



UMA SOLUÇÃO PARA O PROBLEMA DE ROTEAMENTO DE EMBARCAÇÕES DE APOIO “OFFSHORE” ATRAVÉS DA METAHEURÍSTICA RTR

Pedro Henrique Maciel Lopes

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Luiz Pereira Calôba

Rio de Janeiro
Outubro de 2011

UMA SOLUÇÃO PARA O PROBLEMA DE ROTEAMENTO DE EMBARCAÇÕES DE
APOIO “OFFSHORE” ATRAVÉS DA METAHEURÍSTICA RTR

Pedro Henrique Maciel Lopes

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Luiz Pereira Calôba, Ph.D.

Prof. José Gabriel Rodriguez Carneiro Gomes, Ph.D.

Prof. Laura Silvia Bahiense da Silva Leite, D.Sc.

Prof. Luiz Satoru Ochi, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

OUTUBRO DE 2011

Lopes, Pedro Henrique Maciel

Uma Solução para o Problema de Roteamento de Embarcações de Apoio “Offshore” através da Metaheurística RTR/ Pedro Henrique Maciel Lopes. – Rio de Janeiro: UFRJ/COPPE, 2011.

XIII, 73 p.: il.; 29,7 cm.

Orientador: Luiz Pereira Calôba

Dissertação (mestrado) – UFRJ/ COPPE/ Programa de Engenharia Elétrica, 2011.

Referencias Bibliográficas: p. 71 – 73.

1. Busca Local. 2. Apoio “Offshore” 3. Metaheurística RTR. I. Calôba, Luiz Pereira. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

À minha esposa

AGRADECIMENTOS

À minha esposa, pelo carinho e pela compreensão incondicionais.

Aos meus pais, pelo exemplo e pela formação que me proporcionaram, e aos meus irmãos pelo apoio e descontração.

Ao meu orientador, Professor Luiz Calôba, pela paciência e pelos valiosos ensinamentos ao longo de todo o curso.

Aos professores José Gabriel Gomes, Laura Silvia Leite e Luiz Satoru Ochi, por se disponibilizarem a participar da Banca Examinadora e pelas valiosas sugestões de melhoria do texto.

Ao Mayron de Almeida, colega de Petrobras, por sua contribuição direta em grande parte dos resultados dessa dissertação.

À Petrobras, pela liberação em tempo parcial para a conclusão do estudo. Ao meu chefe, Luiz Claudio, e aos amigos de gerência, em especial ao Marcelo e ao Victor, pelo incentivo.

À UFRJ, por me aceitar novamente como seu aluno. Em especial à Daniele, funcionária do PEE, pelo profissionalismo e agilidade na prestação de informações.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UMA SOLUÇÃO PARA O PROBLEMA DE ROTEAMENTO DE EMBARCAÇÕES DE APOIO “OFFSHORE” ATRAVÉS DA METAHEURÍSTICA RTR

Pedro Henrique Maciel Lopes

Outubro/2011

Orientador: Luiz Pereira Calôba

Programa: Engenharia Elétrica

Esse trabalho tem como objetivo enunciar e resolver o Problema de Roteamento de Embarcações de Apoio Offshore da Petrobras através da metaheurística RTR. Após simplificações, o problema é enquadrado na literatura como Problema de Roteamento de Veículos de Frota Heterogênea Limitada. Para solucioná-lo, a biblioteca de Buscas Locais VRPH é adaptada e os resultados obtidos são comparados com os melhores conhecidos. Adicionalmente, uma instância baseada no problema real é gerada e o resultado alcançado é comparado com o obtido pela solução adotada na empresa.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A SOLUTION BASED ON THE RTR METAHEURISTIC TO SOLVE THE OFFSHORE
SUPPORT VESSELS ROUTING PROBLEM

Pedro Henrique Maciel Lopes

October/2011

Advisor: Luiz Pereira Calôba

Department: Electrical Engineering

This work aims to formulate and solve the Offshore Support Vessels Routing Problem arising at Petrobras through the RTR metaheuristic. After simplifications, the problem is framed in the literature as Heterogeneous Limited Fleet Vehicle Routing Problem. In order to solve it the Local Search Library VRPH is adapted and the obtained results are compared to the best known solutions. Additionally, an instance inspired on the real problem is generated and the obtained result is compared to the solution adopted by the company.

SUMÁRIO

CAPÍTULO I INTRODUÇÃO.....	1
I.1 Descrição do Negócio	2
I.2 Produtos Transportados.....	4
I.3 Unidades Marítimas	6
I.3.1 Sondas de Perfuração	6
I.3.2 Unidades de Produção.....	8
I.3.3 Barcos Especiais	9
I.3.4 Embarcações	9
I.4 Bases de Apoio	12
I.5 Objetivo.....	13
I.6 Delimitação do Escopo	14
I.7 Estrutura da Dissertação.....	15
CAPÍTULO II REVISÃO DE LITERATURA	17
II.1 Descrição do Problema	17
II.2 Extensões	20
II.3 Abordagens de Solução.....	21
II.3.1 Métodos Exatos	22
II.3.2 Heurísticas.....	22
II.3.2.1 Heurísticas de Construção de Rotas.....	23
II.3.2.2 Heurísticas de Busca Local	23
II.3.3 Metaheurísticas	27
II.3.3.1 Algoritmo Genético de Almeida [16]	29
CAPÍTULO III METODOLOGIA DA PESQUISA	34
III.1 Algoritmo RTR de Li [27].....	34
III.1.1 Estrutura do Algoritmo HRTR.....	35
III.1.2 Método de Geração da Solução Inicial.....	38
III.1.3 Método de Perturbação	39
III.1.4 Métodos de Busca Local.....	41
III.2 Biblioteca VRPH.....	42
III.2.1 Estrutura.....	42

III.2.2 Características	44
III.2.2.1 Codificação da Solução Atual	44
III.2.2.2 Métodos de Geração da Solução Inicial	46
III.2.2.3 Método de Criação da Vizinhança	47
III.2.2.4 Métodos de Busca Local.....	47
III.2.3 Outras Funcionalidades	48
III.2.4 Modificações Necessárias.....	48
III.3 Instâncias Teste	49
III.3.1 Instâncias <i>Benchmark</i>	49
III.3.1.1 Instâncias de Taillard [38].....	49
III.3.1.2 Instâncias de Li [27]	51
III.3.2 Instância Real	54
CAPÍTULO IV RESULTADOS E DISCUSSÕES	57
IV.1 Resultados Obtidos para as Instâncias <i>Benchmark</i>	57
IV.2 Análises de Sensibilidade	62
IV.2.1 Métodos de Busca Local.....	62
IV.2.2 Espaço de Busca.....	63
IV.3 Resultados Obtidos para Instância Real	64
CAPÍTULO V CONCLUSÕES	68
IV.1 Conclusões.....	68
IV.2 Sugestões de Trabalhos Futuros.....	69
REFERÊNCIAS BIBLIOGRÁFICAS.....	71

ÍNDICE DE FIGURAS

Figura I-1 – Produção nacional de petróleo e LGN por origem. Dados de Agosto de 2010.	1
Figura I-2 – Esquema representando o fluxo de materiais da cadeia de apoio <i>offshore</i> da Petrobras.....	3
Figura I-3 – Exemplos de carga geral em contentor (a) e em <i>skid</i> (b) extraídos de [3].....	4
Figura I-4 – Mapa com as unidades da Petrobras na Bacia de Campos.	6
Figura I-5 – Navio-Sonda NS-17, Leo Segerius [5].	7
Figura I-6 – Unidade Estacionária de Produção P-54 da Petrobras [1].	8
Figura I-7 – Navio de lançamento de linhas Acergy Pertinacia [6] (a) e navio de estimulação Blue Shark [7] (b).	9
Figura I-8 – Tanques e convés do PSV [8]	10
Figura I-9 – Entrega de carga sob mau tempo na Bacia de Campos extraído de [10].....	11
Figura I-10 – Terminal Alfandegado de Imbetiba, Macaé, Rio de Janeiro [2].	12
Figura I-11 – Mapa com os locais das principais bases de apoio <i>offshore</i> operadas pela Petrobras [2].	13
Figura II-1 – Exemplo de solução para o Problema de Roteamento de Veículos	18
Figura II-2 – Exemplo do movimento de um ponto	24
Figura II-3 – Exemplo do movimento de dois pontos	24
Figura II-4 – Exemplo do 2-Opt intra-rota.....	25
Figura II-5 – Exemplo do 2-Opt inter-rotas	25
Figura II-6 – Exemplo do Or-Opt.....	26
Figura II-7 – Exemplo do 3-Opt intra-rota.....	26
Figura II-8 – Exemplo do movimento de três pontos	27
Figura II-9 – Exemplo da Troca Cruzada.....	27
Figura II-10 – Esquema exibindo os passos do Algoritmo Genético	30
Figura II-11 – Exemplo de codificação baseada em [16]	31
Figura II-12 – Exemplo de codificação baseada em [41]	32
Figura II-13 – Interface de testes do Roteirizador Marítimo da Petrobras	33
Figura III-1 – Estrutura resumida do HRTR	36
Figura III-2 – Esquema representando a inserção do cliente i na rota do veículo t	40

Figura III-3 – Diagrama de Classes da VRPH elaborado pelo autor e modificações propostas no presente trabalho	43
Figura III-4 – Disposição física dos nós das instâncias T_13 e T_20	51
Figura III-5 – Disposição física dos nós das instâncias H1 e H5 de Li	52
Figura III-6 – Disposição geográfica dos pontos de atendimento	55
Figura IV-1 – Resultado da Instância T_13	58
Figura IV-2 – Evolução do custo na instância T_13	59
Figura IV-3 – Resultado da instância H1	59
Figura IV-4 – Evolução do custo na instância H1	60
Figura IV-5 – Heurística de Inserção de Menor Custo aplicada a H1	60
Figura IV-6 – Primeira solução obtida para a instância real	64
Figura IV-7 – Resultado baseado no Roteirizador Marítimo de [16]	65
Figura IV-8 – Melhor solução obtida para a instância real	66

ÍNDICE DE TABELAS

Tabela I-1 – Quadro contendo características de projeto de 4 embarcações de transporte..	12
Tabela I-2 – Escopo do presente trabalho	14
Tabela III-1 – Esquema de funcionamento do algoritmo RTR [46].....	35
Tabela III-2 – Estrutura do algoritmo de Li [43].....	37
Tabela III-3 – Estrutura do algoritmo de geração da solução inicial.....	38
Tabela III-4 – Estrutura do método de perturbação de Li.....	40
Tabela III-5 – Estrutura do movimento de um ponto na Fase de Uphill [43]	41
Tabela III-6 – Exemplos de rotas.....	44
Tabela III-7 – Codificação do exemplo	44
Tabela III-8 – Novas rotas após inserção	45
Tabela III-9 – Codificação das novas rotas	45
Tabela III-10 – Utilização da frota nas instâncias de Taillard.....	50
Tabela III-11 – Dados de veículos nas instâncias de Taillard	50
Tabela III-12 – Utilização da frota nas instâncias de Li	52
Tabela III-13 – Dados de veículos nas instâncias de Li.....	53
Tabela III-14 – Pseudo-código do gerador de instâncias [27].....	53
Tabela III-15 – Parâmetros das instâncias de Li.....	54
Tabela III-16 – Embarcações disponíveis para o caso real	56
Tabela IV-1 – Comparativo de resultados obtidos	57
Tabela IV-2 – Movimentos e avaliações realizadas nas instâncias T_13 e H1	61
Tabela IV-3 – Impacto das buscas locais no <i>Uphill</i>	62
Tabela IV-4 – Impacto das buscas locais no <i>Downhill</i>	63
Tabela IV-5 – Impacto da utilização de vizinhos	63
Tabela IV-6 – Esquema de funcionamento do movimento de concatenação elaborado no presente trabalho	66

ÍNDICE DE ABREVIATURAS

AG	Algoritmo Genético;
COIN-OR	Computational Infrastructure for Operations Research
DSV	Diving Support Vessel
FPSO	Floating, Production, Storage and Offloading Unit
FSMVRP	Problema de Dimensionamento de Frota e Mix
HRTR	Algoritmo RTR para Roteamento de Veículos de Frota Heterogênea
HVRP	Problema de Roteamento de Veículos de Frota Heterogênea Limitada
IACM	Índice de Atendimento de Carga Marítima
LGN	Líquido de Gás Natural
PLV	Pipe Laying Vessel
PSV	Platform Supply Vessel
RT	Requisição de Transporte
RTR	Record-to-Record Travel Algorithm
RV	Research Vessel
TAI	Terminal Alfandegado de Imbetiba
TPB	Tonelada de Porte Bruto
TSP	Problema do Caixeiro Viajante
UEP	Unidade Estacionária de Produção
UM	Unidade Marítima
US-LOG	Unidade de Serviços de Logística
UT	Embarcação de Transporte Expresso – Utility
VRP ou CVRP	Problema de Roteamento de Veículos Capacitado
VRPH	Biblioteca de Buscas Locais da COIN-OR
WSV	Well Stimulation Vessel

CAPÍTULO I INTRODUÇÃO

A indústria do Petróleo possui uma complexa cadeia produtiva. Alguns de seus processos são destacadamente conhecidos como as atividades de Exploração e Produção e Refino de petróleo.

A atividade de Exploração e Produção está relacionada às operações de sísmica, perfuração e completação de poços e produção de petróleo. Por sua vez, o Refino de petróleo trata da separação e transformação dos diversos componentes do petróleo em produtos preparados para o consumo na forma de combustíveis (gasolina, diesel, querosene de aviação), óleos básicos lubrificantes e muitas outras frações com as mais variadas funções.

O presente trabalho se encaixa num setor de suporte às etapas de Exploração e Produção, o Apoio Marítimo. No Brasil, grande parte das reservas e da produção de petróleo está localizada em alto mar a uma lâmina d'água que pode chegar a 2000 m. Como evidencia a Figura I-1, 89% da produção de petróleo no Brasil advém de instalações localizadas no mar sendo que 83% provém da Bacia de Campos [1]. Dessa forma, é fundamental constantemente construir e manter infra-estrutura, plataformas e equipamentos submarinos, que viabilizem a extração do petróleo.

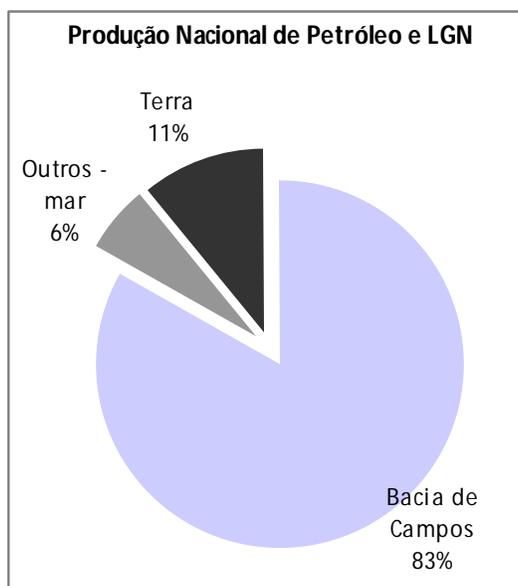


Figura I-1 – Produção nacional de petróleo e LGN por origem. Dados de Agosto de 2010.

Toda a infra-estrutura *offshore* precisa ser abastecida com insumos (tubos, produtos químicos, rancho, equipamentos e ferramentas). Além disso, há operações especiais como o manuseio de âncoras e a instalação de equipamentos submarinos que também compõem o Apoio Marítimo.

O maior complexo produtivo *offshore* instalado no Brasil (um dos maiores do mundo) é o da Petrobras na Bacia de Campos. São mais de 150 embarcações destinadas a atender cerca de 70 unidades marítimas. Os custos dessa operação são extremamente relevantes para o setor de óleo e gás. Na Petrobras, por exemplo, o custo total anual das operações de transporte é da ordem de US\$ 2 bilhões. Com a perspectiva de crescimento do setor de óleo e gás no Brasil devido às recentes descobertas do Pré-Sal nas Bacias de Santos e do Espírito Santo, a atividade de Apoio *Offshore* é diretamente afetada. A estimativa é de que essa operação dobre de tamanho até 2017.

Diante desse cenário, esse capítulo tem como objetivo definir e delimitar o problema abordado nesse trabalho. Começaremos por definir o sistema a ser analisado.

I.1 Descrição do Negócio

O sistema logístico de apoio *offshore* é composto por um fluxo de fornecimento que começa com os fornecedores nacionais e estrangeiros, passa pelos armazéns e terminais portuários e termina com o atendimento das unidades marítimas como exibido na Figura I-2 obtida de [2].

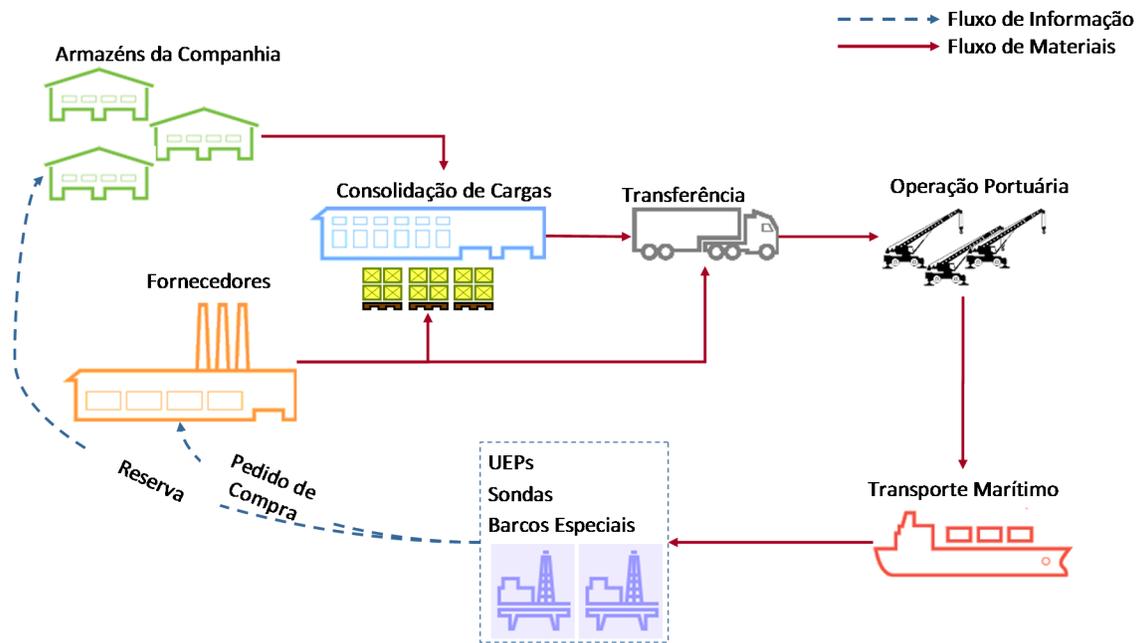


Figura I-2 – Esquema representando o fluxo de materiais da cadeia de apoio offshore da Petrobras.

Esse fluxo de materiais é acompanhado por um fluxo de informação que começa com o pedido da Unidade Marítima (UM) ao detentor do material (onde o mesmo se encontra estocado). Os materiais podem estar armazenados nas instalações da Petrobras ou de fornecedores. Tanto os armazéns de fornecedores quanto os da Petrobras têm como missão disponibilizar os materiais, através do transporte terrestre executado com carretas, até as bases de suprimento indicadas pela Petrobras.

Cada pedido em sua origem é transformado em uma ou mais Requisições de Transporte (RTs). Depois de tratadas – coletadas, unitizadas e transportadas até o terminal portuário – as RTs são levadas ao status de “Liberado para Programação”. As RTs liberadas para programação sofrem um processo de alocação a uma das embarcações de transporte disponíveis.

Algumas informações são importantes nesse processo de programação. O peso e as dimensões (largura e comprimento ou volume) das RTs limitam a quantidade de carga a ser colocada no convés ou nos tanques da embarcação. Além disso, é imprescindível levar em consideração o prazo de entrega de cada RT. A cada uma delas é atribuída uma data mais cedo e uma data mais tarde para a entrega. Se a entrega for efetuada antes da data mais cedo ou depois da data mais tarde (ou seja, fora da janela de entrega), a mesma não

contribuirá para o IACM (Índice de Atendimento de Carga Marítima). Esse indicador mede a quantidade de RTs atendidas dentro da janela em relação ao total de RTs atendidas.

Diante dessa breve descrição do sistema logístico, cabe detalhar seus principais elos para que se tenha uma noção mais completa da atividade de Exploração e Produção e suas interfaces com a atividade de Apoio Marítimo. Serão descritos as Unidades Marítimas, as embarcações de transporte, as bases de apoio e os principais produtos transportados ao longo dessa cadeia.

I.2 Produtos Transportados

Há uma grande gama de insumos necessários à operação *offshore*. Entre os produtos transportados podemos destacar:

- Carga Geral é toda a carga carregada no convés das embarcações. Pode ser dividida em 4 principais tipos: rancho, tubos, produtos químicos e carga geral comum. São apresentados na Figura I-3 exemplos de contentores de carga geral.



(a)



(b)

Figura I-3 – Exemplos de carga geral em contentor (a) e em *skid* (b) extraídos de [3].

- Granéis Líquidos são armazenados em tanques no porão das embarcações. Os principais tipos de granéis líquidos são: água, diesel marítimo, fluidos de poços, parafina e salmoura.

- Granéis Sólidos são também armazenados em tanques no porão das embarcações. Os principais tipos de granéis sólidos são: cimento, baritina e bentonita.
- Os fluidos de poços têm funções variadas. Entre elas podemos destacar: garantia da integridade e temperatura da broca, controle da pressão da coluna de perfuração e limpar o fundo do poço de cascalhos e transportá-los até a superfície. Já a salmoura, a baritina e a bentonita ajudam a controlar as propriedades físicas (salinidade, densidade, viscosidade e teor de sólidos ativos, por exemplo) desses fluidos ao serem misturados a eles.
- Os tubos, por sua vez, são utilizados tanto durante a fase de perfuração (*drill pipes*) quanto para o revestimento do poço. Cimento também tem função estrutural e é utilizado após o revestimento de modo a fixar a tubulação e evitar que haja migração de fluidos por entre os tubos de revestimento.

Quanto à carga geral, os insumos químicos (unitizados como na Figura I-3 (a)) mais consumidos pelas unidades de produção são os antiespumantes, desemulsificantes, inibidores de corrosão e de hidratos. Seus usos estão diretamente relacionados à garantia de escoamento e ao processamento primário de fluidos. Além disso, outros insumos como o rancho, a carga comum (material de escritório, por exemplo) além de ferramentas e equipamentos voltados para a manutenção são extensivamente utilizados. Para informações mais detalhadas sobre esses itens consultar [4].

Além disso, existe um grande volume de carga que retorna das plataformas, chamada de *backload*. O *backload* faz o fluxo inverso ao material de entrega e é constituído, por exemplo, de lixo, sucata, resíduos químicos, tanques e contentores vazios e tubos usados, sendo representativo na forma de carga geral (cerca de 45% de toda a carga geral movimentada) e granel líquido (70% do fluido de poços volta para ser reprocessado). Como se pode observar, o *backload* é composto tanto de inservíveis quanto de material que será reaproveitado num momento oportuno.

I.3 Unidades Marítimas

As unidades atendidas pelo Apoio Marítimo podem ser classificadas em Sondas de Perfuração, Unidades Estacionárias de Produção e os Barcos Especiais. Essas unidades são responsáveis pelos pedidos de materiais.

Cada um desses tipos de unidade tem necessidades diferentes em termos das cargas necessárias, urgência e incerteza na demanda. Dessa forma, cabe descrever cada um desses tipos.

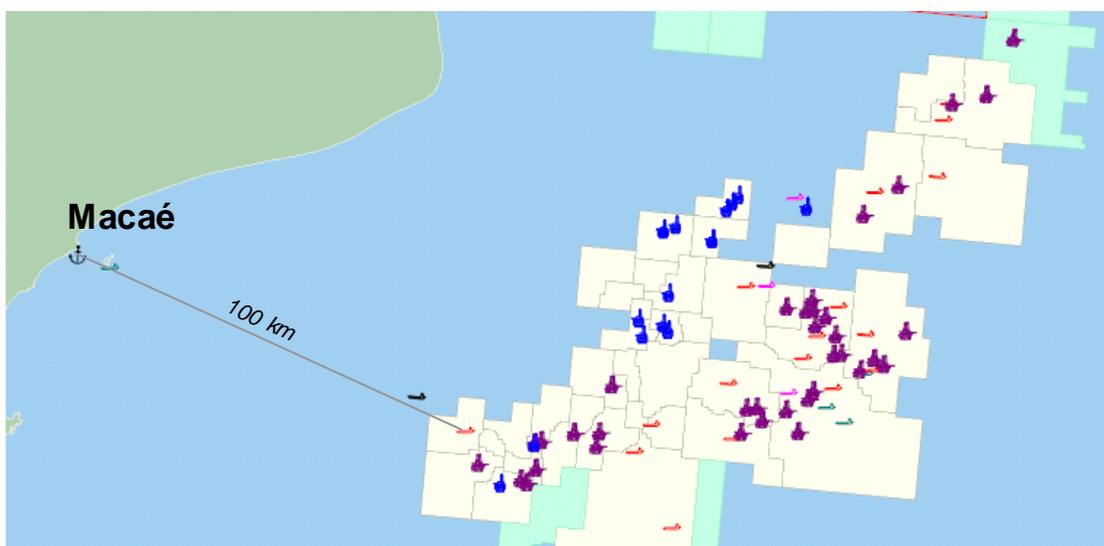


Figura I-4 – Mapa com as unidades da Petrobras na Bacia de Campos.

A título de ilustração, a Figura I-4, obtida a partir de [2] evidencia as unidades marítimas que compõem os clientes do sistema logístico considerado e o ponto de partida dos atendimentos, a cidade de Macaé. A unidade mais próxima se encontra a 100 km de distância.

I.3.1 Sondas de Perfuração

As sondas de Perfuração são unidades destinadas a perfurar e completar poços. Os poços podem ser injetores ou produtores dependendo da estratégia de recuperação a ser adotada. Para a atividade de perfuração, alguns insumos são essenciais como os fluidos de

perfuração, a salmoura, a baritina e a bentonita e os tubos conforme detalhado na Seção I.2.



Figura I-5 – Navio-Sonda NS-17, Leo Segerius [5].

Do ponto de vista do Apoio Marítimo, um ponto fundamental inerente à operação das sondas de perfuração (como a NS-17, da Figura I-5) é a incerteza em relação ao seu consumo. A sonda está constantemente sendo mobilizada para novas áreas, onde o conhecimento do subsolo pode ser relativamente pequeno. Somando isso à baixa capacidade de armazenamento na plataforma, a consequência obtida é uma elevada quantidade de pedidos de emergência.

Além disso, as sondas de perfuração são as unidades marítimas mais caras entre todas as que operam no ambiente *offshore*. A diária de um navio-sonda alugado à Petrobras, por exemplo, custa em torno de US\$ 500.000. Esse valor dá uma idéia do que significa que o custo de atraso de um dia num insumo fundamental para seu funcionamento.

I.3.2 Unidades de Produção

As unidades de produção (como a P-54 na Figura I-6) são responsáveis pela extração do petróleo e gás natural desde os reservatórios e pelo processamento primário dos mesmos. O transporte do petróleo até o continente pode ser feito tanto por navios aliviadores como dutos, dependendo da infra-estrutura instalada para o escoamento do óleo.

Ao contrário das unidades de perfuração, as unidades de produção possuem um padrão de consumo mais estável. Além disso, seu posicionamento tende a ser fixo durante toda a vida útil do campo de petróleo, diferentemente das sondas que estão em constante deslocamento.



Figura I-6 – Unidade Estacionária de Produção P-54 da Petrobras [1].

Um fator que impacta o padrão de consumo das unidades de produção é a sua idade. Plataformas entrando em operação (com até 6 meses de idade) precisam de uma grande quantidade de recursos mobilizados para suprir eventuais serviços de construção e montagem que tenham ficado pendentes da época em que estava no estaleiro. No outro extremo, unidades muito antigas (com mais de 10 anos) demandam muita manutenção. Há também paradas programadas nas unidades de todas as idades. Com exceção a esses casos, a demanda tende a ser bastante previsível, com o consumo mais destacado de produtos químicos e rancho.

I.3.3 Barcos Especiais

Uma série de serviços auxiliares são realizados para a instalação e manutenção dos sistemas produtivos. Entre esses barcos especiais podemos destacar:

- Embarcações de Apoio ao Mergulho (*Diving Support Vessels – DSV*)
- Embarcações de Lançamento de Linhas (*Pipe Laying Vessels – PLV*)
- Navio de Estimulação de Poços de Petróleo (*Well Stimulation Vessel – WSV*)
- Navio de Pesquisa Sísmica (*Research Vessel – RV*)

Essas embarcações (como as exibidas na Figura I-7) se caracterizam por virem à base eventualmente. Mesmo assim, os insumos necessários para o seu trabalho são trazidos pelas embarcações de transporte. Assim como as sondas, os barcos especiais são deslocados constantemente.



(a)



(b)

Figura I-7 – Navio de lançamento de linhas Acergy Pertinacia [6] (a) e navio de estimulação Blue Shark [7] (b).

I.3.4 Embarcações

O trecho marítimo da viagem até a plataforma é usualmente feito através de embarcações denominadas *Platform Supply Vessels* (PSV, conforme exibidos na Figura I-8 e Figura I-9) que fazem a ligação das bases de Apoio *Offshore* até as unidades marítimas. As unidades marítimas são atendidas regularmente, de acordo com os pedidos colocados à US-LOG, Unidade de Serviços de Logística, órgão da Petrobras responsável pelo transporte das cargas aos seus respectivos destinos.

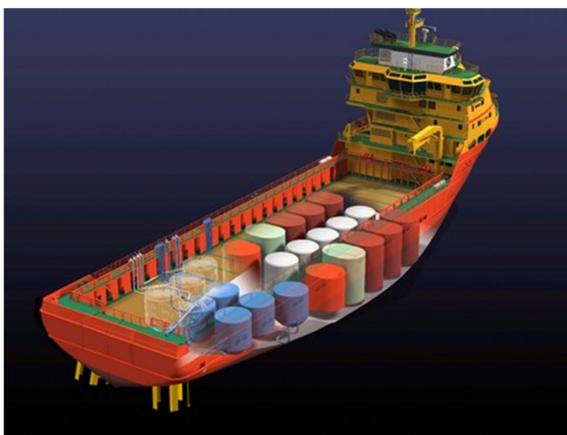


Figura I-8 – Tanques e convés do PSV [8]

Esse, no entanto, não é o único tipo de embarcação disponível. Há embarcações menores como o UT (*Utility*) reservado para transportes expressos devido a sua maior velocidade.

Dessa forma, cabe explicitar as principais embarcações utilizadas, suas funções e principais características. É necessário ressaltar que muitas das embarcações envolvidas nas atividades de apoio offshore não serão tratadas aqui. Para maiores informações, sugere-se consultar [9].

Os PSVs são embarcações de transporte multi-propósito. Seu porte geralmente varia de 1000 toneladas de porte bruto (TPB) até 4500 TPB. O porte impacta a quantidade de tanques e sua capacidade de armazenagem. Além disso, tem reflexos sobre o tamanho da área de convés, que pode variar entre 180 a 940 m², e sobre o consumo de combustível.

Pela grande quantidade de produtos a fornecer, a Petrobras adotou como política segmentar o atendimento por tipo de carga. Assim, há PSVs dedicados ao transporte de fluidos e baritina e bentonita (PSV Fluideiro Graneleiro), outros dedicados ao transporte de cimento (PSV Cimenteiro) e outros a carga geral, a água e óleo diesel.



Figura I-9 – Entrega de carga sob mau tempo na Bacia de Campos extraído de [10].

A Utility (UT) é uma embarcação de pequeno porte com velocidade de serviço que pode chegar a 30 nós. É significativamente mais rápida que o PSV, cuja velocidade é de 10 nós. Possui menor resistência do piso o que implica que não pode levar cargas pesadas. A UT não pode levar tubos nem produtos químicos, por exemplo. Seu objetivo é, portanto, suprir as unidades marítimas de carga geral de pequeno porte (ferramentas, equipamentos) que sejam urgentes e impactem diretamente na operação das unidades.

A Tabela I-1 resume as principais características de 4 embarcações de diferentes tipos contratadas pela Petrobras. Seus nomes são: Fast Vinicius [11], Astro Vermelho [12], SeaBulk Brasil [13] e Far Supporter [14].

Tabela I-1 – Quadro contendo características de projeto de 4 embarcações de transporte.

	UT	PSV 1500	PSV 3000	PSV 4500
Embarcação	Fast Vinicius	Astro Vermelho	SeaBulk Brasil	Far Supporter
Calado (m)	2.7	4.1	5.8	6.2
Boca (m)	9.0	12.0	16.0	18.8
Comprimento (m)	44.0	66.0	71.9	84.0
TPB (t)	370	1,047	3,250	4,680
Área de Convés	160	266	688	955
Água (m3)	90	776	1,078	1,385
Diesel (m3)	64	735	1,010	1,980

I.4 Bases de Apoio

As Bases de Apoio são responsáveis pelo suprimento das embarcações. Nas bases, as embarcações de transporte descarregam o *backload* e carregam carga geral, água e diesel, fluidos de perfuração e granéis sólidos que suprirão as unidades marítimas.

No Brasil, a maior base de apoio *offshore* é o Terminal Alfandegado de Imbetiba (TAI, exibido na Figura I-10), que abastece toda a Bacia de Campos. Construído em 1978, tem capacidade de receber 6 embarcações simultaneamente (sem considerar as operações à contrabordo) com calado de até 8 metros. É necessário ressaltar que este é um terminal privativo, no qual apenas embarcações autorizadas pela Petrobras podem operar.



Figura I-10 – Terminal Alfandegado de Imbetiba, Macaé, Rio de Janeiro [2].

As embarcações ficam fundeadas à espera de programação. A partir do momento em que são programadas, elas esperam vaga em um dos berços do terminal para efetuar as operações de descarga do *backload* e carregamento dos itens programados.

Os terminais em que a Petrobras opera estão dispostos no mapa da Figura I-11.

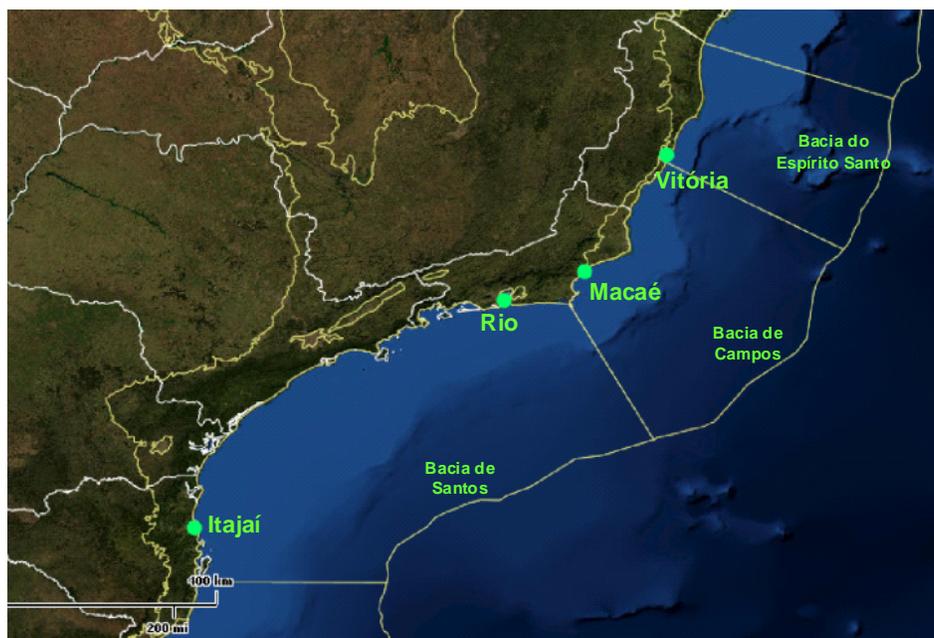


Figura I-11 – Mapa com os locais das principais bases de apoio *offshore* operadas pela Petrobras [2].

I.5 Objetivo

O presente trabalho trata justamente desse conjunto de decisões que compõem a programação das embarcações: alocação de cargas e da atividade subsequente, a definição das rotas marítimas de atendimento. Ambas as atividades estão relacionadas visto que as cargas alocadas a uma embarcação determinam os pontos (Unidades Marítimas) que precisam ser visitados em sua rota.

Em linhas gerais, o estudo pretende desenvolver um método de solução eficiente para o problema descrito de modo a:

- Complementar a abordagem já utilizada na empresa para que esta se torne ainda melhor
- Contribuir para o desenvolvimento de trabalhos acadêmicos nessa mesma linha ao enunciar o problema e indicar possíveis caminhos para sua solução

I.6 Delimitação do Escopo

A solução do problema real de roteamento de embarcações *offshore* envolve:

- Minimizar os custos fixos (função da necessidade de frota) e variáveis (consumo de combustível – função da distância percorrida)
- Respeitar os prazos de entrega das cargas ou, caso não seja possível, minimizar os atrasos nas entregas
- Respeitar a capacidade dos diferentes tipos de embarcações para os diversos tipos de carga geral, granéis sólidos e líquidos levando em consideração as operações de *upload* e *backload*
- Não exceder a quantidade de embarcações disponíveis de cada tipo

Grande parte da complexidade do problema real não foi considerada no desenvolvimento do presente trabalho. A Tabela I-2 ilustra essa situação:

Tabela I-2 – Escopo do presente trabalho

	Problema Clássico de Roteamento de Veículos	Escopo do Trabalho	Problema de Roteamento de Embarcações Offshore
Janela de Entrega	✗	✗	✓
Coleta e Entrega	✗	✗	✓
Multi-produtos	✗	✗	✓
Frota Heterogênea	✗	✓	✓
Limitação da Frota	✗	✓	✓

Ou seja, apenas a heterogeneidade e limitação da frota são consideradas na modelagem da solução. São as características mais importantes para aproximar o modelo do problema real devido às grandes diferenças de custos e capacidades entre as embarcações.

A questão dos múltiplos produtos se refere a uma extensão natural do modelo sendo a parte mais fácil de ser incorporada em novas versões do algoritmo desenvolvido.

Já incorporar a Janela de Entrega e a Coleta e Entrega (problemas correlatos) constitui um desafio adicional devido à criação de relações de precedência entre clientes (na forma de novas restrições de atendimento). Tais restrições podem acarretar maior dificuldade de se encontrar soluções viáveis e, conseqüentemente, impactar negativamente o desempenho computacional.

Adicionalmente, é necessário ressaltar que algumas operações características da atividade de apoio *offshore* não serão tratadas nesse trabalho, como o transbordo de carga entre plataformas, o manuseio de âncoras e espias e alívio de FPSOs, por exemplo. Para o detalhamento dessas operações sugere-se consultar [9] e trabalhos nessa linha, [15].

Outro ponto importante de se destacar é o fato de o escopo do trabalho estar restrito à Bacia de Campos embora seja possível expandi-lo às demais bacias em que a Petrobras opera no Brasil. Essa limitação não restringe o trabalho visto que as demais bacias replicam o modelo de operação existente na Bacia de Campos. Isso significa que ganhos obtidos em mudanças no modelo de negócios atual têm um potencial de serem ainda maiores ao serem replicados para outros locais.

I.7 Estrutura da Dissertação

Após essa breve introdução que teve como objetivo contextualizar o problema a ser tratado, o Capítulo 2 traz uma definição formal do problema de roteamento de veículos e as abordagens encontradas na literatura para tratá-lo, com especial ênfase na aplicação de heurísticas e metaheurísticas. O tratamento dado por Almeida em [16], solução adotada pela empresa e que discute o problema já adaptado para o negócio de apoio offshore será descrito.

O Capítulo 3 provê uma abordagem diferente de solução com o objetivo de avaliar possíveis melhorias ao algoritmo genético de [16] que possam futuramente ser incorporadas a ele através da adaptação de mecanismos presentes nos melhores trabalhos encontrados na literatura. Adicionalmente, nesse capítulo são descritas as instâncias utilizadas.

Os resultados do método proposto serão apresentados no Capítulo 4 juntamente com a discussão sobre os mesmos e a comparação com os melhores resultados conhecidos. Já no Capítulo 5 são apresentadas as considerações finais do presente trabalho e as sugestões de trabalhos futuros.

CAPÍTULO II REVISÃO DE LITERATURA

O problema descrito no capítulo anterior pode ser encaixado na literatura como sendo um problema de roteamento de veículos (VRP). O caso típico desse problema pode ser descrito como traçar rotas de custo mínimo a partir de um depósito para pontos espalhados geograficamente (cidades, lojas, escolas, clientes). Esse problema recebeu grande atenção nas últimas décadas e a evolução dos métodos de solução desse problema pode ser encontrada em Cordeau [17].

Diante disso, esse capítulo tem como objetivo descrever formalmente o VRP e indicar as principais abordagens de solução encontradas na literatura. Além disso, é necessário destacar que são exibidos com mais detalhe os métodos que servirão de base para a solução do HVRP (VRP com frota heterogênea limitada) através da metaheurística RTR (*Record-to-Record Travel Algorithm*). O RTR só será descrito no Capítulo III.

II.1 Descrição do Problema

O VRP (*Vehicle Routing Problem*) foi inicialmente descrito em [18]. Neste trabalho o VRP foi considerado como uma generalização do problema do caixeiro viajante (TSP). Assim como o TSP, o VRP é NP-difícil [19], e, por conseguinte, também o é sua versão capacitada (*Capacitated Vehicle Routing Problem – CVRP*). A complexidade desses problemas sugere a utilização de métodos heurísticos, algoritmos aproximativos ou elaborados métodos exatos de solução, como os citados na Seção II.3.

O problema do VRP simétrico é definido num grafo não-direcionado $G(V, E)$. O conjunto $V = \{0, \dots, n\}$ de vértices define a origem ou depósito (0) e os clientes (1 até n). Esses nós estão conectados através do conjunto E de arestas cujos custos são dados por c_{ij} (custo de viajar do ponto i até o ponto j).

Cada nó, à exceção do depósito, requer uma demanda q_i que necessita ser carregada ou descarregada do nó i a ser embarcada num dos m veículos de capacidade Q disponíveis para o transporte.

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (1)$$

de tal forma que $\sum_{i \in V} x_{ij} = 1, \forall j \in V - \{0\}$ (2)

$$\sum_{j \in V} x_{ij} = 1, \forall i \in V - \{0\} \quad (3)$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S), \forall S \subseteq V - \{0\}, S \neq \emptyset \quad (4)$$

$$x_{ij} \in \{0,1\}, \forall i, j \in V \quad (5)$$

A função objetivo (1) define que o custo da solução é o somatório do custo das arestas ativas (representadas por x_{ij}). As restrições (2) e (3) e (5) garantem que apenas duas arestas estão ativas para cada nó (uma para entrada e uma para saída). As restrições em (4) por sua vez visam a garantir a conectividade do grafo da solução e o limitante de capacidade do veículo.

O conjunto S é o conjunto das partes de $V - \{0\}$. Dessa forma, $r(S)$ é número mínimo de veículos necessários para atender uma determinada instância de S . Pode ser substituído pelo somatório da demanda da instância de S dividida pela capacidade Q dos veículos.

As restrições do tipo (4) podem ser reescritas, na sua formulação mais conhecida, como as Restrições Gerais de Eliminação de Subciclo:

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - r(S) \text{ para todo } S \subseteq V - \{0\}, S \neq \emptyset \quad (6)$$

Ambas as restrições, no entanto, tem cardinalidade exponencialmente crescente em relação a n . Isso pode gerar deficiências de desempenho ao se tentar resolver instâncias do VRP com grande número de vértices.

De todo modo, essa formulação é base para outras, como a de Laporte [21], por exemplo, e para o desenvolvimento de métodos exatos baseados em Programação Mista Inteira devido à sua capacidade gerar de bons limites inferiores.

É necessário ressaltar que os métodos exatos não fazem parte do escopo desse trabalho. Para uma extensa referência sobre o assunto, sugere-se consultar [17].

II.2 Extensões

O CVRP, conforme descrito na seção anterior, resolve uma grande gama de problemas. No entanto, há problemas de roteamento de veículos com características particulares não contempladas pela formulação apresentada na Seção II.1. Entre essas características podemos citar:

- Janelas de Entrega

As janelas de entrega impõem restrições adicionais ao VRP. Através delas, cada veículo deve chegar a cada um de seus clientes depois de uma data mais cedo e antes de uma data mais tarde. No caso de janelas de entrega flexíveis, paga-se uma penalidade caso não sejam respeitadas.

- Frota Heterogênea (limitada ou ilimitada)

Nem sempre a frota de transporte é homogênea, ou seja, com a mesma capacidade Q e os mesmos custos variáveis. No caso da frota heterogênea, veículos de portes diferentes e que podem transportar diferentes tipos de cargas estão disponíveis.

A frota pode ainda estar limitada a uma quantidade de veículos pré-definida, cabendo à abordagem de solução fazer o melhor uso possível dela. Outra extensão, geralmente relacionada a problemas de planejamento, não considera a limitação da quantidade de veículos. Essa abordagem é interessante quando o objetivo é definir uma política de atendimento, ou seja, com que tipo de veículo e com que frequência atender cada cliente. O resultado esperado é o dimensionamento da frota necessária que precisa ser contratada. Nesse caso o problema é dito *Fleet Size and Mix* VRP (FSMVRP).

O presente trabalho trata justamente do caso oposto, em que a frota é limitada. Nesse caso, o problema é abreviado como HVRP (*Heterogeneous Fleet* VRP).

- Entregas Parciais (*Split Deliveries*)

Cada ponto de demanda pode ser abastecido por mais de um veículo. Além disso, a demanda de cada ponto pode ser maior ou menor do que a capacidade de um veículo.

- Múltiplos Depósitos

Há vários depósitos de origem. Os veículos devem sair de um dos depósitos da rede e retornar a qualquer um deles.

- Coleta e Entrega

O problema de coleta e entrega pode prever relações de precedência entre clientes no qual a tarefa de coleta de carga em um cliente deve preceder a entrega do material no ponto de destino.

Esse problema, comparando à atividade offshore, equivale ao transbordo de cargas entre unidades marítimas, que compartilham os mesmos equipamentos, ou ainda a operação de retorno de carga para o porto.

II.3 Abordagens de Solução

A partir de sua definição, o VRP recebeu grande atenção devido às possibilidades de aplicação real. O algoritmo de Clarke e Wright [22] (também conhecido como *Savings*) e o método de Gillet e Miller [23] (também conhecido como *Sweep*) são alguns dos trabalhos mais conhecidos, desenvolvidos nas décadas de 60 e 70, respectivamente.

Acompanhando o aumento da capacidade computacional, abordagens mais modernas para a resolução do VRP foram propostas, notadamente nas últimas duas décadas. Dentre esses métodos, podem-se destacar os métodos exatos baseados em Programação Inteira e as Metaheurísticas (Algoritmos Genéticos e Busca Tabu, por exemplo).

Segundo [17], as abordagens de solução podem ser divididas em três principais grupos descritos a seguir: os métodos exatos, as heurísticas e as metaheurísticas.

II.3.1 Métodos Exatos

Para uma consulta mais detalhada sobre métodos exatos aplicados ao CVRP sugere-se [17], embora os próprios autores tenham atentado para o fato de que os melhores algoritmos exatos dificilmente conseguem resolver instâncias maiores do que 100 vértices. Por outro lado, esses métodos provêm tratamento matemático mais rígido e geralmente há a garantia da resolução do problema até o ótimo global.

Métodos exatos promissores desenvolvidos recentemente por Lysgaard et al. [24] e Fukusawa et al. [25] se valem de variações do *Branch and Bound* denominadas *Branch and Cut* e *Branch and Cut and Price*.

Através dos métodos baseados em *Branch and Cut*, resolve-se um problema relaxado (sem as restrições de capacidade, por exemplo, como em [21]). O limite inferior encontrado é continuamente melhorado através da inserção das restrições violadas (cortes) ao problema relaxado de modo a limitar o espaço de busca sem eliminar soluções inteiras válidas. A abordagem de [24], por exemplo, utiliza várias desigualdades válidas adicionais com o intuito de produzir limites inferiores de boa qualidade e acelerar o processo de busca.

Já os métodos baseados em *Branch and Cut and Price* como o de [25] se valem também da geração de colunas, no intuito de gerar limites inferiores ainda melhores que os de [24]. Como resultado, esse método obteve soluções ótimas para diversas instâncias de até 135 nós.

II.3.2 Heurísticas

As heurísticas podem ser divididas em 2 grupos: heurísticas de construção de rotas e heurísticas de melhoramento.

Esse tipo de abordagem tem como característica obter bons resultados com baixo esforço computacional. Além disso, são componentes importantes (como operadores) das metaheurísticas, seja na composição da solução inicial ou no posterior melhoramento de soluções promissoras, através da busca local.

II.3.2.1 Heurísticas de Construção de Rotas

As heurísticas de construção de rotas usualmente selecionam nós sequencialmente até que uma solução viável seja criada. Os nós são escolhidos geralmente segundo algum critério de minimização de custo desde que não violem a capacidade do veículos.

Dentro dessa categoria, o método mais conhecido (e um dos mais antigos) é o Método das Economias (*Savings*) de Clarke e Wright [22].

Outro método de destaque é a heurística de inserção de menor custo. Pela flexibilidade que o método propicia, foi adaptado para várias das extensões do VRP (para um exemplo consultar a heurística de inserção I1 de Solomon [26]) e funciona como método de inicialização de diversas metaheurísticas como em [27] e [28]. Seu funcionamento envolve dois grandes passos: escolher o cliente a ser inserido na rota e decidir a posição em que ele será inserido.

Entre os clientes ainda não pertencentes à rota, pode-se escolher o mais distante ou o mais urgente (em relação ao prazo de entrega) ou ainda o cliente pode ser escolhido aleatoriamente. Após esse passo, se decide em qual posição (entre os clientes já alocados na rota) é melhor inserir esse cliente. Para isso, faz-se um laço entre todas as posições do vetor que compõe a rota calculando a distância antes e depois da inserção. A alternativa que implica em menor custo de inserção é escolhida.

II.3.2.2 Heurísticas de Busca Local

Outros métodos de solução, alguns adaptados do TSP, são as heurísticas de busca local. Estas podem ser utilizadas dentro de uma rota ou em várias rotas simultaneamente. A idéia inicial destes métodos advém do mecanismo λ -opt de Lin originalmente descrito em [29]. Por esse mecanismo, λ arestas não adjacentes do grafo que compõe o problema (seja este VRP ou TSP) são removidas e reconectadas de todas as formas possíveis. Se alguma reconexão obtida for benéfica (podendo ser a primeira ou a melhor de todas), o grafo é modificado.

Muitas variantes desse mecanismo foram propostas. A mais bem sucedida delas segundo análise empírica de Johnson e McGeoch em [30] é a de Lin e Kernighan [31].

A seguir são descritos os principais métodos de busca local encontrados na literatura [32] e que são utilizados como operadores de muitas metaheurísticas como em [27] e [28]. Essas heurísticas foram utilizados no método de solução utilizado no presente trabalho.

- Movimento de um ponto

Esse método avalia a inserção do nó j antes ou depois do nó b . A Figura II-2 a seguir ilustra a inserção após b . Desse modo a aresta $b \rightarrow c$ passa a ser a seqüência $b \rightarrow j \rightarrow c$.

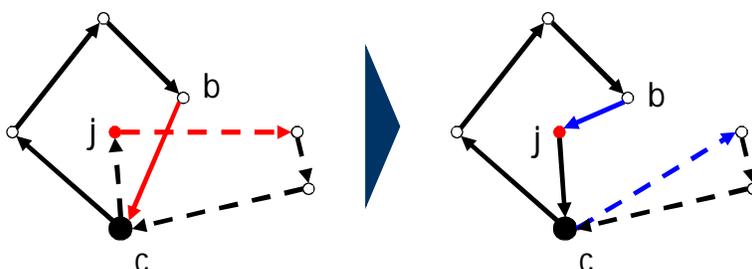


Figura II-2 –Exemplo do movimento de um ponto

- Movimento de dois pontos

Esse movimento avalia a troca de posição de dois nós (j e b) na solução atual conforme Figura II-3. Logo, se antes tínhamos $i \rightarrow j \rightarrow k$ e $a \rightarrow b \rightarrow c$, passamos a ter $i \rightarrow b \rightarrow k$ e $a \rightarrow j \rightarrow c$.

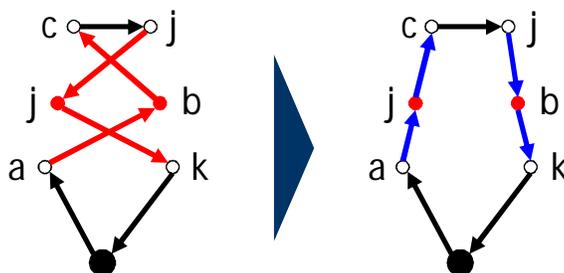


Figura II-3 – Exemplo do movimento de dois pontos

- 2-Opt

O movimento pode ser feito intra ou inter rotas. No caso intra-rota (Figura II-4), considera a destruição de duas arestas não adjacentes da solução atual ($a \rightarrow b$ e $c \rightarrow d$) e a reconexão dos nós de modo que $a \rightarrow c$ e $b \rightarrow d$. O trecho entre os nós b e c é revertido.

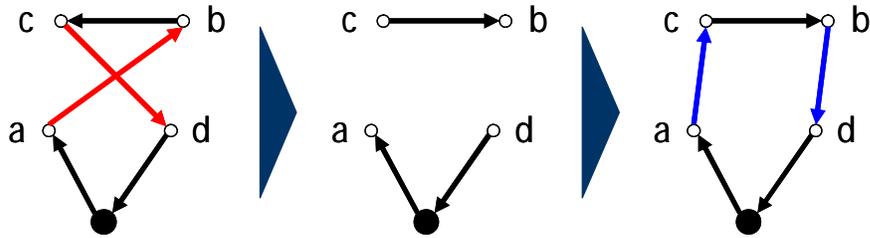


Figura II-4 – Exemplo do 2-Opt intra-rota

Já no 2-Opt inter rotas (Figura II-5), as arestas não adjacentes $a \rightarrow b$ e $c \rightarrow d$ são destruídas e os trechos finais das rotas originais (de b e d em diante) são trocados de modo que $a \rightarrow d$ e $c \rightarrow b$.

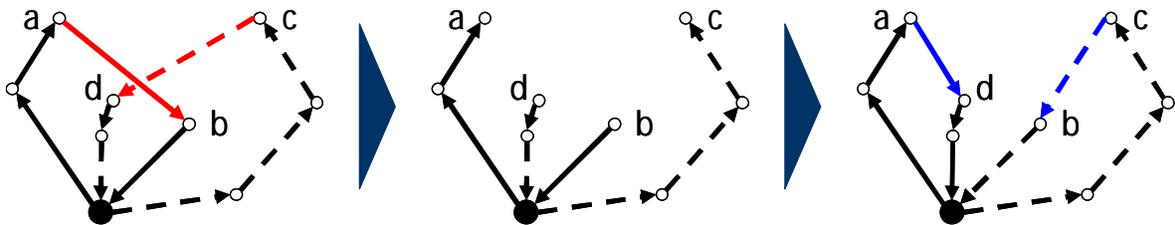


Figura II-5 – Exemplo do 2-Opt inter-rotas

- Or-Opt

Movimento similar ao de um ponto porém, em vez de apenas um nó, um trecho (*string*) de nós é considerado de uma só vez para ser reinserido. Dessa forma, no caso da reinserção do trecho $i \rightarrow j \rightarrow k$ na aresta $c \rightarrow d$ conforme Figura II-6, temos:

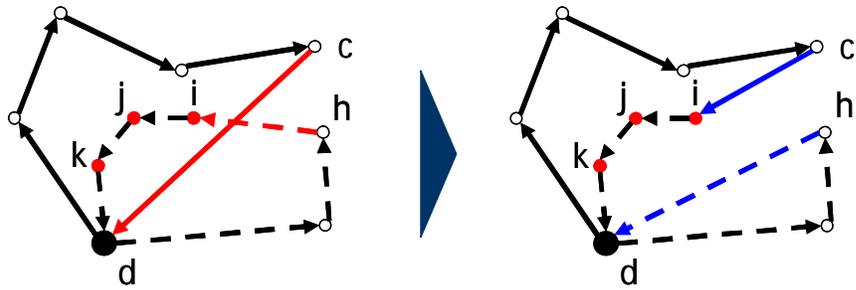


Figura II-6 – Exemplo do Or-Opt

- 3-Opt

O 3-Opt é similar ao 2-Opt embora nesse caso 3 arestas não adjacentes sejam destruídas em vez de 2. Nessa configuração, o número de recombinações possíveis é de 7. No caso da Figura II-7, por exemplo, o nó *a* foi reconectado ao nó *e* porém o mesmo poderia ter sido com *c*, *d* e *e*.

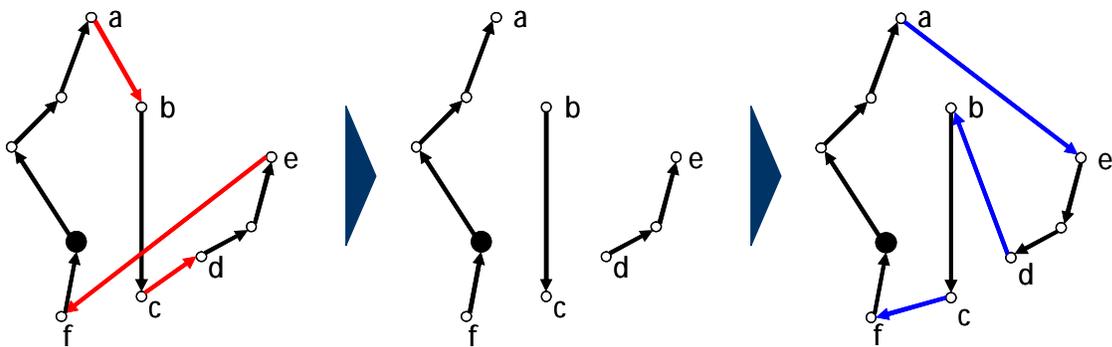


Figura II-7 – Exemplo do 3-Opt intra-roteira

- Movimento de três pontos

Nessa busca local, um nó *b* é trocado de posição com uma aresta $i \rightarrow j$ conforme o exemplo da Figura II-8.

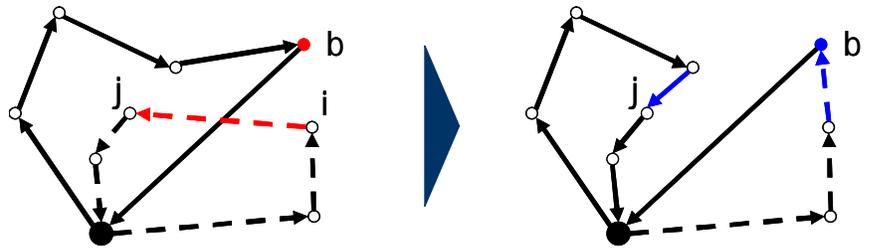


Figura II-8 – Exemplo do movimento de três pontos

- Troca Cruzada

O movimento prevê que as arestas $i_1 \rightarrow i_2$, $j_1 \rightarrow j_2$, $k_1 \rightarrow k_2$ e $l_1 \rightarrow l_2$ sejam substituídas pelas arestas $i_1 \rightarrow j_2$, $j_1 \rightarrow i_2$, $k_1 \rightarrow l_2$ e $l_1 \rightarrow k_2$ como ilustra a Figura II-9:

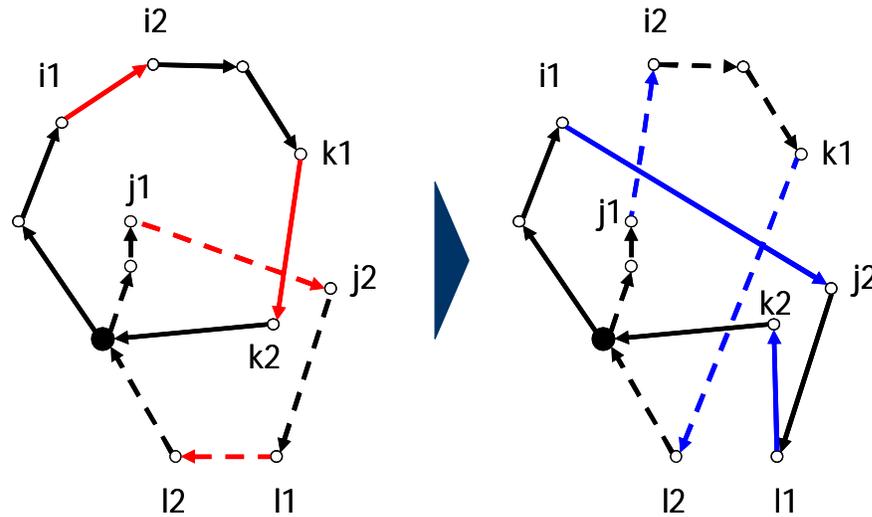


Figura II-9 – Exemplo da Troca Cruzada

II.3.3 Metaheurísticas

Segundo a definição de Glover em [33] e [34], as metaheurísticas se referem a uma estratégia mestre que guia e modifica outras heurísticas com o intuito de produzir soluções melhores do que seriam geradas utilizando-se apenas métodos de busca local. Comparativamente aos métodos exatos como o *Branch-and-Bound*, as metaheurísticas não garantem uma busca sistemática de todo o espaço de soluções.

Dessa forma, as metaheurísticas visam percorrer um exame amplo do espaço de busca, tentando evitar a convergência para ótimos locais ainda distantes do ótimo global.

Com vistas a esse fim, as metaheurísticas (segundo Crainic em [35]) podem ser divididas em dois grandes grupos: métodos baseados em população e métodos baseados em vizinhança.

Os métodos baseados em população, entre os quais se pode destacar Algoritmos Genéticos, Colônia de Formigas, Enxame de Partículas e Busca Dispersa (*Scatter Search*), utilizam simultaneamente um conjunto de soluções. Novas soluções são obtidas a partir da recombinação de elementos do conjunto que compõe a solução atual.

Já os métodos baseados em vizinhança definem mecanismos para guiar métodos de busca local que exploram o espaço de busca na vizinhança da solução atual. Dessa maneira, esses métodos costumam modificar a trajetória de apenas uma solução.

Nesse caso, por vizinho pode-se depreender todo o resultado de um possível movimento da busca local. Entre os principais métodos baseados em vizinhança pode-se destacar: Busca Tabu, Recozimento Simulado, *Guided Local Search*, *Variable Neighborhood Search* (VNS), *Greedy Randomized Adaptive Search Procedure* (GRASP) e *Iterated Local Search*.

No que tange ao HVRP (com frota heterogênea limitada), há poucos trabalhos desenvolvidos. Pode-se destacar dois métodos: o HCG (*Heuristic Column Generation*) de Taillard [38] (primeiro método desenvolvido) e o Algoritmo Genético de [16], adaptado da solução adotada pela empresa e descrita na Seção II.3.3.1 a seguir. No Capítulo III é dada ênfase ao método adotado no presente trabalho.

O método HCG propõe tratar o HVRP através da solução de sucessivos VRPs (um para cada tipo de veículo). A idéia é se valer do procedimento de memória adaptativa (AMP) e da Busca Tabu, métodos reconhecidamente eficientes para resolver o VRP.

É necessário ressaltar também os trabalhos [28] e [39] que endereçam o HVRP através de novas metaheurísticas, baseadas critérios de aceitação por limiar (*threshold accepting*). Nesses casos, a solução inicial é criada com a heurística de inserção de menor custo. Em seguida, é desencadeado um procedimento de busca iterativo em que uma nova solução é gerada aleatoriamente na vizinhança da solução atual (o movimento é escolhido estocasticamente) e valor limite (diferença entre as duas soluções) é calculado. Uma solução de qualidade inferior pode ser aceita desde que a variação desfavorável esteja

dentro de um limiar. A forma de atualização do limiar é a principal diferença entre os dois trabalhos.

Seu objetivo é, analogamente ao Recozimento Simulado, poder aceitar movimentos de busca local que piorem o resultado na expectativa de escapar de ótimos locais distantes do ótimo global.

II.3.3.1 Algoritmo Genético de Almeida [16]

Nessa seção, as principais características dos Algoritmos Genéticos são descritas. Dentre todos os métodos populacionais, o Algoritmo Genético foi escolhido por estar presente na ferramenta de roteirização desenvolvida na empresa para o problema de programação das embarcações *offshore*. As características aqui descritas estão em linha com essa ferramenta. Uma tela do ambiente de testes dessa ferramenta pode ser observada na Figura II-13.

O Algoritmo Genético (AG) é uma metaheurística inspirada na dinâmica genética de uma população. O AG evolui uma população de indivíduos codificados como cromossomos ao produzir novas gerações filhas de indivíduos através de um processo iterativo até que algum critério de convergência seja atingido.

A criação de uma nova geração envolve quatro grandes passos: avaliação, seleção, recombinação (*crossover*) e mutação conforme representado no esquema da Figura II-10.

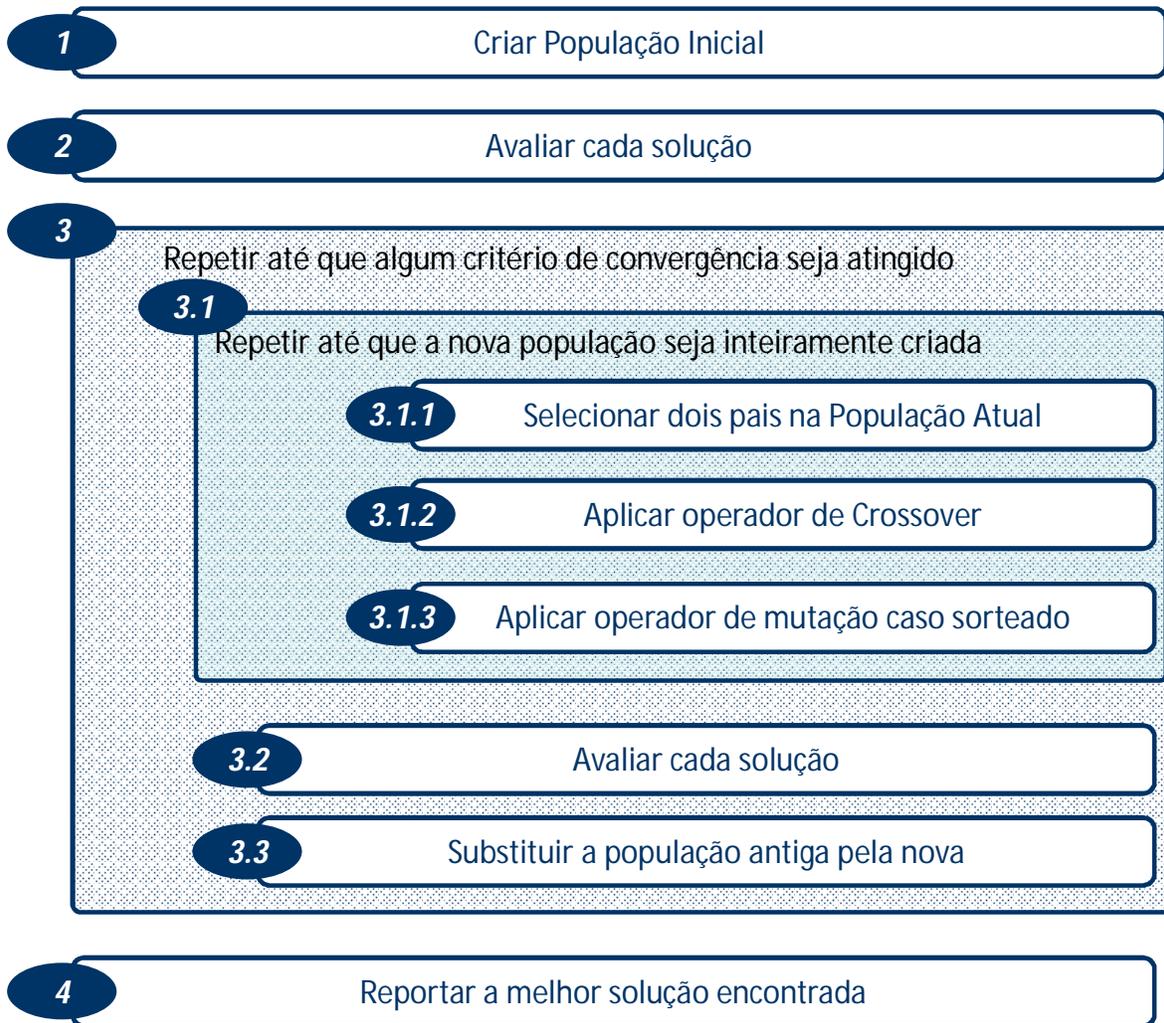


Figura II-10 – Esquema exibindo os passos do Algoritmo Genético

A característica mais marcante de [16] é a forma de representação da solução. Desenvolvida para dar flexibilidade ao algoritmo, trata também extensões do VRP como a de Coleta e Entrega. Para atingir esse objetivo foi utilizada uma representação indireta baseada em ordem composta por dois grupos de números inteiros (conforme exemplo da Figura II-11). O primeiro grupo contém a ordem de programação dos veículos e o segundo contém a ordem de programação dos clientes.

Através dessa codificação, o cromossomo é considerado como uma matriz de n linhas (número de clientes) por v colunas (número de veículos). Para o exemplo considerado:

		Veículos		
		1	2	3
Clientes	1	<u>1</u>	<u>5</u>	<u>2</u>
	3	<u>3</u>	<u>6</u>	<u>8</u>
	4	<u>4</u>	<u>9</u>	<u>10</u>
	7	<u>7</u>	1	5
	2	2	8	1
	9	9	4	3
	6	6	7	4
	10	10	2	7
	8	8	3	6
	5	5	10	9

Figura II-11 – Exemplo de codificação baseada em [16]

A decodificação dos indivíduos é feita percorrendo-se o segmento de veículos (linha 1) e para cada elemento desse segmento percorre-se a coluna (que contém os clientes) correspondente. Ao percorrer a coluna do veículo v , os pedidos dos clientes são alocados a sua rota.

Para o exemplo da Figura II-11, os elementos das rotas estão sublinhados. Os demais elementos serão eliminados na fase de decodificação por exceder a capacidade da embarcação ou não cumprir uma janela de tempo, por exemplo. Quando a capacidade do veículo se exaure, o mesmo volta à origem.

Desse modo, as rotas ficam:

- Rota do Veículo 1: 0→1→3→4→7→0
- Rota do Veículo 2: 0→5→6→9→0
- Rota do Veículo 3: 0→2→8→10→0

Através dessa representação, operadores de *crossover* e mutação tradicionais para a solução do TSP podem ser utilizados. Entre os operadores de *crossover* utilizados pode-se destacar o PMX (*Partially-Mapped Crossover*), *Order Crossover* e o *Cycle Crossover*. Já

entre os operadores de mutação destaca-se a *Inversion Mutation* e a *Insertion Mutation*. Para maiores informações sobre esses operadores sugere-se consultar [40].

Outra alternativa de codificação (similar a essa), obtida a partir de [37] e [41], codifica o cromossomo do VRP da mesma forma que o TSP. A fase de decodificação fica responsável por executar um algoritmo, denominado *Splitting*, que quebra o cromossomo nas respectivas viagens. Considerando o exemplo utilizado nessa seção, o cromossomo fica:

1	3	4	7	5	6	9	2	8	10
---	---	---	---	---	---	---	---	---	----

Figura II-12 – Exemplo de codificação baseada em [41]

Essa codificação também tem como vantagem a possibilidade de utilização de operadores genéticos importados do TSP sem a necessidade de adaptações. Por outro lado, o desempenho do AG se torna dependente do algoritmo *Splitting*, etapa mais custosa dentro da avaliação das soluções.

Para o CVRP, esse problema é contornado visto que a implementação do *Splitting* para esse caso é inspirado no algoritmo de caminho mais curto de Bellman-Ford [42]. Esse algoritmo, baseado em Programação Dinâmica, é extremamente eficiente propiciando grande velocidade ao *Splitting* também (ainda mais porque que o grafo considerado nesse método é acíclico). Além disso, o *Splitting* garante as melhores quebras possíveis para cromossomo a ele submetido, respeitando a restrição de capacidade dos veículos. Dessa forma, as soluções obtidas são sempre viáveis.

No entanto, as extensões desse algoritmo para casos mais complexos, como o de frota heterogênea limitada, o tornam muito mais custoso, especialmente se houver uma grande gama de tipos de veículos diferentes. Mesmo assim, Prins em [41] mostrou obter bons resultados para instâncias de 50 a 100 nós e utilizando 3 a 6 tipos de veículos.

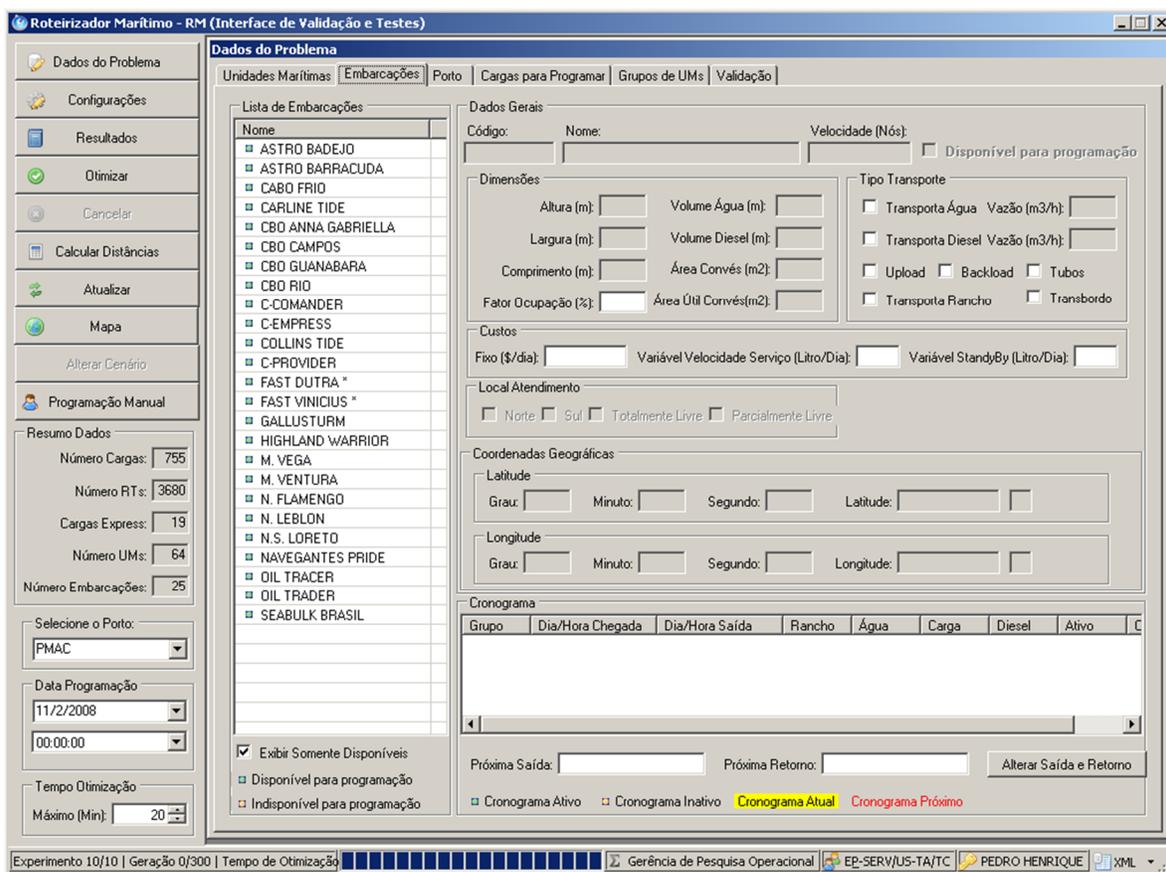


Figura II-13 – Interface de testes do Roteirizador Marítimo da Petrobras

CAPÍTULO III METODOLOGIA DA PESQUISA

Há dois trabalhos fundamentais em que o presente trabalho se baseia para resolver o Problema de Roteamento de Veículos de Frota Heterogênea Limitada (HVRP). A proposta de Li [43] e [27] atingiu excelentes resultados. Mais recentemente, Gröer [32] e [44] desenvolveu e disponibilizou uma biblioteca denominada VRPH para resolver o CVRP. Vários dos métodos e algoritmos dessa biblioteca foram derivados do trabalho de Li em [45] embora haja algumas diferenças marcantes entre os dois trabalhos.

Para viabilizar a utilização da biblioteca no HVRP é necessário adaptar o código fonte disponibilizado. A restrição de limitação de frota e sua heterogeneidade demandam adaptações no método de geração da solução inicial (que necessitou ser substituído) e nos métodos de busca local que precisam considerar os diferentes atributos (custos e capacidades) dos veículos.

As principais alterações, necessárias ao método de solução empregado, são discutidas nas próximas seções (III.2 sobre a VRPH e III.1 sobre o algoritmo de Li) nas quais se descreve a contribuição decisiva de ambos os trabalhos.

III.1 Algoritmo RTR de Li [27]

Li [27] adaptou o algoritmo RTR (*Record-to-Record Travel Algorithm*) de Dueck [46] para tratar o VRP e muitas de suas variações (Problema do Caixeiro Viajante Dependente do Tempo e VRP aberto, por exemplo). O trabalho [45], por exemplo, tratou problemas de roteamento de grandes proporções (até 1200 nós).

Na versão original do RTR (evidenciado na Tabela III-1), a nova configuração da solução é gerada através de uma pequena perturbação estocástica a partir da configuração anterior. O algoritmo de Li, por sua vez, é determinístico, gerando sempre a mesma solução para uma dada instância, seguindo uma variação determinística do RTR para o TSP (também descrita em [46]). Em ambos os casos, movimentos que pioram a solução atual também são aceitos na expectativa de escapar de ótimos locais ainda distantes do ótimo global.

Tabela III-1 – Esquema de funcionamento do algoritmo RTR [46]

Algoritmo Record-to-Record Travel para Minimização	
1	Geração da Solução Inicial
2	Atribua RECORD = Custo da Solução Inicial
3	Atribua DESVIO
4	Opt: escolha uma nova configuração através de uma pequena perturbação da configuração antiga
5	Atribua E = Custo da Nova Solução
6	se $E < RECORD + DESVIO$ então
7	Atribua configuração antiga = configuração nova
8	se $E < RECORD$ então
9	Atribua RECORD = E
10	se critério de convergência é atingido então finalize o algoritmo
11	GOTO Opt

Mesmo sem ter perturbações estocásticas (mecanismo importante de muitas metaheurísticas), o algoritmo consegue chegar a soluções muito boas, atingindo e, em alguns casos, superando os melhores resultados conhecidos das instâncias *benchmark*.

Em [43] e [27], o algoritmo adaptado para tratar o HVRP (denominado HRTR) é descrito e a principal diferença em relação a [45] é o método de geração da solução inicial visto que o algoritmo de Clarke-Wright (usado em [45]) desconsidera a limitação do número de rotas.

III.1.1 Estrutura do Algoritmo HRTR

O algoritmo está organizado em três grandes blocos conforme Figura III-1. Em dois deles, *Uphill* e *Downhill*, ocorre a aplicação dos métodos de busca local com a diferença de que na fase de *Uphill* são aceitos movimentos que pioram a solução atual. No bloco de *Downhill*, apenas movimentos que aproximem a solução de um mínimo local são aceitos. O último bloco diz respeito ao algoritmo de perturbação determinística da solução atual.

Há quatro parâmetros no algoritmo que determinam quantas iterações de cada laço são efetuadas: I , K , M e $NBListSize$. Embora boas soluções sejam obtidas para qualquer combinação desses parâmetros, para se obter os melhores resultados publicados pode ser necessário ajustá-los.

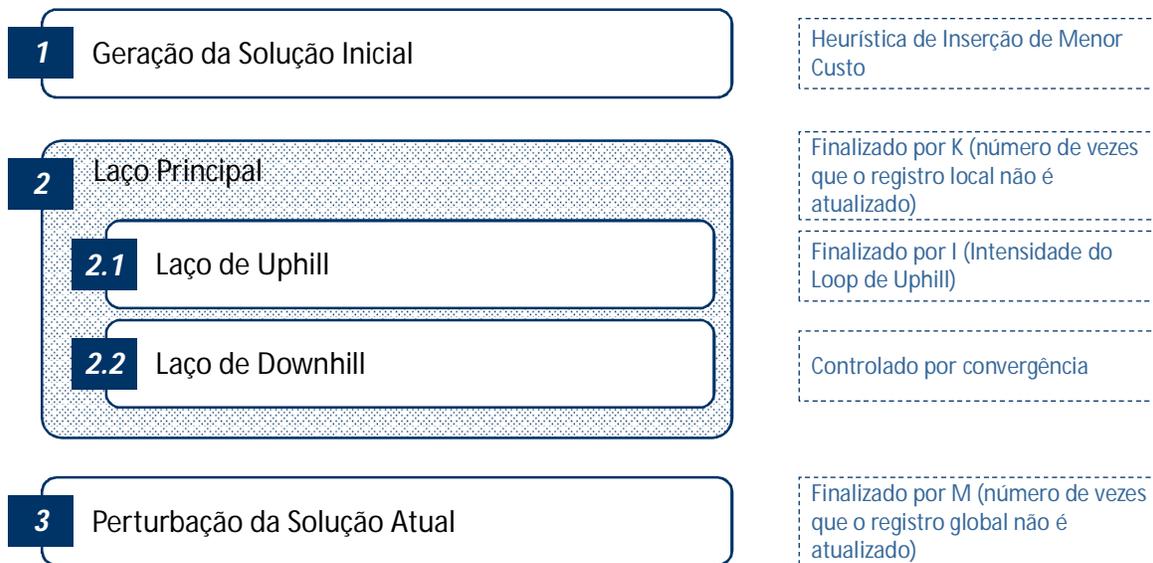


Figura III-1 – Estrutura resumida do HRTR

- I : número de iterações do tipo *Uphill* na solução atual. Valor padrão: $I=30$
- K : número máximo de iterações consecutivas permitidas sem que o registro local seja atualizado. Valor padrão: $K=5$
- M : número máximo de iterações consecutivas permitidas sem que o registro global seja atualizado. Valor padrão: $M=\max(\text{Número de Nós}/2, 30)$
- $NBListSize$: tamanho da lista de vizinhos (*neighbor list*) para cada nó. Valor padrão: $NBListSize=20$

A Tabela III-2 apresenta o pseudo-código do algoritmo HRTR. Nota-se que a maioria dos laços é finalizada por convergência e não por um número especificado de iterações. Isso justifica a inexistência de parâmetros a ajustar ao se processar instâncias de diferentes tamanhos.

Tabela III-2 – Estrutura do algoritmo de Li [43]

Algoritmo *Record-to-Record Travel* para o HVRP

```
1  Atribua  $I = 30$ ,  $K = 5$ ,  $M = \max(\text{Número de Nós}/2, 30)$  e  $NBListSize = 20$ 
2  Atribua Registro Global =  $\infty$ , Desvio Global =  $0.01 \times$  Registro Global
3  Geração da Solução Inicial com a Heurística de Inserção de Menor Custo
4  Atribua Registro = Custo da Solução Atual
5  Atribua Desvio =  $0.01 \times$  Registro
6  Atribua itr = 0
7  enquanto itr  $\leq M$  faça
8      Atribua contador = 0
9      enquanto contador  $\leq K$  faça
10     uphill:
11         para i = 1 até I faça
12             Movimento de 1 Ponto com RTR
13             Movimento de 2 Pontos com RTR
14             Movimento 2-opt com RTR
15             se nenhum movimento viável for feito então
16                 vá para downhill
17             fim se
18             se um novo Registro for produzido então
19                 atualize Registro e Desvio
20                 Atribua contador=0
21             fim se
22         fim para
23     downhill:
24         aplique Movimento de 1 Ponto (intra e entre rotas),
25         Movimento de 2 Pontos entre rotas, Movimento 2-opt (intra e entre rotas) e
26         Or-opt (intra e entre rotas). Apenas movimentos Downhill são aceitos.
27         se um novo Registro for produzido então
28             atualize Registro e Desvio
29             Atribua contador = 0
30         fim se
31         contador = contador + 1
32     fim enquanto
33     se solução atual  $\leq$  melhor solução global então
34         atualize Registro e Desvio
35         Atribua itr = 0
36     fim se
37     Perturbe a solução e vá para uphill
38     se solução perturbada  $\leq$  melhor solução global então
39         atualize Registro Global e Desvio Global
40         Atribua itr = 0
41     fim se
42     Aceite a solução perturbada se solução perturbada  $\leq$  Registro Global + Desvio Global
43     itr = itr + 1
44 fim enquanto
45 Reporte a Melhor Solução
```

III.1.2 Método de Geração da Solução Inicial

A geração da solução inicial é endereçada através da heurística de inserção de menor custo, adaptada de Solomon [47]. A Tabela III-3 evidencia a estrutura do algoritmo conforme desenvolvida por Li [27].

É importante salientar que esse procedimento é eficiente sendo adequado para tratar instâncias grandes do HVRP. Destaca-se também que o método não estava implementado na biblioteca VRPH.

Tabela III-3 – Estrutura do algoritmo de geração da solução inicial

Algoritmo Geração da Solução Inicial para HVRP

```
1  Atribua  $T$  = Número de Veículos,  $N$  = Número de Clientes
2  para  $t = 1$  até  $T$  faça
3    Atribua  $rota[t] = \emptyset$ 
4    Atribua  $rota.veiculo = Veiculo[t]$ 
5  fim para
6  Ordene clientes em ordem decrescente de demanda
7  para  $n = 1$  até  $N$  faça
8    Atribua menor custo de inserção =  $\infty$ 
9    para  $t = 1$  até  $T$  faça
10     Ache a posição do menor custo de inserção na  $rota[t]$ 
11     se menor custo de inserção menor na  $rota[t] <$  menor custo de inserção então
12       Atribua menor custo de inserção = menor custo de inserção na  $rota[t]$ 
13       Atribua melhor rota =  $rota\ t$ 
14     fim se
15   fim para
16   Insira  $n$  na melhor rota
17 fim para
```

É necessário destacar o procedimento de ordenação da linha 6 da Tabela III-3. Seu objetivo é maximizar as chances de se obter uma solução viável ao problema, dando prioridade aos pedidos maiores (mais difíceis de acomodar).

Adicionalmente, o procedimento da linha 10 (achar o menor custo dentro da $rota(t)$) não foi detalhado no pseudo-código. Em linhas gerais, ele consiste em um laço por todos os nós da rota, calculando os custos de inserção do nó n entre um cliente i da rota t e o cliente $i + 1$ (subseqüente da mesma rota). O custo da inserção é dado por:

$$Custo\ Inserção = (d(i, n) + d(n, i + 1) - d(i, i + 1)) \times Custo\ Variável\ do\ veículo\ t$$

Nesse caso, a inserção envolve quebrar a aresta $i \rightarrow i+1$. Por isso, o custo de inserção envolve acrescentar o novo caminho $i \rightarrow n \rightarrow i+1$ e desconsiderar a aresta $i \rightarrow i+1$. Isso deve ser ponderado pelo custo do veículo, necessário para o HVRP. A variável d denota a distância entre dois nós.

Caso não haja nenhum cliente na rota considerada, a operação a ser avaliada é a criação de uma rota que vá do depósito ao cliente n e retorne ao depósito. Ou seja:

Custo Inserção

$$= (d(0, n) + d(n, 0)) \times \text{Custo Variável do veículo } t \\ + \text{Custo Fixo do Veículo } t$$

Nesse caso, a criação da rota envolve um veículo que não fazia parte da solução. Logo, o custo fixo desse veículo deve ser incorporado também à função objetivo.

III.1.3 Método de Perturbação

Adicionalmente ao *Uphill* e conforme a estrutura do algoritmo descrita na Seção III.1.1, a fase de perturbação visa ao escape de um ótimo local de baixa qualidade. Uma característica marcante da perturbação proposta por Li é o fato de não ser estocástica.

O esquema de perturbação é diretamente relacionado à heurística de inserção de menor custo. Considerando-se cada nó i , seu antecessor $i-1$ e seu sucessor $i+1$ deve-se calcular o custo de inserção s do nó i na configuração atual. O nó i está na rota do veículo t .

$$s(i) = (d(i-1, i) + d(i, i+1) - d(i-1, i+1)) \times \text{Custo Variável do Veículo } t$$

A Figura III-2 ilustra essa operação:

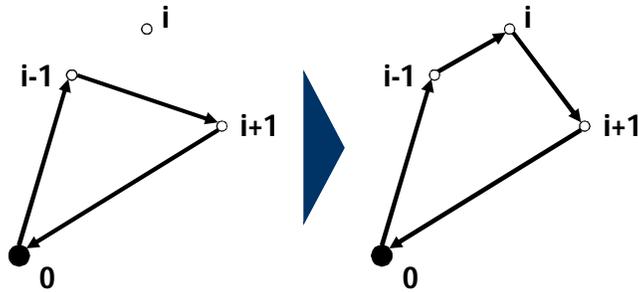


Figura III-2 – Esquema representando a inserção do cliente i na rota do veículo t

Adicionalmente, computa-se $r(i)$ a partir de $d(i)$ (demanda do nó i) e $s(i)$:

$$r(i) = d(i)/s(i)$$

A razão $r(i)$ definirá a prioridade da reinserção do nó i . Quanto maior a demanda em relação ao seu custo de inserção menor será a prioridade de movimentação desse cliente. De forma análoga, clientes com proporcionalmente pequena demanda e custo muito alto de entrega devem ser realocados.

Diante disso, o método segue a estrutura descrita na Tabela III-4:

Tabela III-4 – Estrutura do método de perturbação de Li

Método de Perturbação para HVRP

-
- 1 Atribua N = Número de Clientes
 - 2 **para** $i = 1$ até N **faça**
 - 3 Calcule $s(i)$
 - 4 Calcule $r(i)$
 - 5 **fim para**
 - 6 Ordene clientes em ordem crescente de $r(i)$
 - 7 Atribua $M = \min(20, N/10)$
 - 8 **para** $i = 1$ até M **faça**
 - 9 Aplique Inserção de Menor Custo ao nó i
 - 10 **fim para**
-

Aplicar a Inserção de Menor Custo é equivalente a executar o procedimento da Tabela III-2 das linhas 8 a 16.

III.1.4 Métodos de Busca Local

Li [43] aplica quatro métodos de busca local: Movimento de um ponto, Movimento de dois pontos, 2-Opt e Or-Opt.

O movimento Or-Opt só é utilizado na fase de *Downhill*. Provavelmente, o fato de o Or-opt ter maior complexidade e envolver maior número de nós não o faz ser apropriado a gerar pequenas perturbações, típicas da fase de *Uphill*.

O pseudo-código da Tabela III-5 detalha o Movimento de um ponto dentro da fase de Uphill referente à linha 12 da Tabela III-2.

Tabela III-5 – Estrutura do movimento de um ponto na Fase de Uphill [43]

Movimento de Um Ponto na fase de Uphill

1	Para cada nó i dentro da solução atual faça (loop I)
2	Para cada nó j dentro da lista de vizinhos de i faça (loop J)
3	Calcule a economia do Movimento de Um Ponto envolvendo i e j
4	se o movimento é viável então
5	se economia > 0 então faça o movimento e volte ao loop I
6	se economia $<$ Desvio e economia $>$ maior economia então
7	Armazene o movimento
8	fim se
9	Faça o movimento armazenado de maior economia
10	fim loop J
11	fim loop I

A lógica de funcionamento dos outros movimentos na fase de *Uphill* é análoga à descrita na Tabela III-5. De forma geral, todo movimento do tipo *Downhill* (economia maior do que zero) é aceita. Movimentos que piorem a solução também são aceitos desde que não sejam maiores do que o Desvio (1% do Registro) e caso nenhum movimento *Downhill* seja encontrado. Dentre todos os movimentos que pioram a solução dentro do *loop I*, apenas o melhor (que menos piora a solução atual) é efetuado.

III.2 Biblioteca VRPH

A biblioteca VRPH é um conjunto de métodos e algoritmos desenvolvidos em C++ por Chris Gröer [32] para resolver o CVRP, disponível livremente na internet através da iniciativa COIN-OR (*Computational Infrastructure for Operations Research*) [48]. Possui também conexão com a biblioteca gráfica PLplot [49] para a visualização das soluções geradas.

O foco da biblioteca é a disponibilização de sete métodos de busca local de forma orientada a objetos numa estrutura de dados eficiente. A orientação a objetos permite um maior nível de abstração na construção de metaheurísticas.

Adicionalmente, são fornecidos dois algoritmos para geração de solução inicial e mais duas metaheurísticas, chamadas *solvers*, baseadas nos sete métodos de busca local.

III.2.1 Estrutura

A VRPH está estruturada conforme diagrama de classes da Figura III-3. Muitas das funcionalidades da biblioteca foram suprimidas em benefício da brevidade. As classes em cinza foram inseridas para adaptar a biblioteca para tratar o HVRP no presente trabalho.

O módulo principal é a classe VRP. Na execução de qualquer algoritmo é necessário criar uma instância dessa classe. Nela, localiza-se a codificação da solução atual bem como todas as informações relativas a ela (*route_num*, vetor de rotas entre outros). Além disso, os cadastros de todas as informações necessárias ao problema são carregados nessa instância.

Adicionalmente, os *solvers* (Recozimento Simulado e RTR) e os algoritmos de perturbação da solução atual também são procedimentos dessa classe.

As metaheurísticas da VRP chamam durante a execução “classes amigas” referentes aos métodos de busca local, que são responsáveis pela investigação da vizinhança da solução atual. Cada classe de busca local possui métodos de busca, avaliação e movimento.

A busca percorre todos os nós vizinhos de determinado nó da solução atual. Caso a operação de busca local seja avaliada como interessante, esse movimento é confirmado e a solução atual é modificada.

Os métodos de busca local exibidos (Movimento de um ponto, Movimento de dois pontos, 2-Opt e Or-Opt) desdobram suas ações em etapas mais simples, compostas dos movimentos elementares *Presert* e *Postsert*, responsáveis pela inserção antes ou depois de um nó especificado.

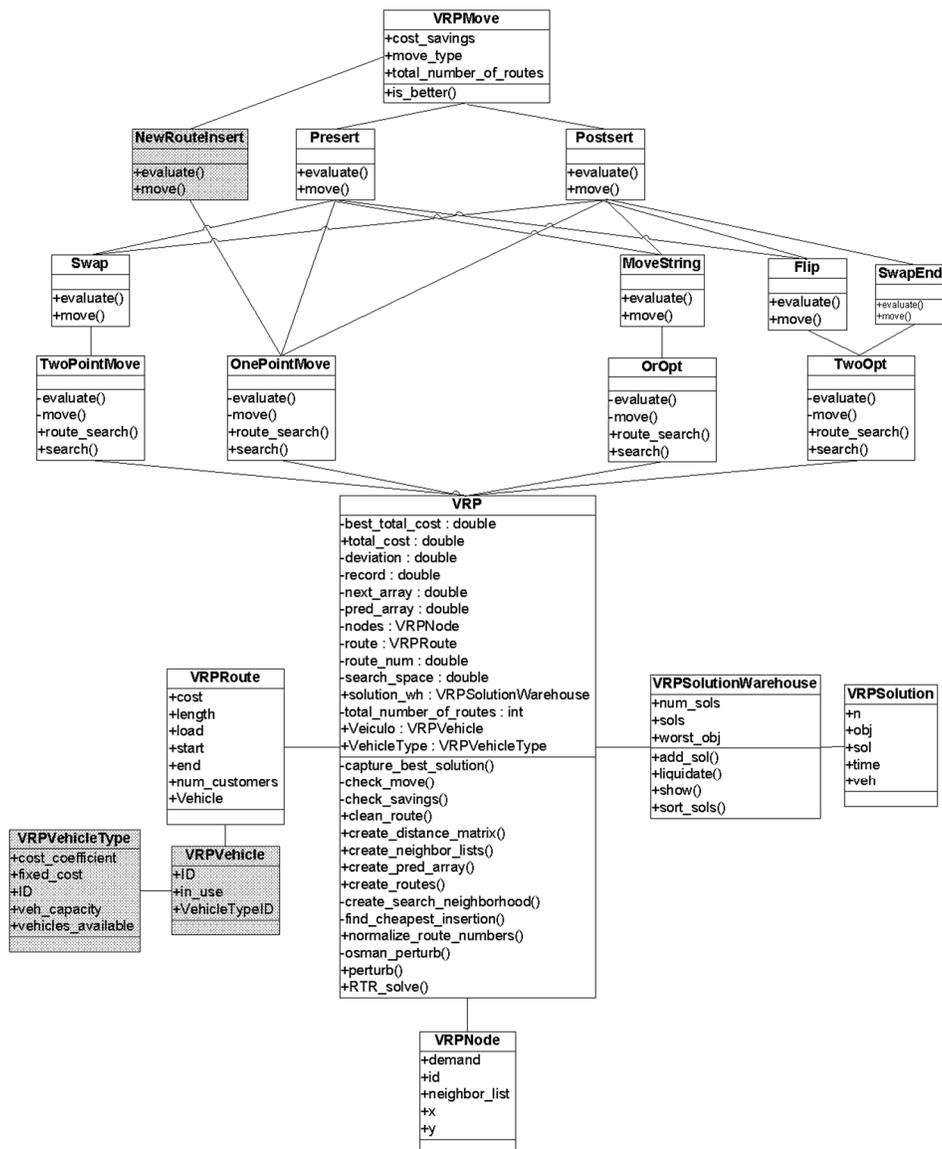


Figura III-3 – Diagrama de Classes da VRPH elaborado pelo autor e modificações propostas no presente trabalho

III.2.2 Características

Algumas das características mais importantes da VRPH são descritas ao longo dessa seção.

III.2.2.1 Codificação da Solução Atual

Considerando um problema de n nós, a codificação de uma solução viável em qualquer estágio do algoritmo de solução na VRPH se dá através de uma lista duplamente ligada contendo dois vetores de tamanho $n+1$: *next_array* e *pred_array*.

O exemplo a seguir possui dez clientes (numerados de um a dez) mais o depósito (de onde saem todas as rotas). As rotas em dado momento são dadas por:

Tabela III-6 – Exemplos de rotas

Rota	Trajeto
1	0-2-5-8-0
2	0-1-3-4-0
3	0-6-7-9-10-0

A codificação dessa solução fica:

Tabela III-7 – Codificação do exemplo

i	0	1	2	3	4	5	6	7	8	9	10
<i>next_array</i>	-2	3	5	4	-6	8	7	9	-1	10	0
<i>pred_array</i>	-10	-8	0	1	3	2	-4	6	5	7	9

A linha 1 contém o índice referente ao depósito e a cada um dos clientes. Pode-se observar alguns números negativos em ambos os vetores, representando o início (em *next_array*) e final (em *pred_array*) de uma rota. Esse recurso foi incorporado à representação da solução com o objetivo de deixá-la mais compacta, similarmente à representação de [50].

Assim, para recompor a rota 1, por exemplo, devemos partir da posição 0 do *next_array*, cujo valor é igual a -2. Esse valor remeterá, portanto, à segunda posição desse vetor e ainda indica o início de uma rota. O valor dessa posição é 5 (segundo cliente da rota 1). Indo para *next_array[5]*, há o valor 8 que, por sua vez, remete à posição 1 e com o sinal de negativo na frente. Isso indica que a rota terminou no cliente 8. O raciocínio é análogo para a decodificação do *pred_array*.

É necessário ressaltar que o *next_array* do último cliente da última rota direcionará sempre para o depósito e o *pred_array* do depósito remeterá a esse cliente. Seu valor será acrescido de um sinal negativo para sinalizar que este é o último cliente de uma rota.

Essa codificação permite manipulações rápidas no conteúdo desses vetores. Segundo [32], essa abordagem é ainda potencialmente mais adequada para esse fim do que à utilização de uma tradicional lista duplamente ligada baseada em ponteiros.

Dessa forma, operações na solução do tipo inserção se tornam extremamente eficientes e a modificação da solução ocorre em tempo constante. Na Seção III.2.1, pode-se verificar que esse tipo de movimento é a base para a construção de todos os métodos de busca local.

No exemplo a seguir, supomos que se deseja movimentar o cliente 9 da rota 3 para rota 1 entre os clientes 5 e 8. Ou seja:

Tabela III-8 – Novas rotas após inserção

Rota	Trajeto
1	0-2-5-9-8-0
2	0-1-3-4-0
3	0-6-7-10-0

A codificação fica:

Tabela III-9 – Codificação das novas rotas

<i>i</i>	0	1	2	3	4	5	6	7	8	9	10
<i>next_array</i>	-2	3	5	4	-6	9	7	10	-1	8	0
<i>pred_array</i>	-10	-8	0	1	3	2	-4	6	9	5	7

Pode-se notar que apenas 6 modificações foram necessárias para efetuar a reformulação da codificação. Foram alterados o *next_array[5]* que passou a ser igual a 9, o *next_array[9]* que passou a ser 8 e o *next_array[7]* que passou a ser igual a 10. No vetor *pred_array* houve outras três alterações. O cliente 8 passou a ser precedido pelo 9, que passa a ser precedido pelo 5. Já o cliente 10 passa a ser precedido pelo cliente 7.

O número de modificações é sempre pequeno e independe do tamanho do problema ou do número de clientes das rotas envolvidas no movimento. Sem uma lista ligada, essa operação envolveria “empurrar” pelo menos todos os nós posteriores à inserção na rota considerada. Laço semelhante seria necessário para reajustar as posições dos nós remanescentes na rota de onde o nó movimentado foi retirado. Kytöjoki [50] pontua que acrescentar um cliente em uma estrutura tabular tradicional requer mudanças de índices dos clientes posteriores, envolvendo cálculos de complexidade $O(n)$, onde n é o número de clientes.

Para completar a representação da solução, há ainda mais dois objetos. O vetor *route_num* associa clientes às suas rotas. Logo, para o exemplo considerado *route_num[2]* é igual a 1. Além disso, é mantido um vetor de objetos da classe *VRPRoute* onde são armazenados os custos, tamanho, número de clientes, nó inicial e final de cada rota.

III.2.2.2 Métodos de Geração da Solução Inicial

Há dois algoritmos de Geração de Solução Inicial:

- Método das Economias de Clarke-Wright [22]
- Algoritmo de Varredura (*Sweep*) [23]

No algoritmo de Clarke-Wright é possível ajustar parâmetros que geram soluções de diferentes qualidades através da variação do peso atribuído à economia obtida ao se concatenar rotas.

Nenhum dos dois é utilizado para a solução das instâncias apresentadas nesse trabalho.

III.2.2.3 Método de Criação da Vizinhança

A distância é o critério utilizado para definir quais são os pontos que entram no vetor *NBList* de cada nó. É necessário ressaltar que a lista de vizinhos é preenchida em ordem de distância. Dessa forma, o *NBList[0]* do nó *i* é o vizinho mais próximo do nó *i*.

III.2.2.4 Métodos de Busca Local

O principal mérito da biblioteca é disponibilizar sete métodos de busca local de acordo com a estrutura de dados eficiente mencionada em III.2.1. Os métodos disponíveis são:

- Movimento de Um Ponto
- Movimento de Dois Pontos
- 2-Opt
- Or-Opt
- Troca cruzada
- Movimento de Três Pontos
- 3-Opt

Cada um desses métodos pode ser utilizado dentro ou entre rotas, considerando sempre a interação de um cliente com cada um de seus vizinhos, um de cada vez. Esses métodos foram apresentados na Seção II.3.2.2

Além disso, os métodos podem ser chamados de forma determinística ou aleatória. A diferença reside na ordem em que se chama cada um dos vizinhos para o método de avaliação. No presente trabalho apenas a versão determinística (seqüencial) de chamada dos vizinhos foi utilizada.

III.2.3 Outras Funcionalidades

A VRPH dispõe de um método para codificar rotas e soluções numa tabela *Hash*. Essa funcionalidade é particularmente útil para definir de forma rápida a unicidade de uma rota ou solução. Essa funcionalidade é particularmente interessante para desenvolvimento de algoritmos baseados em Busca Tabu. Embora a metaheurística não esteja implementada, é possível restringir os métodos de busca local para que não efetuem movimentos pertencentes a uma Lista Tabu.

Outra funcionalidade interessante é a possibilidade de integração da biblioteca com o SYMPHONY [51], software gratuito para a solução de problemas de programação mista-inteira. Como tanto o SYMPHONY quanto a VRPH lêem arquivos em formato TSPLIB (padrão para o TSP e VRP), a melhor solução obtida pela VRPH pode ser transmitida para o *solver* de problemas de roteamento de veículos do SYMPHONY e servir de ponto de partida para uma solução ainda melhor.

III.2.4 Modificações Necessárias

Houve a necessidade de criar mais um movimento elementar, o *NewRouteInsert*. Seu objetivo é tratar de forma mais rígida a criação de rotas, ou seja, a inserção de nós em rotas inexistentes. Essa alteração foi necessária para lidar de forma rigorosa com a limitação da frota, problema inexistente no CVRP.

Mais duas classes foram acrescentadas para lidar com a heterogeneidade da frota, *VRPVehicle* e *VRPVehicleType*. Dessa forma, uma instância com t veículos disponíveis terá um vetor de tamanho t de objetos do tipo *VRPVehicle*, cada elemento do vetor contendo capacidade e custos (fixo e variável) de cada um dos veículos.

É necessário ressaltar que cada veículo pode ter seu próprio tipo. Isso significa que todos os veículos podem ser diferentes uns dos outros, sem prejuízo em esforço computacional ou complexidade de geração da solução.

Adicionalmente, o método de geração da solução inicial utilizado em Li [27] (descrito na Seção III.1.2) foi implementado devido à inaplicabilidade dos métodos existentes na biblioteca.

Adaptações foram feitas também nas buscas locais para que as mesmas passassem a considerar as diferentes capacidades e custos dos diversos veículos.

III.3 Instâncias Teste

Com vistas a avaliar a efetividade do HRTR implementado dentro da VRPH, diferentes instâncias *benchmark* (disponíveis na literatura) são submetidas ao algoritmo.

Por fim, na Seção III.3.2, uma instância inspirada no roteamento de embarcações de apoio *offshore* é apresentada para ser submetida ao HRTR.

III.3.1 Instâncias *Benchmark*

O trabalho de Taillard [38] adapta as 8 maiores instâncias de Golden [52] (construídas para avaliar o FSMVRP) para tratar o HVRP, constituindo o principal meio de comparação de diferentes algoritmos encontrado na literatura.

Adicionalmente, Li [27] propõe novas instâncias, de maior porte (de 200 a 360 clientes) para avaliar o comportamento do HRTR sujeito a problemas de grandes proporções.

III.3.1.1 Instâncias de Taillard [38]

As instâncias de Taillard variam de 50 a 100 nós, sendo, portanto, instâncias de pequeno porte.

Sua principal característica é o baixo grau de flexibilidade, considerando a demanda de todos os nós e capacidade de todos os veículos disponíveis, para efetuar modificações nas rotas e mesmo assim manter a solução viável como pode ser observado na Tabela III-10.

Tabela III-10 – Utilização da frota nas instâncias de Taillard

Instancia	<i>Demanda Total</i>	<i>Capacidade Total</i>	<i>% Utilização da Frota</i>
13	973	1020	95.4%
14	973	1100	88.5%
15	777	820	94.8%
16	777	820	94.8%
17	1364	1430	95.4%
18	1364	1430	95.4%
19	1458	1900	76.7%
20	1458	1520	95.9%

Como se pode observar, a flexibilidade de realocação de um pedido de um veículo a outro é muito pequena visto que, em média, todos estarão muito ocupados. Além disso, a expectativa é de que todos os veículos disponíveis sejam utilizados.

Na Tabela III-11 são apresentadas as características dos veículos considerados nas instâncias:

Tabela III-11 – Dados de veículos nas instâncias de Taillard

Instancia	n	Q_A	f_A	α_A	η_A	Q_B	f_B	α_B	η_B	Q_C	f_C	α_C	η_C	Q_D	f_D	α_D	η_D	Q_E	f_E	α_E	η_E	Q_F	f_F	α_F	η_F
13	50	20	20	1.0	4	30	35	1.1	2	40	50	1.2	4	70	120	1.7	4	120	225	2.5	2	200	400	3.2	1
14	50	120	100	1.0	4	160	1500	1.1	2	300	3500	1.4	1												
15	50	50	100	1.0	4	100	250	1.6	3	160	450	2.0	2												
16	50	40	100	1.0	2	80	200	1.6	4	140	400	2.1	3												
17	75	50	25	1.0	4	120	80	1.2	4	200	150	1.5	2	350	320	1.8	1								
18	75	20	10	1.0	4	50	35	1.3	4	100	100	1.9	2	150	180	2.4	2	250	400	2.9	1	400	800	3.2	1
19	100	100	500	1.0	4	200	1200	1.4	3	300	2100	1.7	3												
20	100	60	100	1.0	6	140	300	1.7	4	200	500	2.0	3												

n : número de clientes; Q_t : capacidade do veículo do tipo t ($t = A, B, C, D, E, F$); f_t : custo fixo do tipo de veículo t
 α_t : custo variável por unidade de distância do tipo de veículo t ; η_t : número de veículos do tipo t disponíveis

O número de tipos de veículos varia de 3 a 6 e o número total de veículos de 13 a 17. A Figura III-4 apresenta como os nós estão dispostos fisicamente nas instâncias T_13 e T_20:

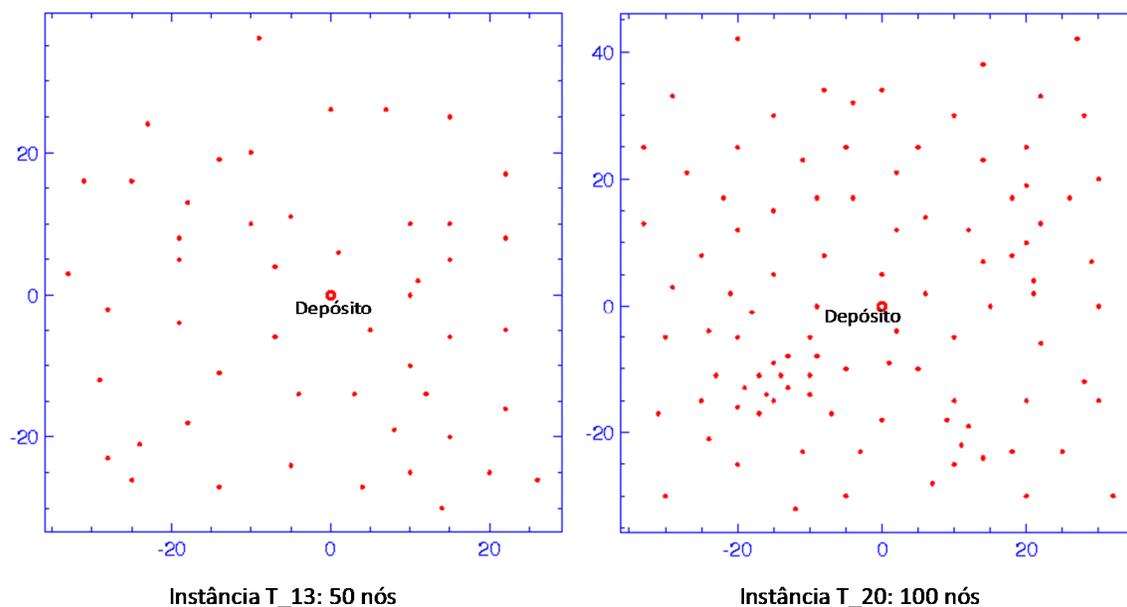


Figura III-4 – Disposição física dos nós das instâncias T_13 e T_20

III.3.1.2 Instâncias de Li [27]

As instâncias propostas variam de 200 a 360 clientes. Embora a geração dos clientes de forma aleatória represente melhor as situações encontradas na prática, o método proposto tem a seu favor permitir avaliar visualmente a qualidade da solução gerada.

A Figura III-5 apresenta a menor (200 clientes) e a maior (360 clientes) instância proposta para o HVRP.

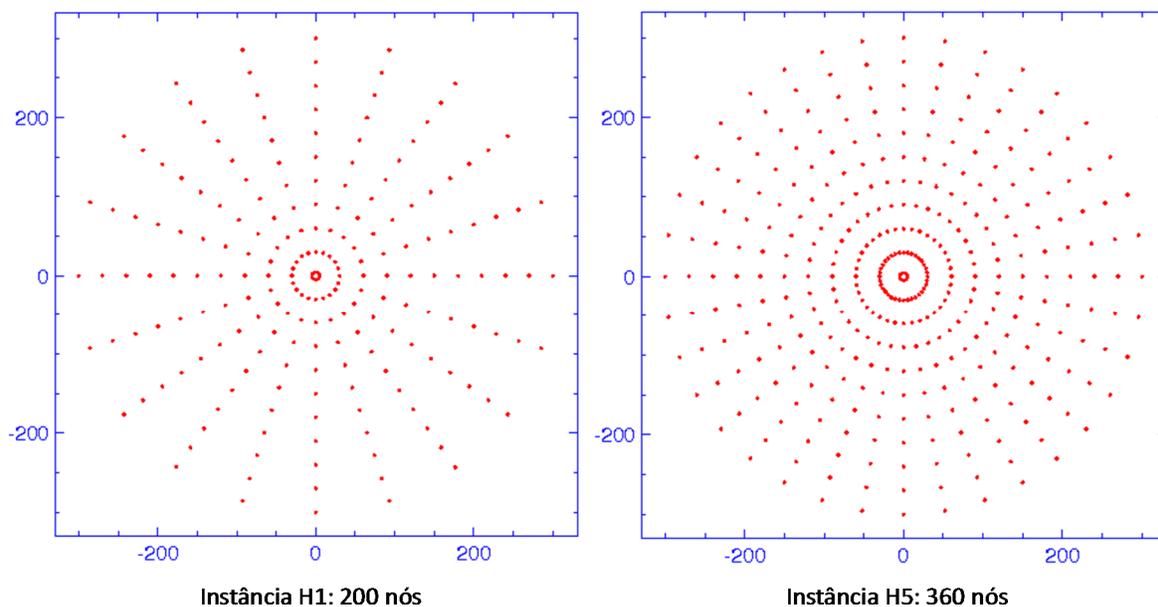


Figura III-5 – Disposição física dos nós das instâncias H1 e H5 de Li

A baixa flexibilidade (veículos cheios) também está presente nas instâncias de Li conforme a Tabela III-12:

Tabela III-12 – Utilização da frota nas instâncias de Li

Instancia	<i>Demanda</i> <i>Total</i>	<i>Capacidade</i> <i>Total</i>	<i>% Utilização</i> <i>da Frota</i>
H1	4000	4300	93.0%
H2	4800	5000	96.0%
H3	5600	6000	93.3%
H4	6400	6800	94.1%
H5	7200	7800	92.3%

Os dados dos veículos referentes às instâncias H1 a H5 são apresentados na Tabela III-13:

Tabela III-13 – Dados de veículos nas instâncias de Li

Instância	n	Q_A	f_A	α_A	η_A	Q_B	f_B	α_B	η_B	Q_C	f_C	α_C	η_C	Q_D	f_D	α_D	η_D	Q_E	f_E	α_E	η_E	Q_F	f_F	α_F	η_F
H1	200	50	20	1.0	8	100	35	1.1	6	200	50	1.2	4	500	120	1.7	3	1000	225	2.5	1				
H2	240	50	100	1.0	10	100	1500	1.1	5	200	3500	1.2	5	500	120	1.7	4	1000	225	2.5	1				
H3	280	50	100	1.0	10	100	250	1.1	5	200	50	1.2	5	500	120	1.7	4	1000	225	2.5	2				
H4	320	50	100	1.0	10	100	200	1.1	8	200	400	1.2	5	500	120	1.7	2	1000	225	2.5	2	1500	250	3.0	1
H5	360	50	25	1.0	10	100	80	1.2	8	200	150	1.5	5	500	320	1.8	1	1500	225	2.5	2	2000	250	3.0	1

n : número de clientes; Q_t : capacidade do veículo do tipo t ($t = A, B, C, D, E, F$); f_t : custo fixo do tipo de veículo t ; α_t : custo variável por unidade de distância do tipo de veículo t ; η_t : número de veículos do tipo t disponíveis

O método proposto para a geração das instâncias segue a estrutura descrita na Tabela III-14:

Tabela III-14 – Pseudo-código do gerador de instâncias [27]

Geração de Instância para o VRP

1	Atribua $\omega = 0$
2	Atribua $x(\omega) = 0, y(\omega) = 0$ e $q(\omega) = 0$
3	para $k=1$ até B faça
4	$\gamma = 30k$
5	para $i=1$ até A faça
6	Atribua $\omega = \omega + 1$
7	Atribua $x(\omega) = \gamma \cos[2(i-1)\pi / A]$
8	Atribua $y(\omega) = \gamma \sin[2(i-1)\pi / A]$
9	se $i \bmod 4 = 2$ ou 3 então
10	Atribua $q(\omega) = 30$
11	caso contrário
12	Atribua $q(\omega) = 10$
13	fim se
14	fim para

O conjunto $x(i), y(i)$ representa as coordenadas do ponto i . A demanda de i , $q(i)$, pode ser igual a 10 ou 30. O total de clientes n é dado por $n = A \times B$. A e B são parâmetros que determinam a forma do problema. Escolhidos arbitrariamente, A e B têm os seguintes valores para a geração das instâncias:

Tabela III-15 – Parâmetros das instâncias de Li

Instância	A	B	Número de Clientes
H1	20	10	200
H2	40	6	240
H3	28	10	280
H4	40	8	320
H5	36	10	360

III.3.2 Instância Real

O caso real submetido ao algoritmo possui estrutura bastante diferente das instâncias *benchmark*. Proporcionalmente, há um maior número de veículos disponíveis, em parte devido às simplificações necessárias para adaptar o problema real a uma instância que pudesse ser solucionada pelo algoritmo. Ou seja, vários dos pedidos presentes na base de dados da Petrobras foram excluídos restando apenas os pedidos para entrega de carga geral (218 pedidos de um total de 715). Adicionalmente, as datas de entrega dos pedidos foram ignoradas.

Mesmo diante dessas simplificações a instância proporciona complexidade adicional devido ao elevado número de veículos e grande distância entre a origem (Porto de Macaé) e a região de atendimento composta pelas plataformas e sondas conforme a Figura III-6.

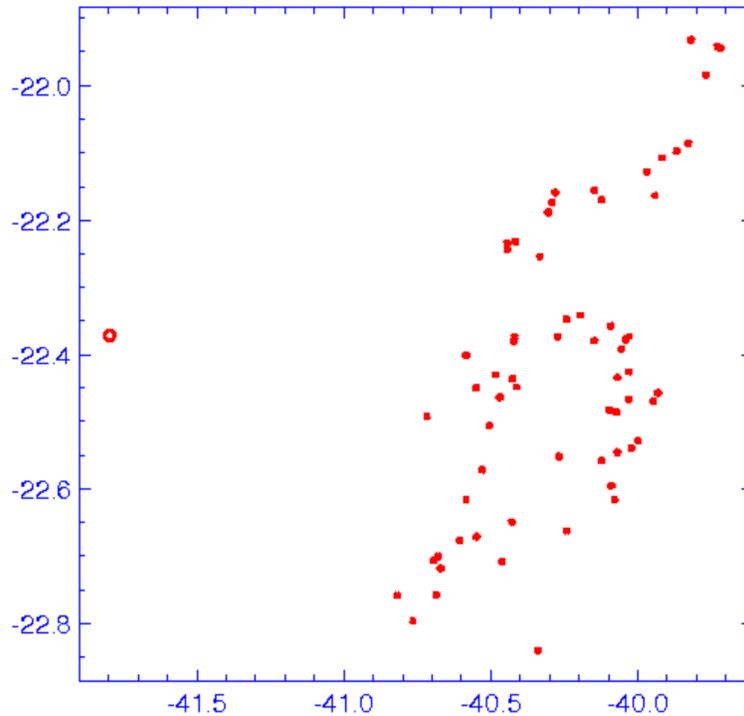


Figura III-6 – Disposição geográfica dos pontos de atendimento

É necessário ressaltar que embora haja 218 pedidos de entrega, há 65 unidades a serem atendidas. Ou seja, há em média 3,3 pedidos por unidade. Os pedidos da mesma unidade podem ser atendidos por diferentes embarcações mesmo tendo as mesmas coordenadas geográficas.

Para realizar esses atendimentos são disponibilizados 21 veículos de 4 tipos diferentes conforme a Tabela III-16. Esses veículos, mesmo sendo do mesmo tipo, podem ter capacidades de carga, em área de convés, e custos diferentes. Isso se deve ao fato de o tipo ser definido pelo TPB (Tonelada de Porte Bruto) e não pelo tamanho da área de convés.

É necessário ressaltar que apenas os custos variáveis das embarcações foram considerados porque a frota disponível já está contratada pela empresa. Dessa forma, o custo fixo associado a cada embarcação será pago pela empresa mesmo que parte da frota não seja utilizada.

Tabela III-16 – Embarcações disponíveis para o caso real

Veículo	Tipo	Área Convés	R\$/km
1	UT 4.000	180	13.50
2	UT 4.000	180	13.50
3	PSV 1.500	280	18.90
4	PSV 3.000	380	27.00
5	PSV 1.500	386	31.59
6	PSV 1.500	451	20.79
7	PSV 3.000	540	45.90
8	PSV 3.000	540	49.41
9	PSV 3.000	540	45.90
10	PSV 3.000	546	40.50
11	PSV 3.000	560	39.15
12	PSV 3.000	561	45.14
13	PSV 3.000	563	45.14
14	PSV 3.000	612	44.49
15	PSV 3.000	630	24.30
16	PSV 3.000	644	52.65
17	PSV 3.000	688	32.40
18	PSV 3.000	700	36.99
19	PSV 4.500	940	54.00
20	PSV 4.500	940	54.00
21	PSV 4.500	940	54.00

Ao analisar a capacidade de carga total disponível nota-se que a situação é bastante diferente das instâncias *benchmark*. A demanda é apenas 14% da capacidade oferecida. Logo, a escolha dos veículos confere um desafio adicional à solução do caso real como poderá ser observado no Capítulo IV.

CAPÍTULO IV RESULTADOS E DISCUSSÕES

O algoritmo RTR foi executado com bases nas instâncias teste de Taillard e Li e na instância real obtendo bons resultados, em média a 1,4% das melhores soluções conhecidas, levando em consideração os trabalhos [43], [28] e [39]. Os tempos computacionais também foram bastante reduzidos variando de 5 segundos a menos de 10 minutos. Adicionalmente, são apresentados os resultados obtidos a partir do algoritmo adaptado do Roteirizador Marítimo de [16] para resolver o HVRP.

IV.1 Resultados Obtidos para as Instâncias *Benchmark*

Os resultados obtidos estão descritos na Tabela IV-1:

Tabela IV-1 – Comparativo de resultados obtidos

Problema	Pedro*		Li**		Mayron_AG***		Melhor Solução Conhecida	Gap p/ Melhor Solução
	Solução	Tempo (s)	Solução	Tempo (s)	Solução	Tempo (s)		
T_13	1.511,80	5,1	1.517,84	358	1.619,07	55	1.517,84	-0,4%
T_14	608,20	6,9	607,53	141	627,98	43	607,53	0,1%
T_15	1.015,60	7,3	1.015,29	166	1.083,77	31	1.015,29	0,0%
T_16	1.143,00	6,3	1.144,94	188	1.203,91	19	1.144,94	-0,2%
T_17	1.066,70	8,2	1.061,96	216	1.133,36	54	1.061,96	0,4%
T_18	1.827,90	32,1	1.823,58	366	2.214,35	90	1.823,58	0,2%
T_19	1.127,20	18,8	1.120,34	404	1.208,40	136	1.117,51	0,9%
T_20	1.541,60	40,0	1.534,17	447	1.751,58	311	1.534,17	0,5%
H1	12.074,70	95,0	12.067,65	687,82	13.899,37	242	12.067,65	0,1%
H2	10.461,50	162,8	10.234,40	995,27	10.895,30	248	10.234,40	2,2%
H3	16.746,40	269,1	16.231,80	1437,56	19.374,99	433	16.231,80	3,2%
H4	18.149,50	547,4	17.576,10	2256,35	19.804,99	146	17.576,10	3,3%
H5	23.469,00	534,2	21.850,41	3276,91	26.147,95	222	21.850,41	7,4%

* Resultados do presente trabalho – Intel Celeron 1,73 GHz 512 MB RAM

** Resultados obtidos em [43] – Athlon 1GHz 256 MB RAM

*** Resultados baseados em [16] – Intel Core 2 Duo 2,80 GHz 3,2 GB RAM

Os resultados obtidos estão em linha com os melhores resultados publicados, ficando muito próximos (porém piores) do que os obtidos em comparação a Li [43]. A diferença principal entre os dois reside no tempo computacional reduzido no primeiro

decorrente da codificação utilizada. Além disso, estão cerca de 10% melhores que os obtidos pelo método utilizado atualmente na empresa.

A solução obtida para a instância 13 com 50 nós é apresentada na Figura IV-1:

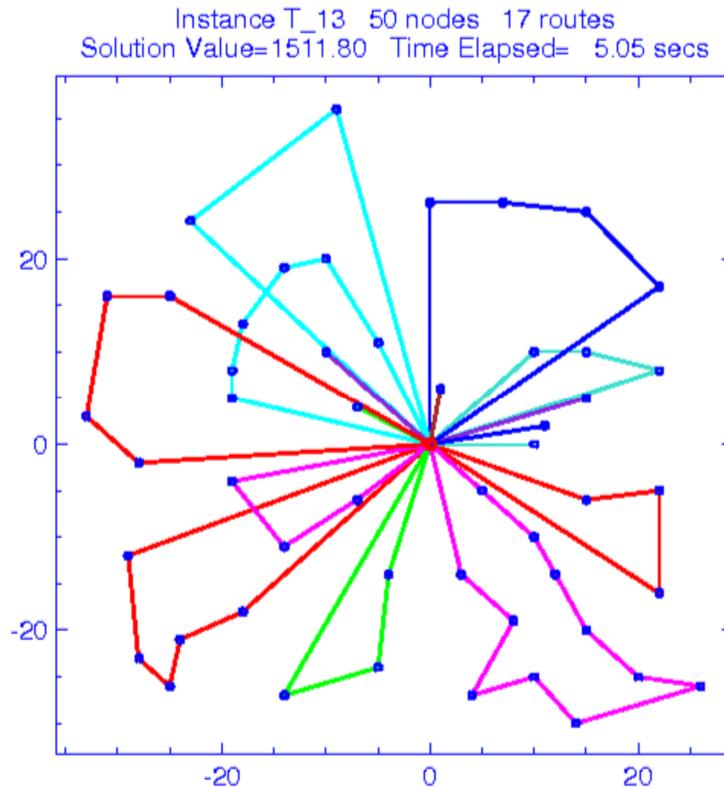


Figura IV-1 – Resultado da Instância T_13

A evolução da solução pode ser observada na Figura IV-2. É possível notar os picos formados pelas perturbações da solução atual e pequenas variações decorrentes da fase de *Uphill* do algoritmo. Para essa instância o algoritmo realiza 221 iterações do *loop* principal. Cada iteração refere-se a uma chamada ao *Uphill* ou *Downhill*. O número de iterações do *loop* principal está diretamente relacionado à convergência para um mínimo local. Caso haja muitos escapes para mínimos locais cada vez menores, o algoritmo tende a realizar mais iterações.

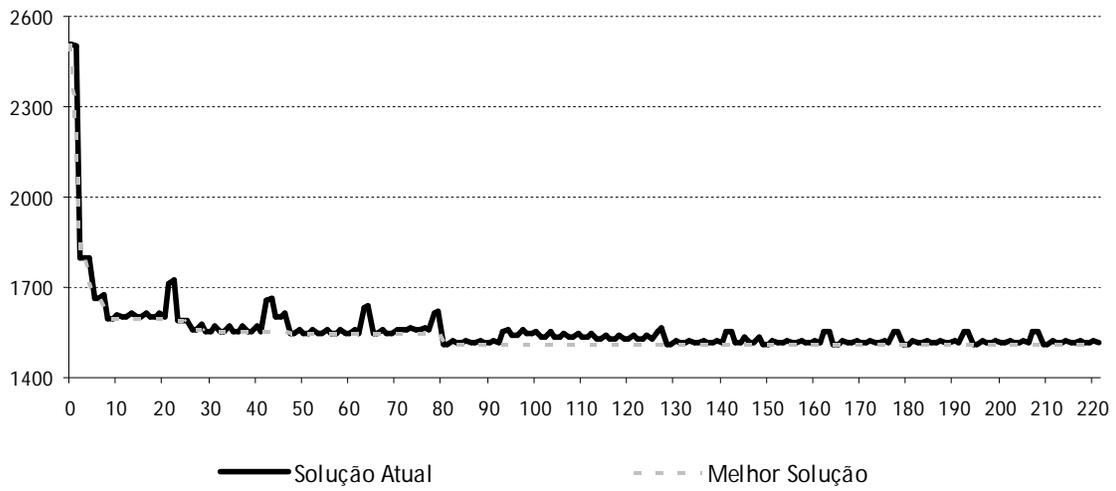


Figura IV-2 – Evolução do custo na instância T_13

Instâncias grandes como a H1, de 200 nós, têm resultado exposto na Figura IV-3 e na Figura IV-4.

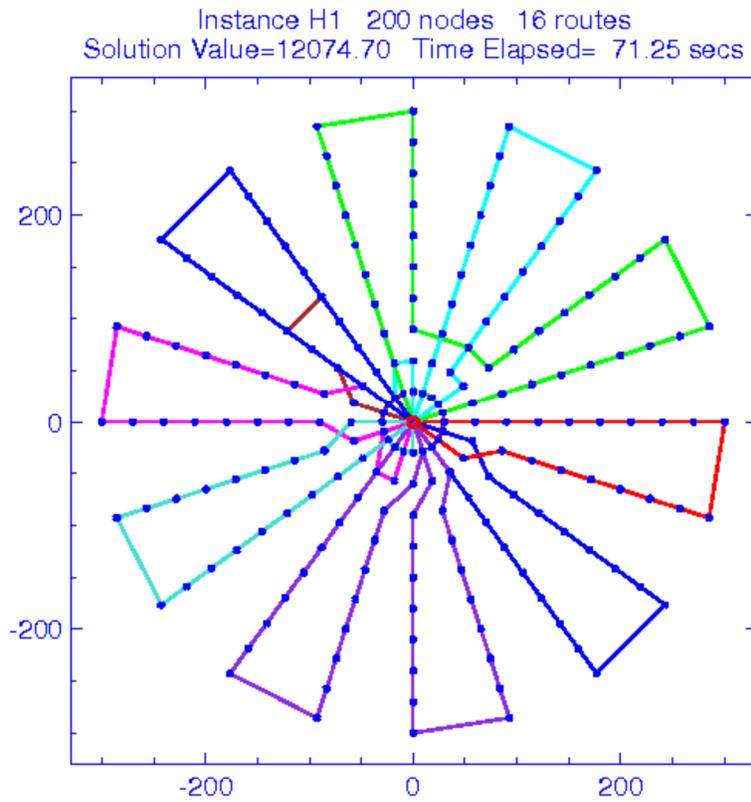


Figura IV-3 – Resultado da instância H1

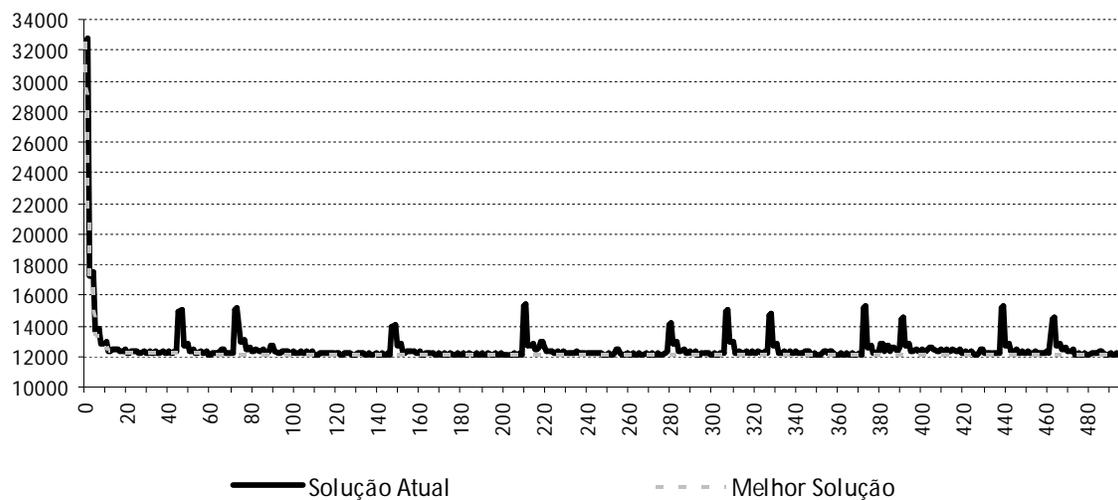


Figura IV-4 – Evolução do custo na instância H1

O algoritmo nesse caso executou 506 iterações do loop principal. É necessário destacar que para as instâncias grandes (de 200 a 360 nós) a heurística de inserção de menor custo utilizada gera um resultado de baixa qualidade (quase 3 vezes pior do que a solução encontrada), deixando grande parte do trabalho a cargo do HRTR.

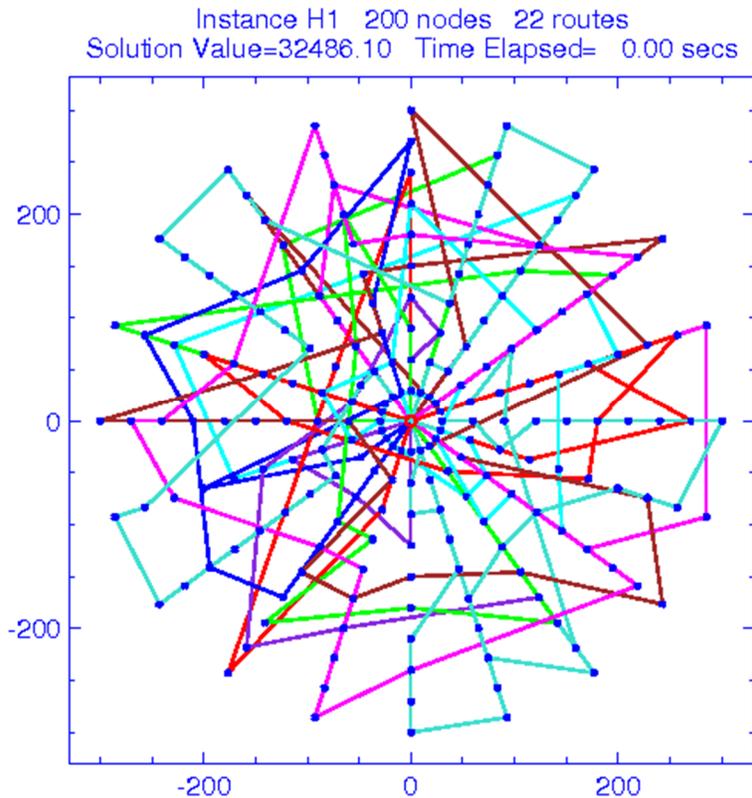


Figura IV-5 – Heurística de Inserção de Menor Custo aplicada a H1

Nas instâncias T_13 e H1 bem como em todas as demais, a quantidade de movimentos aceitos é muito pequena. A Tabela IV-2 ilustra esse comportamento:

Tabela IV-2 – Movimentos e avaliações realizadas nas instâncias T_13 e H1

Método	T_13			H1		
	Movimentos	Avaliações	Aproveitamento	Movimentos	Avaliações	Aproveitamento
Movimento de 1 Ponto	34.068	7.476.386	0,46%	87.822	158.596.542	0,06%
Movimento de 2 Pontos	4.364	6.832.965	0,06%	60.700	137.345.789	0,04%
2-OPT	45.866	34.283.199	0,13%	125.386	644.254.561	0,02%
OR-OPT	0	326.499	0,00%	123	42.916.088	0,00%

Como se pode perceber menos de 1% das avaliações (alguns milhões delas são efetuadas) transforma a solução. Os movimentos do tipo Or-opt são os de menor taxa de sucesso.

A taxa de sucesso cai para instâncias maiores principalmente porque o número de avaliações aumenta muito. O número de avaliações, por sua vez, aumenta por causa do maior número de nós.

Para obter os resultados apresentados nessa seção foram considerados os seguintes parâmetros:

- Intensidade do Laço de *Uphill*: são realizadas 40 iterações para instâncias menores (50 a 75 nós), 30 para instâncias de 100 nós e 20 iterações para as instâncias grandes
- Número de Perturbações: 7 perturbações são permitidas sem que a melhor solução global seja melhorada. Se a solução não decrescer o algoritmo considera que houve convergência para um mínimo local e finaliza
- Não há geração de vizinhança. Todas as buscas locais são efetuadas com todos os nós. Essa questão será abordada na Seção IV.2.2
- Máximo de Iterações sem Redução da Solução Atual: 5 chamadas à fase de *Uphill* são efetuadas antes de executar uma perturbação

IV.2 Análises de Sensibilidade

Nas Seções IV.2.1 e IV.2.2 são discutidos alguns parâmetros que podem influenciar a qualidade da solução. Na Seção IV.2.1, diferentes configurações de busca local são testadas para as instâncias de Taillard para justificar a escolha do conjunto de métodos utilizados em Li [43] e também no presente trabalho.

Já na Seção IV.2.2, dada a configuração de métodos escolhida, são testados diferentes valores para os principais parâmetros do algoritmo de modo a ilustrar a importância de cada um deles para as instâncias de Li.

IV.2.1 Métodos de Busca Local

Foram adicionados aos métodos já presentes no algoritmo (Movimento de Um Ponto, Movimento de Dois Pontos, 2-Opt e Or-Opt) os demais métodos disponíveis na VRPH: Troca Cruzada, Movimento de Três Pontos e 3-Opt.

Cada um deles foi incorporado ao algoritmo separadamente ora na fase de *Uphill* ora na fase de *Downhill*. O movimento Or-Opt foi incluído apenas no *Uphill* visto que o mesmo já fazia parte do algoritmo original.

Pela Tabela IV-3 observa-se o aumento do custo percentual em relação aos resultados obtidos no presente trabalho com o algoritmo em sua configuração inicial quando da inserção de uma das buscas locais na fase de *Uphill*.

Tabela IV-3 – Impacto das buscas locais no *Uphill*

Uphill	T_13	T_14	T_15	T_16	T_17	T_18	T_19	T_20	Média
Troca Cruzada	3,8%	0,6%	1,1%	0,1%	2,8%	1,4%	-1,5%	0,2%	1,1%
Movimento de 3 Pontos	2,3%	-0,5%	1,7%	-0,2%	1,3%	0,2%	-2,2%	0,8%	0,4%
3-OPT	2,2%	0,8%	-0,6%	0,0%	1,9%	0,6%	2,7%	0,3%	1,0%
OR-OPT	2,7%	2,3%	-0,5%	0,3%	0,6%	1,4%	0,5%	1,3%	1,1%

Embora haja diminuição dos custos obtidos em algumas instâncias, os ganhos não são consistentes, havendo aumento do custo em média.

A inserção dos métodos na fase de Downhill está resumida na Tabela IV-4:

Tabela IV-4 – Impacto das buscas locais no *Downhill*

Downhill	T_13	T_14	T_15	T_16	T_17	T_18	T_19	T_20	Média
Troca Cruzada	0,0%	-0,5%	0,0%	0,0%	0,0%	0,0%	-1,0%	0,9%	-0,1%
Movimento de 3 Pontos	2,1%	3,4%	1,1%	0,2%	0,0%	1,1%	0,4%	0,2%	1,1%
3-OPT	0,0%	1,2%	0,0%	0,1%	1,2%	1,1%	-0,2%	0,4%	0,5%

Nesse caso, podem-se perceber ganhos consistentes quando a Troca Cruzada é inserida embora o ganho seja pequeno – 0,1%. O motivo para não inserir esse método no algoritmo para a rodada do caso real se deve à complexidade da Troca Cruzada. Ela cresce rapidamente com o número de pedidos de uma mesma rota, configurando uma limitação de desempenho computacional em instâncias grandes.

IV.2.2 Espaço de Busca

A utilização do método de geração de vizinhança, apresentado na Seção III.2.2.3 , restringe o espaço de busca e proporciona grande redução nos tempos computacionais. Isso pode ser observado na Tabela IV-5:

Tabela IV-5 – Impacto da utilização de vizinhos

Instância	Custo			Tempo (s)		
	Sem Vizinhos	Com Vizinhos	%	Sem Vizinhos	Com Vizinhos	%
H1	12.074,7	12.060,3	-0,1%	95,0	48,6	-48,9%
H2	10.461,5	10.645,6	1,8%	162,8	38,1	-76,6%
H3	16.746,4	17.261,8	3,1%	269,1	45,9	-83,0%
H4	18.149,5	18.003,2	-0,8%	547,4	39,4	-92,8%
H5	23.469,0	23.576,4	0,5%	534,2	49,1	-90,8%

Como se pode observar os resultados pioram em média 1%. Em compensação, os tempos computacionais para chegar a essas soluções são 80% menores.

É necessário ressaltar que para chegar a esses resultados foram utilizados 30 vizinhos para cada nó.

IV.3 Resultados Obtidos para Instância Real

Aplicando o algoritmo (com 30 vizinhos) à instância descrita na Seção III.3.2, obtém-se:

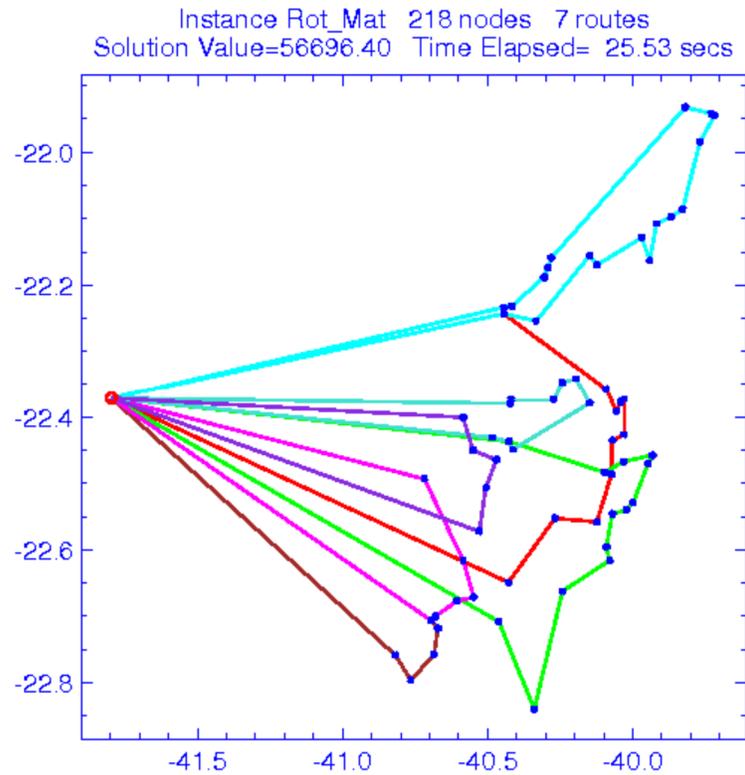


Figura IV-6 – Primeira solução obtida para a instância real

A solução da Figura IV-6 é muito inferior à obtida pela solução adotada atualmente na empresa como se pode observar pela Figura IV-7:

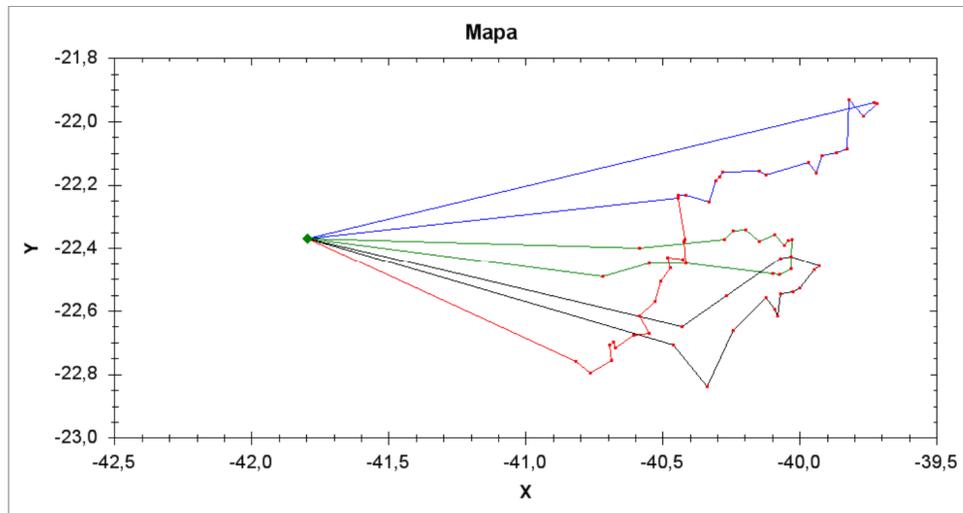


Figura IV-7 – Resultado baseado no Roteirizador Marítimo de [16]

A diferença entre as soluções (custo igual a R\$ 56.696,40 no presente trabalho contra R\$ 47.856,07 da solução da empresa) se deve principalmente ao número de rotas visto que grande parte do custo das rotas se deve ao deslocamento de ida e volta para o porto. Adicionalmente, o elevado número de pedidos por rota (mais de 30) dificulta o algoritmo a gerenciar o número de rotas (trocar veículos utilizados por outros maiores ociosos, por exemplo) de forma adequada.

Para superar essa dificuldade é utilizado mais um operador presente na VRPH ainda não mencionado: o Operador de Concatenação. Esse operador na biblioteca só é utilizado na Heurística de Clarke-Wright, não participando de nenhuma metaheurística.

Houve a necessidade de adaptar o operador para torná-lo um método de busca, ou seja, foi necessário criar um procedimento de busca por rotas para que uma vez que um bom movimento fosse encontrado, o operador pudesse efetuar as modificações na solução conforme Tabela IV-6.

Tabela IV-6 – Esquema de funcionamento do movimento de concatenação elaborado no presente trabalho

Movimento de Concatenação na fase de Uphill	
1	Para cada rota i faça (loop I)
2	Ache a rota j mais próxima da rota i
3	Se Qtde de Veículos > Número de Rotas então
4	Para cada Veículo v faça (loop V)
5	Calcule a economia do Movimento envolvendo v e as rotas i e j
6	se o movimento é viável então
7	se economia > 0 então faça o movimento e volte para loop I
8	se economia < Desvio e economia > maior economia então
9	Armazene o movimento
10	fim se
11	Faça o movimento armazenado de maior economia
12	fim loop V
13	fim se
14	fim loop I

É necessário ressaltar que a rota mais próxima é encontrada computando-se o centro de massa de cada rota e escolhendo o mais próximo do centro de massa da rota i .

Com o novo método foi possível um resultado de melhor qualidade:

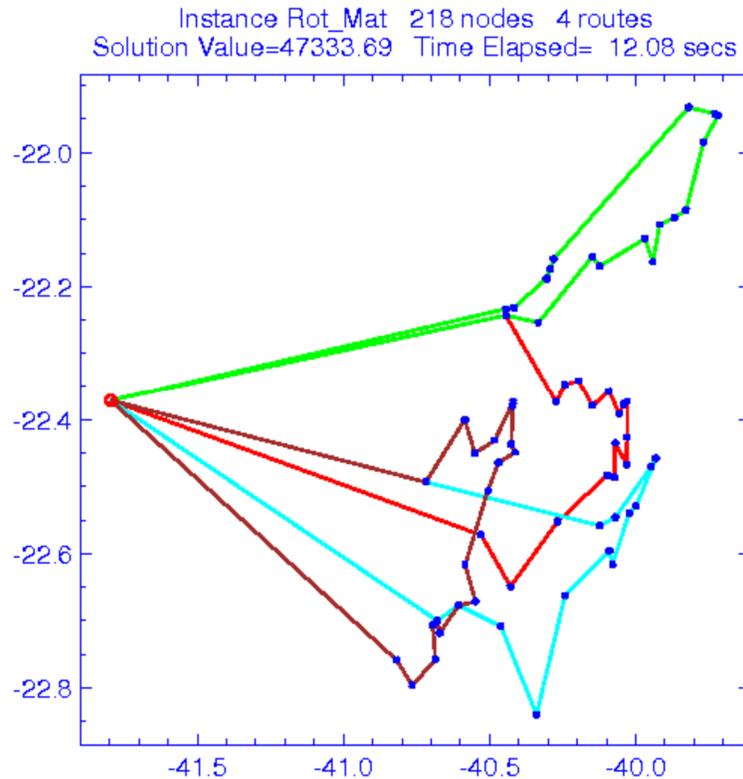


Figura IV-8 – Melhor solução obtida para a instância real

O resultado da Figura IV-8 é 17% mais barato que o obtido com as mesmas configurações (sem o operador de concatenação) e é ainda ligeiramente melhor do que o obtido pela solução atual da empresa.

CAPÍTULO V CONCLUSÕES

IV.1 Conclusões

O objetivo do presente trabalho foi propor um método diferente do utilizado atualmente na empresa para solucionar o problema de programação de embarcações de apoio offshore. Para isso, o problema é descrito e delimitado, sendo enquadrado como um Problema de Roteamento de Veículos de Frota Heterogênea e Limitada.

Os diferentes métodos de solução são enunciados dando-se destaque às metaheurísticas. As principais características do Algoritmo Genético utilizado pela empresa são descritas. Em seguida, o método de solução utilizado no presente trabalho é enunciado.

Deu-se destaque à biblioteca VRPH, conjunto de rotinas de busca local em C++ da COIN-OR, que constituiu o ambiente de desenvolvimento utilizado. A codificação eficiente da solução (em linha com o roteamento de veículos de grande escala) e os diversos métodos de busca disponíveis foram descritos.

Adicionalmente, os métodos desenvolvidos por Li [43], que compõem o algoritmo HRTR e estão presentes na VRPH, foram detalhados. As idéias presentes nesse trabalho nortearam o algoritmo utilizado e as modificações feitas na VRPH para tratar o HVRP. O método de geração da solução inicial (Heurística de Inserção de Menor Custo) foi implementado bem como um movimento específico para tratar a limitação da frota em relação à criação de rotas (que precisam estar associadas a um veículo) e adaptações em todos os métodos para tratar a heterogeneidade (diferentes custos e capacidades).

Diante disso, a expectativa era produzir um método de solução eficiente e determinístico para a solução do HVRP. Tal expectativa pôde ser comprovada através da execução do algoritmo nas instâncias benchmark para o HVRP e comparação com os melhores resultados já publicados. A compilação dos resultados mostrou uma diferença média de 1,4% o que pode ser considerado um bom resultado. Se for levado em consideração o fato de os melhores resultados publicados serem produzidos geralmente por alguma replicação de um algoritmo estocástico (executado várias vezes) em comparação

ao método determinístico utilizado, as soluções obtidas no presente trabalho são ainda mais animadoras.

Tendo o algoritmo HRTR já calibrado a partir das instâncias *benchmark*, pôde-se avaliá-lo em situações mais próximas da realidade das embarcações de apoio *offshore*. Nesse caso, a qualidade da solução foi avaliada em comparação à solução utilizada na empresa, obtendo resultados ligeiramente superiores. Para isso, foi necessário adaptar a estrutura da busca por concatenação com o objetivo de aumentar a chance de que um veículo ocioso (fora da solução) fosse incorporado. Tal adaptação mostrou uma grande melhoria nos resultados obtidos para casos mais próximos da realidade *offshore* (porto distante dos clientes).

IV.2 Sugestões de Trabalhos Futuros

Tratar o problema real segundo a abordagem proposta exige ainda grande trabalho. Desse modo, uma extensão natural seria a ampliação do algoritmo para tratar problemas com janelas de entrega, coleta e entrega e múltiplos produtos.

Com vistas a equacionar o desenvolvimento das funcionalidades Janelas de Entrega e Coleta e Entrega, o trabalho de Kindervater e Savelsbergh [53], que propõe um esquema de pré-processamento e a estratégia de busca lexicográfica, pode ser bastante útil ao reduzir a complexidade de avaliação dos movimentos. Além disso, é esperado que esse tipo de abordagem possa aumentar proporcionalmente o número de avaliações que são transformadas em movimentos, conforme pontuado em [32].

Outra extensão possível é aplicar para o mesmo problema (HVRP) outras metaheurísticas que possam se valer da estrutura já montada na VRPH e eventualmente fazer melhor uso de algumas das buscas locais desenvolvidas e que não obtiveram bons resultados no presente trabalho. Considerando todo o desenvolvimento já feito na biblioteca, evoluí-la para a Busca Tabu, por exemplo, não seria tão custoso.

Adicionalmente, a investigação de heurísticas de busca local que possibilitem a manipulação de rotas e veículos (como a realizada nesse trabalho com a busca por concatenação) pode ser promissora para problemas de frota heterogênea limitada. Nesse caso, uma extensão natural é implementar uma heurística que teste a troca de um veículo

em uso por um ocioso, por exemplo. Outras configurações envolvendo a quebra de rotas em outras menores também poderiam ser testadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] **Site da Petrobras**. Acesso em: 15 out. 2010.
- [2] Material Interno. Petrobras, [S.l.], 2010.
- [3] **Site da TigerRentank**. Disponível em: <www.rentank.com.br>. Acesso em: 19 out. 2010.
- [4] THOMÁS, J. E. **Fundamentos de Engenharia de Petróleo**. 2a Edição. ed. Rio de Janeiro: Interciência, 2004.
- [5] **Site da Keppel Fels**. Disponível em: <www.keppcorp.com>. Acesso em: 18 out. 2010.
- [6] **Site da Huisman**. Disponível em: <www.huisman.com>. Acesso em: 16 out. 2010.
- [7] **Site da Shipspotting**. Disponível em: <www.shipspotting.com>. Acesso em: 15 out. 2010.
- [8] **Site da Ostensjo Rederi AS**. Disponível em: <www.ostensjo.no>. Acesso em: 7 Julho 2011.
- [9] DA SILVEIRA, M. M. **Introdução ao Apoio Marítimo**. Rio de Janeiro: [s.n.], 2002.
- [10] **Site de Fotos Panoramio**. Disponível em: <www.panoramio.com>. Acesso em: 15 out. 2010.
- [11] **Site da Breauxboats**. Disponível em: <www.breauxboats.net>. Acesso em: 12 out. 2010.
- [12] **Site da Astromarítima**. Disponível em: <www.astromaritima.com.br>. Acesso em: 12 out. 2010.
- [13] **Site da Seacormarine**. Disponível em: <www.seacormarine.com>. Acesso em: 12 out. 2010.
- [14] **Site da Farstad**. Disponível em: <www.farstad.no>. Acesso em: 12 out. 2010.
- [15] SHYSHOU, A.; GRIBKOVSKAIA, I.; BARCELÓ, J. "A simulation study of the fleet sizing problem arising in offshore anchor handling operations". *European Journal of Operational Research*, v. 203, n. 1, pp. 230-240, Maio 2010.
- [16] DE ALMEIDA, M. R. "Algoritmos Genéticos Aplicados Programação de Embarcações de Apoio às Operações "Offshore"". In: *XLI Simpósio Brasileiro de Pesquisa Operacional*, pp. 1015-1026, [S.l.], 2009.
- [17] CORDEAU, J.-F. et al. "Vehicle Routing". In: BARNHART, C.; LAPORTE, G. (eds), *Handbook in OR & MS*. v. 14, Cap. 6, pp. 367-428 [S.l.]: Elsevier, 2007.
- [18] DANTZIG, G.; RAMSER, J. "The Truck Dispatching Problem". *Management Science*, v. 6, n. 1, pp. 80-91, 1959.
- [19] LENSTRA, J.; KAN, A. R. "Complexity of vehicle routing and scheduling". *Networks*, v.1, n. 1, pp. 221-227, 1981.
- [20] TOTH, P.; VIGO, D. (Eds.). **The Vehicle Routing Problem**. Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, 2002.
- [21] LAPORTE, G.; NOBERT, Y.; DESROCHERS, M. "Optimal routing under capacity and distance restrictions". *Operations Research*, v. 33, pp. 1050-1073, 1985.
- [22] CLARKE, G.; WRIGHT, J. "Scheduling of vehicles from a central depot to a number of delivery points". *Operations Research*, v. 12, pp. 568-581, 1964.
- [23] GILLET, B.; MILLER, L. "A heuristic algorithm for the vehicle dispatch problem".

- Operations Research*, v. 22,pp. 240-349, 1974.
- [24] LYSGAARD, J.; LETCHFORD, A.; EGGLESE, R. A new branch-and-cut algorithm for the capacitated vehicle routing problem. **Mathematical Programming Series**, v. 100, p. 423–445.
- [25] FUKASAWA, R. et al. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. **Lecture Notes in Computer Sciences**, v. 3064, p. 1-15.
- [26] BRÄYSY, O.; GENDREAU, M. "Vehicle Routing with Time Windows, Part I: Route Construction and Local Search Algorithms". **Transportation Science**, v. 39, n. 1,pp. 104-118, 2005.
- [27] LI, F. *Modeling and Solving Variants of the Vehicle Routing Problem: Algorithms, Test Problems, and Computational Results*. Ph.D. dissertation, University of Maryland, College Park, Estados Unidos, 2005.
- [28] TARANTILIS, C.; KIRANOUDIS, C.; VASSILIADIS, V. "A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem". **Journal of the Operational Research Society**, v. 54,pp. 65-71, 2003.
- [29] LIN, S. "Computer solutions of the traveling salesman problem". **The Bell System Technical Journal**, v. 44,pp. 2245-2269, 1965.
- [30] JOHNSON, D.; MCGEOCH, L. "The Traveling Salesman Problem: A Case Study in Local Optimization". In: AARTS, E.; LENSTRA, J. K. (eds), *Local Search in Combinatorial Optimization*. 1. ed., Cap. 8, London: Wiley, 1997.
- [31] LIN, S.; KERNIGHAN, B. "An Effective Heuristic Algorithm for the Traveling-Salesman Problem". **Operations Research**, v. 21, n. 2,pp. 498-516, Março 1973.
- [32] GRÖER, C.; GOLDEN, B.; WASIL, E. "A library of local search heuristics for the vehicle routing problem". **Mathematical Programming Computation**, v. 2,pp. 79-101, Abril 2010.
- [33] GLOVER, F.; LAGUNA, M. **Tabu Search**. 1. ed. Boston: Kluwer Academic Publishers, 1997.
- [34] GLOVER, F.; KOCHENBERGER, G. (Eds.). **Handbook of Metaheuristics**. 1. ed. Nova Iorque: Kluwer Academic Publishers, 2003.
- [35] CRAINIC, T. *Parallel Solution Methods for Vehicle Routing Problems*. Cirrelet, Quebec, Canadá, 2007.
- [36] TAILLARD, E. "A heuristic column generation method for the heterogeneous fleet VRP". **RAIRO-Operations Research**, v. 33, n.1,pp. 1-14, 1999.
- [37] TARANTILIS, C.; KIRANOUDIS, C.; VASSILIADIS, V. "A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem". **European Journal of Operational Research**, v. 152,pp. 148-158, 2004.
- [38] GEN, M.; CHENG, R. **Genetic Algorithms and Engineering Design**. Nova Iorque: John Wiley & Sons, 1996.
- [39] PRINS, C. "A simple and effective evolutionary algorithm for the vehicle routing problem". **Computers & Operations Research**, 31,pp. 1985-2002, 2004.
- [40] PRINS, C. "Two memetic algorithms fo heterogeneous fleet vehicle routing problems". **Engineering Applications of Artificial Intelligence**, 22,pp. 916-928, 2009.
- [41] CORMEN, T.; LEISERSON, C.; RIVEST, R. **Introduction to Algorithms**. 2nd. ed.

- Cambridge: MIT Press, 2001.
- [42] LI, F.; GOLDEN, B.; WASIL, E. "A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem". *Computers & Operations Research*, v. 34, pp. 2734-2742, 2007.
 - [43] GRÖER, C. *Parallel and Serial Algorithms for Vehicle Routing Problems*. University of Maryland, College Park, Estados Unidos, 2008.
 - [44] LI, F.; GOLDEN, B.; WASIL, E. "Very large-scale vehicle routing: new test problems, algorithms, and results". *Computers & Operations Research*, v. 32, pp. 1165-1179, 2005.
 - [45] DUECK, G. "The Great Deluge Algorithm and Record-to-Record Travel". *Journal of Computational Physics*, v. 104, n. 86, pp. 86-91, 1993.
 - [46] SOLOMON, M. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Windows Constraints". *Operations Research*, v. 35, n. 2, pp. 254-265, Abril 1987.
 - [47] **Site da COIN-OR**. Disponível em: <www.coin-or.org>. Acesso em: 1 Março 2011.
 - [48] **Site da PLplot**. Disponível em: <plplot.sourceforge.net>. Acesso em: 30 jan. 2011.
 - [49] KYTÖJOKI, J. et al. "An efficient variable neighborhood search heuristic for very large scale vehicle routing problems". *Computers & Operations Research*, v. 34, pp. 2743-2757, 2007.
 - [50] **Site do Symphony**. Disponível em: <projects.coin-or.org/SYMPHONY>. Acesso em: 1 Junho 2011.
 - [51] GOLDEN, B. et al. "The fleet size and mix vehicle routing problem". *Computers & Operations Research*, v. 11, pp. 49-66, 1984.
 - [52] KINDERVATER, G. A. P.; SAVELSBERGH, M. "Vehicle Routing: handling edge exchanges". In: AARTS, E. H. L.; LENSTRA, J. K. (eds), *Local Search in Combinatorial Optimization.*, Cap. 10, pp. 337-360 Nova Iorque: John Wiley & Sons, 1997.