.956	2333.18	3290.59
Tempo	00:00:01	00:00:01	00:00:07	00:00:58	00:01:27
TME	00:22:36	00:09:22	00:06:05	00:03:10	00:02:35

Tabela 7.3 – 100 iterações

L.R.C	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
1	446.42	1007.65	1515.17	2270.9	2975.93
2	444.47	955.14	1450.15	1922.20	2927.28
5	442.26	870.51	1471.09	2039.18	2873.85
10	441.74	871.15	1496.75	2056.52	2755.71
10000	441.74	883.90	1626.90	2252.73	3342.44
Tempo	00:00:01	00:00:01	00:00:16	00:01:48	00:02:58
TME	00:22:15	00:12:10	00:06:00	00:02:36	00:02:58

Tabela 7.4 – 200 iterações

L.R.C	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
1	446.42	1007.65	1515.17	2270.9	3110.82
2	441.74	915.172	1436.834	1934.49	2835.49
5	441.74	878.298	1441.87	2005.07	2821.28
10	441.74	813.678	1451.92	2083.89	2863.59
10000	441.74	819.564	1567.34	2258.79	3360.25
Tempo	00:00:01	00:00:03	00:00:31	00:03:26	00:05:40
TME	00:22:53	00:10:50	00:06:07	00:02:45	00:02:34

Tabela 7.5 – 500 iterações

L.R.C	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
1	446.42	1007.65	1515.17	2270.9	3110.82
2	441.74	902.24	1435.23	1908.28	2793.68
5	441.74	874.0125	1425.96	1984.02	2793.01
10	441.74	867.9575	1447.43	2075.20	2783.89
10000	441.74	813.22	1518.55	2209.5925	3202.65
Tempo	00:00:01	00:00:06	00:01:18	00:08:51	00:13:12
TME	00:22:30	00:09:32	00:06:09	00:02:45	00:02:26

Na instância de 10 clientes (Inst1_10) a demanda foi atendida por um veículo, a instância de 20 clientes (Inst1_20) e de 50 clientes (Inst1_50) a demanda foi atendida por dois veículos. A instância seguinte de 100 clientes (Inst1_100) foi atendida por três veículos, e a última de 120 clientes (Inst1_120) foi atendida por quatro. Em todas as situações todos os clientes foram roteados.

A tabela 7.6 mostra o resumo dos melhores resultados a partir da média dos 5 resultados gerados, encontrados para cada variação do número de iterações, percebe-se que não houve altas variações no resultado, isso fica mais evidenciado no gráfico abaixo (Fig. 7.2). Sugerindo que o número de iterações não precisa ser alto para que se obtenha uma solução com boa qualidade.

Tabela 7.6 – Resultados Médios – Iterações

Iterações	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
50	442.02	816.482	1467.62	1966.03	2788.23
100	<u>441.74</u>	870.51	1450.15	1922.20	<u>2755.71</u>
200	441.74	813.678	1436.83	1934.49	2821.28
500	441.74	<u>813.22</u>	<u>1425.96</u>	<u>1908.28</u>	2783.89

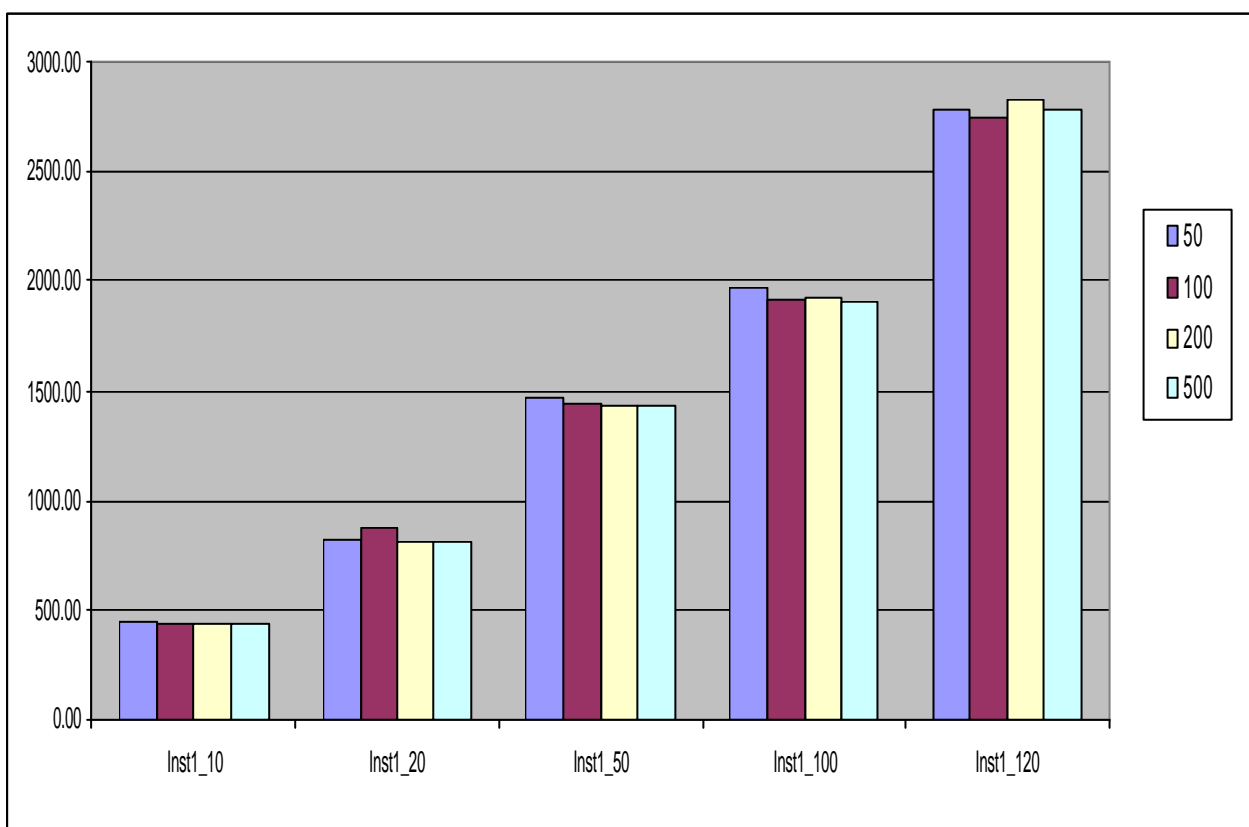


Fig. 7.2 – Gráfico Comparativo – Iterações

A tabela 7.7 mostra os melhores resultados a partir da média dos 5 resultados gerados para cada variação da lista restrita de candidatos, percebe-se que os

melhores resultados foram obtidos com valores entre 2, 5 e 10 com uma leve tendência para 5. Os piores resultados foram obtidos na solução gulosa ($\lambda=1$) e quase sempre na totalmente aleatória ($\lambda=100000$). O gráfico da figura 7.3 demonstra a diferença dos resultados obtidos em utilizar $\lambda=1, 2, 5, 10$ e 100000.

Tabela 7.7 – Resultados Médios – L.R.C.

L.R.C	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
1	446.42	1007.65	1515.17	2270.90	2975.93
2	441.74	888.72	1435.23	1908.28	2793.68
5	441.74	816.48	1425.96	1984.02	2788.23
10	441.74	813.68	1447.43	2056.52	2755.71
100000	441.74	813.22	1518.55	2209.59	3202.65

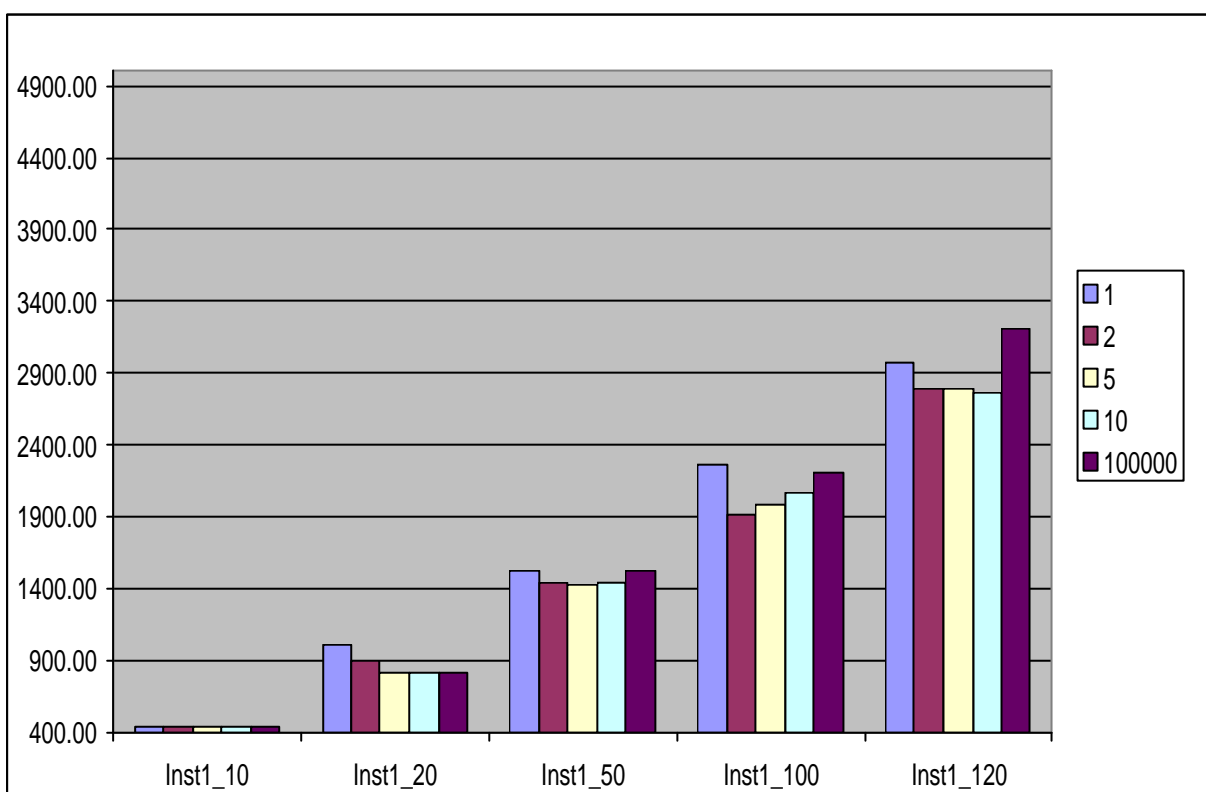


Fig. 7.3 – Gráfico Comparativo – Lista Restrita de Candidatos

A tabela 7.8 resume os resultados médios para cada instância, apesar da ausência de outras pesquisas para que seja feita uma comparação, o algoritmo apresenta eficiência em tempo computacional, tornando viável sua aplicação em problemas reais. Pode-se observar também que o Tempo Médio de Espera (TME) vai diminuindo na medida que o número de clientes é maior.

Tabela 7.8 – Resultados Médios			
Instância	Custo	Tempo	TME
10	442.68	00:00:01	00:22:34
20	867.95	00:00:03	00:10:29
50	1468.47	00:00:33	00:06:05
100	2085.86	00:03:46	00:02:49
120	2903.24	00:05:49	00:02:38

Para avaliar os parâmetros δ_1 , δ_2 , δ_3 e δ_4 foi fixado o número de iterações (T) em 100 e a lista restrita de candidatos (λ) em 5. Para cada variação dos parâmetros (δ_1 , δ_2 , δ_3 e δ_4) foram gerados cinco resultados, a Tabela 7.9 demonstra o resultado médio dos testes.

Tabela 7.9 – Avaliação dos parâmetros δ_1 , δ_2 , δ_3 e δ_4 (Inst_1)					
$d_1; d_2; d_3; d_4$	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
0.1 / 0.6 / 0.1 / 0.2	442.26	870.51	1471.09	2039.18	2873.85
0.0 / 0.0 / 0.5 / 0.5	383.44	855.50	1711.35	2394.32	3033.97
0.0 / 0.6 / 0.0 / 0.4	714.95	901.23	1380.41	1842.07	2862.24
0.0 / 0.6 / 0.4 / 0.0	340.04	901.93	1382.54	1905.57	2885.22
0.0 / 0.7 / 0.0 / 0.3	714.95	901.49	1397.91	1953.00	2779.98
0.0 / 0.7 / 0.1 / 0.2	339.63	901.23	1396.01	1821.01	2810.10
0.1 / 0.7 / 0.0 / 0.2	441.74	858.37	1441.88	1981.61	2813.56

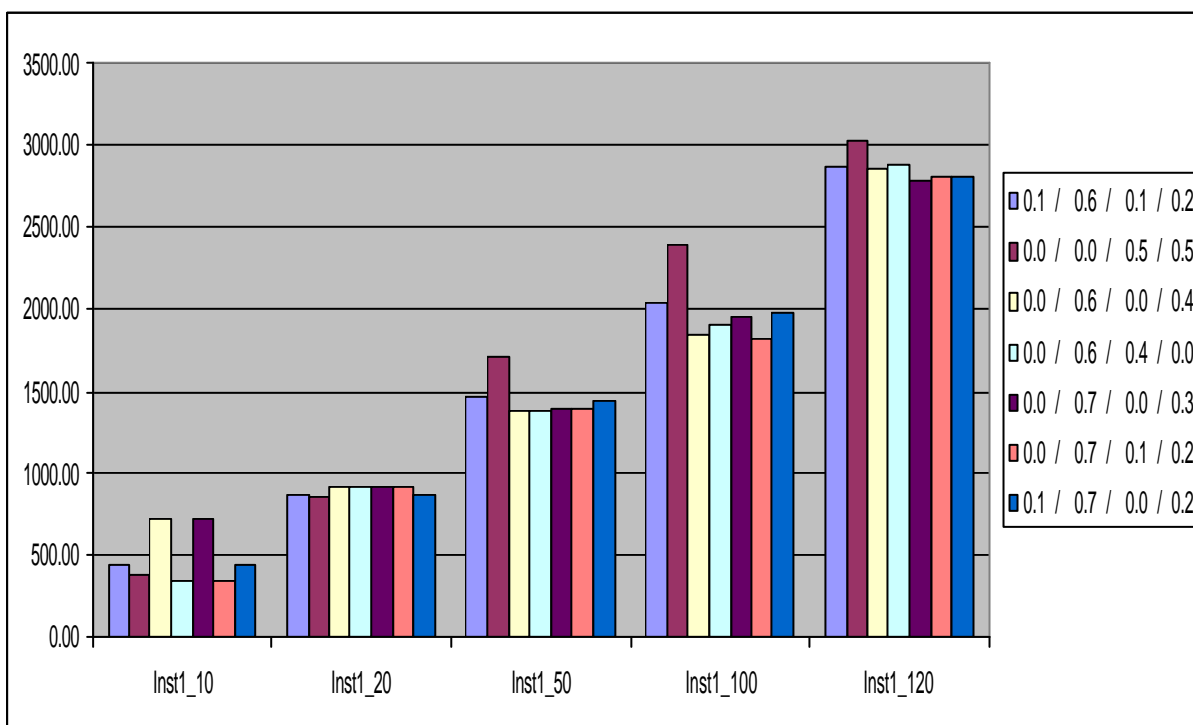


Fig. 7.4 – Gráfico Comparativo – Parâmetros δ_1 , δ_2 , δ_3 e δ_4

Conforme fig. 7.4, a heurística com os parâmetros $\delta_1= 0.0$ $\delta_2= 0.0$, $\delta_3= 0.5$ e $\delta_4= 0.5$ apresentou as piores soluções, principalmente nos problemas com mais clientes (50, 100 e 120) onde há um aumento de deslocamento dos horários dos clientes. Percebe-se também que os casos com melhor solução foram os que o peso para distância percorrida foi mais elevado, indicando a importância da distância percorrida para a qualidade da solução. Outro aspecto observado é que na maioria dos casos o tempo e o custo fixo tiveram a mesma importância na solução final. Nas figuras 7.5, 7.6, 7.7, 7.8 e 7.9 estão demonstrados os gráficos que também representam estes testes.

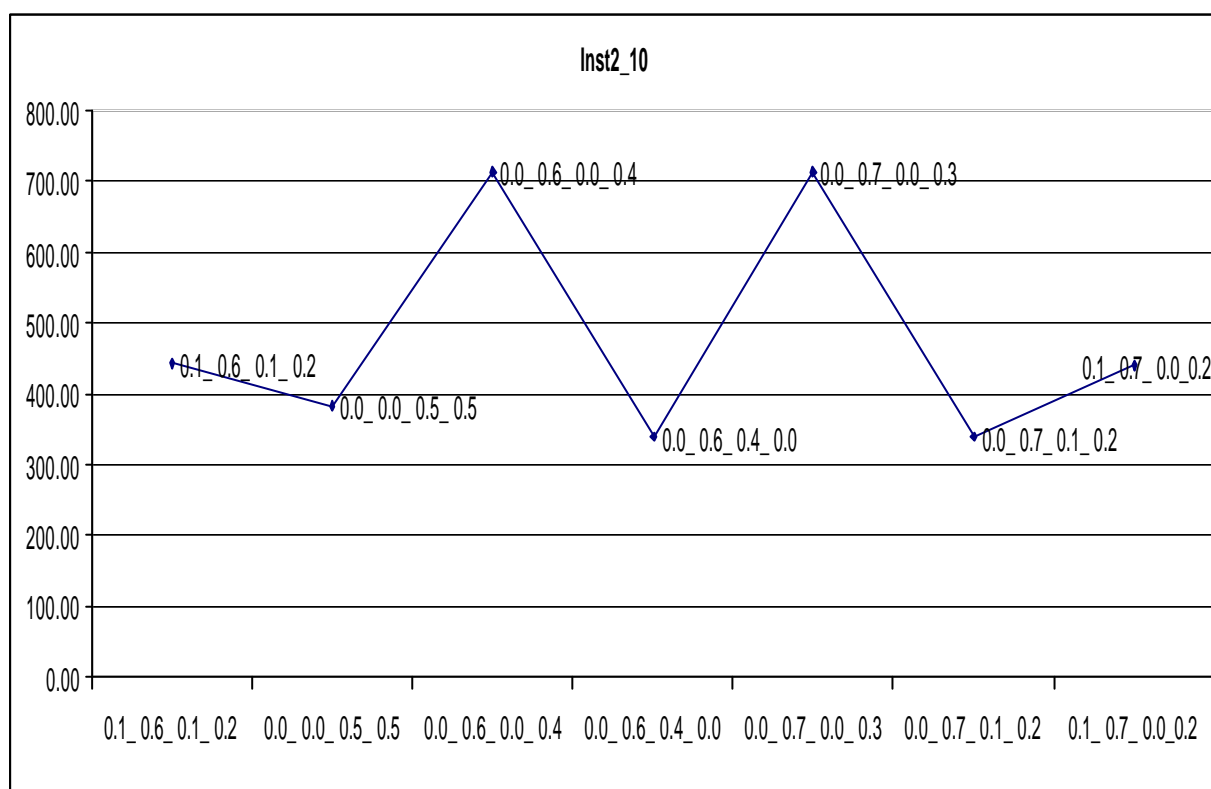


Fig. 7.5 – Variação da função objetivo - δ_1 , δ_2 , δ_3 e δ_4 (Inst_1_10)

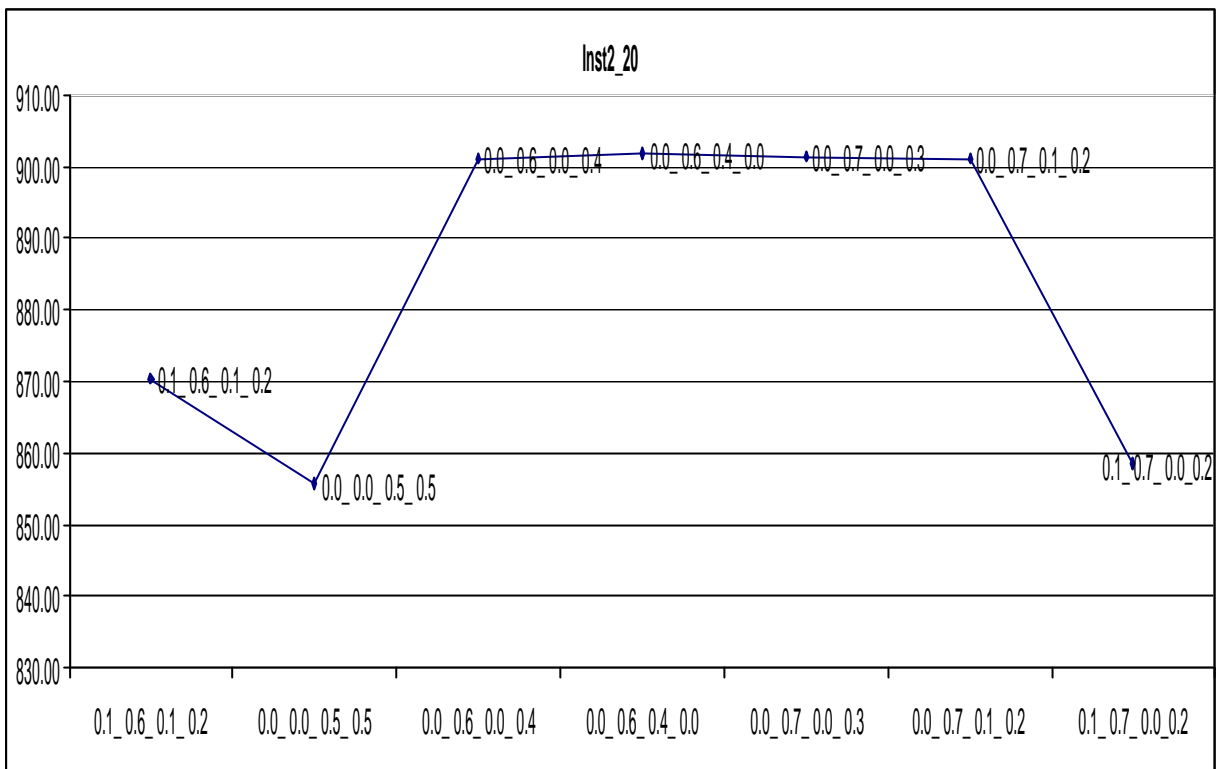


Fig. 7.6 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_20)

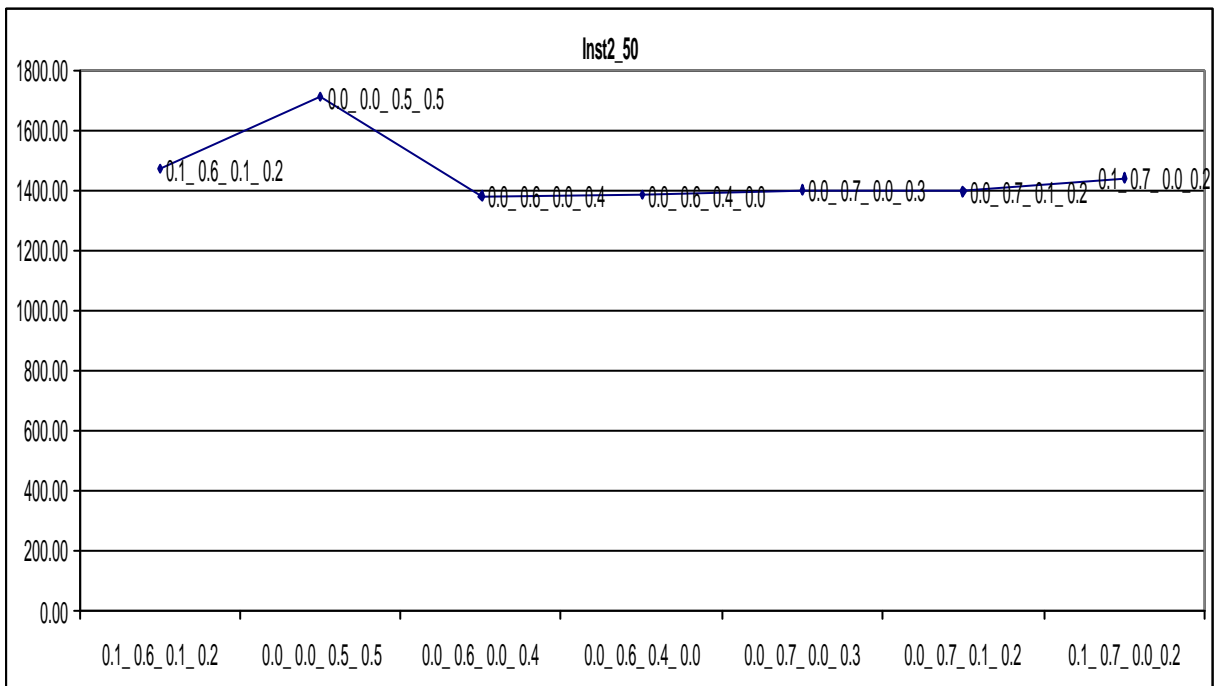


Fig. 7.7 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_50)

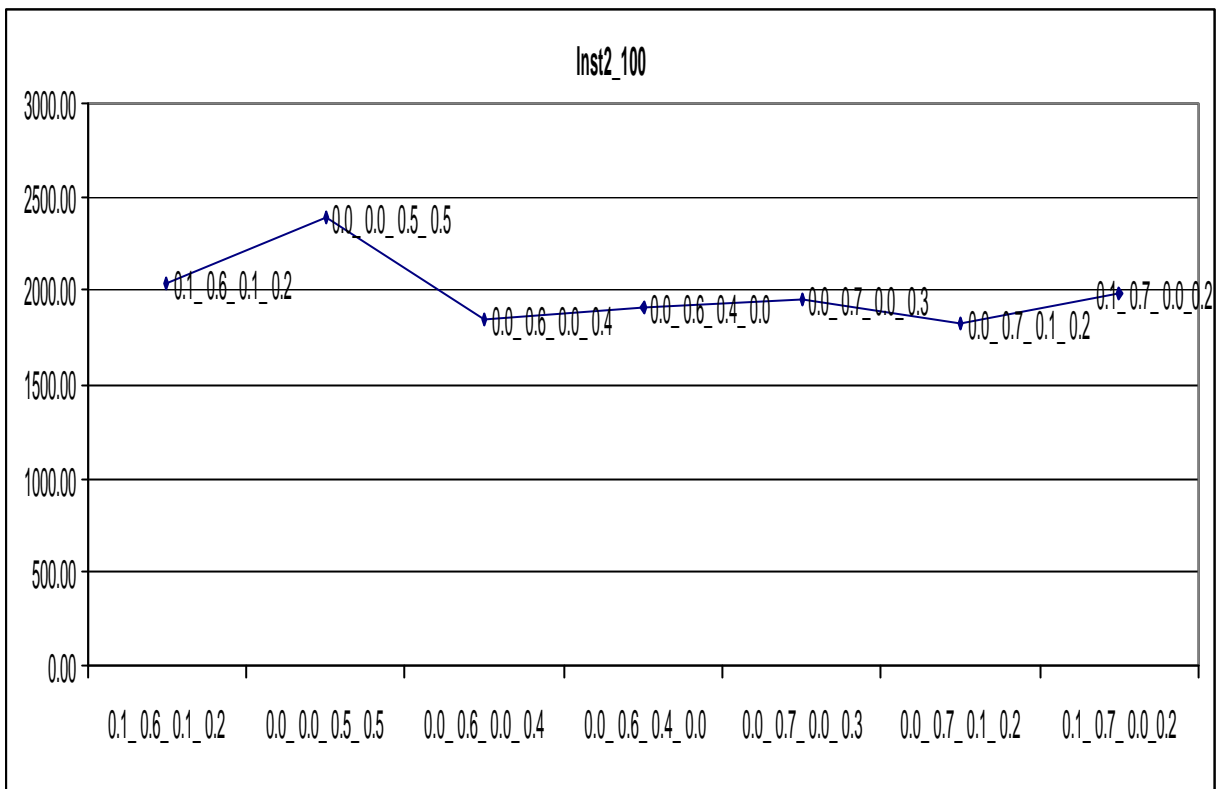


Fig. 7.8 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_100)

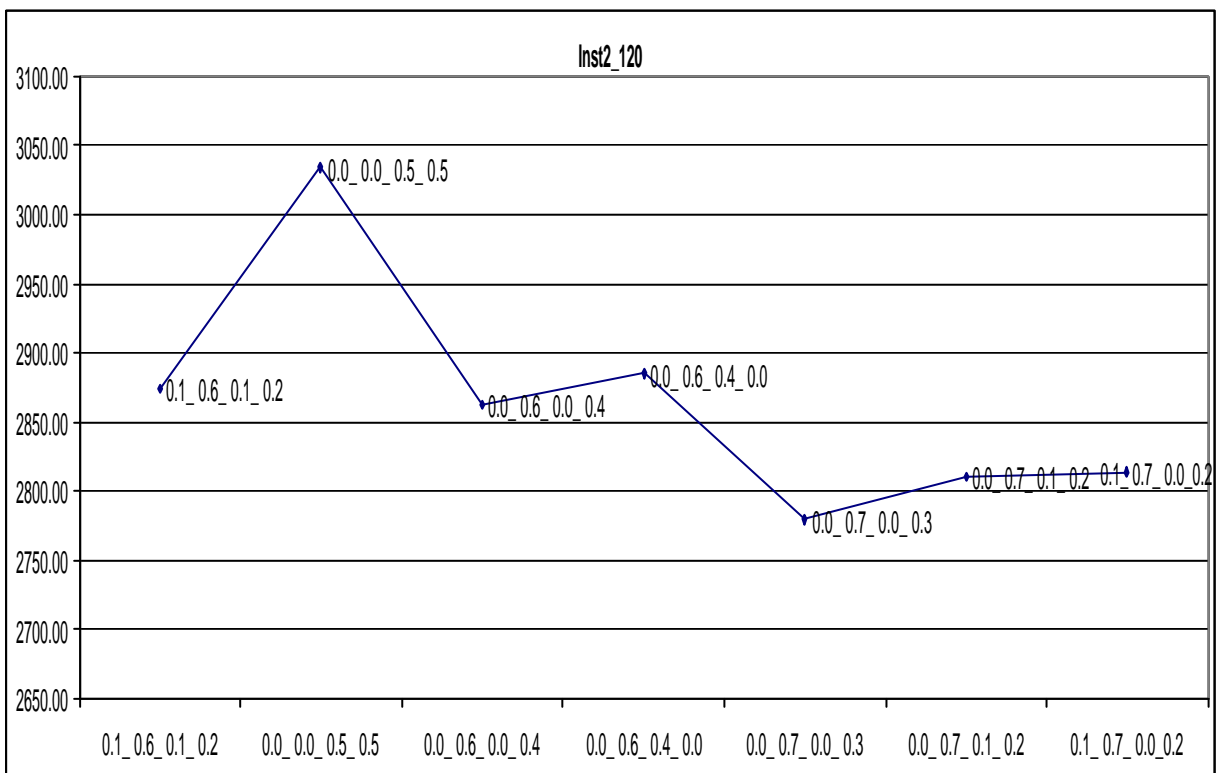


Fig. 7.9 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_1_120)

Para melhor analisar os parâmetros foram feitos mais testes com o número de iterações fixado em $T= 200$ e a lista restrita de candidatos (λ) foi testada com os

seguintes valores: 2, 5, 6, 7, 9 e 10. Foram descartados os valores de $\lambda = 1$ (totalmente guloso) e $\lambda = 100000$ (totalmente aleatório). Foram realizados cinco testes para cada variação de λ . A tabela 7.10 demonstra a média obtida de cada resultado, já as figuras 7.10, 7.11, 7.12, 7.13 e 7.14 demonstram a variação da função objetivo em relação a λ .

Tabela 7.10 – Resultados Médios L.R.C – Inst_1

L.R.C (λ)	Inst_1_10	Inst_1_20	Inst_1_50	Inst_1_100	Inst_1_120
2	441.74	915.17	1436.83	1934.49	2835.49
5	441.74	878.30	1441.87	2005.07	2821.28
6	441.74	878.45	1461.11	2020.63	2752.44
7	441.74	845.87	1424.29	2036.94	2865.59
9	441.74	852.90	1447.03	2044.28	2790.94
10	441.74	813.68	1451.92	2083.89	2863.59

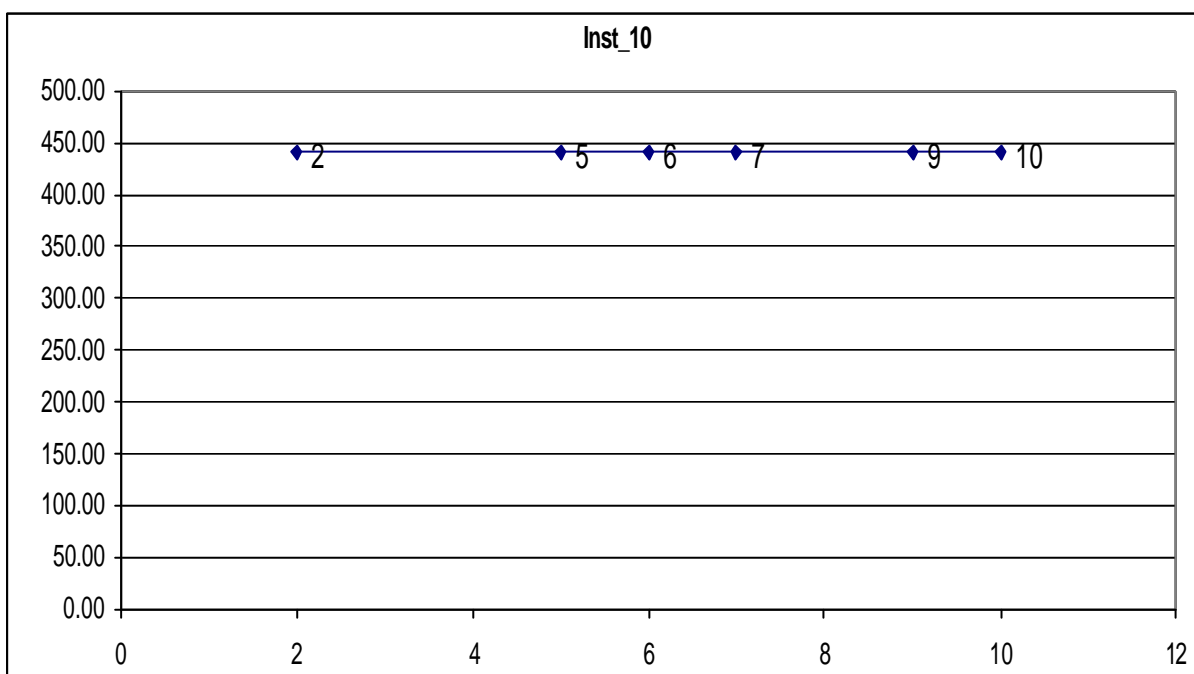


Fig. 7.10 – Variação da função objetivo – L.R.C – Inst_1_10 Clientes

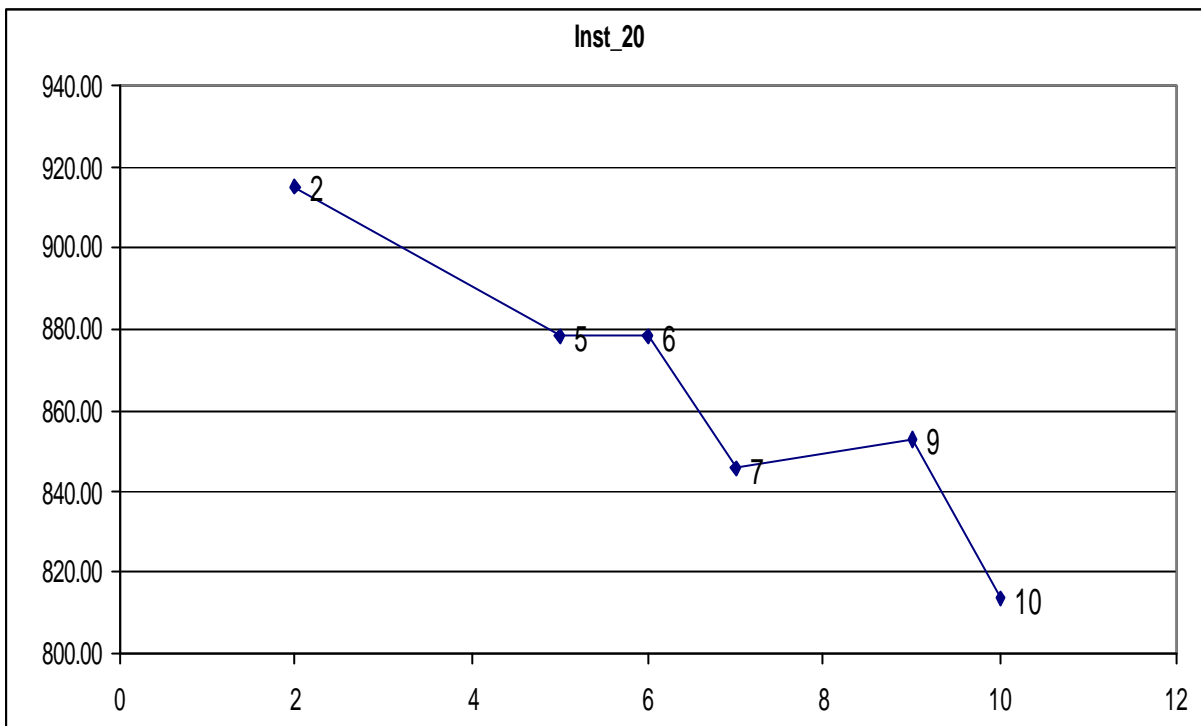


Fig. 7.11 – Variação da função objetivo – L.R.C – Inst_1_20 Clientes

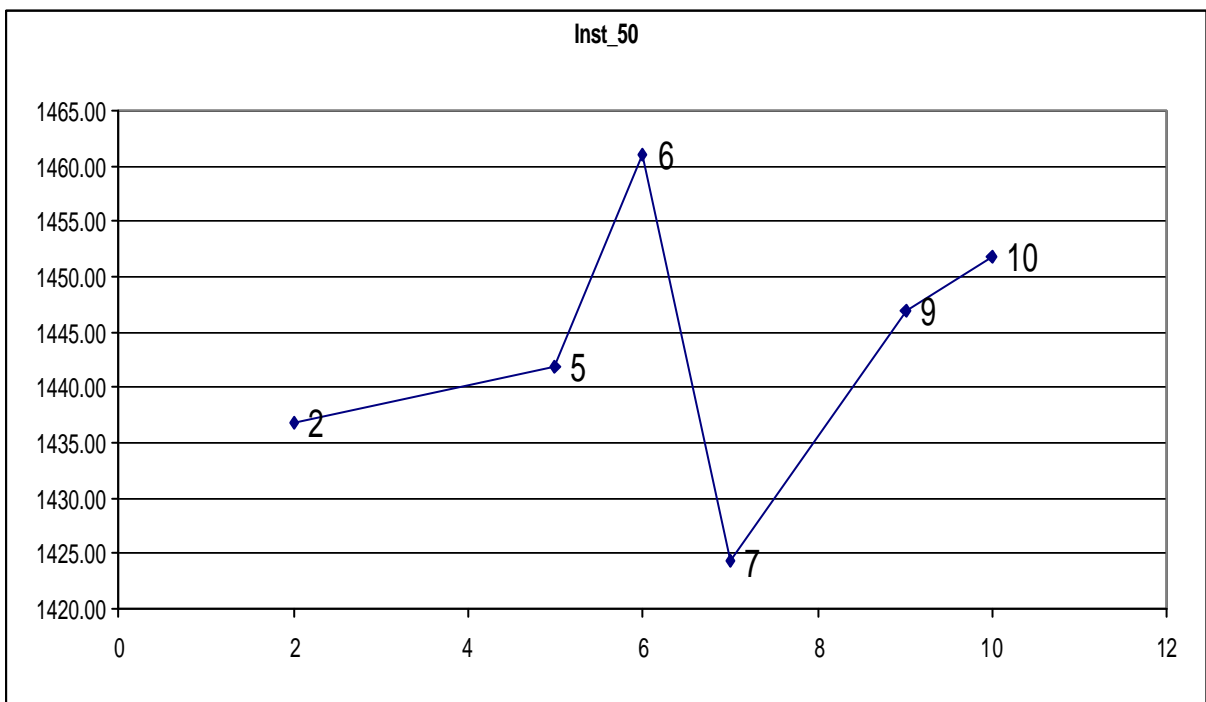


Fig. 7.12 – Variação da função objetivo – L.R.C – Inst_1_50 Clientes

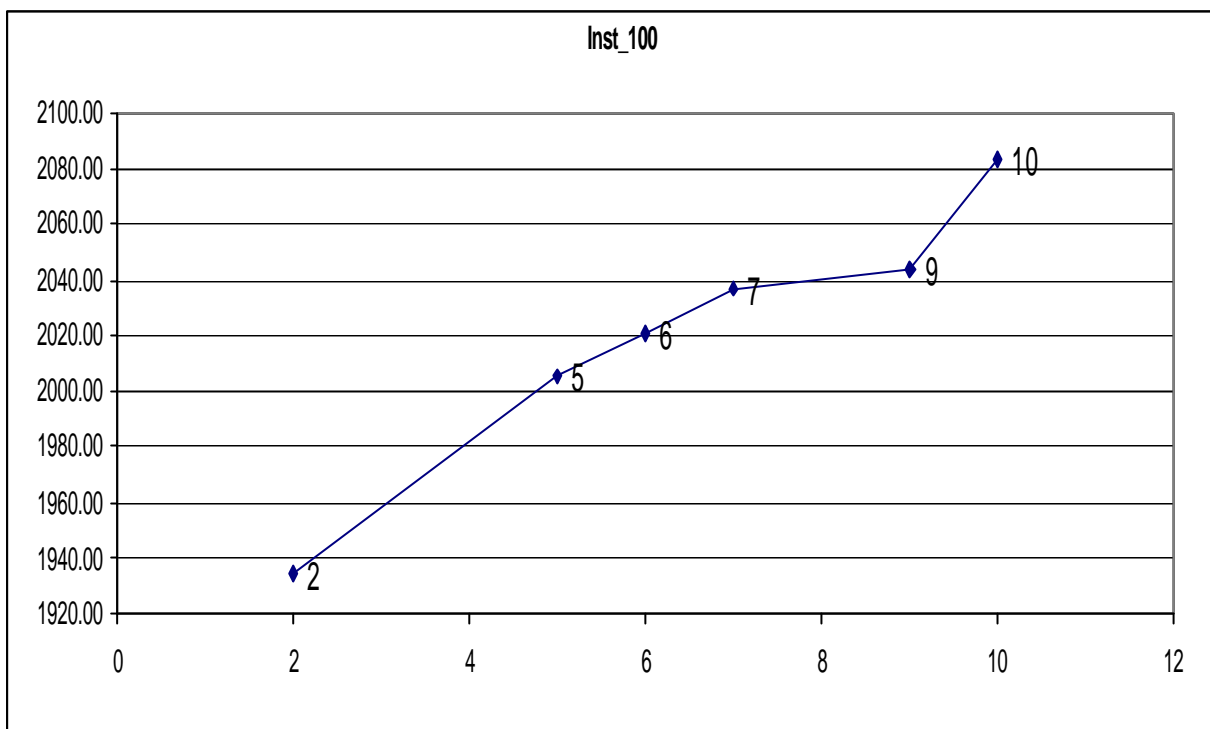


Fig. 7.13 – Variação da função objetivo – L.R.C – Inst_1_100 Clientes

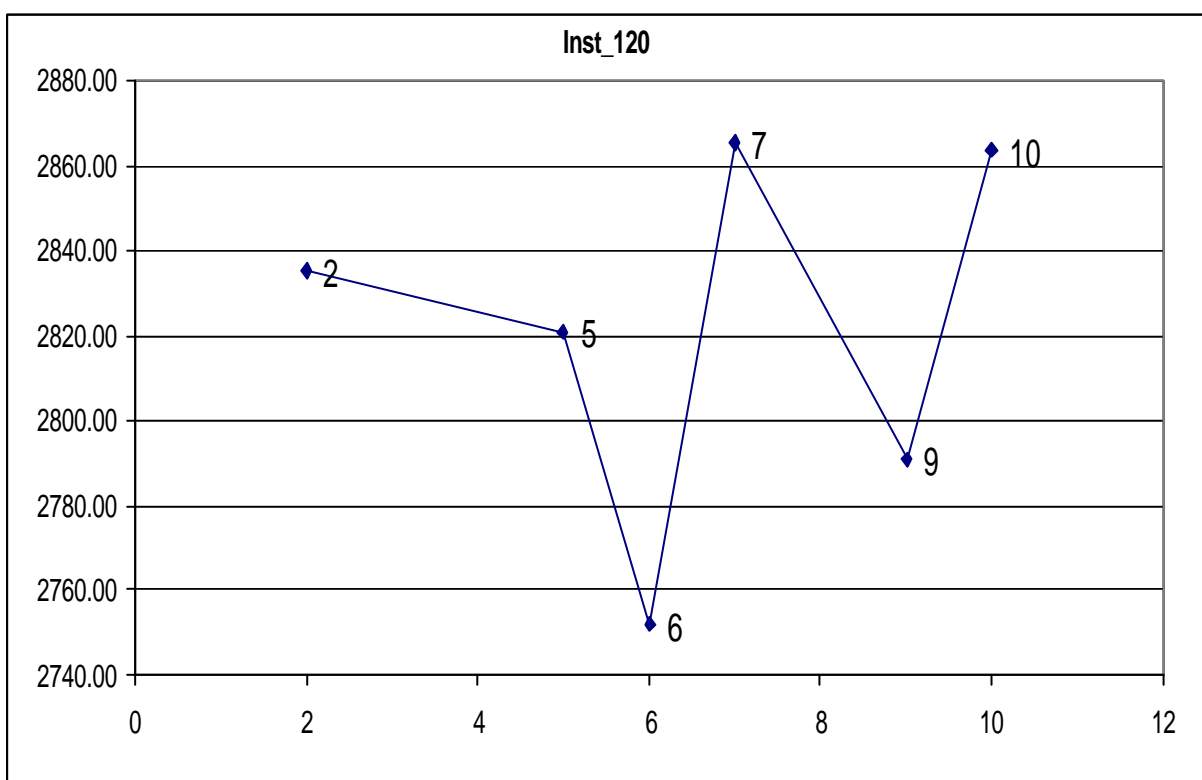


Fig. 7.14 – Variação da função objetivo – L.R.C – Inst_1_120 Clientes

Analisando os gráficos acima, percebe-se que na maioria dos casos as melhores soluções apareceram para valores mais alto da lista restrita de candidatos (λ), mas mesmo assim a variação em torno do melhor resultado é baixa.

7.1.3 Testes com Instâncias Aleatórias – Três Compartimentos

Os testes também foram feitos com problemas contendo 10, 20, 50, 100, 120 clientes, e com as mesmas instâncias baseadas nos problemas de Chiristofides *et al.* (1979). Assim como no exemplo anterior foram geradas de forma aleatória as demandas de cada compartimento, a cubagem da carga e a janela de tempo de cada cliente, o custo fixo, o custo variável e a velocidade de cada veículo, bem como a capacidade de carga e a cubagem máxima de cada compartimento.

Para atender o conjunto de problemas Inst_2 foram gerados veículos com capacidade de carga e volume de cubagem menor, e clientes com janela de tempo também menor. Para as instâncias de 10 e 20 clientes utilizaram-se três veículos com três compartimentos e com capacidade de carga e cubagem de cada compartimento variando entre 1000 a 2000, já para a instância de 50 clientes foi utilizado uma frota de cinco veículos com três compartimentos, com capacidade de carga e cubagem de cada compartimento variando entre 1000 a 2000. Para as instâncias de 100 e 120 clientes foram utilizados frota com dez veículos e três compartimentos, com as mesmas características de carga e cubagem anteriores, exceto na frota utilizada pela instância de 120 clientes a capacidade de carga variou entre 1000 e 3000. Todos os veículos tiveram suas velocidades definidas aleatoriamente entre 70 e 90. Os horários de abertura e fechamento de janela foram gerados entre 08:00 e 18:00.

Assim como na primeira instância (Inst_1), o número de iterações (**T**) foi testado em 50, 100, 200 e 500. Já a lista restrita de candidatos (**l**) foi testada com os seguintes valores: 1 (totalmente guloso), 2, 5, 10 e 10000 (totalmente aleatório). Os valores de d_1 , d_2 , d_3 e d_4 foram fixados em 0.1, 0.6, 0.1 e 0.2. As Tabelas 7.11, 7.12, 7.13 e 7.14 mostram os resultados médios dos testes.

Tabela 7.11 – 50 Iterações

L.R.C	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
1	680.90	1143.23	2585.97	4087.27	4411.29
2	607.46	1020.972	2095.49	3754.51	3787.26
5	648.03	1023.764	2199.65	3866.42	3938.87
10	612.26	1045.606	2270.762	3930.81	3934.04
10000	620.02	1031.76	2389.568	4426.82	4619.60
Tempo	00:00:01	00:00:01	00:00:03	00:00:17	00:00:31
TME	00:17:35	00:36:24	00:23:08	00:26:12	00:13:15

Tabela 7.12 – 100 iterações

L.R.C	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
1	646.94	1062.68	2256.26	3980.49	4065.91
2	611.62	<u>1031.90</u>	<u>2121.15</u>	<u>3741.45</u>	<u>3792.51</u>
5	607.21	1045.62	2175.96	3813.73	3844.89
10	<u>587.25</u>	1046.61	2320.17	3909.40	4030.74
10000	596.43	1047.24	2382.58	4417.08	4506.13
Tempo	00:00:01	00:00:01	00:00:08	00:00:34	00:00:40
TME	00:18:12	00:36:26	00:23:21	00:27:45	00:13:55

Tabela 7.13 – 200 iterações

L.R.C	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
1	646.94	1062.68	2256.26	3980.49	4065.91
2	610.842	<u>1020.442</u>	<u>2029.82</u>	<u>3869.70</u>	<u>3749.31</u>
5	<u>592.606</u>	1021.346	2108.15	3927.48	3810.86
10	586.138	1028.046	2193.40	3901.16	4777.69
10000	594.074	1047.782	2360.90	4398.37	4397.57
Tempo	00:00:01	00:00:01	00:00:14	00:01:08	00:03:19
TME	00:18:24	00:41:32	00:24:17	00:27:02	00:13:25

Tabela 7.14 – 500 iterações

L.R.C	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
1	646.94	1062.68	2256.26	3980.49	4065.91
2	608.57	1022.722	<u>2096.72</u>	3801.19	<u>3734.63</u>
5	<u>578.17</u>	<u>1019.476</u>	2128.58	<u>3795.53</u>	3757.83
10	578.17	1029.612	2251.93	3931.12	3931.92
10000	578.17	1034.50	2336.28	4358.602	4361.82
Tempo	00:00:01	00:00:01	00:00:39	00:03:00	00:05:05
TME	00:17:02	00:40:32	00:22:28	00:27:38	00:13:16

Na instância de 10 clientes (Inst1_10) a demanda foi atendida por dois veículos, a instância de 20 clientes (Inst1_20) a demanda foi atendida por três veículos, a de 50 clientes (Inst1_50) a demanda foi atendida por cinco veículos. A instância seguinte de 100 clientes (Inst1_100) foi atendida por dez veículos, e a última de 120 clientes (Inst1_120) foi atendida por oito. Em todas as situações todos os clientes foram roteados.

A tabela 8.15 mostra o resumo dos melhores resultados a partir da média dos 5 resultados gerados, encontrados para cada variação do número de iterações, assim como nos testes da Inst_1 não houve altas variações no resultado, isso fica mais evidenciado no gráfico a seguir (Fig. 7.15). Sugerindo também que o número de iterações não precisa ser alto.

Tabela 7.15 – Resultados Médios - Iterações

Iterações	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
50	607.46	1020.972	2095.49	3754.51	3787.26
100	587.25	1031.90	2121.15	3741.45	3792.51
200	584.074	1020.442	2029.82	3869.70	3749.31
500	578.17	1019.476	2096.72	3795.53	3734.63

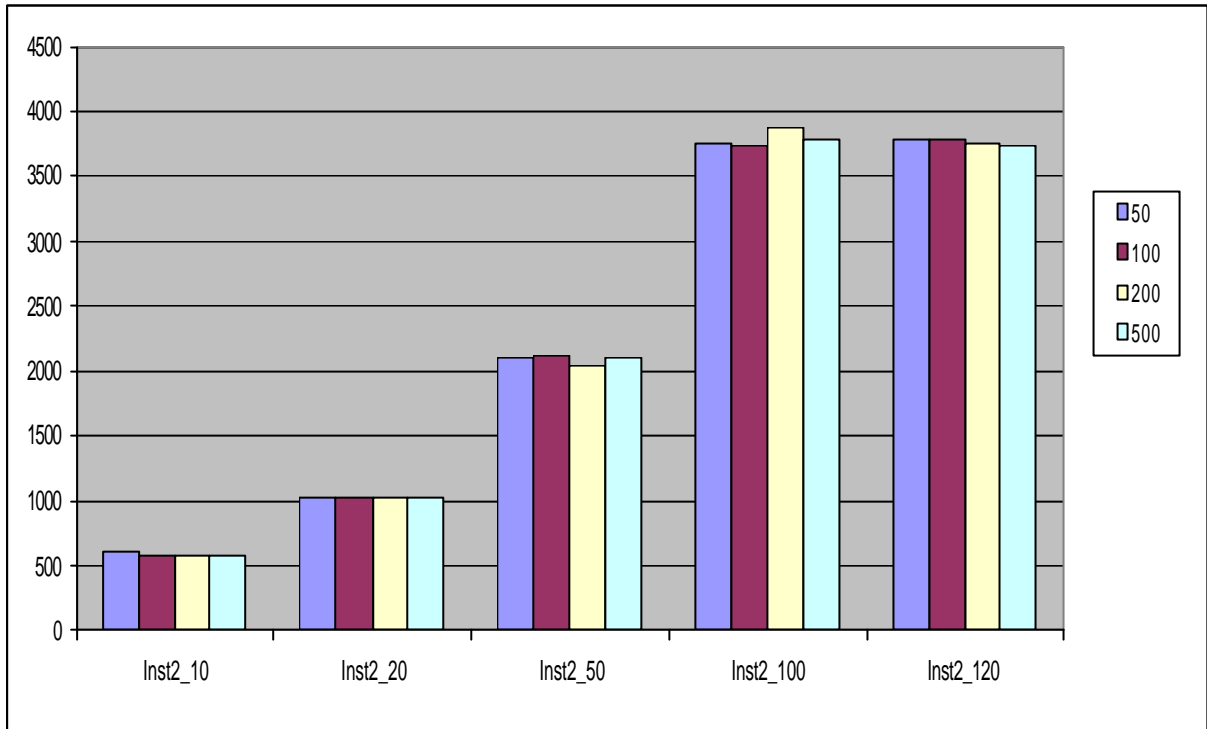


Fig. 7.15 – Gráfico Comparativo – Iterações

A Tabela 7.16 mostra os melhores resultados a partir da média dos 5 resultados gerados para cada variação da lista restrita de candidatos, percebe-se que os melhores resultados foram obtidos com valores entre 2 e 5 com uma leve tendência para 2. Os piores resultados foram obtidos na solução gulosa ($\lambda=1$) e quase sempre na totalmente aleatória ($\lambda=100000$). O gráfico 8.16 demonstra a diferença dos resultados obtidos em utilizar $\lambda=1, 2, 5, 10$ e 100000.

Tabela 7.16 – Resultados Médios – L.R.C

L.R.C	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
1	646.94	1062.68	2256.26	3980.49	4065.91
2	607.46	1020.44	2029.82	3741.45	3734.63
5	578.17	1019.48	2108.15	3795.53	3757.83
10	578.17	1028.05	2193.40	3901.16	3931.92
100000	578.17	1031.76	2336.28	4358.60	4361.82

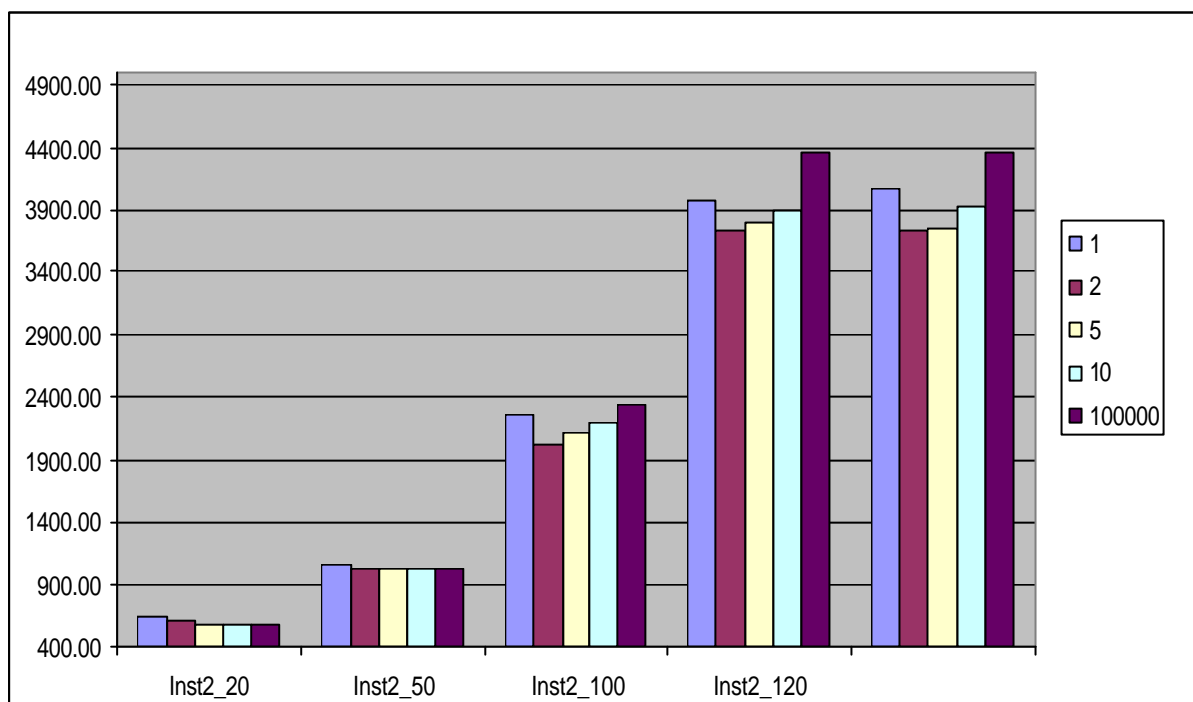


Fig. 7.16 – Gráfico Comparativo – Lista Restrita de Candidatos

A tabela 7.17 resume os resultados médios para cada instância. Pode-se observar também que o tempo de espera vai diminuindo com o aumento do número de clientes do problema. Se compararmos estes TME's com os tempos dos testes com dois compartimentos iremos observar que neste caso os tempos são maiores, isto porque como mencionado as restrições temporais nos testes com três compartimentos foram mais severas.

Tabela 7.17 – Resultados Médios

Instância	Custo	Tempo	TME
10	597.78	00:00:01	00:17:48
20	1032.48	00:00:01	00:38:44
50	2184.78	00:00:16	00:23:18
100	3955.45	00:01:15	00:27:09
120	3970.42	00:02:24	00:13:28

Neste grupo de problemas também foram avaliados os parâmetros δ_1 , δ_2 , δ_3 e δ_4 , para tal foi fixado o número de iterações (T) em 100 e a lista restrita de candidatos (λ) em 5. Para cada variação dos parâmetros (δ_1 , δ_2 , δ_3 e δ_4) foram gerados cinco resultados, a Tabela 7.18 demonstra o resultado médio dos testes.

Tabela 7.18 – Avaliação dos parâmetros δ_1 , δ_2 , δ_3 e δ_4 (Inst_2)

$d_1; d_2; d_3; d_4$	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
0.1 / 0.6 / 0.1 / 0.2	607.21	1045.62	2175.96	3813.73	3844.89
0.0 / 0.0 / 0.5 / 0.5	627.05	1119.35	2551.39	4703.85	4624.73
0.0 / 0.6 / 0.0 / 0.4	602.42	1022.30	2170.17	3854.23	3788.07
0.0 / 0.6 / 0.4 / 0.0	582.79	1031.54	2179.33	3876.40	3736.25
0.0 / 0.7 / 0.0 / 0.3	601.32	1028.93	2179.50	3839.13	3781.83
0.0 / 0.7 / 0.1 / 0.2	596.66	1028.67	2199.49	3823.27	3771.95
0.1 / 0.7 / 0.0 / 0.2	578.17	1033.30	2222.44	3888.05	3824.49

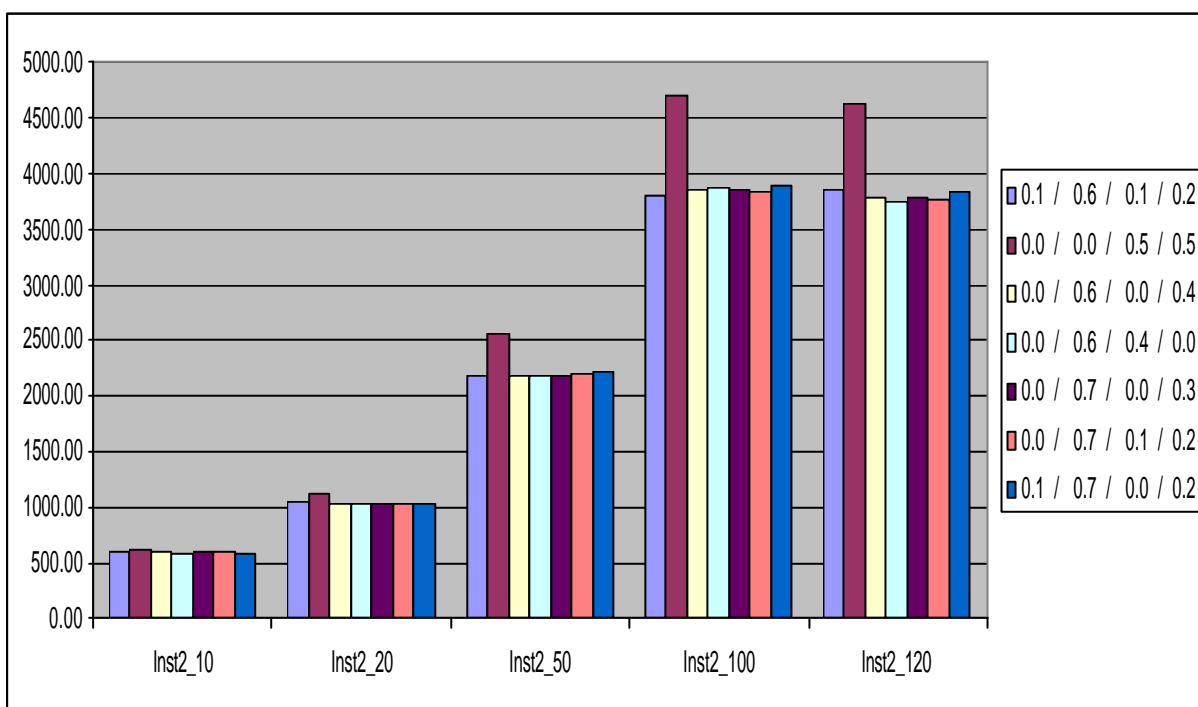


Fig. 7.17 – Gráfico Comparativo – Parâmetros δ_1 , δ_2 , δ_3 e δ_4

Assim como nos testes da primeira instância (Inst_1) a heurística com os parâmetros $\delta_1= 0.0$ $\delta_2= 0.0$, $\delta_3= 0.5$ e $\delta_4= 0.5$ apresentou as piores soluções (Fig. 8.17), e mais uma vez para os problemas com mais clientes (50, 100 e 120) onde também irá ocorrer o aumento de deslocamento dos horários dos clientes. Percebe-se também que os casos com melhor solução também foram aqueles em que o peso para distância percorrida foi mais elevado, e também na maioria dos casos o tempo

e o custo fixo tiveram a mesma importância na solução final. Nas figuras 7.18, 7.19, 7.20, 7.21, 7.22 estão demonstrados os gráficos destes testes.

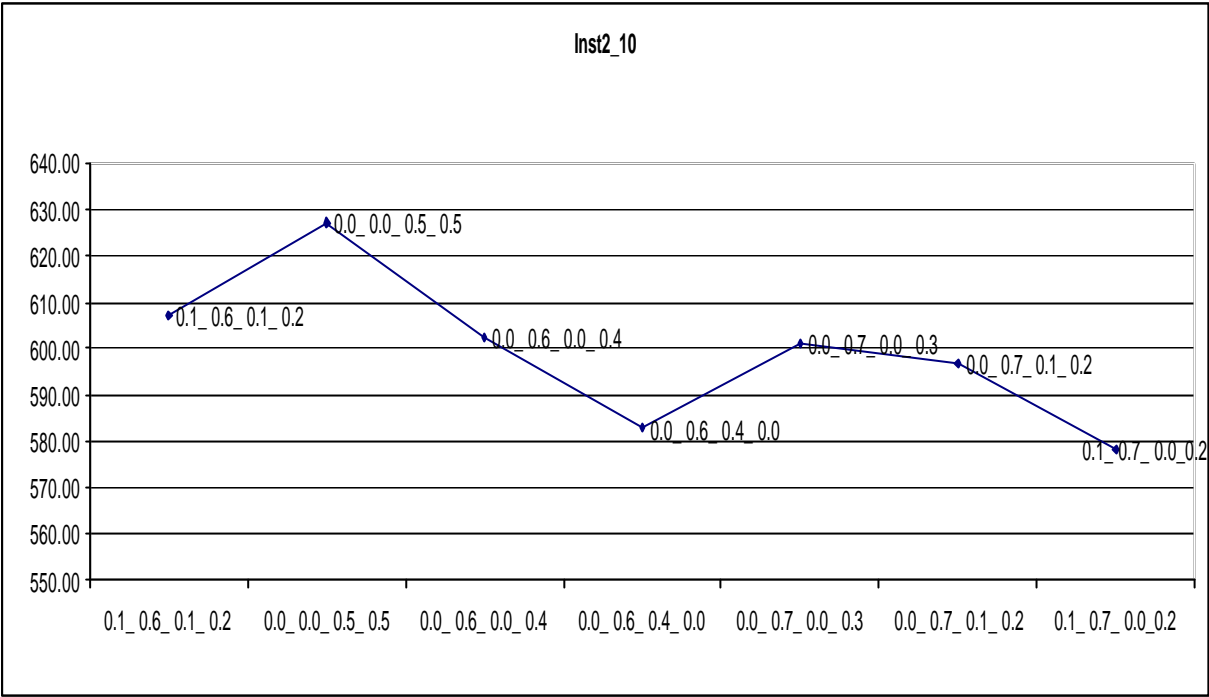


Fig. 7.18 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_10)

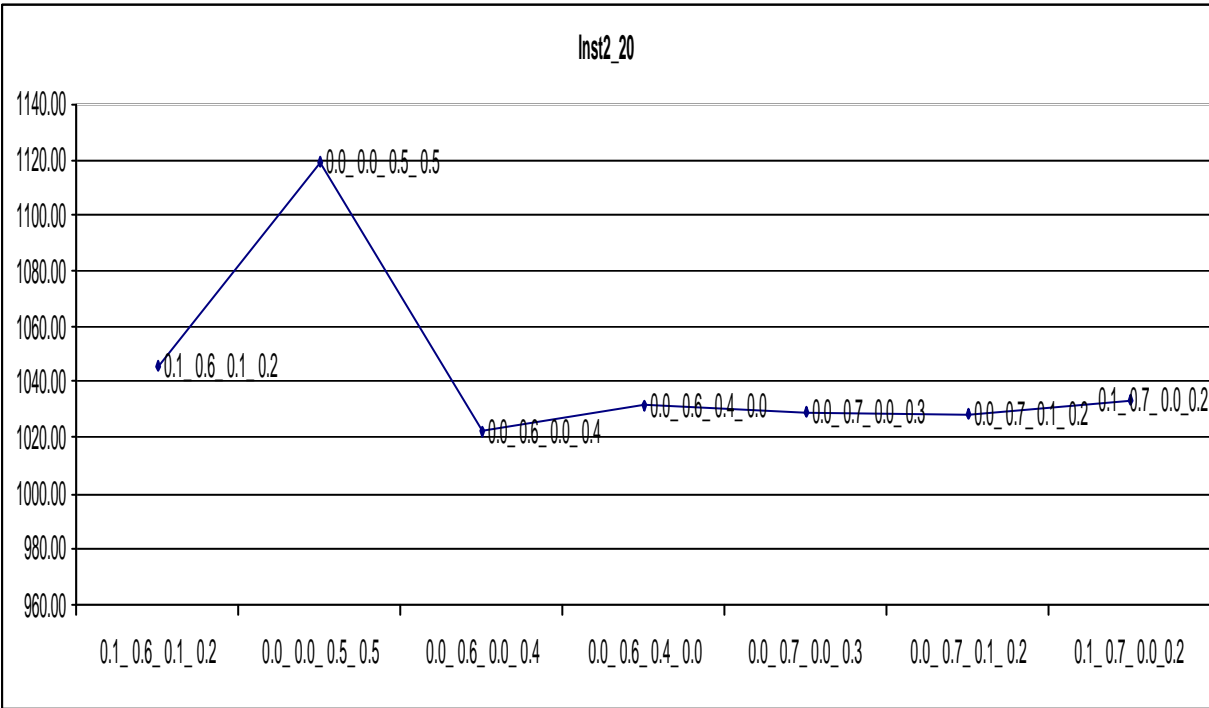


Fig. 7.19 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_20)

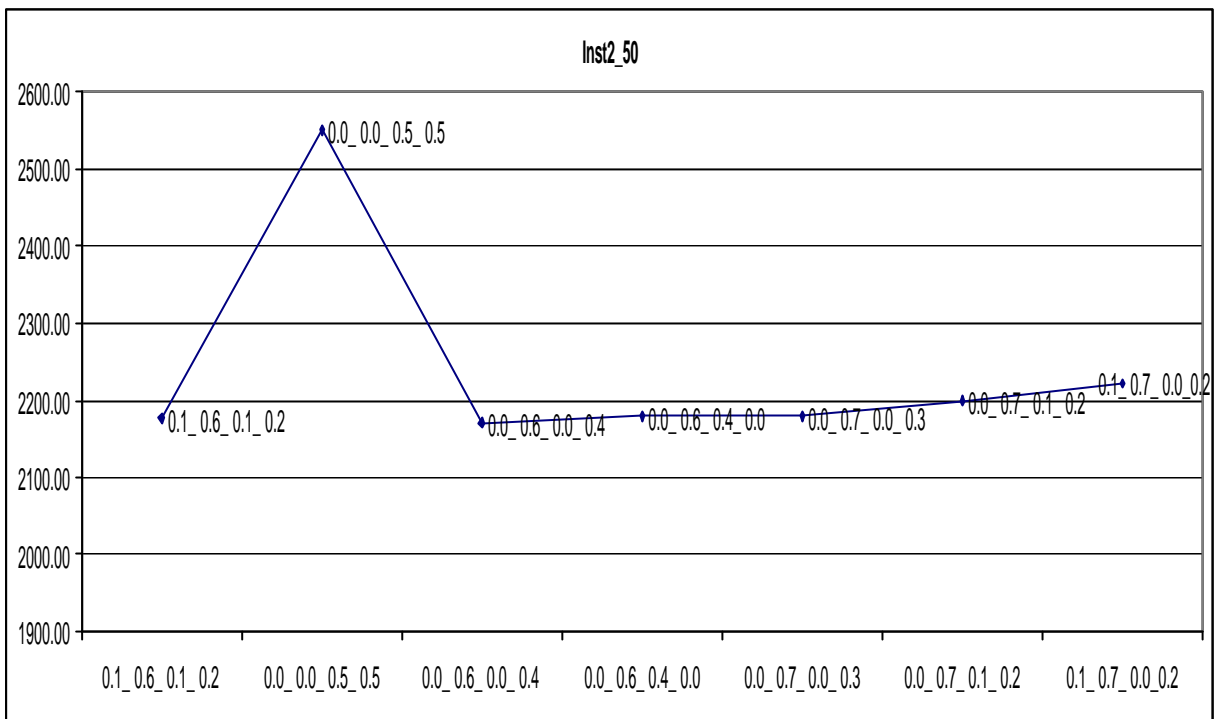


Fig. 7.20 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_50)

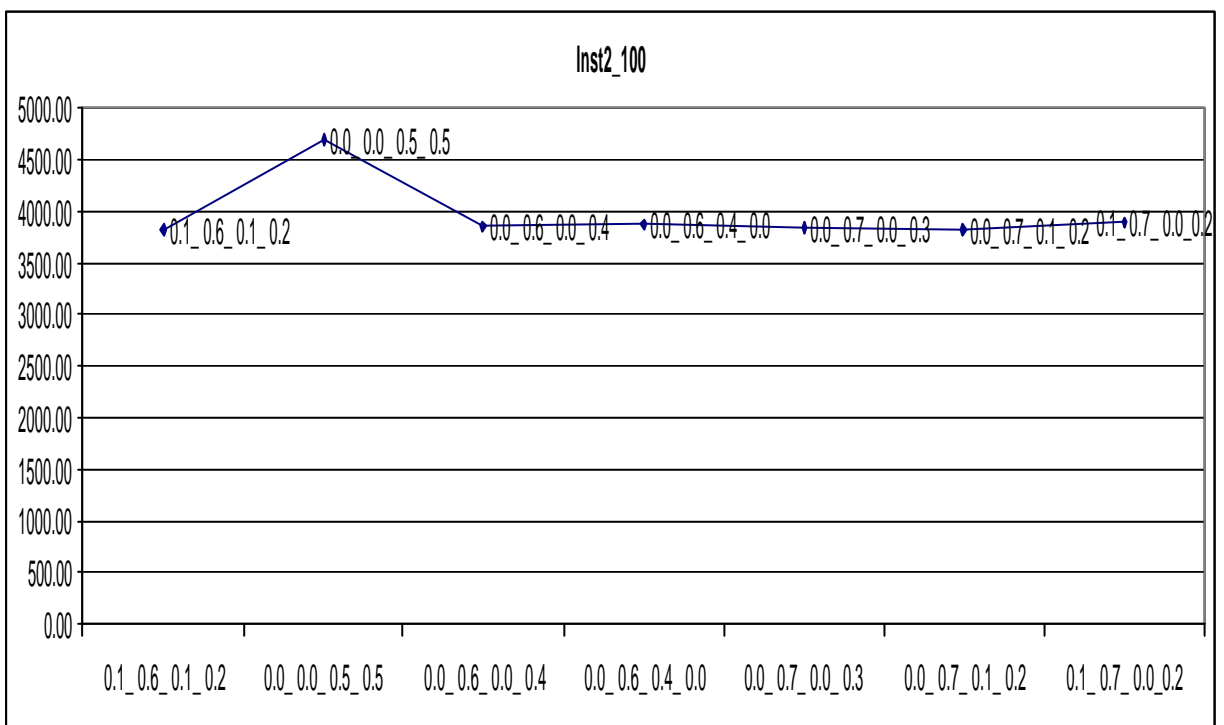


Fig. 7.21 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_100)

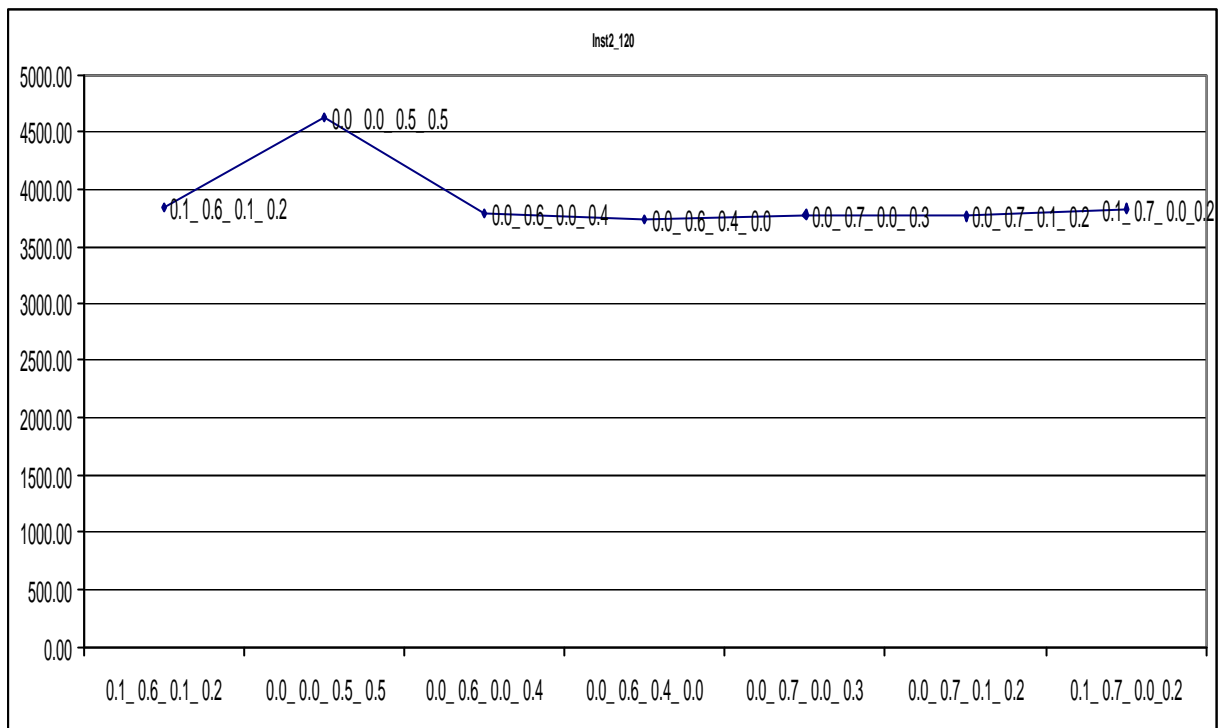


Fig. 7.22 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_2_120)

A seguir estão demonstrados os testes da Inst_2 com os parâmetros $T=200$ e a lista restrita de candidatos (λ) com os seguintes valores: 2, 5, 6, 7, 9 e 10. Assim como na Inst1 foram descartados os valores de $\lambda=1$ (totalmente guloso) e $\lambda=10000$ (totalmente aleatório). A Tabela 7.19 demonstra a média obtida de cada resultado, já as figuras 7.23, 7.24, 7.25, 7.26 e 7.27 demonstram a variação da função objetivo em relação a λ .

Tabela 7.19 – Resultados Médios L.R.C – Inst_2

L.R.C (λ)	Inst_2_10	Inst_2_20	Inst_2_50	Inst_2_100	Inst_2_120
2	611.9	1020.44	2029.82	3889.7	4563.67
5	594.84	1021.14	2108.15	3927.48	4751.57
6	591.56	1025.43	2110.98	3936.97	4774.41
7	584.07	1018.81	2136.71	3865.37	4713.78
9	582.98	1022.15	2148.92	4016.13	4755.99
10	589.00	1028.05	2193.4	3901.16	4777.69

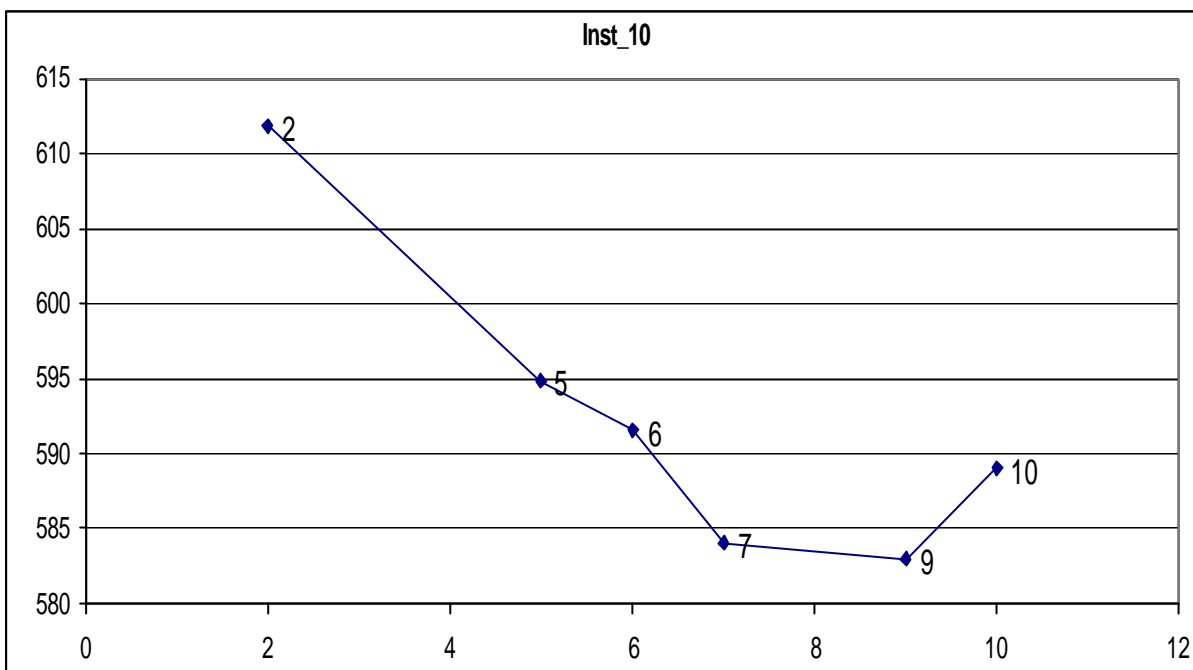


Fig. 7.23 – Variação da função objetivo – L.R.C – Inst_2_10 Clientes

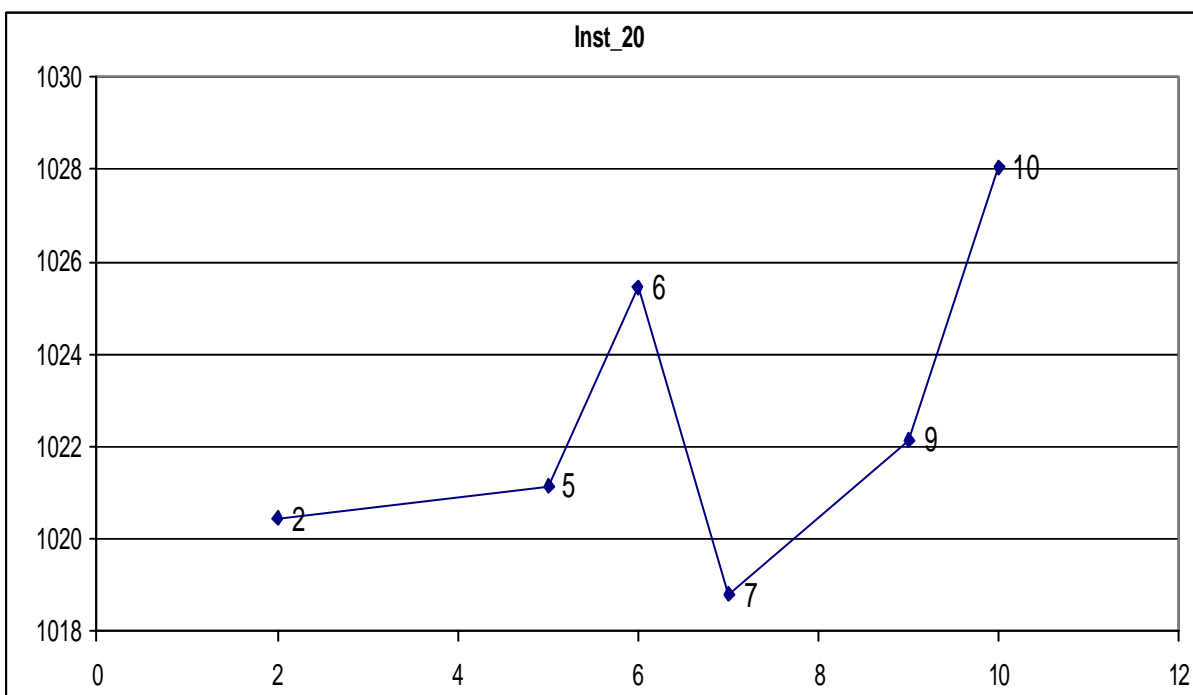


Fig. 7.24 – Variação da função objetivo – L.R.C – Inst_2_20 Clientes

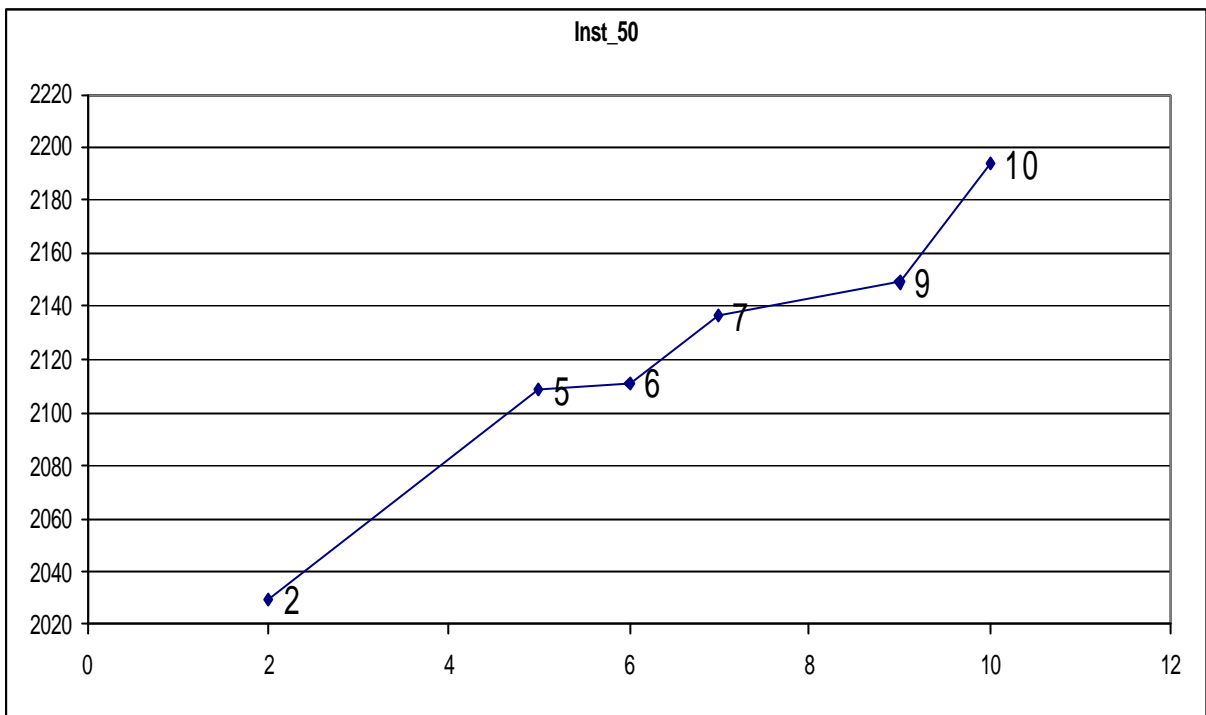


Fig. 7.25 – Variação da função objetivo – L.R.C – Inst_2_50 Clientes

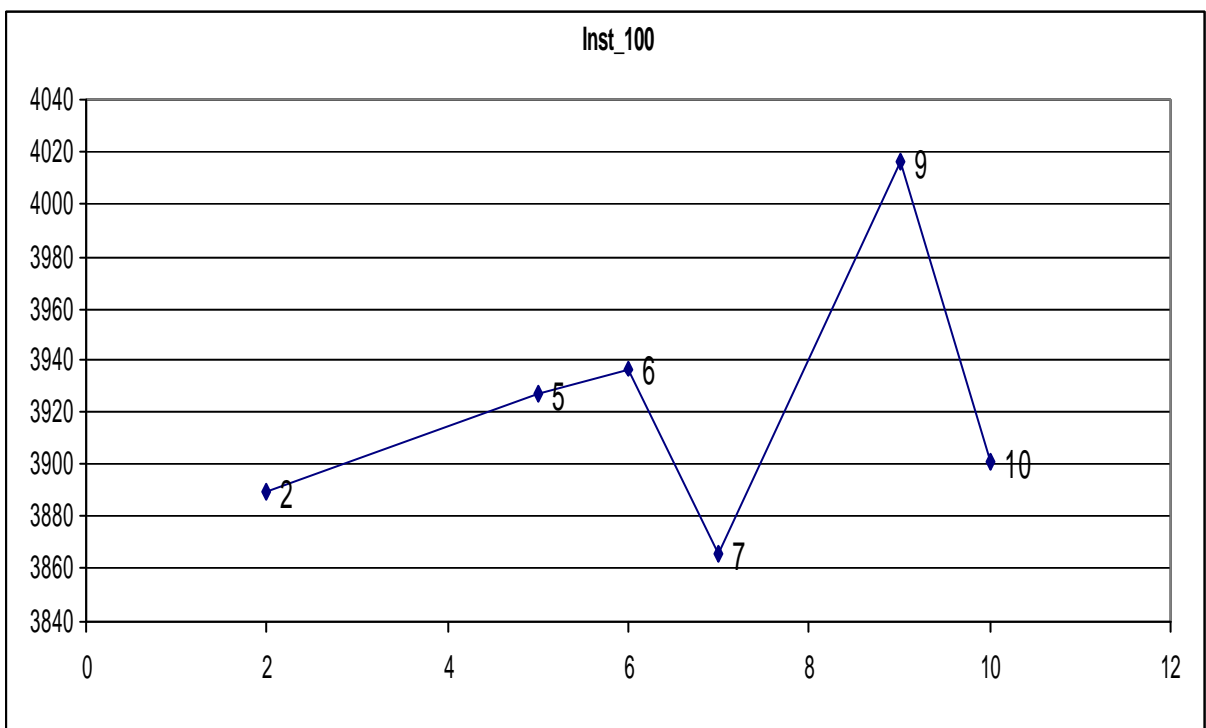


Fig. 7.26 – Variação da função objetivo – L.R.C – Inst_2_100 Clientes

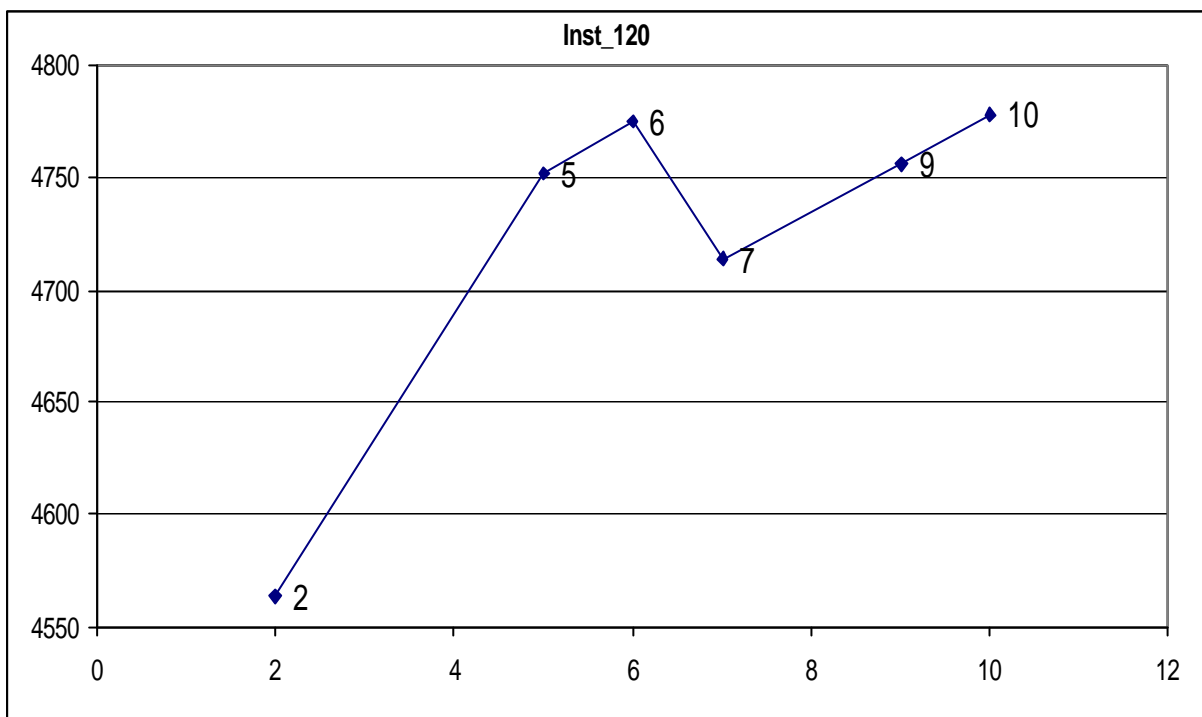


Fig. 7.27 – Variação da função objetivo – L.R.C – Inst_2_120 Clientes

Nestes testes percebe-se que na maioria dos casos as melhores soluções apareceram para valor igual a 7 para a lista restrita de candidatos (λ), mas assim como nos testes da Inst_1 a variação em torno do melhor resultado é baixa.

7.1.4 Testes com Instâncias Aleatórias – Quatro Compartimentos

Nesta fase os testes continuaram com problemas contendo 10, 20, 50, 100, 120 clientes, e com as mesmas instâncias baseadas nos problemas de Christofides *et al.* (1979). Assim como nos exemplos anteriores foram geradas de forma aleatória as demandas de cada compartimento, a cubagem da carga e a janela de tempo de cada cliente, o custo fixo, o custo variável e a velocidade de cada veículo, bem como a capacidade de carga e a cubagem máxima de cada compartimento.

Para atender o conjunto de problemas Inst_3 também foram gerados veículos com capacidade de carga e volume de cubagem semelhantes aos gerados para a Inst_1, e clientes com janela de tempo também menor. Para as instâncias de 10 e 20 clientes utilizaram-se três veículos com quatro compartimentos, e com capacidade de carga variando entre 2500 a 6000 e a cubagem de cada compartimento variando entre 2000 a 3000. Para as instâncias de 50 e 100 clientes utilizou-se frota com 5 veículos diferentes com quatro compartimentos com capacidade de carga variando entre 2500 a 6000 e a cubagem de cada compartimento variando entre 2000 a 3000,

e para a instância de 120 clientes foi utilizada frota com 10 veículos diferentes com quatro compartimentos com capacidade de carga variando entre 2500 a 6000 e a cubagem de cada compartimento variando entre 2000 a 3000. Em todos os veículos a velocidade foi definida aleatoriamente variando entre 70 a 90. A geração da janela de tempo de todos os clientes se deu da seguinte forma, horário de abertura da janela (a_i) variando entre 08:00 e 14:00, e tempo de fechamento da janela variando entre 10:00 e 18:00. O tempo de serviço para todos os clientes foi considerado nulo.

Assim como nas primeiras instâncias (Inst_1 e Inst_2), o número de iterações (T) foi testado em 50, 100, 200 e 500. Já a lista restrita de candidatos (l) foi testada com os seguintes valores: 1 (totalmente guloso), 2, 5, 10 e 10000 (totalmente aleatório). Os valores de d_1 , d_2 , d_3 e d_4 foram fixados em 0.1, 0.6, 0.1 e 0.2. As Tabelas 7.20, 7.21, 7.22 e 7.23 mostram os resultados médios dos testes.

Tabela 7.20 – 50 iterações

L.R.C	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
1	566.60	933.3	1341.52	2710.68	3632.16
2	553.43	888.78	1196.37	2339.23	3320.89
5	553.17	890.306	1197.29	2304.78	3317.04
10	560.09	908.338	1221.76	2356.40	3482.20
10000	562.11	916.418	1346.318	2791.28	4053.72
Tempo	00:00:01	00:00:01	00:00:07	00:00:34	00:00:50
TME	00:22:10	00:13:06	00:06:35	00:05:32	00:05:35

Tabela 7.21 – 100 iterações

L.R.C	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
1	566.60	933.30	1341.52	2710.68	3632.16
2	554.44	888.78	1161.93	2339.44	3216.80
5	553.17	892.43	1161.82	2269.02	3298.84
10	563.05	904.94	1233.44	2319.25	3425.64
10000	557.83	918.24	1334.33	2742.90	3974.83
Tempo	00:00:01	00:00:01	00:00:12	00:01:09	00:01:39
TME	00:22:17	00:12:42	00:06:41	00:05:23	00:05:35

Tabela 7.22 – 200 iterações

L.R.C	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
1	566.6	933.30	1341.52	2710.68	3632.16
2	553.17	888.78	1161.68	2314.99	3209.75
5	550.14	888.94	1157.29	2311.81	3223.69
10	557.73	901.98	1205.18	2306.59	3359.29
10000	561.12	919.19	1310.73	2734.19	3863.97
Tempo	00:00:01	00:00:02	00:00:25	00:02:13	00:03:18
TME	00:22:30	00:12:50	00:06:39	00:05:21	00:05:19

Tabela 7.23 – 500 Iterações

L.R.C	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
1	566.6	933.3	1341.52	2710.68	3632.16
2	554.18	888.78	1150.64	2292.54	3168.45
5	554.18	887.88	1148.41	2258.04	3238.01
10	555.19	889.36	1181.57	2306.29	3324.24
10000	554.18	909.72	1280.42	2711.882	3861.28
Tempo	00:00:01	00:00:06	00:00:59	00:05:28	00:07:58
TME	00:22:32	00:12:32	00:06:40	00:05:26	00:05:32

Na instância de 10 clientes (Inst1_10) a demanda foi atendida por um veículo, a instância de 20 clientes (Inst1_20) e de 50 clientes (Inst1_50) a demanda foi atendida por dois veículos. A instância seguinte de 100 clientes (Inst1_100) foi atendida por quatro veículos, e a última de 120 clientes (Inst1_120) foi atendida por cinco. Em todas as situações todos os clientes foram roteados.

A tabela 7.24 mostra o resumo dos melhores resultados a partir da média dos cinco resultados gerados, encontrados para cada variação do número de iterações, assim como nos testes da Inst_1 e Inst_2 não houve altas variações no resultado, isso fica mais evidenciado no gráfico abaixo (Fig. 7.28). Sugerindo também que o número de iterações não precisa ser alto.

Tabela 7.24 – Resultados Médios – Iterações

Iterações	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
50	560.09	890.306	1196.37	2339.23	3317.04
100	563.05	892.43	1161.93	2339.44	3425.64
200	561.12	901.98	1161.68	2314.99	3223.69
500	554.18	909.72	1148.41	2292.54	3324.24

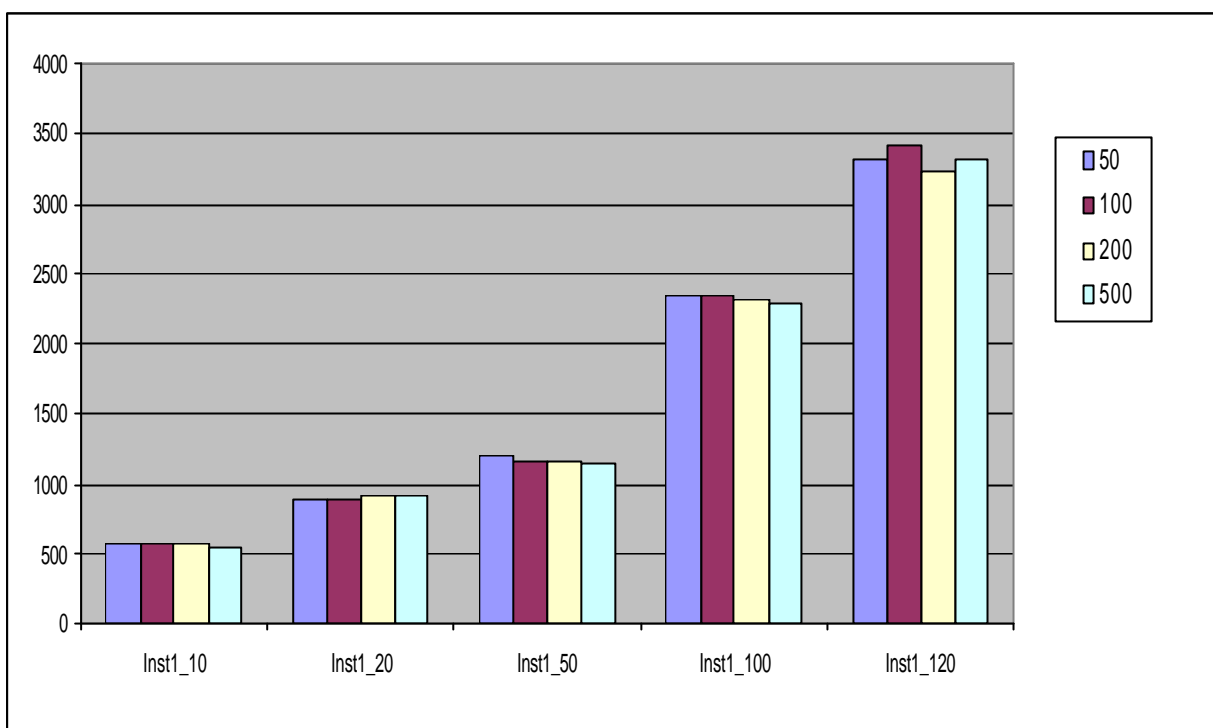


Fig. 7.28 – Gráfico Comparativo – Iterações

A Tabela 7.25 mostra os melhores resultados a partir da média dos 5 resultados gerados para cada variação da lista restrita de candidatos, percebe-se que os melhores resultados foram obtidos com valores entre 2 e 5 com uma leve tendência para 5. Os piores resultados foram obtidos na solução gulosa ($\lambda=1$) e quase sempre na totalmente aleatória ($\lambda=100000$). O gráfico da figura 7.29 demonstra a diferença dos resultados obtidos em utilizar $\lambda=1, 2, 5, 10$ e 100000.

Tabela 7.25 – Resultados Médios – L.R.C

L.R.C	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
1	566.60	933.30	1341.52	2710.68	3632.16
2	553.17	888.78	1150.64	2292.54	3168.45
5	550.14	887.88	1148.41	2258.04	3223.69
10	555.19	889.36	1181.57	2306.29	3324.24
100000	554.18	909.72	1280.42	2711.88	3861.28

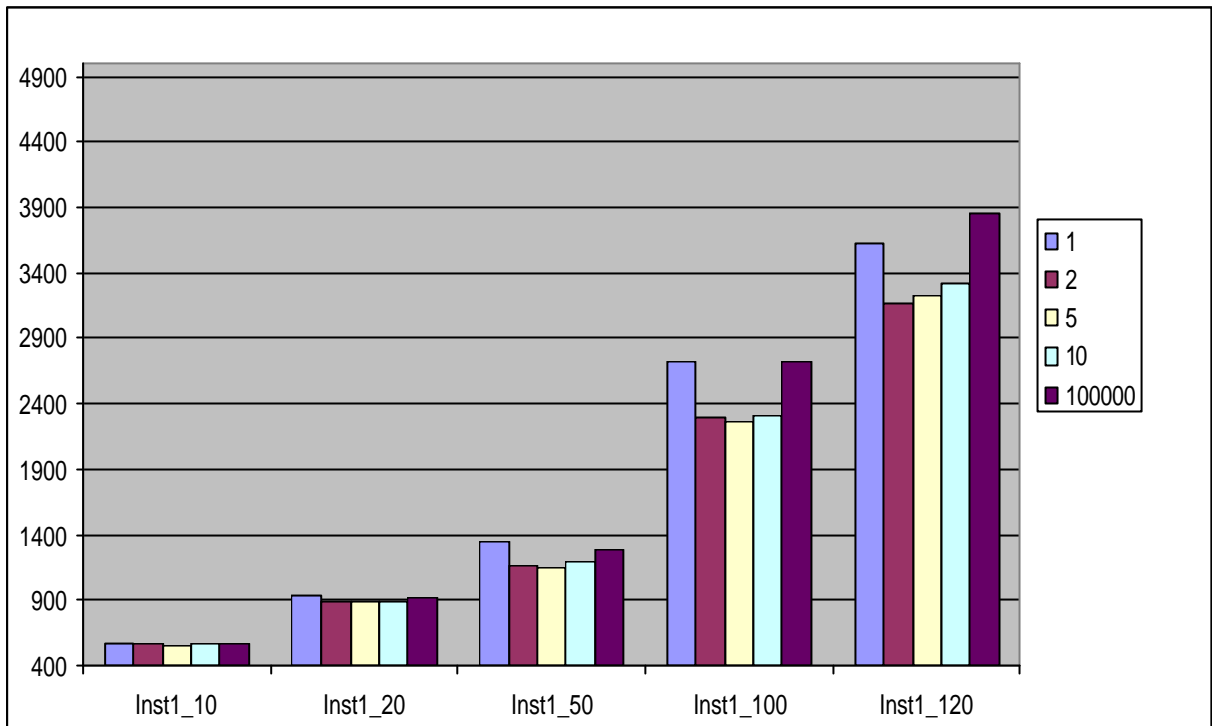


Fig. 7.29 – Gráfico Comparativo – Lista Restrita de Candidatos

A tabela 7.26 resume os resultados médios para cada instância. Nesta tabela também verifica - se a tendência de redução do tempo de espera com o aumento do número de clientes do problema.

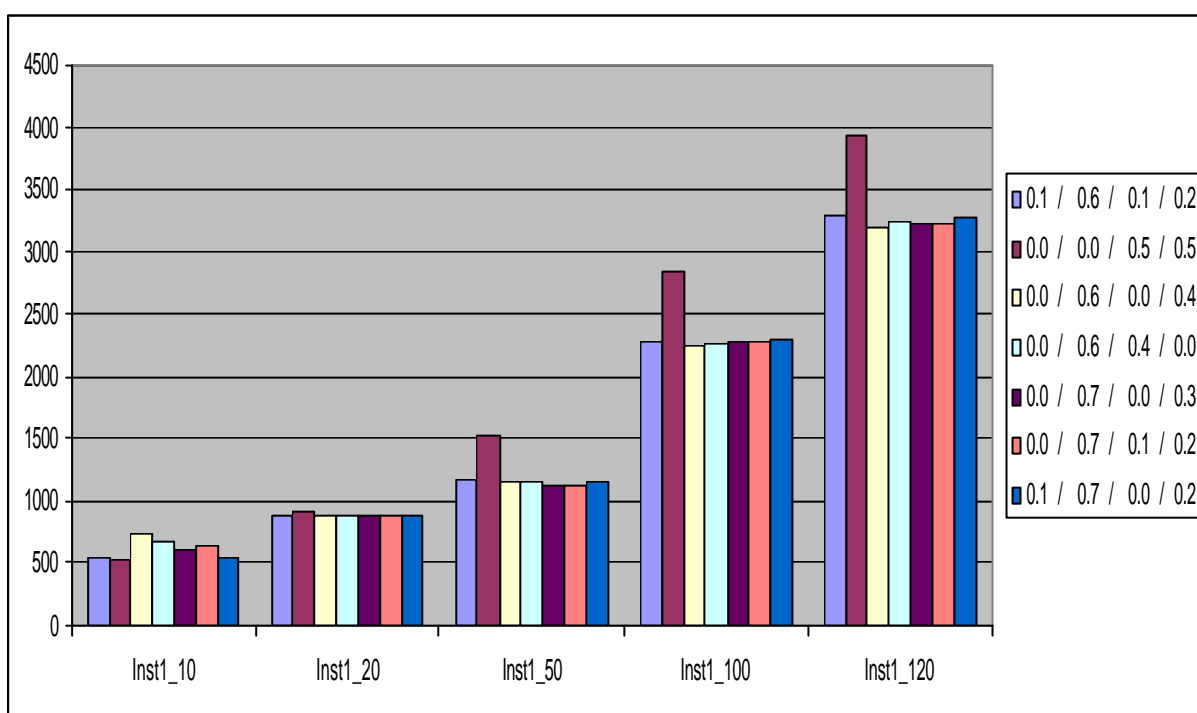
Tabela 7.26 – Resultados Médios

Instância	Custo	Tempo	TME
10	555.86	00:00:01	00:22:22
20	901.81	00:00:02	00:12:47
50	1220.51	00:00:26	00:06:39
100	2455.89	00:02:21	00:05:26
120	3441.96	00:03:26	00:05:30

Neste grupo de problemas também foram avaliados os parâmetros δ_1 , δ_2 , δ_3 e δ_4 , para tal foi fixado o número de iterações (T) em 100 e a lista restrita de candidatos (λ) em 5. Para cada variação dos parâmetros (δ_1 , δ_2 , δ_3 e δ_4) foram gerados cinco resultados, a Tabela 7.27 demonstra o resultado médio dos testes.

Tabela 7.27 – Avaliação dos parâmetros $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3)

$d_1; d_2; d_3; d_4$	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
0.1 / 0.6 / 0.1 / 0.2	553.17	892.43	1161.82	2269.02	3298.84
0.0 / 0.0 / 0.5 / 0.5	527.12	913.91	1520.84	2835.25	3931.96
0.0 / 0.6 / 0.0 / 0.4	731.74	886.91	1139.58	2236.07	3209.96
0.0 / 0.6 / 0.4 / 0.0	687.24	888.69	1143.99	2258.11	3233.48
0.0 / 0.7 / 0.0 / 0.3	607.51	888.69	1131.26	2270.07	3214.31
0.0 / 0.7 / 0.1 / 0.2	648.32	888.96	1125.24	2286.33	3222.08
0.1 / 0.7 / 0.0 / 0.2	553.17	890.07	1154.58	2297.65	3278.05

Fig. 7.30 – Gráfico Comparativo – Parâmetros $\delta_1, \delta_2, \delta_3$ e δ_4

Assim como nos testes das instâncias anteriores (Inst_1, Inst_2) a heurística com os parâmetros $\delta_1=0.0$, $\delta_2=0.0$, $\delta_3=0.5$ e $\delta_4=0.5$ apresentou as piores soluções (Fig. 7.30), e mais uma vez para os problemas com mais clientes (50, 100 e 120) onde também ocorre uma influência maior no resultado do deslocamento dos horários dos clientes. Percebe-se também que os casos com melhor solução também foram aqueles em que o peso para distância percorrida foi mais elevado, e também na maioria dos casos o tempo e o custo fixo tiveram a mesma importância na solução final. Nas figuras 7.31, 7.32, 7.33, 7.34, 7.35 estão demonstrados os gráficos destes testes.

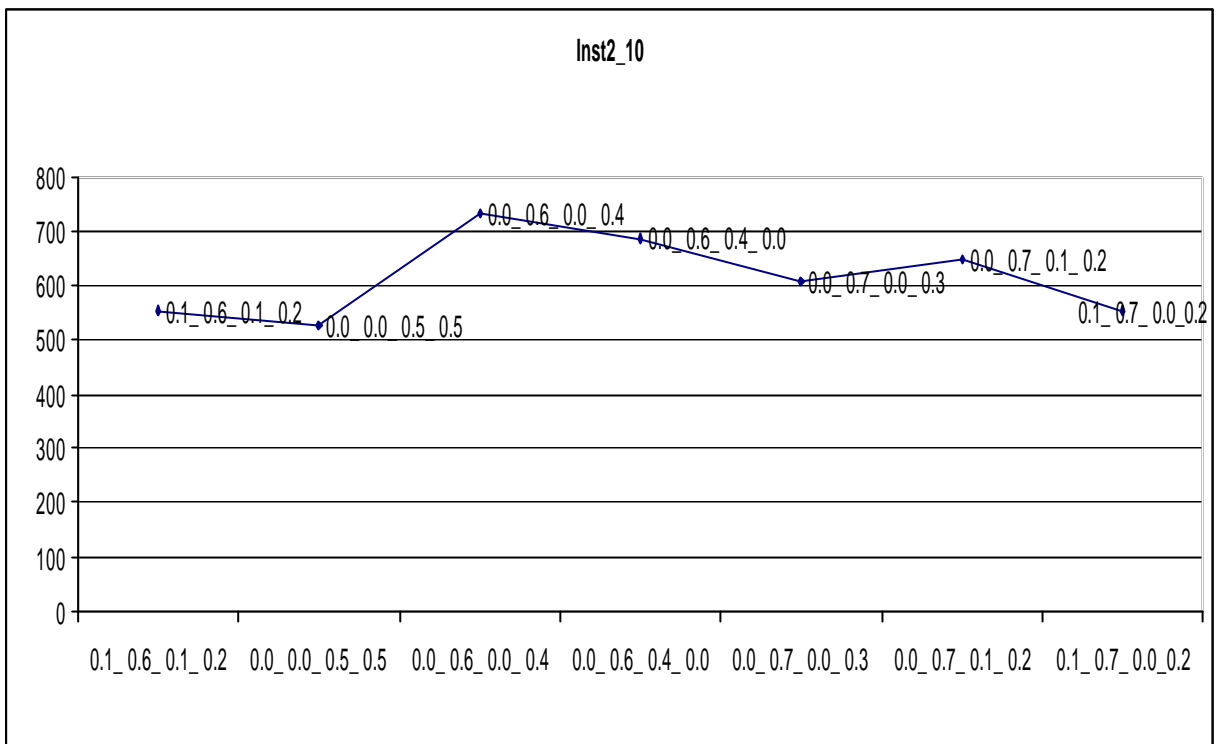


Fig. 7.31 – Variação da função objetivo - δ_1 , δ_2 , δ_3 e δ_4 (Inst_3_10)

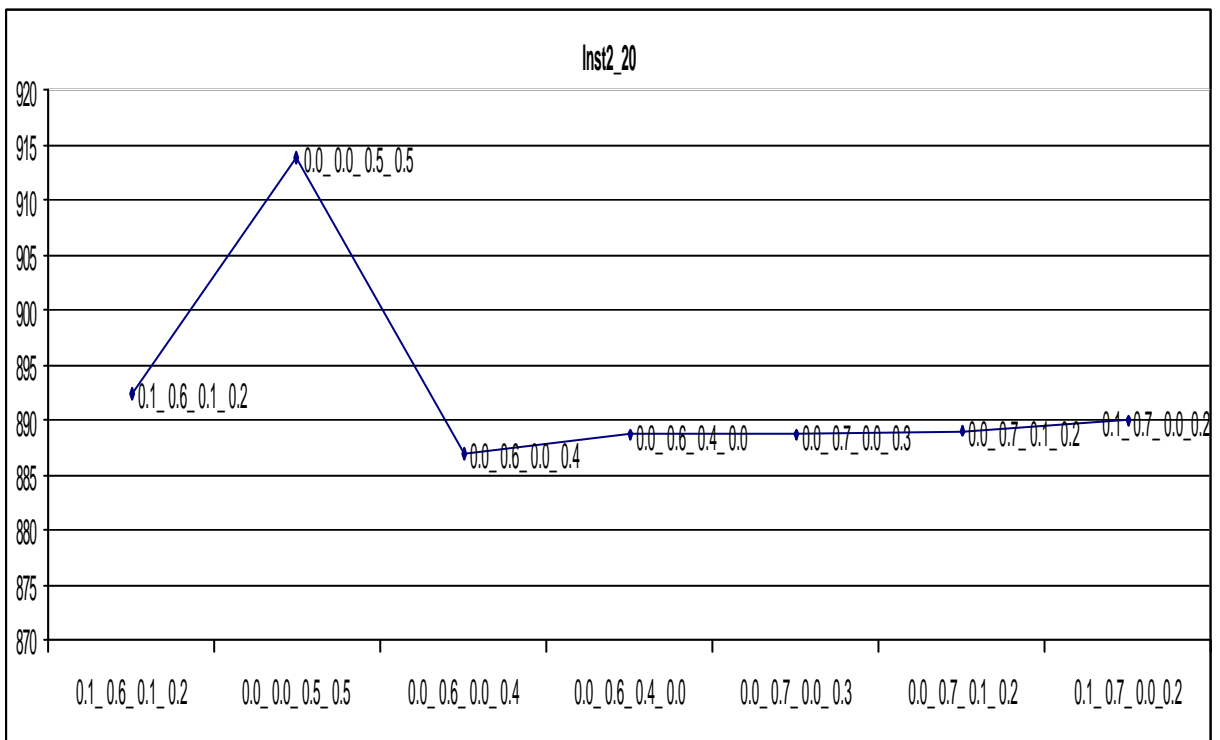


Fig. 7.32 – Variação da função objetivo - δ_1 , δ_2 , δ_3 e δ_4 (Inst_3_20)

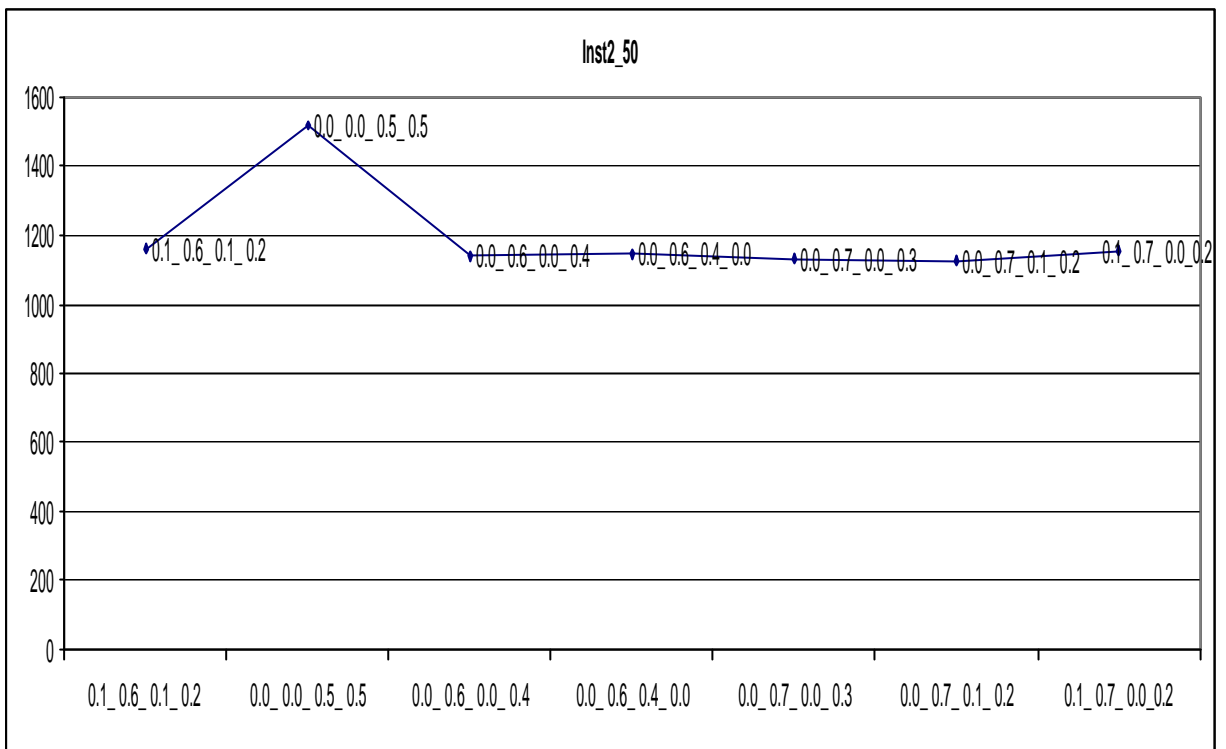


Fig. 7.33 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_50)

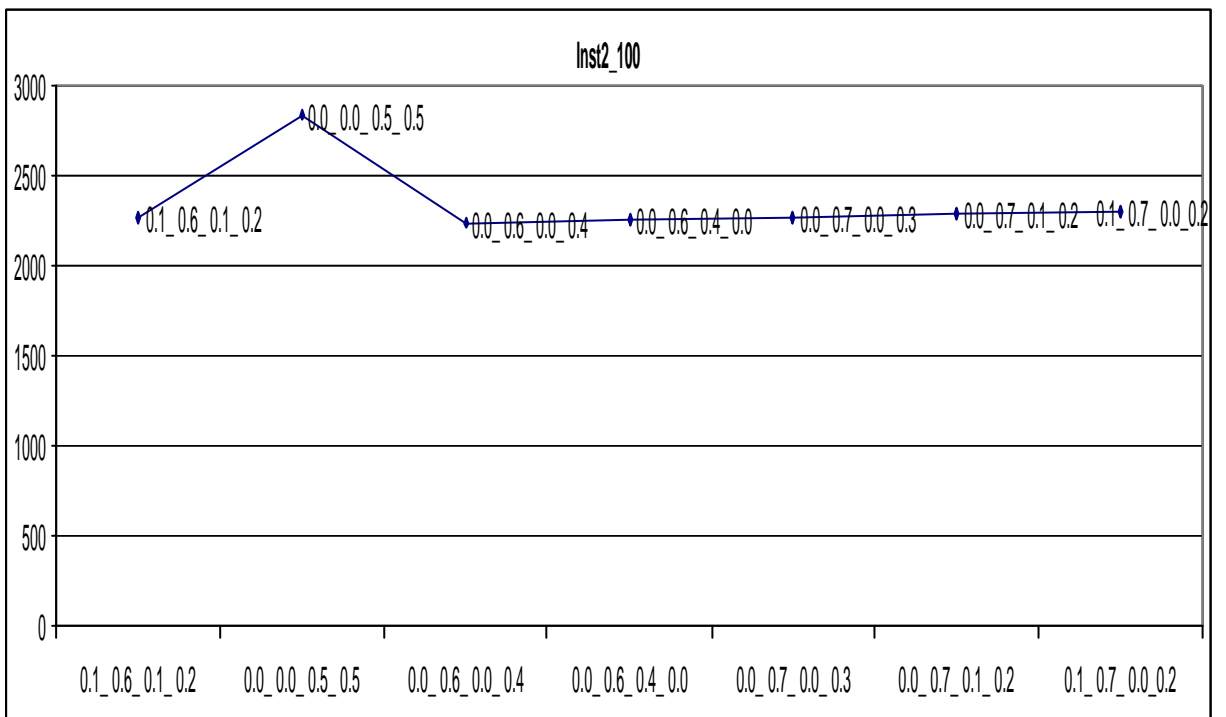


Fig. 7.34 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_100)

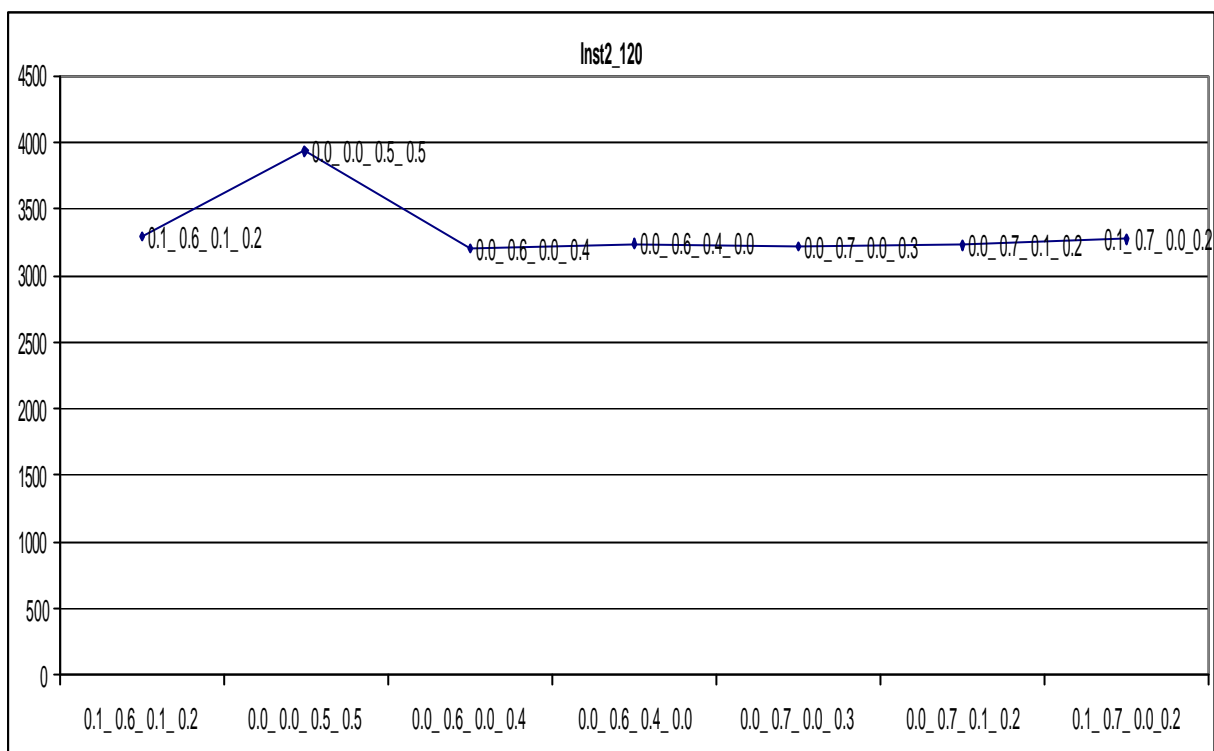


Fig. 7.35 – Variação da função objetivo - $\delta_1, \delta_2, \delta_3$ e δ_4 (Inst_3_120)

A seguir estão demonstrados os testes da Inst_3 com os parâmetros $T=200$ e a lista restrita de candidatos (λ) com os seguintes valores: 2, 5, 6, 7, 9 e 10. Assim como nos testes das Inst_1 e Inst_2 foram descartados os valores de $\lambda=1$ (totalmente guloso) e $\lambda=10000$ (totalmente aleatório). A Tabela 7.28 demonstra a média obtida de cada resultado, já as figuras 7.36, 7.37, 7.38, 7.39 e 7.40 demonstram a variação da função objetivo em relação a λ .

Tabela 7.28 – Resultados Médios L.R.C – Inst_3

L.R.C (l)	Inst_3_10	Inst_3_20	Inst_3_50	Inst_3_100	Inst_3_120
2	553.17	888.78	1161.68	2314.99	3209.75
5	550.14	888.94	1157.29	2311.81	3223.69
6	555.45	889.64	1167.46	2322.44	3281.16
7	554.18	889.98	1168.41	2312.54	3270.81
9	553.17	892.31	1200.50	2323.68	3303.51
10	557.73	901.98	1205.18	2306.59	3359.29

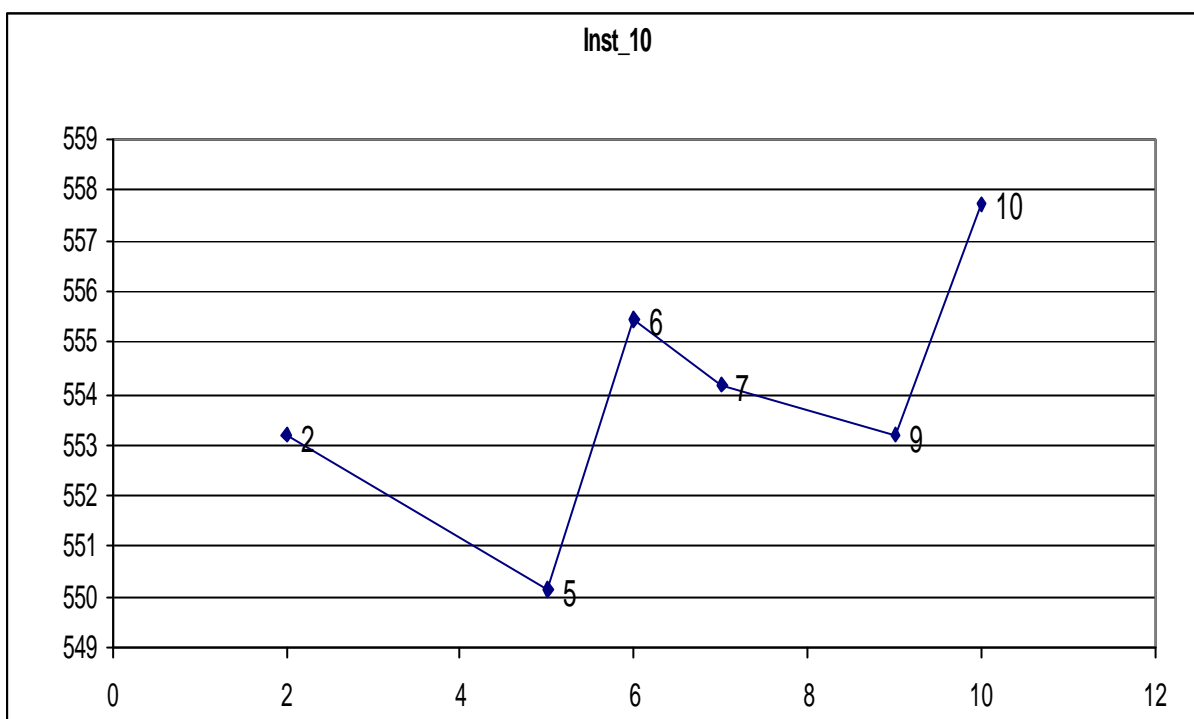


Fig. 7.36 – Variação da função objetivo – L.R.C – Inst_3_10 Clientes

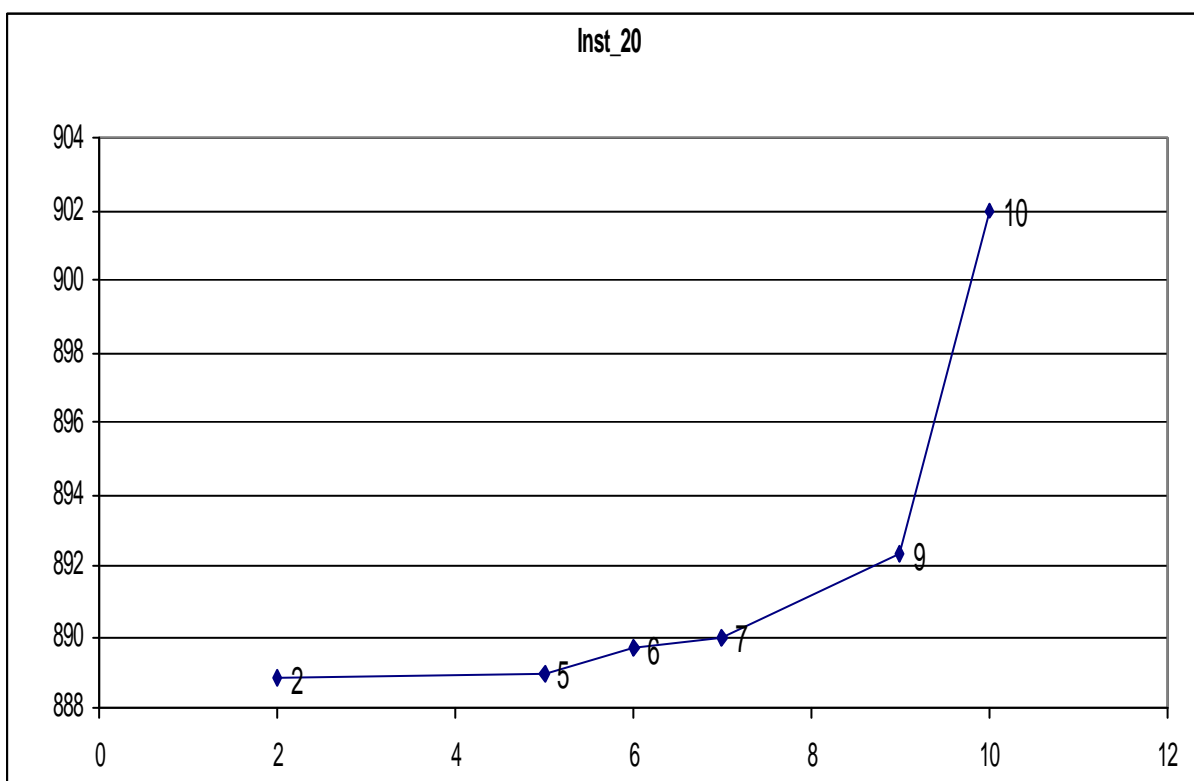


Fig. 7.37 – Variação da função objetivo – L.R.C – Inst_3_20 Clientes

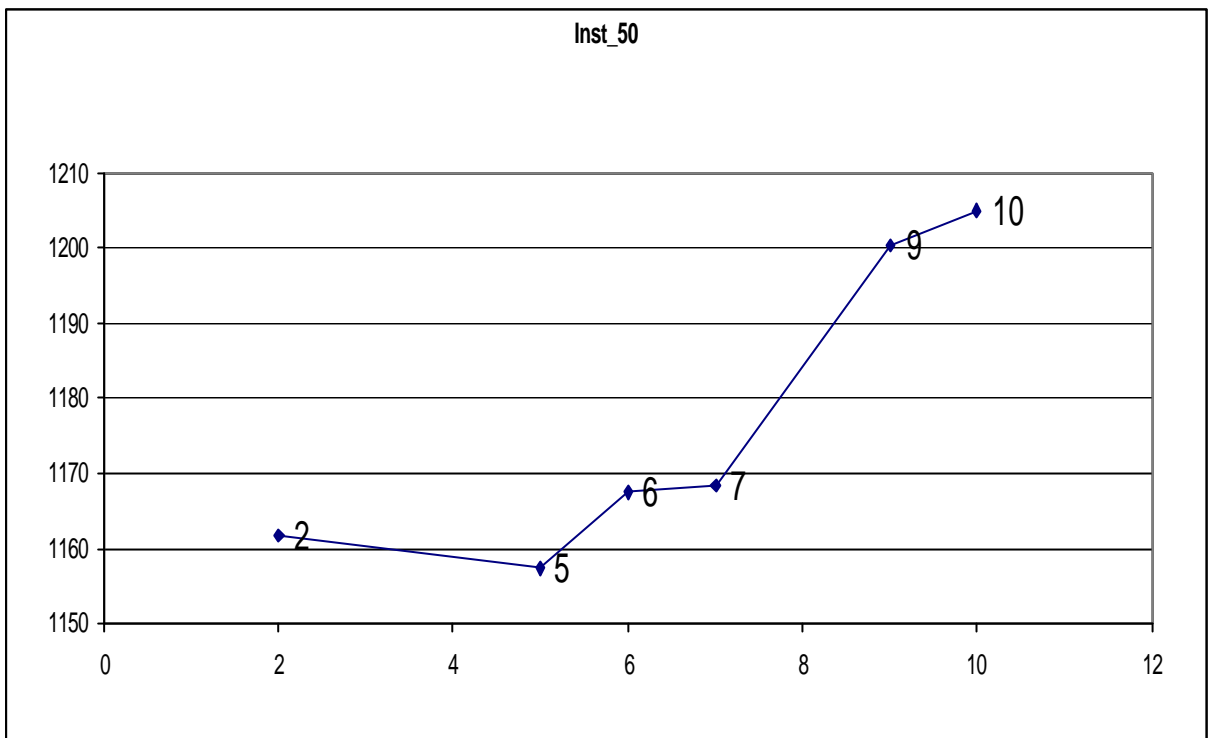


Fig. 7.38 – Variação da função objetivo – L.R.C – Inst_3_50 Clientes

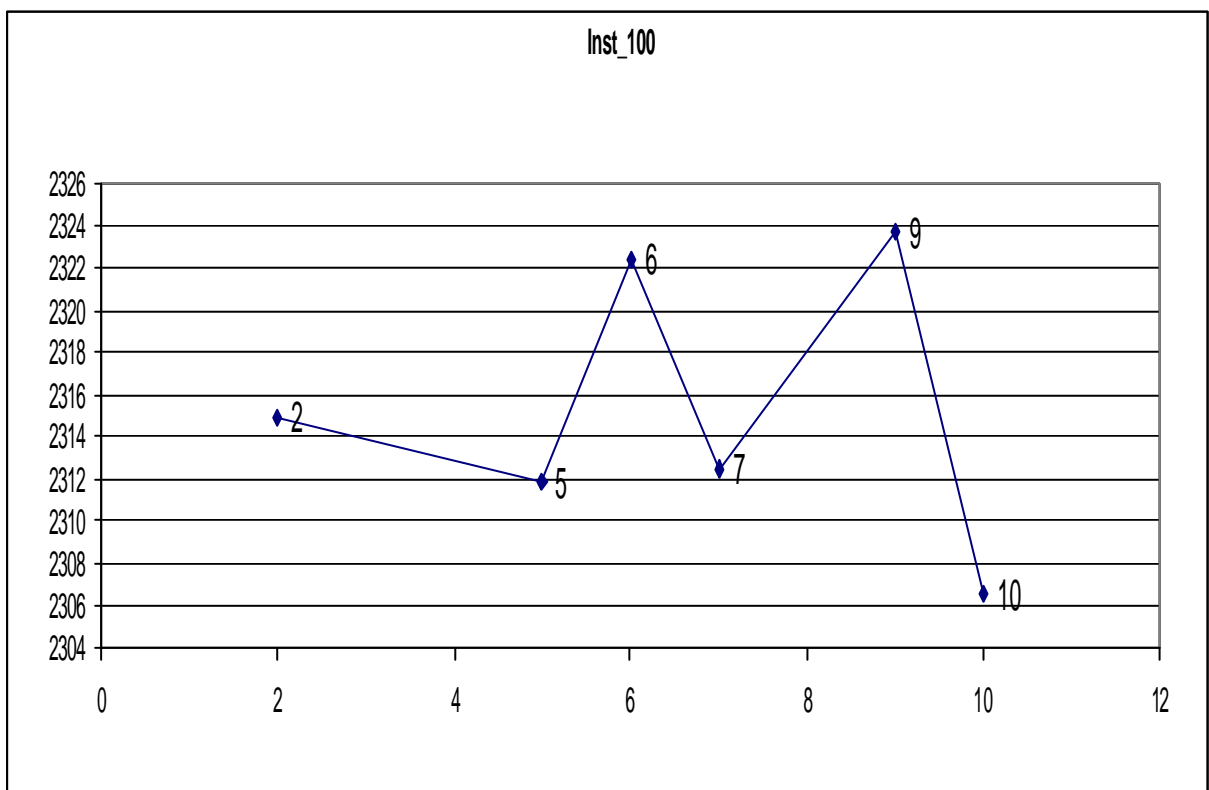


Fig. 7.39 – Variação da função objetivo – L.R.C – Inst_3_100 Clientes

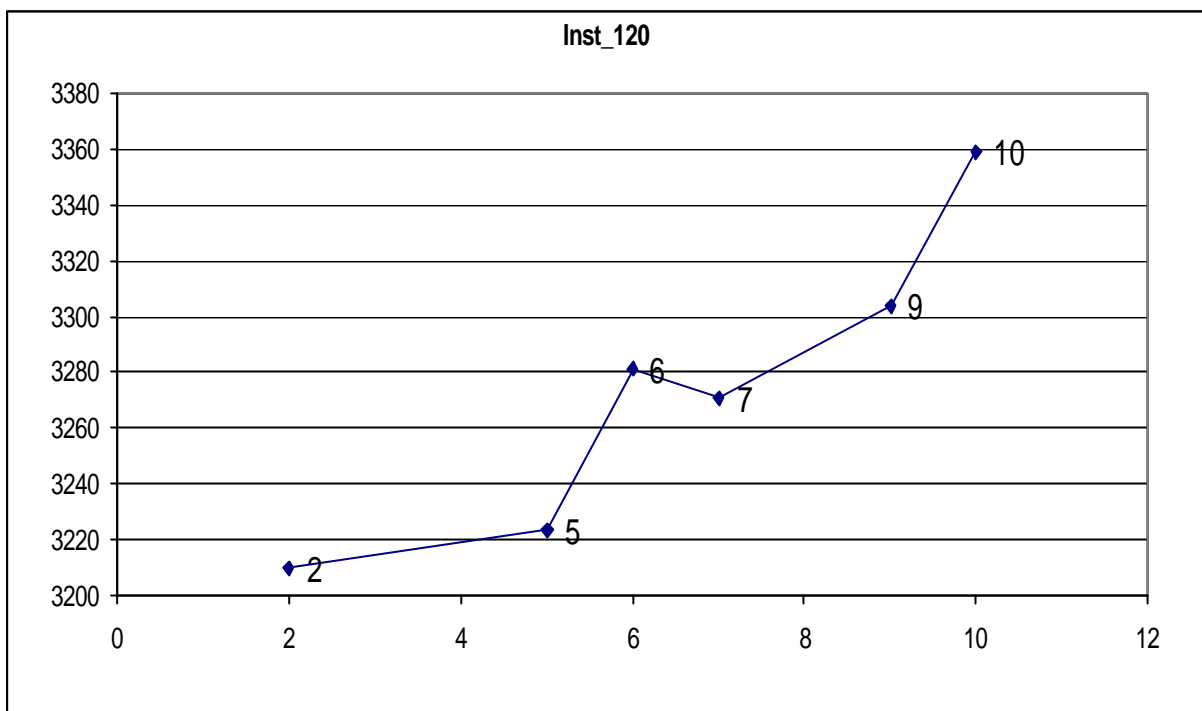


Fig. 7.40 – Variação da função objetivo – L.R.C – Inst_3_120 Clientes

Nestes testes percebe-se que na maioria dos casos as melhores soluções apareceram para valores iguais a 2 e 5 para a lista restrita de candidatos (λ), mas assim como nos testes anteriores a variação em torno do melhor resultado é baixa.

7.1.5 Testes com Instância Maior Aleatória – Dois Compartimentos

Foi gerada uma instancia de 534 clientes (Inst_4) com o objetivo de avaliar o comportamento da heurística em instâncias maiores. Os seguintes parâmetros foram considerados para o gerador aleatório, frota com 15 veículos e 2 compartimentos, com capacidade de carga de cada compartimento variou de 2000 a 3000 e a cubagem variou de 2500 a 6000.

O número de iterações foi fixado em $T = 5$, já a lista restrita de candidatos (λ) foi testada com os seguintes valores: 1, 2, 5, 6, 7, 9, 10. Foram realizados cinco testes para cada variação de λ . O tempo médio de processamento foi de 00:02:28. A Tabela 7.29 demonstra a média obtida de cada resultado, bem como a diferença dos resultados, já a Figura 7.41 demonstra a variação da função objetivo em relação a λ .

Tabela 7.29 – Resultados Médios L.R.C – Inst_4	
L.R.C	Inst_4
1	6544,84
2	6546,16
5	6572,35
6	6512,49
7	6584,19
9	6768,39
10	6651,25

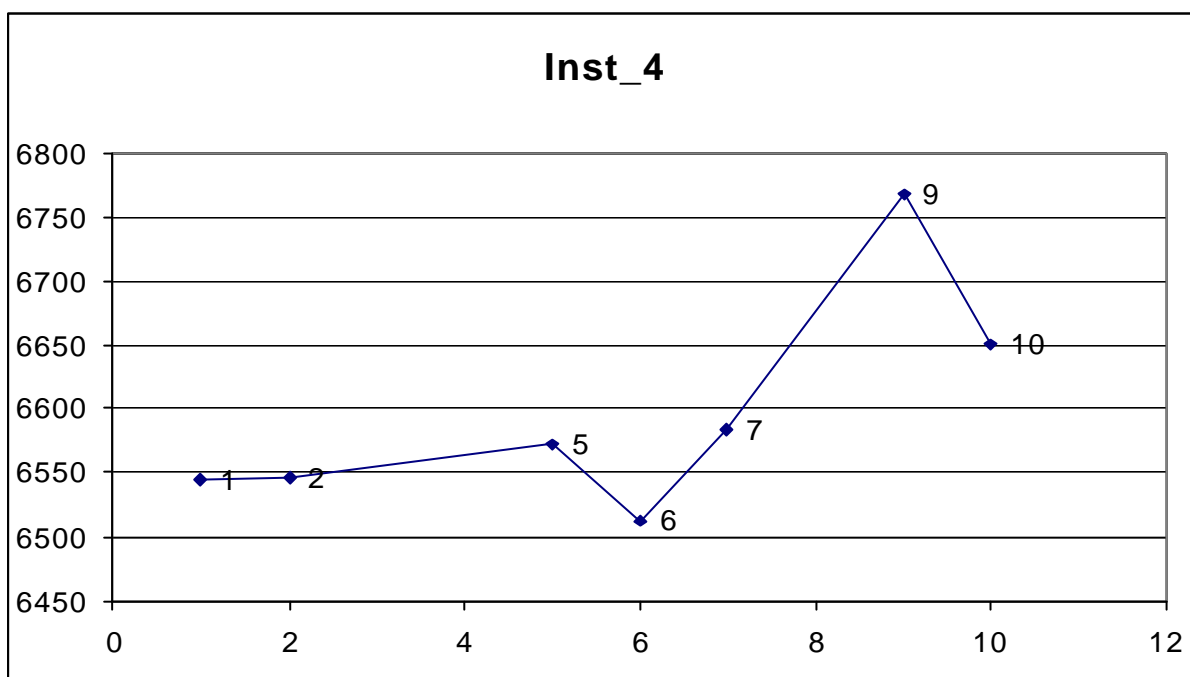


Fig. 7.41 – Variação da função objetivo – L.R.C – Inst_4_534 Clientes

Analisando o gráfico acima, percebe-se que o melhor resultado foi obtido com $\lambda=6$, mas a variação dos resultados não é muito elevada.

Para avaliar o comportamento dos parâmetros δ_1 , δ_2 , δ_3 e δ_4 foram realizados testes variando seus valores. A Tabela 7.30 resume a média dos cinco testes realizados para cada variação dos parâmetros.

Tabela 7.30 – Avaliação dos parâmetros d_1 , d_2 , d_3 e d_4 (Inst_4)	
d_1 ; d_2 ; d_3 ; d_4	Inst_4
0.1; 0.6; 0.1; 0.2	6572,35
0.0; 0.0; 0.5; 0.5	10275,79
0.0; 0.6; 0.0; 0.4	6533,98
0.0; 0.6; 0.4; 0.0	6575,57
0.0; 0.7; 0.0; 0.3	6583,47
0.0; 0.7; 0.1; 0.2	6567,43
0.1; 0.7; 0.0; 0.2	6476,31

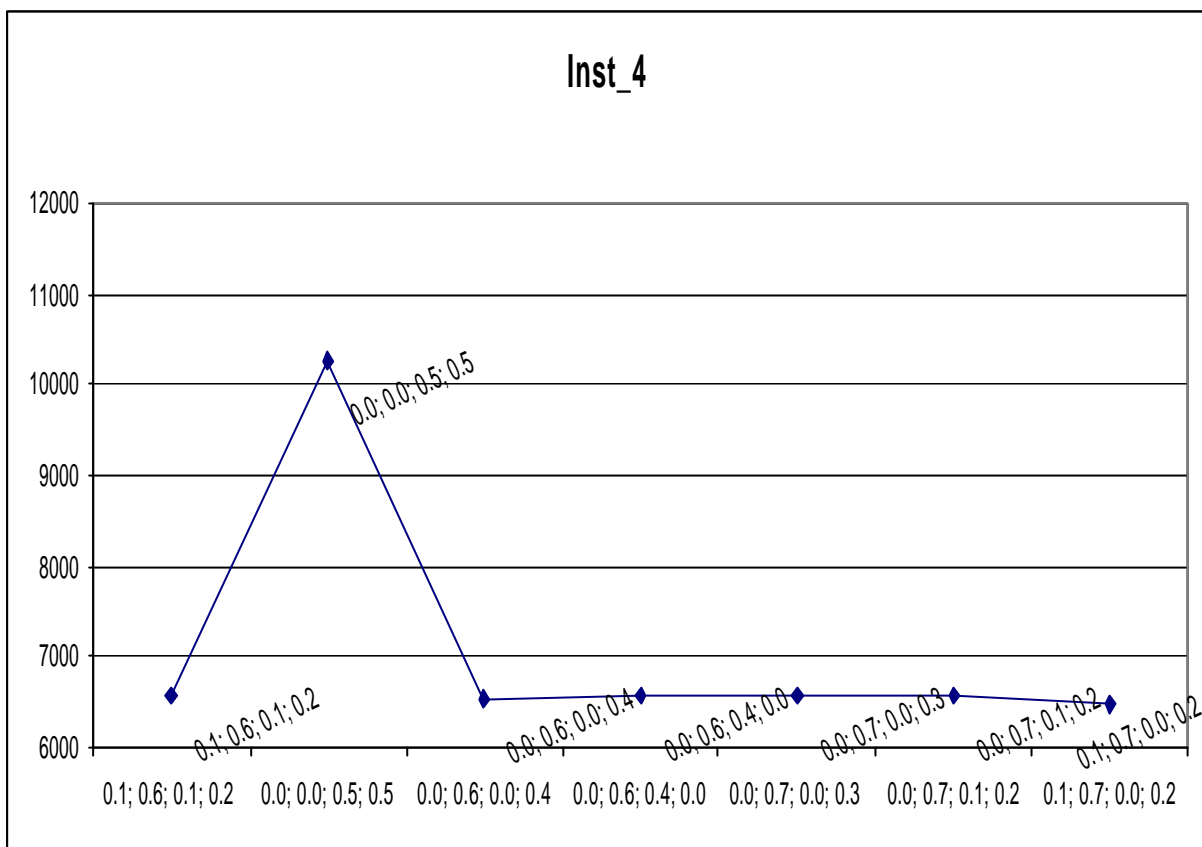


Fig. 7.42 – Variação da função objetivo - δ_1 , δ_2 , δ_3 e δ_4 (Inst_4_534)

O melhor resultado foi obtido com $\delta_1 = 0.1$, $\delta_2 = 0.7$, $\delta_3 = 0.0$ e $\delta_4 = 0.2$. Na figura 7.42 pode-se verificar que a variação dos resultados é muito pequena, a variação de maior relevância ocorre para a combinação de parâmetros $\delta_1 = 0.0$, $\delta_2 = 0.0$, $\delta_3 = 0.5$ e $\delta_4 = 0.5$. Percebe-se aqui a tendência observada nos testes das instâncias anteriores (Inst_1, Inst_2 e Inst_3), ou seja, o aumento do número de clientes, e conseqüentemente uma necessidade maior de deslocamento de horário, provoca uma piora da solução quando o parâmetro relacionado a restrição temporal é de maior valor.

8. CONCLUSÃO

8.1 CONCLUSÕES

Esta pesquisa trata de um problema específico de roteamento de veículos que pode ser encontrado na realidade das empresas, onde se tem à necessidade de atender clientes com restrições temporais e demanda por produtos que necessitam e/ou podem ser transportados por uma frota de veículos com multi-compartimentos. O objetivo principal proposto no Capítulo 1 foi alcançado com o desenvolvimento de uma heurística GRASP para o Problema de Roteamento de Veículos com Multi-compartimentos e Restrição de Janela de Tempo que contribui de forma direta para os processos de roteamento de veículos relacionados.

A metaheurística GRASP desenvolvida para o problema de roteamento de veículos com multi-compartimentos com restrição de janela de tempo demonstrou ser bastante robusta para solução de problemas reais. Apesar de não existir na literatura modelos matemáticos e heurísticas que tratem do problema pesquisado nesta dissertação, as soluções obtidas são de qualidade. Esta afirmação está baseada na comparação feita entre as soluções do modelo matemático com as soluções obtidas pela metaheurística GRASP, e também no tempo reduzido para soluções, e que na prática atenderia a definição de um roteamento diário. Sendo assim, a metaheurística oferece conteúdo informacional de qualidade para dar suporte a decisão sobre as rotas e horários em que os clientes serão atendidos.

O programa desenvolvido permite a definição de rotas tanto em trechos urbanos e intermunicipais, bastando para isso calibrar os parâmetros com as características pertinentes de cada caso.

Em todos os testes todos os clientes foram roteados, na maioria deles as melhores soluções foram obtidas com λ igual a 2 e 5. Os resultados também demonstraram que ocorre a redução do tempo de espera, para início do atendimento, com o aumento do número de clientes a ser atendido.

A orientação a objetos mostrou-se como uma ótima metodologia de modelagem e implementação da mesma. Assim como foi demonstrada a capacidade de reutilização e adequação do código elaborado por Póvoa (2005), o código desenvolvido nesta pesquisa também permite sua reutilização, com modificação ou expansão, em outros modelos e variações do Problema de Roteamento de Veículos.

O algoritmo de Graham mostrou-se de fácil implementação e com resultado ótimo na obtenção do fecho convexo. Em todas as instâncias testadas a separação obtida entre clientes no fecho convexo (*convex hull*) e clientes que não estão no fecho convexo foi de 100 % de acerto.

8.2 SUGESTÕES PARA TRABALHOS FUTUROS

Em relação ao algoritmo desenvolvido, fica como sugestão a implementação de algoritmos que contemplem a restrição temporal de forma *soft*, ou seja, que seja permitido a violação da mesma.

Outro ponto importante é buscar metodologias que defina qual o valor ideal dos parâmetros utilizados na dissertação. Pode-se destacar o tamanho da Lista Restrita, uma correlação de tamanhos de lista com número de clientes poderia ser útil nesta definição.

Por fim, fica o incentivo à implementação dos casos que não foram apresentados neste trabalho, como por exemplo: roteamento com multi-depósitos, adaptação do algoritmo para o roteamento de frota de embarcações que atendem plataformas de petróleo etc.

9. REFERÊNCIAS BIBLIOGRÁFICAS

- Agarwal, Y., Mathur, K., Salkin, H. M. Set Partitioning approach to vehicle routing. **Networks**, v. 7, p 731-749, 1989.
- Alvarenga, G. B., Silva, R. M. A., Sampaio, R. M. A hybrid algorithm for the vehicle routing problem with time window. **Revista Infocomp Universidade Federal de Lavras**, v. 4, n. 2, p. 9-16, 2005.
- Balinski, M. L., Quandt, R. E. On an integer program for a delivery problem. **Operations Research**. v. 12, p. 300-304, 1964.
- Ballou, R. H. **Logística Empresarial: transporte, administração de materiais e distribuição física**. Edit. Atlas. 388 p, 1993.
- Ballou H. R. **Gerenciamento da Cadeia de Suprimentos: Planejamento, Organização e Logística Empresarial**. 4. ed. Porto Alegre: Bookman, 532 p, 2001.
- Berger, J., Barkaoui, M. A parallel hybrid genetic algorithm for the vehicle routing problem with time window. **Computers & Operations Research**. 2002.
- Bodin, L., Golden, B., Assad, A., Ball, M. Routing and scheduling of vehicles and crews: the state of the art. **Computers & Operations Research**, v.9, p. 63-212, 1983
- Bräysy, O., Hasle, G., Dullaert, W. A Multi-Start Local Search Algorithm for the Vehicle Routing Problem With Time Windows. **European Journal of Operational Research**, 2003.
- Breedam, V. A. (2001) Comparing descent heuristics e metaheuristics for the vehicle routing problem. **Computer & Operational Research**, V.28, p. 289-315, 2001
- Brown, G. G., Graves, G. W. Real-Time Dispatch of Petroleum Tank Trucks. **Management Science**, v 27: 19-32, 1981.

- Chajakis, E. D., Guignard, M. Scheduling Deliveries in Vehicles with Multiple Compartments. **Journal of Global Optimization**, v.26, p.43-78, 2003.
- Chaovalitwongse, W., Kim, D., Pardalos, P. M. GRASP with a New Local Search Scheme for Vehicle Routing Problems with Time Windows. **Journal of Combinatorial Optimization**, v. 7, p 179-207, 2003.
- Chapleau, L., Ferland, J. Rousseau, J. M. Clustering for routing in dense area. University of Montreal. **Transportation Research Center Publication**, n.206, 1981.
- Christopher, M. **Logística e Gerenciamento da Cadeia de Suprimentos: Estratégias para a Redução de Custos e Melhoria dos Serviços**; São Paulo; 2p, 1997.
- Clark & Wright, J. Scheduling of Vehicles from a central Depot to a Number of Delivery Points. **Operation Research**. 12:856-861, 1964
- Cook, W. J., Cunningham, W. H., Pulleyblank, W. R., Schrijver, A. Combinatorial Optimization. **Inter-Science**, 1998.
- Desrochers, M. Lenstra, J. K., Savelsbergh, M. W. P. A classification schema for vehicle routing and scheduling problems. **European Journal of Operational Research**, v.46, p.322-332, 1990.
- Dorigo, M., Caro, G. D. Ant algorithms for discrete optimization. **Artificial Life**, v. 5, n. 3, p. 137-172, 1999
- Evans, S. R., Norback, J. P. The Impact of a Decision-Support System for Vehicle Routeing in a Foodservice Supply Situation. **Management Science**, v 36: 467-472, 1985.
- Feo, T. A., Resende, M.G.C. Greedy Randomized Adaptive Search Procedures. **Journal of Global Optimization**, v. 6, p. 109-133, 1995.

- Gaskell, T. J. Bases for vehicle fleet scheduling. **Operational Research**, n.18, p.281-295, 1967.
- Gillet, A., Johnson, J. Multi-Terminal vehicle dispatch problem. **Omega**, n.4, p.711-718, 1976.
- Gillet, B. Miller, L. A heuristic algorithm for the vehicle-dispatch problem. **Operational Research**, n.22, p.340-349, 1974.
- Glover, F. Tabu Search – part I. **ORSA Journal on Computing**, 1:3, 190-206, 1989.
- _____ Tabu Search – part II. **ORSA Journal on Computing**, 2:1, 4-32, 1990.
- Golden, B., Wasil, E. A. Computerized Vehicle Routing in the Soft Drink Industry. **Operations Research**, v 35: 6-18, 1987.
- Graham, R. (1972). An efficient algorithm for determining the convex hull of a finite planar point set. **Info. Proc. Letters**. V. 1: 132-133, 1972.
- Ho S. C., Haugland D. A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. **Computers & Operations Research**, 2003.
- Holland, J. H. **Adaptation in Natural and Artificial Systems**. Ann Arbor, University of Michigan Press, 1975.
- Ioannou, G., Kritikos, M., Prastacos, G. A Problem Generator-Solver Heuristic for Routing With Soft Time Windows. **The International Journal of Management Science**. V 31: 41-53, 2003
- Jarvis, R. A. On the identification of the convex hull of a finite set of points in the plane. **Info. Proc. Letters**. V. 2: 18-21, 1973.
- Júnior, A. C. G., Souza, M. J. F., Martins, A. X. Um algoritmo simulated annealing eficiente para o problema de roteamento de veículos com janela de tempo.

Encontro Nacional de Engenharia de Produção 25, Porto Alegre: Pontifícia Universidade Católica, 2005.

Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P. Optimization by Simulated Annealing. **Science**, n. 220, p.671-680, 1983.

Kontoravdis G., Bard, J. F. A GRASP for the Vehicle Routing Problem with Time Window. **Journal on Computing**, v. 7, n. 1, p 10-22, 1994.

Laporte, G. M. Gendreau, J.Y. Potvin e F. Semet Classical and modern heuristics for the vehicle routing problem, **International Transaction in Operational Research**, v.7, n.4/5, p.285-300, 2000

Lau, H. C., Sim, M., Teo, K. M. (2003). Vehicle routing problem with time window and a limited number of vehicles. **European Journal of Operational Research**, v.148, p. 559-569, 2003.

Lima, M. P. Custos Logísticos na Economia Brasileira. **Revista Tecnológica**. 01:p 64-69, 2006.

Lin, S. Computer solutions of the traveling salesman problem. **Bell System Computer Journal**, n.44, p.2245-2269, 1965.

Lisboa Filho, J. **Projeto Conceitual de Banco de Dados Geográficos Através da Reutilização de Esquemas, Utilizando Padrões de Análise e um Framework Conceitual**. Tese de Doutorado. Porto Alegre – RS, Universidade Federal do Rio Grande do Sul – UFRGS, 2000.

Mladenovic, N., Hansen, P. Variable neighborhood search. **Computers Operational Research**, v. 24, p. 1097-1100, 1997.

Moscato, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. **Caltech Concurrent Computation Program**, report 826, 1989.

- Nelson, M., Nygard, K., Griffin, j. Shereve, W. Implementation techniques for the vehicle routing problem. **Computers & Operational Research**, v.12, p 273-283, 1985.
- Noronha T. F. (2001), Uma Abordagem sobre Estratégias Metaheurísticas. **Revista Eletrônica de Iniciação Científica**, v 1, n 1, 2001.
- Oliveira, H. C. B., Souza M. M., Alvarenga, G. B., Silva, R. M. A. Adaptação do Algoritmo Genético no Problema de Roteamento de Veículos com Janela de Tempo. **Revista Infocomp Universidade Federal de Lavras**, v. 3, n. 2, p. 51-58, 2001.
- Oliveira, J. L., Pires, F., Medeiros, C. An environment for modeling and design of geographic applications. **Geoinformatica**, Boston, n.1, 29-58, 1997.
- Paessens, H. Saving algorithms for the vehicle routing. *European Journal of Operational Research*, v.34, pp.336-344, 1988.
- Póvoa, C. L. **Heurística GRASP para o problema do roteamento de veículos com multi-compartimentos e sua integração com o sistema de informação geográfica Geo-Rota**. Tese de Doutorado em Engenharia de Produção - Campos dos Goytacazes - RJ. Universidade Estadual do Norte Fluminense - UENF, 183 p, 2005.
- Preparata, F. P., Shamos, M. I. **Computacional Geometry: na Introduction**. New York: Springer – Verlag, 1985.
- Pureza, V. M., França, P. M. Vehicle routing problems via tabu search metaheuristic. Montreal. Centre de recherche su les transports, **Report CRT – 747**, 1991.
- Santana, J. B., Rodrigues, C. C., López, F. G., Torres, M. G., Batista, B. M., Pérez, J. A. M., Vega, J. M. M. Metaheurísticas: Uma revision actualizada. **Departamento de Estatística, Investigacion Operativa y Computacion**, n. 2, 2004.

Solomon, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations Research**, 35:2, 254-265, 1987.

Viana, G. V. R. (1996) **Meta-heurísticas para solução de problemas de otimização combinatória**. Tese de Mestrado em Ciência da Computação – Fortaleza – CE, Universidade Federal do Ceará – UFC, 1996.

Znamesnki, A.; Cunha, C. B. (2001) Um Modelo para o Problema de Roteirização e Programação do Trsnporte de Deficientes. São Paulo, **Makron Books do Brasil**, São Paulo, p.1-12, 2001.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)