UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA

KELVIN YUSO TAMURA

Heurísticas de programação linear inteira para resolução de problemas de programação de frota com restrições de sincronização

São Paulo
2019

KELVIN YUSO TAMURA

Heurísticas de programação linear inteira para resolução de problemas de programação de frota com restrições de sincronização

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

Área de Concentração:
Engenharia Naval e Oceânica

Orientador:
Prof. Dr. André Bergsten Mendes

São Paulo
2019

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo,_____ de _____ de _____

Assinatura do autor: _____

Assinatura do orientador: _____

Catalogação-na-publicação

Tamura, Kelvin Yuso

Heurísticas de programação linear inteira para resolução de problemas de programação de frota com restrições de sincronização / K. Y. Tamura – versão corr. -- São Paulo 2019.

148 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Naval e Oceânica.

1.Modelagem matemática 2.Heurística híbrida 3.Programação de veículos 4.Programação linear inteira I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Naval e Oceânica II.t.

AGRADECIMENTOS

Agradeço ao meu orientador Professor Doutor André Bergsten Mendes pela disponibilidade e paciência, pelo companheirismo nos momentos difíceis e pelas boas conversas repletas de orientações, ideias e motivações.

Agradeço ao meu amigo e mentor Bruno Bandarovsky, por compartilhar seus conhecimentos e experiências e por ter sido o grande incentivador para o desenvolvimento deste trabalho.

Agradeço ao Departamento de Engenharia Naval e Oceânica da USP e ao corpo docente pela oportunidade de desenvolvimento deste trabalho.

Agradeço aos amigos, tanto pelos momentos especiais e felizes, como também pelos difíceis. Um agradecimento especial ao meu grande amigo Cezar Augusto Bellezi, por toda a ajuda, motivação e pelos vários finais de semana dedicados aos estudos.

Agradeço à família, por estar sempre ao meu lado, pelo carinho, incentivo e suporte ao longo de todos estes anos, e por me fornecer todas as condições necessárias para que fosse possível a minha caminhada até aqui.

RESUMO

A presente pesquisa aborda um problema de programação de veículos rico, em que a característica mais importante é a demanda de múltiplas embarcações para atendimento a uma única tarefa. Trata-se de uma aplicação real do setor de apoio marítimo “offshore”, das embarcações que fazem o reboque e o lançamento de linhas de ancoragem de sondas de perfuração e unidades de produção. Como método de solução, aplicaram-se duas heurísticas com uma abordagem híbrida que incluem uma inserção baseada em programação linear inteira, visando a minimização do custo total da operação, dentro de um tempo de processamento aceitável.

Palavras-chave: Modelagem matemática. Heurística híbrida. Programação de veículos. Programação linear inteira.

ABSTRACT

This research deals with a rich vehicle scheduling problem, having as the most important feature the demand of multiple vessels per task. It is a real problem present in the oil industry related to the vessels that undertake the towing and the launching of mooring lines of drilling and production units. As a solution method, two heuristics with a hybrid approach were applied which include an insertion based on integer linear programming, aiming at minimizing the total cost of the operation, within an acceptable processing time.

Keywords: Mathematical modelling. Matheuristic. Vehicle scheduling. Integer linear programming.

Lista de Figuras

Figura 1-1 - Metodologia para Resolução de Problemas em Pesquisa Operacional....16
Figura 2-1 – Exemplo de planejamento das rotas das embarcações.....25
Figura 2-2 – Exemplo de planejamento de rota.....26
Figura 4-1 - Algoritmo 1 – Pseudo-código para Procedimento de Construção de Solução Aleatória.....46
Figura 4-2 - Algoritmo 2 – Pseudo-código para Algoritmo de Múltiplos Recomeços Aleatório 47
Figura 4-3 – Fluxograma da heurística – Visão global 50
Figura 4-4 – Fluxograma – Heurística fase construtiva.....52
Figura 4-5 – Fluxograma função "selectVesselSets" 56
Figura 4-6 - Fluxograma – Heurística Fase Busca Local – Método 1.....60
Figura 4-7 - Pseudo código do algoritmo da rotina iterativa de permutações de inserção 62
Figura 4-8 - Fluxograma – Heurística Fase Busca Local – Método 2.....67
Figura 5-1 – Gráfico resultados Cenário 1 - 10 tarefas - Penalidade 87
Figura 5-2 – Gráfico resultados Cenário 1 – 20 tarefas – Penalidade 88
Figura 5-3 – Gráfico resultados Cenário 1 – 10 tarefas – Instante de Liberação 90
Figura 5-4 – Gráfico resultados Cenário 1 - 20 tarefas – Instante de Liberação.....91
Figura 5-5 – Gráfico resultados Cenário 2 - 10 tarefas – Penalidade.....94
Figura 5-6 – Gráfico resultados Cenário 2 - 20 tarefas – Penalidade.....95
Figura 5-7 – Gráfico resultados Cenário 2 - 10 tarefas – Instante de Liberação.....96
Figura 5-8 – Gráfico resultados Cenário 2 - 20 tarefas –Instante de Liberação.....97
Figura 5-9 – Gráfico resultados Cenário 3 - 10 tarefas – Penalidade.....100
Figura 5-10 – Gráfico resultados Cenário 3 - 20 tarefas – Penalidade.....101
Figura 5-11 – Gráfico resultados Cenário 3 - 10 tarefas – Instante de Liberação.....102
Figura 5-12 – Gráfico resultados Cenário 3 - 20 tarefas – Instante de Liberação.....103
Figura 5-13 – Gráfico resultados Cenário 4 - 10 tarefas – Penalidade.....106

Figura 5-14 – Gráfico resultados Cenário 4 - 20 tarefas – Penalidade.....107
Figura 5-15 – Gráfico resultados Cenário 4 - 10 tarefas – Instante de Liberação.....108
Figura 5-16 – Gráfico resultados Cenário 4 - 20 tarefas – Instante de Liberação.....109
Figura 5-17 – Gráfico resultados Cenário 1 - 10 tarefas – Instante de Liberação.....115
Figura 5-18 – Gráfico resultados Cenário 1 – 20 tarefas – Instante de Liberação.....116
Figura 5-19 – Gráfico resultados Cenário 2 - 10 tarefas – Instante de Liberação.....117
Figura 5-20 – Gráfico resultados Cenário 2 - 20 tarefas – Instante de Liberação.....118
Figura 5-21 – Gráfico resultados Cenário 3 - 10 tarefas – Instante de Liberação.....121
Figura 5-22 – Gráfico resultados Cenário 3 - 20 tarefas – Instante de Liberação.....122
Figura 5-23 – Gráfico resultados Cenário 4 - 10 tarefas – Instante de Liberação.....124
Figura 5-24 – Gráfico resultados Cenário 4 - 20 tarefas – Instante de Liberação.....125
Figura 5-25 – Gráfico resultados Finais – 10 tarefas – Método 1.....131
Figura 5-26 – Gráfico resultados Finais – 20 tarefas – Método 1.....132
Figura 5-27 – Gráfico Resultados Finais – 10 tarefas – Método 2 134
Figura 5-28 – Gráfico Resultados Finais – 20 tarefas – Método 2 135

Lista de Tabelas

Tabela 5-1 – Parâmetros dos cenários 80
Tabela 5-2 – Cenário 1: Tarefas candidatas 81
Tabela 5-3 – Cenário 2: Conjuntos de embarcações candidatas 82
Tabela 5-4 – Cenário 3: Iterações sem melhoria 82
Tabela 5-5 – Cenário 4: Tarefas excluídas 83
Tabela 5-6 – Resultados do Processamento do Modelo Matemático 84
Tabela 5-7 – Resultados Cenário 1 - 10 tarefas - Penalidade 86
Tabela 5-8 – Resultados Cenário 1 - 20 tarefas - Penalidade 87
Tabela 5-9 – Resultados Cenário 1 – 10 tarefas – Instante de Liberação 89
Tabela 5-10 – Resultados Cenário 1 – 20 tarefas – Instante de Liberação 90
Tabela 5-11 – Resultados Cenário 2 - 10 tarefas – Penalidade 93
Tabela 5-12 – Resultados Cenário 2 - 20 tarefas – Penalidade 94
Tabela 5-13 – Resultados Cenário 2 - 10 tarefas – Instante de Liberação 95
Tabela 5-14 – Resultados Cenário 2 - 20 tarefas – Instante de Liberação 96
Tabela 5-15 – Resultados Cenário 3 - 10 tarefas – Penalidade 99
Tabela 5-16 – Resultados Cenário 3 - 20 tarefas – Penalidade 100
Tabela 5-17 – Resultados Cenário 3 - 10 tarefas – Instante de Liberação 101
Tabela 5-18 – Resultados Cenário 3 - 20 tarefas – Instante de Liberação 102
Tabela 5-19 – Resultados Cenário 4 - 10 tarefas – Penalidade 106
Tabela 5-20 – Resultados Cenário 4 - 20 tarefas – Penalidade 107
Tabela 5-21 – Resultados Cenário 4 - 10 tarefas – Instante de Liberação 108
Tabela 5-22 – Resultados Cenário 4 - 20 tarefas – Instante de Liberação 109
Tabela 5-23 – Cenário 1 – Tarefas Candidatas – Valores Médios 110
Tabela 5-24 – Cenário 2 – Conjuntos de Embarcações – Valores Médios 111
Tabela 5-25 – Cenário 3 – Iterações sem Melhoria – Valores Médios 111
Tabela 5-26 – Cenário 4 – Tarefas Excluídas – Valores Médios 112
Tabela 5-27 - Resultados Cenário 1 - 10 tarefas – Instante de Liberação 114

Tabela 5-28 – Resultados Cenário 1 – 20 tarefas – Instante de Liberação 115
Tabela 5-29 – Resultados Cenário 2 - 10 tarefas – Liberação para Início 117
Tabela 5-30 – Resultados Cenário 2 - 20 tarefas – Instante de Liberação..... 118
Tabela 5-31 – Resultados Cenário 3 - 10 tarefas – Instante de Liberação..... 120
Tabela 5-32 – Resultados Cenário 3 - 20 tarefas – Instante de Liberação..... 121
Tabela 5-33 – Resultados Cenário 4 - 10 tarefas – Instante de Liberação..... 124
Tabela 5-34 – Resultados Cenário 4 - 20 tarefas – Instante de Liberação..... 125
Tabela 5-35 – Cenário 1 – Tarefas Candidatas - Valores Médios 127
Tabela 5-36 – Cenário 2 – Conjuntos de Embarcações - Valores Médios 127
Tabela 5-37 – Cenário 3 – Iterações sem Melhoria - Valores Médios 127
Tabela 5-38 – Cenário 4 – Tarefas Excluídas - Valores Médios 127
Tabela 5-39 – Resultados Finais – 10 tarefas – Método 1 130
Tabela 5-40 – Resultados Finais – 20 tarefas – Método 1 131
Tabela 5-41– Resultados Finais – 10 tarefas – Método 2 134
Tabela 5-42 – Resultados Finais – 20 tarefas – Método 2 135
Tabela 5-43 – Cenário 1 – Tarefas Candidatas – Valores Médios 137
Tabela 5-44 – Cenário 2 – Conjuntos de Embarcações – Valores Médios..... 137
Tabela 5-45 – Cenário 3 – Iterações sem Melhoria – Valores Médios 137
Tabela 5-46 – Cenário 4 – Tarefas Excluídas – Valores Médios 138
Tabela 5-47 – Resultados Finais – Comparação Método 1 e Método 2 – 10 tarefas ... 140
Tabela 5-48 – Resultados Finais – Comparação Método 1 e Método 2 – 20 tarefas ... 141

SUMÁRIO

1 INTRODUÇÃO 13

1.1 DESCRIÇÃO SUCINTA DO PROBLEMA 13

1.2 MOTIVAÇÃO DO TEMA 15

1.3 OBJETIVOS 15

1.4 METODOLOGIA 16

1.5 ESTRUTURA DA DISSERTAÇÃO 17

2 REVISÃO DA LITERATURA 19

2.1 LOGÍSTICA OFFSHORE 19

2.2 ROTEIRIZAÇÃO DE VEÍCULOS COM MÚLTIPLOS RECURSOS 23

2.3 MÉTODOS HÍBRIDOS PARA PROBLEMAS DE ROTEIRIZAÇÃO DE VEÍCULOS 30

3 MODELAGEM MATEMÁTICA 35

3.1 DESCRIÇÃO DETALHADA DO PROBLEMA 35

3.2 MODELAGEM MATEMÁTICA 37

3.2.1 Conjuntos e Índices 38

3.2.2 Parâmetros 39

3.2.3 Variáveis de decisão 41

3.2.1 Modelo Matemático 41

4 MÉTODOS DE SOLUÇÃO 45

4.1 INTRODUÇÃO 45

4.2 MÉTODOS PROPOSTOS 48

4.2.1 Método 1 – Seleção Aleatória 51

4.2.1.1 Heurística construtiva 52

4.2.1.2 Busca local – exclusão aleatória 58

4.2.2 Método 2 – Seleção por Penalidade Aplicada à Tarefa 64

4.2.2.1 Heurística construtiva 64

4.2.2.2 Busca local – exclusão por penalidade aplicada à tarefa 65

4.3 MODELO MATEMÁTICO DE INSERÇÃO 68

4.3.1 Conjuntos e Índices 68

4.3.2 Parâmetros 70

4.3.3 Variáveis 71

4.3.4 Modelo *72*

5 APLICAÇÃO DOS MÉTODOS DE SOLUÇÃO 77

5.1 BASE DE DADOS 78

5.2 CALIBRAÇÃO DOS PARÂMETROS 79

5.3 RESULTADOS MODELO MATEMÁTICO EXATO – BASE DE COMPARAÇÃO 83

5.4 CALIBRAÇÃO DE PARÂMETROS - MÉTODO 1 – SELEÇÃO ALEATÓRIA 85

5.4.1 Resultados Cenário 1 – Tarefas Candidatas *85*

5.4.2 Resultados Cenário 2 – Conjuntos de Embarcações Candidatas *92*

5.4.3 Resultados Cenário 3 – Iterações Sem Melhoria *98*

5.4.4 Resultados Cenário 4 – Tarefas Excluídas *104*

5.4.5 Considerações Finais – Método Seleção Aleatória *110*

5.5 CALIBRAÇÃO DE PARÂMETROS - MÉTODO 2 – SELEÇÃO POR PENALIDADE DAS TAREFAS 112

5.5.1 Resultados Cenário 1 – Tarefas Candidatas *113*

5.5.2 Resultados Cenário 2 – Conjuntos de Embarcações Candidatas *116*

5.5.3 Resultados Cenário 3 – Iterações Sem Melhoria *119*

5.5.1 Resultados Cenário 4 – Tarefas Excluídas *123*

5.5.2 Considerações Finais – Método Seleção Por Penalidade das Tarefas *126*

5.6 RESULTADOS FINAIS 128

5.6.1 Definição dos Parâmetros para as Simulações dos Resultados Finais *128*

5.6.2 Método 1 – Seleção Aleatória *130*

5.6.3 Método 2 – Seleção por Penalidade das Tarefas *133*

5.7 DISCUSSÃO DOS RESULTADOS 136

6 CONCLUSÕES 144

7 REFERÊNCIAS 146

1 INTRODUÇÃO

A finalidade deste capítulo é introduzir o tema a ser abordado nesta pesquisa, cujo título é *Heurísticas de programação linear inteira para resolução de problemas de programação de frota com restrições de sincronização*, apresentar os objetivos da pesquisa, e a metodologia a ser utilizada.

O tema desta pesquisa se enquadra na classe de problemas de roteirização e programação de veículos (PRPVs), o qual envolve a definição de rotas, associado a aspectos temporais das viagens, frota heterogênea, restrições de sincronização, entre outras.

Os PRPVs constituem uma classe de problemas na área da Pesquisa Operacional, a qual estuda a resolução de problemas reais, tendo como foco a tomada de decisões, aplicando conceitos e métodos de várias áreas científicas na concepção, planejamento ou operação de sistemas (SOBRAPO, 2018).

1.1 Descrição Sucinta do Problema

O escopo desta dissertação refere-se à programação da frota de apoio marítimo “offshore”, em particular, de rebocadores oceânicos que prestam diversos serviços, dentre eles o reboque e a ancoragem de sondas de perfuração e plataformas de produção de óleo & gás no mar. Como as companhias do setor de óleo e gás possuem um portfólio de atividades de exploração, existe uma demanda por reboque e ancoragens de sondas que concorre com os serviços prestados às unidades de produção.

Conhecida a demanda pelos diversos serviços, faz-se necessário planejar, de forma eficaz, a alocação dos recursos, sendo esta a questão central desta pesquisa. O tipo de embarcação que efetua estas operações são os rebocadores oceânicos denominados AHTS (“anchor handling tug supply”) dotado de área de convés suficiente para acomodar âncoras, boias, e linhas de ancoragem, e dispositivos para manuseá-los (Ritchie, 2004).

O problema de programação da frota de AHTS consiste em determinar a rota de cada embarcação, a qual é composta por uma sequência de locais associados às tarefas que serão realizadas, observando as seguintes restrições: o instante de liberação de cada tarefa é dinâmico, porém determinístico; a frota é heterogênea; a demanda de cada tarefa é expressa por um conjunto de embarcações de diferentes classes; a execução da tarefa só pode acontecer quando todas as embarcações alocadas a uma tarefa estiverem disponíveis no local de início da tarefa.

Assim como ocorre em outros contextos, a sequenciação do atendimento das tarefas utiliza-se de critérios econômicos. Com isto, o problema reside em minimizar a parcela do custo variável relativa ao consumo de combustível das embarcações, ao realizarem seus deslocamentos, e uma parcela associado ao nível de serviço. Por exemplo, uma sonda de perfuração que demora para ser rebocada para outra localidade em decorrência de um mal planejamento operacional, causa um prejuízo econômico significativo, tendo em vista o custo de afretamento da ordem de US$ 300 mil / dia, segundo Clarksons (2019).

Será adotado, como hipótese, o conhecimento da demanda para um horizonte de curto prazo (30 a 45 dias); também são admitidos como conhecidos os locais e os instantes em que as embarcações estarão disponíveis no início do horizonte de planejamento. Também são premissas desta pesquisa que a frota é fixa, e que tanto os tempos de navegação como os tempos de execução das tarefas são determinísticos.

1.2 Motivação do Tema

O tema abordado nesta pesquisa se enquadra na classe de problemas de roteirização e programação de veículos. Esta é uma área muito rica de pesquisa, conforme demonstram as revisões de Laporte; Osman (1995), Eksiouglu et al. (2009), Lahyani et al. (2015). Contudo, diferentemente dos problemas tratados nestas revisões, o problema desta pesquisa incorpora o aspecto de alocação de múltiplas embarcações para realização de uma mesma tarefa, as quais devem operar de forma sincronizada. A alocação de embarcações operando de forma sincronizada adiciona ao problema estudado restrições de sincronização, que, segundo Drexl (2012), "problemas de roteirização de veículos como múltiplas restrições de sincronização são problemas de roteirização de veículos que exibem requisitos adicionais de sincronização no que diz respeito a aspectos espaciais, temporais e de capacidade".

Já que o caso particular do problema tratado nesta pesquisa, em que apenas um veículo é demandado por tarefa, recai no problema clássico de roteirização, que é NP-Difícil Eksiouglu et al. (2009), o presente problema também é NP-Difícil. A motivação para esta pesquisa também decorre do fato do tema possuir aplicação prática na indústria de óleo & gás, além de possuir relação com outros contextos operacionais, conforme discutido em Drexl (2012).

1.3 Objetivos

O objetivo desta pesquisa consiste em resolver o problema de programação de frota com restrições de sincronização, fazendo uso de métodos heurísticos. Em particular, objetiva-se testar uma abordagem híbrida que inclui uma inserção baseada em programação linear inteira.

1.4 Metodologia

A solução de problemas complexos na área de planejamento de operações é o objeto central da área de Pesquisa Operacional. É uma ciência que aplica o método científico para resolução de diversos problemas, tendo como base teórica áreas do conhecimento de estatísticas, ciência da computação e matemática.

De acordo com (Bertrand; Fransoo, 2002) (Figura 1-1) a Pesquisa Operacional, como ciência, se propõe a resolver problemas na área de operações, usando os passos identificados no ciclo descrito em sua metodologia. O problema real, após a etapa de abstração, é convertido em um modelo conceitual que traz consigo a especificação da extensão ou abrangência do problema real. Em seguida, a fase de modelagem matemática leva à definição das relações causais entre as variáveis do problema, tendo como produto um modelo quantitativo. O próximo passo consiste em resolver o problema representado pelo modelo matemático. Por último, vem a fase de implementação da solução Mendes (2007).

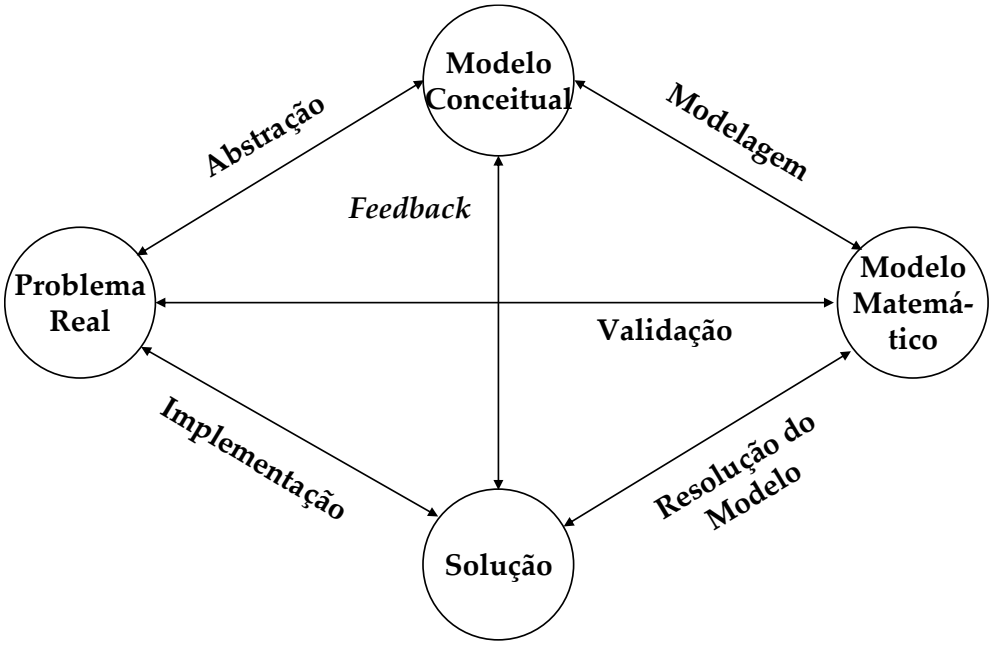

Figura 1-1 - Metodologia para Resolução de Problemas em Pesquisa Operacional

Fonte: Adaptada de (Bertrand; Fransoo, 2002)

Para a presente pesquisa, a abstração, modelagem e resolução do modelo definem o escopo do trabalho. Seguindo nesta linha, primeiramente será apresentada uma descrição detalhada do problema, seguida de uma formulação matemática baseada na abordagem clássica de fluxo em rede. Por fim, serão apresentados os métodos utilizados para solucionar o problema, o qual será testado com instâncias de teste.

Os métodos mais usualmente empregados na solução de problemas de roteirização e programação de veículos podem ser divididos em duas vertentes: os baseados em estratégias otimizantes, como a técnica de decomposição de Dantzig-Wolfe, e sua combinação com o método “branch-and-bound”, resultando no método “branch-and-price”; a outra abordagem largamente empregada se baseia em métodos heurísticos construtivos, de busca local e meta-heurísticos Laporte; Osman (1995) e Eksiouglu et al. (2009). Esta pesquisa, contudo, utilizará modelos heurísticos híbridos, em que uma parte do método de solução consiste em resolver um modelo de programação linear inteira. Este tipo de abordagem permite extrair as potencialidades de cada abordagem, resultando em métodos competitivos, capazes de resolver problemas complexos com baixo tempo de processamento.

1.5 Estrutura da Dissertação

A presente dissertação está estruturada nos seguintes capítulos: ***2. Revisão da Literatura*** – apresenta a revisão dos trabalhos que abordam a logística offshore das operações de ancoragem através da utilização de embarcações do tipo AHTS, trabalhos cujo tema é a roteirização de veículos com múltiplos recursos e por fim trabalhos que abordam problemas de roteirização de veículos resolvidos através de métodos híbridos; ***3. Descrição do Problema*** – apresenta o modelo matemático desenvolvido para este problema, apresentando uma descrição detalhada do problema e cada uma das suas restrições operacionais; ***4. Método de Solução*** – apresenta e discute o método de solução desenvolvido; ***5 Aplicação dos Métodos de Solução*** – Apresenta os resultados obtidos da aplicação dos métodos de solução;

6. Conclusões– contém a conclusão desta pesquisa e indica possíveis extensões dos métodos desenvolvidos;

2 REVISÃO DA LITERATURA

Este capítulo contém a revisão da literatura consultada no decorrer desta pesquisa. Inicialmente, serão apresentados os principais trabalhos que abordam a logística offshore das operações de ancoragem através da utilização de navios do tipo AHTS, tema semelhante ao abordado no presente trabalho. Também foram revisados trabalhos que tratam do problema de roteirização de veículos com múltiplos recursos, que consideram restrições adicionais de sincronização de recursos para atendimento das demandas dos consumidores. Por fim, foram revisados trabalhos que abordam o método de solução com heurísticas híbridas para problemas de roteirização de veículos, que serviu como base para o desenvolvimento dos métodos de resolução do problema abordado neste trabalho.

2.1 Logística Offshore

Operações de perfuração, efetuadas por plataformas de perfuração, são tarefas essenciais para a exploração e produção de gás e petróleo. Estas plataformas de perfuração possuem como principal característica a mobilidade para permitir-lhes trabalhar em diferentes poços, podendo ser utilizadas em diversas fases da vida de um campo de exploração. Na fase de exploração, executando tarefas de perfuração de poços, na fase de desenvolvimento do campo, perfurando e completando os poços que irão drenar o reservatório de acordo com o projeto de exploração, durante a fase produtiva, executando a manutenção dos poços em produção (troca de equipamentos internos, mudanças de zonas de produção, etc.) e na fase final da vida de cada poço, os quais devem ter seus equipamentos retirados e condicionados para o abandono definitivo. Para a execução destas tarefas, é necessário o deslocamento destas plataformas de um poço para o outro, sendo este deslocamento executado com o auxílio de embarcações rebocadoras do tipo AHTS (“Anchor Handling Tug Supply”). Devido ao elevado custo de aquisição destas embarcações de apoio, as companhias usualmente as contratam de companhias de navegação. Os contratos de locação

destas embarcações podem ser efetuados de duas maneiras: contratos de longa duração, ou “Spot”, que são os de curta duração. As taxas praticadas no mercado de locações do tipo Spot são frequentemente muito superiores às taxas praticadas no mercado de locações de longa duração, fazendo com que o aluguel de embarcações por contratos de curto período seja apenas utilizado quando há falta de navios para locação de longo prazo, ou quando se possui uma tarefa urgente para ser realizada, e toda a frota já está alocada em outras tarefas. Sendo assim, os problemas encontrados na logística das operações de ancoragem offshore são, tanto de cunho estratégico, como a definição do número de embarcações do tipo AHTS a serem contratadas por contratos de afretamento de longo prazo, como de cunho operacional, como o planejamento e a roteirização das embarcações, para atendimento a todas as demandas de movimentação, ancoragem, manutenção e reposição de suprimentos das plataformas de perfuração.

A seguir serão apresentados alguns trabalhos que tratam deste tema de logística offshore das operações de ancoragem. A grande maioria dos trabalhos são baseados nas operações de suporte às plataformas de perfuração e produção do Mar do Norte, situado na Noruega, local tido como referência na exploração e produção offshore.

Shyshou et al. (2009) tratam de um problema de dimensionamento de frota de rebocadores do tipo AHTS (Anchor Handling Tug Supply) para operações de movimentação de unidades de perfuração no mar norueguês. Foi desenvolvido um modelo de simulação de eventos discretos, para simular uma sequência de movimentações, por um período de um ano, em que cada evento se inicia em um determinado momento e dispara uma série de operações de ancoragem alterando-se o número de movimentações executadas e o de embarcações utilizadas. O objetivo do estudo foi avaliar o impacto de fatores como o valor das taxas de aluguel de rebocadores para contratos de curto prazo (spot), os impactos das condições climáticas e a quantidade de navios com contratos de longos prazo (long term). No modelo simulado, o plano de movimentação das unidades, contendo as datas de início de cada tarefa, as origens e destinos das unidades e a quantidade de rebocadores necessários é definido, e são introduzidos elementos estocásticos como as condições climáticas, altura das ondas, tempos de mobilização e desmobilização das

embarcações, duração das tarefas, atrasos devido à quebra e taxas de locação praticadas pelo mercado. A partir da data de início de cada tarefa, a alocação da frota é dada de acordo com a disponibilidade dos navios, dando preferência para os navios cuja locação é de longo período e estão localizados a uma menor distância em relação ao local de execução da tarefa. Foram executadas diversas configurações de frotas, variando-se a quantidade de navios com contratos de longo prazo entre zero e treze para se encontrar o melhor mix de embarcações com contratos spot e longo prazo.

Fagerholt e Lindstad (2000) abordam o problema da determinação da política de eficiência para uma operação de suprimentos no mar Norueguês, em que sete instalações offshore devem ser abastecidas por navios de suprimentos, com diferentes capacidades, partindo de um depósito em terra. A proposta do trabalho foi a de avaliar o efeito no custo total de abastecimento caso uma ou todas as instalações offshore forem fechadas para serviço durante a noite e determinar a política de roteirização. O algoritmo utilizado para determinação da frota e dos correspondentes planejamentos semanais foi feito em duas partes, em que na primeira foi gerado para cada navio de suprimento um número de possíveis rotas viáveis e na segunda, foram determinados os tipos de navios a serem utilizados e os seus roteiros semanais. A metodologia utilizada para a primeira etapa foi a do caixeiro viajante em que cada instalação offshore é visitada no máximo vez em cada roteiro, e devido ao pequeno número de instalações estudadas no problema, ele foi solucionado através de programação dinâmica. Para a segunda etapa foi elaborado um modelo de programação inteira em que a função objetivo é a de minimizar o custo total de utilização dos navios e o tempo total de navegação para a frota. Dos seis senários criados, variando o número de serviços semanais e o tempo de operação de cada instalação offshore, foi concluído que a solução ótima inclui dois navios grandes navegando dois roteiros semanais cada, e cada roteiro atendendo todas as instalações.

Halvorsen-Weare e Fagerholt (2011) utilizaram o modelo criado por Fagerholt e Lindstad (2000) como referência para desenvolver um planejamento de frota mais robusto através da combinação do modelo criado com técnicas de simulação, onde foram introduzidos elementos de incerteza ao problema. No Mar do Norte e no Mar Norueguês os serviços de reabastecimentos de suprimentos são altamente afetados

pelas condições climáticas, especialmente durante a estação do inverno, onde há maior probabilidade da ocorrência de atrasos devido às más condições climáticas. Como no modelo de referência foi utilizada uma aproximação determinística para se chegar à solução, planejadores verificaram que os planejamentos de frotas obtidos pelo modelo determinístico geralmente eram difíceis de serem executados, resultando em custos extras de replanejamento. Isto revelou a necessidade de se criar cronogramas mais robustos e resilientes que considerem as condições climáticas vigentes. Os impactos do clima que foram introduzidos ao modelo foram a altura da onda, em que com ondas acima de 4,5m de altura as embarcações não estão autorizadas a operar (Waiting On Weather WOW), e a variação da velocidade das embarcações, de acordo com o estado do clima. O método de simulação otimização utilizado consiste em 1) gerar todos os candidatos de viagens que as embarcações devem navegar; 2) simular cada candidato de viagem e associar uma medida de robustez; 3) resolver o modelo baseado nas viagens com as medidas de robustez associadas a cada viagem. A considerações de incertezas devido às condições climáticas e alteração da velocidade das embarcações são introduzidas no passo 2 e os passos 1 e 3 foram desenvolvidos conforme o trabalho de referência, onde o passo 1 corresponde à primeira etapa da solução do problema, que corresponde à solução do problema do caixeiro viajante com programação dinâmica, e o passo 3 à segunda etapa, que consiste no modelo de simulação inteira. O estudo também avaliou o impacto de se adicionar ganhos extras se a embarcação não navegar mais de duas viagens na semana e adicionar um tempo de espera entre uma viagem e outra, e o impacto de se adicionar custos extras para cada metro quadrado de carga não entregue, sendo estes impactos adicionados como penalidades na função objetivo do passo 3. Foi constatado que ao se adicionar critérios de robustez ao procedimento de otimização, se obtém um menor custo previsto, se comparado ao caso em que este critério não é considerado.

Halvorsen-Weare et al. (2012) posteriormente, também tratam do problema de abastecimento de suprimentos das plataformas offshore, no qual deve-se determinar a frota ótima de embarcações de suprimentos especializadas e o planejamento de viagens e rotas partindo-se de um depósito em terra. Estas embarcações que transportam suprimentos do depósito para as plataformas são um recurso de custo elevado na cadeia de suprimentos offshore, em que um frete diário de uma destas

embarcações pode custar dezenas de milhares de dólares, o que demanda um bom planejamento e um serviço de suprimentos eficiente. O objetivo do planejamento da frota é de minimizar os custos de locações de embarcações e de tempo de viagem, mantendo-se um nível de serviço confiável. As restrições consideradas no modelo são o horário de operação do depósito e das plataformas; o limite de capacidade de atendimento de embarcações do depósito; a demanda semanal das plataformas, a qual é particionada e deve ser atendida por um determinado número de visitas efetuadas pelas embarcações ao longo da semana; o limite da demanda atendida em cada viagem não deve exceder a capacidade da embarcação; as saídas para viagens partindo do depósito devem ser equiespaçadas ao longo da semana; é definida uma duração mínima e máxima das viagens assim como o número mínimo e máximo de plataformas visitadas em uma viagem. Um conceito abordado neste trabalho é a necessidade de se ter as partidas para viagens igualmente espaçadas ao longo da semana, para garantir um abastecimento o mais uniforme possível de cada plataforma, que foi introduzido no modelo através de 5 restrições à função objetivo. Para isto foi elaborado um método de solução baseado em viagens, que consistiu em duas fases, sendo 1) a geração de todos os candidatos de viagens que as embarcações devem navegar, e 2) a solução do modelo baseado em viagens que escolhe as embarcações com melhor custo benefício e as melhores viagens geradas em 1, que atendam às restrições do problema. O primeiro passo foi solucionado através da solução do caixeiro viajante com múltiplas janelas de tempo, e o segundo passo foi modelado por programação inteira e solucionado através do software Xpress-Optimizer.

2.2 Roteirização de Veículos com Múltiplos Recursos

Os problemas de planejamento e roteirização de veículos, mais conhecidos como VRP (“Vehicle Routing Problems”), segundo Eksiouglu et al. (2009), tem sido matéria para muitos estudos ao longo das últimas décadas por ser um tema que possui grandes desafios intelectuais e por apresentar grande relevância e aplicabilidade aos problemas de logística e transporte. A roteirização de veículos com

múltiplos recursos se configura como uma das classes do problema de roteirização na qual, além de considerar a sincronização dos veículos com os consumidores, que é a característica principal dos VRPs, também considera restrições adicionais de sincronização que dizem respeito à sincronização espacial, temporal e de carregamentos. Ao contrário dos problemas padrões de roteirização, em que os veículos são independentes uns dos outros e a mudança de uma rota não afeta as demais, nos problemas com sincronização de múltiplos recursos existe a interdependência dos veículos, ou seja a mudança de uma rota poderá impactar nas demais rotas, e em uma situação extrema, esta mudança poderá tornar todas as demais rotas inviáveis.

O problema abordado neste trabalho consiste no planejamento de rotas de embarcações de apoio marítimo para a execução de tarefas de suporte a plataformas de extração e produção de petróleo em alto mar. Como as tarefas demandam mais de uma embarcação para serem executadas, surge a necessidade de sincronização das embarcações no que diz respeito a tempo, espaço e características das embarcações. A Figura 2-1 apresenta um exemplo simplificado do planejamento das rotas das embarcações para atendimento às demandas das tarefas. Neste exemplo todas as embarcações (representadas pelas flechas) devem sair da base no início do horizonte de planejamento, e ao final, retornar à base. As tarefas (representadas pelos círculos) podem demandar mais de uma embarcação para serem realizadas, como por exemplo, a tarefa 1 demanda três embarcações, a tarefa 2 demanda apenas uma embarcação, a tarefa 3 demanda duas embarcações e a tarefa 4 também demanda duas embarcações.

A embarcação 1 parte da base para atender a tarefa 1. Após a execução desta tarefa a embarcação parte para a execução da tarefa 4 e depois retorna para a base. A embarcação 2 sai da base, ajuda na execução da tarefa 1, depois segue para o local de execução da tarefa 2, executa esta tarefa e retorna à base. A embarcação 3 também parte da base para ajudar na execução da tarefa 1, depois segue para o local de execução da tarefa 3, aonde a executa em conjunto com a embarcação 4 e depois retorna à base. Por fim, a embarcação 4 parte da base para a tarefa 3, depois se

desloca para a execução da tarefa 4 em conjunto com a embarcação 1, retornando para a base após a execução desta tarefa.

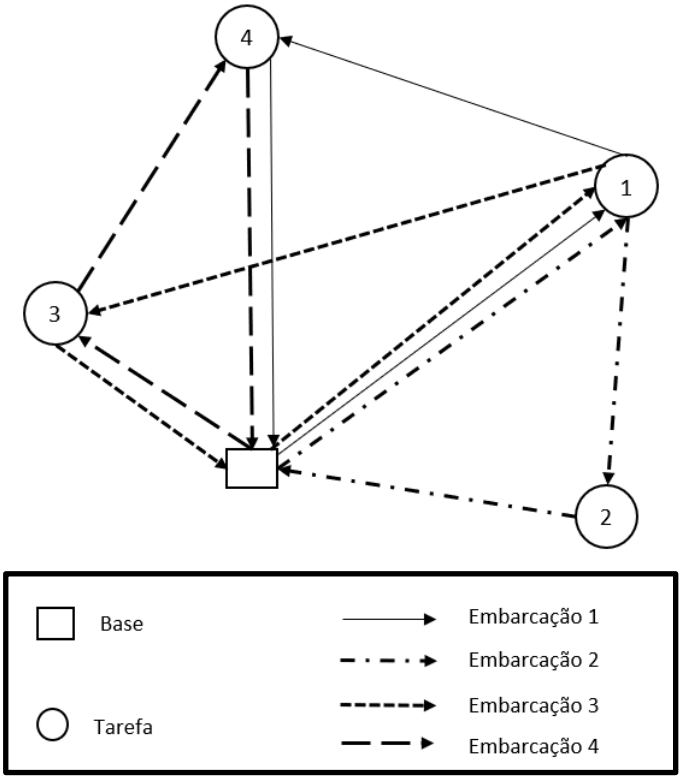

Figura 2-1 – Exemplo de planejamento das rotas das embarcações.

Fonte: Elaborado pelo autor

Um outro exemplo desta variação do problema de roteirização, apresentado por Drexl (2012), consiste no problema de roteirização com trailers e transbordos entre caminhões que contém todos os tipos de sincronização relevantes. O exemplo trata do problema da coleta de leite no sul da Alemanha, em que o leite é coletado nas fazendas e transportado para uma planta de depósito diário todos os dias por uma frota de veículos heterogênea que tem como base um depósito. A frota de veículos é composta por caminhões e cavalos, que são veículos autônomos e podem se deslocar por conta própria, e reboques e semi-reboques que podem se mover no tempo porém precisam ser puxados por veículos autônomos compatíveis, que no caso são os caminhões para os semi-reboques e os cavalos para os reboques. Os caminhões e semi-reboques são equipados para visitar as fazendas e coletar o leite, enquanto os cavalos e reboques podem apenas ser utilizados como veículos de suporte, ou seja, depósitos móveis em que os caminhões e semi-reboques podem transferir a carga em

uma das estações de transbordo. A maioria das fazendas só pode ser visitada por caminhões sem semi-reboques, porém também existem fazendas que podem receber visita do conjunto caminhão + semi-reboque e todas as fazendas devem ser visitadas por apenas um veículo dentro da janela de tempo estipulada. Todos os veículos começam e terminam as suas rotas no depósito central e qualquer veículo não autônomo pode ser puxado ao longo de toda a rota ou em apenas um trecho da rota por um veículo autônomo compatível. Também, qualquer veículo pode transferir a sua carga total ou parcial para qualquer outro veículo, em qualquer estação de transbordo. O problema consiste em se criar rotas de mínimo custo total para todos os veículos em que todo o leite de todas as fazendas seja coletado e entregue ao depósito central. Um exemplo de plano de rotas deste problema de roteirização com trailers e transbordos está apresentado na Figura 2-2. Neste exemplo, são apresentadas as rotas de três caminhões que partem do depósito central, visitam clientes, passam por estações de transbordo para acoplar ou desacoplar um semi-reboque, visitam mais clientes e retornam para o depósito central. Também é apresentada a rota de um cavalo que passa pelas estações de transbordo para se acoplar a um reboque e visitar clientes, retornando para o depósito central ao final de sua rota.

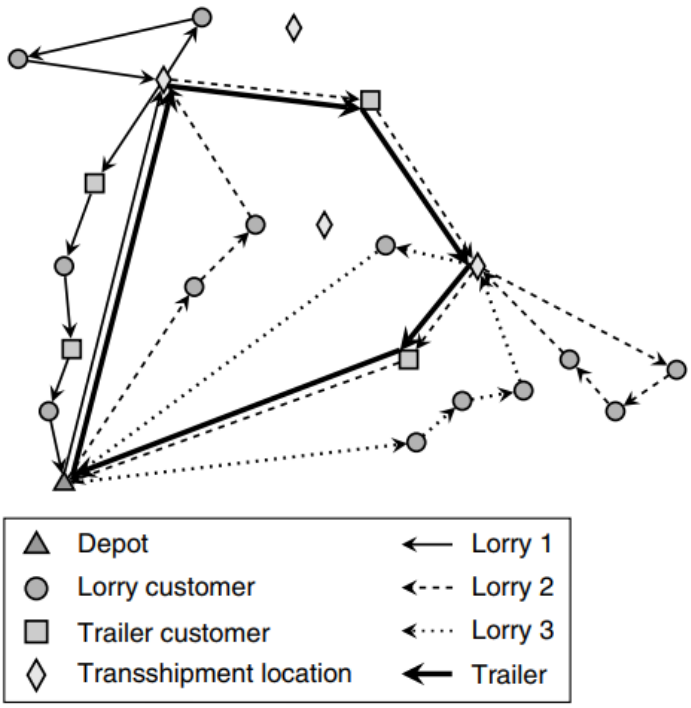

Figura 2-2 – Exemplo de planejamento de rota

Fonte: Dexl (2012)

Os fatores que causam as restrições de sincronização múltiplas em um problema são: (i) possibilidade de particionar a coleta ou entrega no cliente por meio de diversas visitas e diversos veículos; (ii) possibilidade ou necessidade de transbordo de carga ou transferência de pessoas; (iii) necessidade da presença simultânea de veículos em determinado local para executar um serviço; (iv) existência de veículos não-autônomos. Com a adição da sincronização de tarefas nestes fatores de sincronização, os tópicos abordados nas publicações encontradas, tratam de sincronização de carregamentos, recursos, operações, movimento e também da combinação deles. Os trabalhos que utilizaram modelagem em programação inteira mista (MIP) apresentaram diferentes alternativas para o desenvolvimento do modelo, e o espectro de informações, dados e relações que um problema concreto representa variam desde um modelo lógico constituído de variáveis de decisão e restrições até a criação de uma rede altamente elaborada que por si só, assegura a sincronização. Os métodos de solução dos modelos utilizados nos trabalhos pesquisados por Drexl (2012) apresentaram certa recorrência de técnicas algorítmicas, em que para os métodos de solução exata, foram utilizados: discretização do tempo, ramificações nas janelas de tempo (branching), determinação explicita dos requisitos de caminhos e fluxos vinculando-os aos fluxos e caminhos dos veículos, e para as técnicas heurísticas, foram utilizados: decomposição por estágios, determinação explícita dos requisitos de sequenciamento dos trechos, permissão de soluções intermediárias inviáveis e penalização das mesmas na função objetivo, busca indireta e determinação inicial de rotas abstratas e posterior associação a objetos concretos.

Na classe de roteirização de recursos com sincronização de recursos, Hempsch e Irnich (2008) introduziram um problema genérico de roteirização de veículos com restrições entre as rotas, baseado no conceito de representação em rotas gigantes (“giant-tour”) e caminhos com restrição de recursos. As restrições entre rotas são restrições em um problema de roteirização de veículos que apresentam recursos limitados globalmente pelos quais os veículos competem. Alguns exemplos deste tipo de restrição são o número restrito de docas nos depósitos, que limita o serviço nestes locais, e o limitado número de rotas longas, as quais são definidas em função da distância percorrida, do número de paradas, do instante de chegada no depósito, etc. O modelo é baseado em uma plataforma unificada que utiliza a

representação de rotas gigantes e o conceito de caminhos com restrição de recursos, em que são incluídas no modelo, não apenas restrições internas às rotas, como também restrições entre as rotas utilizando-se funções de extensão de recursos (REFs), as quais descrevem a atualização dos recursos ao longo do caminho, isto é, enquanto um veículo viaja de um nó para outro. A dificuldade neste tipo de modelagem é que para um caminho ser viável, não apenas devem ser consideradas as restrições de recursos em uma única rota, como também, as restrições que relacionam todos os conjuntos de rota do problema. O problema foi resolvido através de técnicas de busca local em que procedimentos de processamento de tailor e testes de viabilidade foram combinados em um algoritmo de busca local, as quais se mostraram atrativas para um ponto de vista que considera o pior caso e, na média, foram superiores às técnicas de busca tradicionais.

Uma das classes de problemas de roteirização de veículos com múltiplos recursos contempla a o problema de roteirização com múltiplos estágios. Na logística de transporte de carga, a distribuição da carga pode ser realizada por duas estratégias distintas: a entrega direta e a entrega por múltiplos estágios. Na entrega direta os veículos partem de um depósito e fazem o transporte diretamente para os consumidores finais, enquanto nos sistemas de entrega por múltiplos estágios os produtos são entregues para os consumidores finais, partindo de um depósito principal e passando por depósitos intermediários antes de chegar aos consumidores.

Neste tipo de problema, o objetivo global e garantir a operação do sistema de maneira eficiente e com baixo custo, enquanto a demanda é atendida dentro da janela de tempo disponível e os custos totais de tráfego nas redes é minimizado. A estrutura básica dos problemas de múltiplos estágios pode ser decomposta em $k \geq 2$ estágios:

- O primeiro estágio, que conecta os depósitos com os satélites (depósitos intermediários) de primeiro nível;

- Os $k - 2$ estágios intermediários que interconectam os satélites;

- O último estágio que conecta os satélites aos consumidores finais;

Na linha da roteirização de veículos com múltiplos estágios, Perboli et al. (2008) estudaram o problema de roteirização de veículos capacitados com dois estágios, considerando até 50 consumidores e 4 satélites para testar a efetividade dos métodos utilizados. O problema foi modelado através da representação por “fluxo em rede”, cuja função objetivo visa minimizar a somatória dos custos de viagem e operações de carga e descarga nos depósitos. As restrições aplicadas correspondem ao atendimento da demanda dos consumidores, os requisitos de continuidade de fluxo, locais de início e fim de cada rota e a limitação da quantidade de rotas, de acordo com a quantidade de veículos na frota de cada estágio. Em adição a estas restrições mais comuns para modelos em fluxo em rede, também foram incluídas restrições específicas a este tipo de problema que relacionam o primeiro e o segundo estágio, como: restrições de capacidade dos satélites, de balanço de fluxo nos satélites em que o fluxo é igual à demanda (desconhecida) designada para cada depósito intermediário, assegurando que cada consumidor deve ser atendido por apenas um satélite, e que só permitam que uma rota de segundo estágio se inicie em um satélite, se alguma rota do primeiro estágio atendeu este satélite. Para a solução deste problema foram elaborados dos métodos heurísticos, cujo primeiro método utiliza o procedimento de mergulho, fixando as variáveis em zero e deixando o modelo adaptar o valor das demais variáveis, e o segundo método utiliza programa de resolução de modelos inteiros mistos e seleciona a melhor solução obtida após algumas tentativas. O modelo foi testado com instâncias de benchmarks da literatura de CRVP (“Capacitated Vehicle Routing Problem”), apresentando um bom comportamento para instâncias de pequeno e médio porte.

Prescott-Gagnon et al. (2010) retratam o problema de roteirização de veículos para entrega de óleo, em que um distribuidor fornece um único tipo de óleo para um conjunto de clientes, cujos tanques de armazenagem possuem volumes conhecidos. Para muitos dos clientes, o inventário de óleo é gerenciado pelo distribuidor, o qual, deve garantir, sempre que possível a escassez do produto. O problema da distribuição de óleo é um problema de roteirização que deve ser resolvido diariamente e envolve diversas características de problemas de roteirização de veículos com múltiplos estágios como: frota heterogênea de veículos; múltiplos depósitos de onde podem ser iniciadas as rotas dos veículos; necessidade de abastecimento dos veículos dentro da rota; janelas de tempo para atendimento dos consumidores; turnos de trabalho dos

motoristas, que são janelas de tempo para atendimento dos consumidores, nos casos em que existe uma impossibilidade de acesso do caminhão ao tanque durante o período de trabalho; e dois tipos de consumidores, os consumidores obrigatórios, os quais devem obrigatoriamente ser visitados por algum veículo, e os opcionais, cuja visita, se efetuada, gera um bônus correspondente à aproximação da distância que seria percorrida caso ele fosse visitado nos dias subsequentes. O objetivo do problema é minimizar a distância total, viajada pelos veículos (que está diretamente associada ao custo de deslocamento dos mesmos) menos os bônus adquiridos em visitas a consumidores opcionais. Na solução deste problema foi considerado que os veículos são pré-atribuídos aos motoristas fazendo com que não seja necessária uma restrição de disponibilidade entre veículos e motoristas. A solução do problema foi obtida através de três métodos metaheurísticos: o algoritmo de busca tabu (TS – Tabu Search); a busca em vizinhança (LNS – Large Neighborhood Search) baseada na heurística de busca tabu, e a mesma busca em vizinhança (LNS) baseada na heurística de geração de colunas (CG – Column Generation). O método de busca em vizinhança é um método iterativo que, partindo-se de uma solução inicial, remove elementos da solução corrente (etapa de destruição) e os reinsere (etapa de reconstrução) para encontrar uma solução melhor, e nesta etapa de reconstrução que foram utilizadas as heurísticas de busca tabu e geração de colunas. Os resultados computacionais obtidos de dados reais mostraram que a heurística LNS-CG apresentou os melhores resultados, e a LNS-TS apresentou resultados satisfatórios.

2.3 Métodos Híbridos para Problemas de Roteirização de Veículos

Os trabalhos apresentados a seguir utilizaram métodos híbridos para a resolução de problemas de roteirização de veículos. O método de solução conhecido como híbridos é um algoritmo de otimização que combina elementos de programação matemática exata e algoritmos meta-heurísticos de maneira integrada e cooperativa. Diferentemente dos métodos mais tradicionais para a resolução de problemas de roteirização de veículos, como é o caso dos métodos exatos de programação matemática, os métodos heurísticos e os métodos meta-heurísticos o método híbrido

tem sido aplicado a problemas de roteirização conseguindo obter bons resultados, inclusive em problemas de maior complexidade.

Villegas et al. (2013) utilizaram o método híbrido para a resolução do problema clássico de roteirização de cavalos mecânicos e carretas, em que se tem uma frota composta por m_t cavalos mecânicos e m_r carretas, sendo ($m_r < m_t$) que atendem a uma série de clientes, partindo de um depósito principal. Cada cliente possui uma demanda não negativa e cada cavalo mecânico e cada carreta possui sua capacidade de carga. Alguns clientes possuem espaço limitado para manobras ou acessos estreitos e sinuosos que restringe a passagem de cavalos mecânicos com carretas, fazendo que sejam atendidos apenas por cavalos mecânicos, enquanto outros, que não possuem restrições para a entrega, podem ser atendidos pelo conjunto completo (cavalo mecânico + carreta). Para a resolução do problema, os autores utilizaram a formulação de partição de conjuntos "*set-partitioning*" para o problema em uma heurística híbrida de duas fases. Na primeira fase a heurística preenche um conjunto de rotas com uma solução ótima local obtida através de uma meta-heurística híbrida que contempla GRASP com Busca Local Iterativa (ILS), e na segunda fase as rotas se tornam colunas no problema de partição dos conjuntos, resolvido com um modelo matemático de programação inteira mista (MIP). No processo de construção da solução na primeira fase do algoritmo, as rotas são preenchidas iniciando-se do depósito e em cada iteração um cliente é adicionado à rota. Os clientes mais próximos do ponto em que o veículo se localiza e que não foram atendidos são armazenados em uma lista de clientes candidatos e escolhidos de maneira aleatória para serem incluídos na rota dos veículos. Com esta heurística híbrida de duas fases os autores conseguiram reduzir o tempo de processamento e a diferença entre as soluções mais conhecidas em diversos casos.

Fikar e Hirsch (2015) aplicaram uma heurística híbrida para a resolução do problema de sistemas de transporte para serviços de atendimento de saúde em domicilio, executado por enfermeiras que se deslocam até a casa dos pacientes para atendimento e prestação de serviços. O problema considera a possibilidade de as enfermeiras se deslocarem a pé entre os clientes, interdependências, janelas de tempo e pausas para descanso conforme regulamentação. O objetivo do problema é reduzir o tempo em que os motoristas dirigem e se deslocam e o tempo de trabalho

de todas as enfermeiras, considerando a pausa para descanso. Semelhante às restrições deste presente trabalho, no problema estudado pelos autores Fikar e Hirsch, (2015) os trabalhos exigem determinados níveis de qualificação das enfermeiras para a execução, sendo possível a execução de um trabalho por uma enfermeira mais qualificada do que o requisito exigido no trabalho. Também é utilizado o conceito de janelas de tempo hard para o início da execução dos trabalhos. A heurística desenvolvida para a solução do problema é composta por dois estágios, em que no primeiro estágio são identificados os trechos em que as rotas percorridas a pé são possíveis, criando-se um conjunto de rotas de caminhada viáveis, e, na sequência, através da formulação de partição de conjuntos, são selecionados os conjuntos mais promissores. O segundo estágio parte dos conjuntos das rotas de caminhada e constrói múltiplas soluções iniciais para o planejamento das rotas das enfermeiras e dos veículos através de uma heurística de economia aleatória tendenciosa. Então, é selecionada a melhor solução encontrada, a qual é potencialmente inviável, e se aplica uma busca Tabu para restaurar a viabilidade da solução e otimizá-la. Os resultados desta pesquisa mostraram que o número requerido de veículos poderia ser reduzido substancialmente e que, a utilização de motoristas profissionais, poderia reduzir a pressão por dirigir e estacionar sobre as enfermeiras, que poderiam relaxar durante o percurso e se preparar para o próximo paciente.

Kramer et al. (2015) abordaram o problema de roteirização da poluição (“Polution-Routing Problem PRP”), que se trata de um problema de roteirização de veículos com considerações ambientais, cujo objetivo é minimizar os custos operacionais e ambientais, respeitando restrições de capacidade e janela de tempo. Os custos são baseados nos salários dos motoristas e no consumo de combustível. A particularidade do problema estudado é que a velocidade de deslocamento dos veículos em si é uma variável de decisão, uma vez que a emissão de gases poluentes pelos veículos depende, principalmente, do peso e da velocidade do mesmo. Com isto, o PRP busca encontrar uma matriz de velocidades e um conjunto de rotas para atender a todos os consumidores enquanto minimiza os custos ambientais e operacionais. Os autores utilizaram um método híbrido para a solução deste problema, que consiste na combinação de busca local iterativa, com procedimentos de otimização de velocidade e otimização através de programação inteira sobre uma

formulação de partição de conjuntos. O algoritmo nomeado “Iterated Local Search (ILS) – Set Partitioning (SP) – Speed optimization Algorithm (SOA) é composto das seguintes etapas: 1) Construção da solução inicial através de uma heurística de inserção barata modificada (“modified cheapest insertion heuristic”); 2) Aplicação iterativa da busca local iterativa com três procedimentos de perturbação sobre a solução inicial para minimizar o custo até que não se tenha mais melhoria na solução; 3) Complemento com um procedimento de otimização de velocidade; 4) Utilização de um procedimento de otimização inteira sobre a formulação de partição de conjuntos para gerar possíveis melhores soluções compostas de recombinações de rotas diferentes. Os resultados da aplicação extensiva deste método demonstraram que o algoritmo ILS-SP-SOA é capaz de gerar soluções de alta qualidade de uma maneira bastante consistente.

Mancini e Stecca (2018) estudaram o um caso de roteirização de navios de uma companhia que opera cruzeiros no Mar Mediterrâneo. O planejamento dos itinerários para as companhias de cruzeiros é um problema de decisão estratégico e crítico que afeta a utilização dos recursos e a capacidade da companhia de gerar receita no médio período. O problema de planejamento em questão foi modelado como uma variante de problema de roteirização de veículos considerando restrições específicas como: número fixo de passeios, a visita a todos os nós não é mandatória, é permitida a visita, desde que não sobreposta, a um mesmo nó em diferentes passeios, múltiplas janelas de tempo e possibilidade de escolha de valores de velocidade de viagem. O método de solução proposto foi um modelo misto de programação inteira MIP para minimizar os custos operacionais respeitando as restrições do problema e uma efetiva e eficiente heurística híbrida de busca em grandes vizinhanças “Large Neighborhood Search”. O algoritmo proposto inicia-se de uma solução inicial viável, obtida através da execução do modelo com um pequeno limite de tempo de processamento, impondo que cada nó pode ser visitado apenas uma vez e limitando o número de nós que pode ser selecionado. A melhor solução obtida nesta execução do modelo é a solução inicial do algoritmo. Na sequência, são criados conjuntos de portos, selecionados aleatoriamente, e criadas novas restrições que torna livre a escolha dos navios, de visitar os portos contidos nos conjuntos criados, e então, se roda o modelo resultante da versão com restrições extras com um curto limite de tempo de processamento e se grava a melhor solução. Se a solução

encontrada for melhor do que a melhor solução atual, se define a solução encontrada como melhor solução atual. O mecanismo de parada do algoritmo foi definido como um número máximo de iterações sem melhoria, ou uma quantidade máxima de iterações globais. A heurística e o modelo foram testados com instâncias reais e mostraram que este mecanismo obteve resultados competitivos em termos de eficácia e eficiência.

Akbari e Salman (2017) estudaram o problema de roteirização de arcos para múltiplos veículos sincronizados para restaurar o acesso e conectividade a um local após um desastre. Após um desastre natural as ruas e rodovias podem estar danificadas ou bloqueadas por detritos. Com isto em muitos dos casos o acesso e a comunicação ao local do desastre podem ser comprometidos, então, a primeira e imediata fase no pós-desastre é reunir esforços para desbloquear as vias e restaurar o acesso e a conectividade ao local. Sendo assim, o objetivo da pesquisa desenvolvida pelos autores é prover um método de solução eficiente para gerar e sincronizar o planejamento de trabalho dos times de limpeza e desbloqueio das vias. No problema, os autores otimizaram as rotas dos veículos que fazem a travessia e desbloqueiam um conjunto de vias, encontrando quais caminhos desbloquear e determinando as rotas dos veículos que fazem o desbloqueio, de maneira que as maiores rotas sejam desbloqueadas no menor tempo. Para isto, foi desenvolvido um modelo matemático exato de programação inteira mista e uma heurística híbrida baseada em um modelo MIP relaxado e um algoritmo de busca local. O modelo MIP relaxado resolve uma relaxação do modelo exato em que as condições de tempo entre os veículos são desprezadas. Com isto, uma vez que os parâmetros de tempo são ignorados, as rotas dos veículos podem não estar sincronizadas, e, na próxima etapa foi utilizado um algoritmo de viabilidade que detecta se um veículo atravessa uma via antes do mesmo estar desbloqueado e adiciona um tempo de espera até que a via seja desbloqueada por outro veículo mais próximo, designado para o desbloqueio da via. No terceiro passo da heurística, se utiliza um algoritmo de busca local para melhorar a solução obtida após o algoritmo de viabilidade, que considera mecanismos de remoção, encurtamento e corte das rotas da solução para buscar uma otimização da solução. Com isto, os autores provaram que o gap de otimalidade da solução relaxada é limitado ao número de veículos utilizados.

3 MODELAGEM MATEMÁTICA

Este capítulo apresenta a descrição do problema desta pesquisa e o modelo de programação linear inteira, conforme o trabalho de Mendes (2007). Neste capítulo, o modelo é reproduzido, embora com notação modificada, pelo fato do mesmo ser processado para fins de avaliação do desempenho da heurística proposta.

3.1 Descrição Detalhada do Problema

O problema abordado nesta pesquisa retrata a programação da frota de rebocadores oceânicos, denominados de AHTS (“anchor handling tug supply”), usados em serviços de apoio marítimo da indústria de exploração e produção de óleo e gás no mar. Ritchie (2004) apresenta as características das embarcações e dos subsistemas presentes.

As tarefas desempenhadas por estes rebocadores consistem em uma gama variada de serviços. Tiago (2018) apresenta parte uma lista de serviços comumente realizados pelos AHTS, abaixo transcrita:

i) reboque e na ancoragem de sondas de perfuração, na fase de exploração e desenvolvimento de um campo de petróleo;

ii) reboque e na ancoragem de plataformas de produção de petróleo;

iii) realização de testes para determinar as propriedades do solo em que as âncoras serão instaladas, fornecendo subsídios para o projeto do sistema de ancoragem de unidades de produção;

iv) manutenção e/ou substituição de linhas de ancoragem;

v) lançamento de “torpedos base-guia”, que consistem em torpedos que cravam o leito marinho, facilitando o início do processo de perfuração de uma sonda;

vi) manuseio de linhas de ancoragem de unidades de produção FPSO, que possuem um sistema de amarração concentrado em um único ponto (“turret”);

vii) suporte a operações de lançamento de risers e dutos, que são realizadas pelos navios PLSV (“pipe layer support vessel”);

viii) ocorrência de condições ambientais adversas que colocam em risco as operações de sondas desprovidas de sistemas de ancoragem, que se utilizam de propulsores para mantê-las posicionadas em uma determinada posição - nestes casos, o rebocador tem a função de segurar a sonda até que as condições ambientais severas cessem;

ix) suporte em operações de alívio de FPSO, em que o rebocador deve tracionar um navio aliviador para impedir que haja colisão deste navio com o FPSO.

O horizonte de planejamento considerado é de 30 dias, sendo uma premissa desta pesquisa que a demanda é conhecida. Na ocorrência de novas demandas, o replanejamento do atendimento das tarefas poderá ser feito. As frotas são heterogêneas, sendo a diferenciação mais marcante entre os tipos de embarcações AHTS a sua potência instalada. São encontradas embarcações de 12.000 HP, 15.000 HP, 18.000 HP e 21.000 HP. Apesar da gama variada de embarcações, nesta pesquisa serão adotadas 2 classes distintas de embarcações, sem nenhum prejuízo à generalização ou aplicabilidade dos métodos propostos a um maior número de classes. Por se tratar de um problema de programação, a frota será considerada conhecida, sendo um dado de entrada. Por esta razão, não será considerado o custo do afretamento destas embarcações no horizonte indicado.

As características do problema desta pesquisa são: 1) existe um conjunto conhecido de tarefas que demandam os rebocadores oceânicos tipo AHTS, em um horizonte que varia de 30 a 45 dias; 2) as tarefas possuem instantes de liberação, que indicam o momento em que poderão iniciar (ex: o reboque e a ancoragem de uma plataforma de produção só inicia quando a plataforma é finalizada em um estaleiro); 3) as tarefas necessitam de mais de uma embarcação, eventualmente de classes diferentes, para serem executadas, impondo que as embarcações sejam sincronizadas; 4) como as embarcações de maior potência (isto é, de uma classe superior) são capazes de substituir embarcações de menor potência (isto é, de uma classe superior), será considerada a possibilidade de substituição de uma embarcação por outra de uma classe imediatamente superior; 5) uma tarefa pode iniciar e terminar em locais diferentes (ex: o reboque de uma sonda); 6) cada

embarcação estará disponível no instante e no local em que encerrar a sua última campanha (conjunto de tarefas realizadas no horizonte de planejamento anterior); 7) a demanda deve ser integralmente atendida; 8) cada tarefa possui uma data limite de início, a partir da qual incide uma penalidade proporcional ao atraso (ex: se uma sonda fica aguardando um tempo elevado para ser movimentada, tem-se um custo de oportunidade dado pelo valor do aluguel diário da mesma); 9) diferentemente de Mendes (2007), em que havia um limite superior para iniciar uma operação, nesta pesquisa optou-se por relaxar este limite, tendo em vista que a penalização por atraso será um fator que evitará a postergação do início da tarefa; 10) a função objetivo contempla a parcela das penalidades ponderadas pelo atraso para início, e do custo variável associado ao consumo de combustível para navegação entre as localidades de execução das tarefas; 11) cada embarcação finaliza a sua campanha em um instante de tempo distinto, ficando à disposição no local da conclusão da última tarefa; ao final da execução de todas as tarefas programadas no horizonte de planejamento de cada embarcação, as mesmas devem retornar à base.

3.2 Modelagem Matemática

Para o modelo, foi utilizada a representação clássica de problemas de “fluxo em rede”, em que uma rede orientada R = (V, A), com diversos atributos associados aos seus arcos (custo, distância e tempo) e aos seus nós (demanda por tipo de embarcação, janela de tempo, duração da tarefa e penalidade por atraso) é construída. O objetivo deste modelo é a determinação de um fluxo que atenda à demanda de cada nó da rede com o mínimo custo e respeite às restrições operacionais.

3.2.1 Conjuntos e Índices

A rede utilizada para a representação do problema é definida como R = (V, A), sendo V o conjunto de nós da rede, e A o conjunto de arcos orientados. Os nós da rede são representados pelo conjunto V, dado por $\mathrm{V} = \mathrm{I}^{+} \cup \mathrm{J} \cup \mathrm{I}^{-}$. I^{+} representa o conjunto de origem dos navios, sendo $\mathrm{I}^{+} = \{0_1^{+}, 0_2^{+}, \dots, 0_{\mathrm{m}}^{+}\}$ em que cada elemento equivale ao nó de origem de cada embarcação m, indicando o local onde cada embarcação finalizou a sua última campanha. Por indicarem o posicionamento das embarcações, os elementos de I^{+} possuem índice m, e são referenciados pela notação I^{+}(m). Os elementos do conjunto $J = \{1,2, \dots, \mathrm{n}\}$ representam as tarefas a serem executadas dentro do horizonte de planejamento do problema. O subconjunto $\mathrm{I}^{-} = \{0_1^{-}, 0_2^{-}, \dots, 0_{\mathrm{m}}^{-}\}$ representa os locais de término das campanhas de cada uma das m embarcações, os quais possuem o índice m, e são referenciados pela notação I^{-}(m). Os índices deste conjunto de nós da rede são i, j, m. O conjunto de arcos da rede A, é representado por $\mathrm{A} = (\mathrm{i}, \mathrm{j}) : \mathrm{i}, \mathrm{j} \in \mathrm{V}$. Este conjunto representa todos os arcos, reais e fictícios do modelo. Este conjunto contempla o subconjunto de arcos denominado A1 = {(i, j): i ∈ I^{+}, j ∈ J} que representa o conjunto de arcos que iniciam no nó de origem de cada embarcação, representado pela posição em que cada embarcação finalizou a sua última campanha, e terminam no local de início de uma tarefa da rede. Outro subconjunto contemplado no conjunto de arcos da rede é o A2 = {(i, j): i ∈ J, j ∈ J, i ≠ j}, que é o conjunto de arcos que iniciam no local de término de cada tarefa i e terminam no local de início de cada tarefa j, ou seja, os arcos que unem os “nós de demanda” da rede. Por fim, o terceiro subconjunto do conjunto A é o subconjunto A3 = {(i, j): i ∈ J, j ∈ I^{-}}, que representa os arcos que possuem seu início no local de término da tarefa i e terminam no nó de destino de cada embarcação. Diferentemente do modelo elaborado no trabalho de Mendes (2007) em que este conjunto de arcos era composto por arcos fictícios, com o intuito de auxiliar na modelagem, no presente trabalho, este subconjunto de arcos A3 é composto por arcos que possuem a função de fazer com que as embarcações retornem à sua base ao final do horizonte de planejamento.

As embarcações serão representadas pelo conjunto $\mathrm{M} = \{1,2,\ldots,\mathrm{m}\}$, indexado pelo índice k, e o número total de embarcações, representado por m. Será utilizado o conjunto $\mathrm{C} = \{1,2,\ldots,\mathrm{l}\}$, indexado pelo índice c, para representar cada uma das l classes de embarcações. Também será utilizado o subconjunto $\mathrm{M}^{\mathrm{c}} \subseteq \mathrm{M}$ que engloba apenas as embarcações de uma das c classes. Por fim, por conta da possibilidade de uso de uma embarcação de classe superior, será definido o conjunto $\widetilde{M}^{c}$ contendo os índices das embarcações compatíveis com a classe c, ou seja, contempla todas as embarcações da classe c, como também as embarcações da classe subsequente (c + 1), quando houver.

3.2.2 Parâmetros

Os parâmetros deste problema serão divididos em quatro grupos: referentes às tarefas, referentes às embarcações, referentes ao binômio tarefa – embarcação e referentes aos deslocamentos ou arcos da rede.

Associadas às tarefas, ou nós da rede ($j \in J$) são definidos os seguintes parâmetros:

loc_início(j)	Local de início da tarefa j;
loc_fim(j)	Local de término da tarefa j;
$[a_j^S, b_j^S]$	Janela de tempo 'soft' da tarefa $j \in J$, dentro da qual não há penalização; o valor a_j^S é o instante de liberação da tarefa [dia];
$[a_j^H, -]$	Instante de tempo a partir do qual incide a penalização na tarefa $j \in J$ [dia]. OBS: $a_j^H = b_j^S$
w_j	Penalidade linear por atraso (dentro da janela hard) da tarefa $j \in J$ [$/dia];
$\tilde{t}_j$	Duração da tarefa $j \in J$ [dia];

Os parâmetros relacionados às embarcações são:

classe(k)	Índice da classe à qual a embarcação pertence;
v(k)	Velocidade da embarcação k [milhas/h];
cv(k)	Custo variável diário [$/dia], que engloba o produto do custo do combustível [$/t] pelo consumo diário de combustível [t/dia] na velocidade de serviço (ou de cruzeiro) da embarcação k;
$\bar{a}^k$	Instante de disponibilidade inicial da embarcação k, em função do contrato de afretamento ou do término da campanha anterior;

O parâmetro que relaciona as tarefas às embarcações é a demanda.

d_{jc}	Especifica o número ou a demanda de embarcações da classe $c \in C$ de cada tarefa $j \in J$;

Os parâmetros referentes aos deslocamentos dos navios entre os locais de realização das tarefas, que correspondem aos arcos da rede são o tempo e o custo de deslocamento:

t_{ij}^k	Expressa o tempo de viagem, em dias, entre o local de término da tarefa i e o local de início da tarefa j, para cada arco (i, j) ∈ A, e para cada embarcação k ∈ M. O cálculo do tempo de viagem será expresso por $t_{ij}^k = \mathrm{dist}(i,j)/(24 * v(k))$;

Para os casos em que $i \in I^+$, se $i = I^+(k)$, ou seja o nó i corresponde ao local de origem da embarcação k, o parâmetro de tempo t_{ij}^k será o tempo de deslocamento entre o local de término da última campanha da embarcação k (ou condição inicial da embarcação) e o local de início das tarefas j do problema. Caso $i \neq I^+(k)$, ou seja, se o nó i se referir ao local de início de outra embarcação, o tempo de deslocamento de i para j será igual a +∞. Analogamente, quando $j \in I^-$, se $j = I^-(k)$, ou seja, o nó j corresponde ao local de término da campanha da embarcação k, o parâmetro de tempo t_{ij}^k será o tempo de deslocamento entre o local de término da última tarefa e o local de término da campanha da embarcação, que, por convenção é o local de base

das embarcações. Caso $j \neq I^{-}(k)$, ou seja, se o nó j se referir ao local de término da campanha de outra embarcação, o tempo de deslocamento de i para j será igual a +∞.

c_{ij}^{k}	Expressa o custo para a embarcação k se deslocar entre o local de término da tarefa i e o local de início da tarefa j, para cada arco (i, j) ∈ A, e para cada embarcação k ∈ M. O cálculo do custo de viagem será expresso por $t_{ij}^{k} * cv(k)$;

3.2.3 Variáveis de decisão

Definidos os parâmetros do problema, a seguir serão apresentadas as variáveis de decisão e auxiliares:

x_{ij}^{k}	1, se a embarcação k ∈ M se desloca do local de término da tarefa $i \in I^{+}(k) \cup J$ para o local de início da tarefa $j \in J \cup I^{-}(k)$; 0, em caso contrário;
$\hat{s}_j$	Instante de início da tarefa $j \in J$;
s_{jk}	Instante de chegada do navio k ao local de início da tarefa j;
z_j	Tempo de atraso para o início da execução da tarefa $j \in J$ (corresponde à diferença de tempo entre o momento de início da tarefa j, após a chegada de todas as embarcações necessárias para a execução da tarefa, e o momento desejado para início da tarefa j);

3.2.1 Modelo Matemático

A função objetivo do problema do planejamento da frota de apoio para operações “offshore” é dada por:

$$\min Custo = \sum_{j \in J} z_j w_j + \sum_{i \in V} \sum_{j \in V} \sum_{k} x_{ij}^{k} c_{ij}^{k} \tag{3.1}$$

A equação 3.1, que visa minimizar o custo das operações, é composta das parcelas de custo por penalidade de atraso ao início das tarefas e do custo variável de deslocamento das embarcações entre as tarefas.

$$\sum_{j \in J \cup 0_k^-} x_{0_k^+ j}^{k} = 1 \qquad \forall k \tag{3.2}$$

A restrição 3.2 garante que cada embarcação k ∈ M inicia os serviços saindo do local de sua condição inicial, ou da última campanha, e se desloca para alguma tarefa j ∈ J ∪ I^-.

$$\sum_{j \in 0_k^+ \cup N} x_{j 0_k^-}^{k} = 1 \qquad \forall k \tag{3.3}$$

A restrição 3.3 garante que cada embarcação k ∈ M finaliza os seus serviços no local de término da campanha, partindo de alguma tarefa I^+ ∪ J.

$$\sum_{k \in M^c} \sum_{i \in 0_k^+ \cup J} x_{ij}^{k} \geq d_{jc} \qquad \forall j \in J, \quad \forall c \in C: d_{jc} > 0 \tag{3.4}$$

Esta restrição 3.4 impõe que o número total de embarcações de uma classe c, que chegam ao local de início de uma tarefa j, seja maior ou igual à demanda desta tarefa por embarcações desta classe. A primeira somatória contemplará apenas as embarcações do conjunto M^c, isto é, que são compatíveis com a classe c.

$$\sum_{k \in M} \sum_{i \in 0_k^+ \cup J} x_{ij}^{k} = \sum_{c} d_{jc} \qquad \forall j \in J \tag{3.5}$$

A restrição 3.5 complementa a restrição 3.4, garantindo que o total de embarcações alocadas à uma tarefa j ∈ J seja igual ao total demandado.

$$\sum_{i \in 0_k^+ \cup J} x_{ij}^k = \sum_{i \in J \cup 0_k^-} x_{ji}^k \qquad \forall k, \forall j \in J \qquad (3.6)$$

A restrição 3.6 impõe a continuidade do fluxo, ou seja, assegura que cada embarcação que chegue a um nó, também deixe este local.

$$s_{jk} \geq \bar{a}^k + t_{ij}^k - \left(1 - x_{ij}^k\right)T \qquad \forall k, i = 0_k^+, \forall j \in J \qquad (3.7)$$

A restrição 3.7 define o instante de chegada da embarcação k no local de início da tarefa j, de acordo com o instante de disponibilidade inicial da embarcação e o tempo necessário para deslocamento. A constante T faz com que esta restrição só se aplique às embarcações k que forem alocadas à tarefa j, e pode ser determinada, fixando-se o seu valor igual ao horizonte de planejamento do problema.

$$s_{jk} \geq \hat{s}_i + \tilde{t}_i + t_{ij}^k - \left(1 - x_{ij}^k\right)T \qquad \forall k, \forall i \in J, \forall j \in J \qquad (3.8)$$

Esta restrição 3.8 é complementar à 3.7, definindo o instante de chegada da embarcação k à tarefa j, porém após ter finalizado a tarefa i, ou seja, o instante de chegada da embarcação k à tarefa j deve ser maior ou igual à soma do instante de início da tarefa i, com o tempo de duração da tarefa i e o tempo de deslocamento entre os locais de término da tarefa i e de início da tarefa j.

$$\hat{s}_j \geq s_{jk} \qquad \forall k, \forall j \in J \qquad (3.9)$$

A restrição 3.9 impõe que o momento de início da tarefa j seja maior ou igual ao momento de chegada das embarcações ao local de início da tarefa, ou seja esta restrição assegura que a tarefa j só será iniciada após a chegada de todas as embarcações demandadas por ela.

$$a_j^s \leq \hat{s}_j \qquad \forall j \in J \qquad (3.10)$$

A restrição 3.10 garante que o instante de início da tarefa j deverá ser maior ou igual ao instante início da janela de tempo soft, ou seja, que a execução da tarefa obedeça à janela de tempo para execução da tarefa.

$$z_j \geq \hat{s}_j - a_j^H \qquad \forall j \in J \qquad (3.11)$$

A restrição 3.11 auxilia no cálculo do atraso da tarefa j, sendo maior ou igual ao instante de início da tarefa subtraído do instante desejado para início da tarefa.

$$\hat{s}_j \geq 0, z_j \geq 0 \qquad \forall j \in J \qquad (3.12)$$

$$s_{jk} \geq 0 \qquad \forall j \in J, \forall k \qquad (3.13)$$

$$x_{ij}^k \in \{0, 1\} \qquad \forall k, i \in I^+ \cup J, \forall j \in J \cup I^- \qquad (3.14)$$

As restrições 3.12, 3.13 e 3.14 garantem a não negatividade das variáveis $\hat{s}_j$, z_j, s_{jk}, e garantem também que a variável x_{ij}^k seja binária.

A modelagem de fluxo em rede apresentada é uma variação do modelo clássico encontrado na literatura (Laporte; Osman, 1995), tendo sido acrescentadas as restrições (3.4) e (3.5) referente à necessidade de requisição de múltiplas embarcações. No modelo apresentado, foram excluídas as restrições referentes à imposição do limite superior das janelas hard b_j^H com o intuito de se flexibilizar e simplificar o modelo. Como a partir do início da janela hard a_j^H já existe a incidência de penalidades por atraso do início das tarefas, e o impacto destas penalidades no modelo é bastante representativo, a exclusão das restrições de limite superior das janelas hard deixa o espaço de soluções ampliado, o que permite que ele escolha uma solução com menor custo possível.

4 MÉTODOS DE SOLUÇÃO

Este capítulo tem por objetivo apresentar as *heurísticas de programação linear inteira para resolução de problemas de programação de frota com restrições de sincronização*. O modelo descrito no capítulo anterior consiste na representação matemática do problema desta pesquisa, e serviu como base para o desenvolvimento dos métodos de solução.

4.1 Introdução

O problema desta pesquisa, conforme descrito no capítulo anterior, é representado por um modelo matemático de grande complexidade. Trata-se de um problema combinatório, contendo um grande número de variáveis e algumas restrições que não são usualmente consideradas em problemas correlatos, como o aspecto da requisição simultânea de embarcações e a consequente necessidade de sincronização.

Para solucionar este problema optou-se por desenvolver uma heurística híbrida, composta por um algoritmo de otimização que combina elementos de programação matemática exata e algoritmos de busca local de maneira integrada e cooperativa, e com múltiplos recomeços, em que cada iteração, denominada de global, ocorre uma fase construtiva e uma fase de busca local. Ambas as fases fazem uso de um modelo matemático de programação linear inteira, para fazer a inserção das tarefas nas rotas.

O método de múltiplos recomeços é uma abordagem descrita por Martí et al. (2013), e consiste em um procedimento que constrói uma solução viável através da adição progressiva de elementos da solução, um por vez, quer seja de maneira aleatória ou orientada, até que se tenha uma solução viável para o problema. Um

procedimento de múltiplos recomeços simples aplica o processo de criação de uma nova solução múltiplas vezes, e retorna a melhor solução encontrada nas iterações.

Considerando-se um problema combinatório de otimização definido por um conjunto base finito E = {1, ..., n}, um conjunto de soluções viáveis $F \subseteq 2^E$, e uma função objetivo $f: 2^E \rightarrow \mathbb{R}$. Em sua versão de minimização é buscada uma solução ótima $S^* \in F$ em que $f(S^*) \leq f(S), \forall S \in F$.

Inicializar solução: $S \leftarrow \emptyset$
Inicializar conjunto de candidatos:
$C \leftarrow \{s \in E \setminus S \mid S \cup \{s\}\ não\ inviável\}$
Enquanto $C \neq \emptyset$ faça
Selecionar $s \in C\ de\ maneira\ aleatória$;
Incluir s na solução: $S \leftarrow S \cup \{s\}$;
Atualizar lista de candidatos:
$C \leftarrow \{s \in E \setminus S \mid S \cup \{s\}\ não\ inviável\}$
Fim
Retorna S

Figura 4-1 - Algoritmo 1 – Pseudo-código para Procedimento de Construção de Solução Aleatória

Fonte: Martí et al. (2013)

O Algoritmo 1, presente na Figura 4-1 apresenta um procedimento para a construção de uma solução randômica que inicia com uma solução vazia $S \leftarrow \emptyset$. Em cada passo da construção, seja C um subconjunto de E\S em que, para todo $s \in C$, $S \cup \{s\}$ contém apenas subconjuntos viáveis do conjunto total de elementos. O algoritmo de construção seleciona um $s \in C$ de maneira aleatória e adiciona a S, em que $S \leftarrow S \cup \{s\}$, e atualiza C levando em consideração a inclusão de s em S, terminando quando $C = \emptyset$. O Algoritmo 2, presente na Figura 4-2 apresenta um processo de construção aleatória em um esquema de múltiplos recomeços.

Os métodos de solução propostos neste estudo, aplicados ao problema de programação de frotas com múltiplos recursos alocados às tarefas, se fundamentaram

na metodologia de múltiplos recomeços, em que foi criada uma heurística construtiva em que, a cada iteração da construção da solução, inclui a programação de uma tarefa às rotas das embarcações. Depois de criada uma solução inicial, utilizou-se uma heurística de busca local em que são removidas, e posteriormente reinseridas, um determinado número de tarefas da solução inicial para se buscar uma otimização da solução construtiva. Este algoritmo contemplando uma heurística construtiva e uma, de busca local é executado diversas vezes, selecionando-se a melhor solução encontrada como a solução final, assim como nos métodos de solução de múltiplos recomeços.

$f^* \leftarrow \infty$;
Enquanto o critério de parada não é satisfeito, fazer:
    Construção de solução viável aleatória:
        $S \leftarrow Construção\ de\ Solução\ Aleatória$
    Se $f(S) < f^*$ então:
        $S^* \leftarrow S$;
        $f^* \leftarrow f(S)$;
    Fim
Fim
Retorna S^*;

Figura 4-2 - Algoritmo 2 – Pseudo-código para Algoritmo de Múltiplos Recomeços Aleatório

Fonte: Marti et al. (2013)

Para a inclusão das tarefas nas rotas das embarcações, utilizado tanto na heurística construtiva, como na heurística de busca local, utilizou-se um modelo matemático de otimização orientado pela minimização do custo de inserção de uma tarefa na rota.

4.2 Métodos Propostos

Foram desenvolvidas duas variações de heurísticas, compostas por uma primeira fase construtiva e uma segunda fase de busca local. Observando-se de forma global, a metodologia aplicada ao problema de programação de frota com múltiplos recursos alocados às tarefas se inicia com o carregamento das instâncias do problema, em que são carregados os diversos parâmetros referentes às tarefas a serem programadas, como por exemplo a janela de tempo para execução e a demanda por embarcações, e também parâmetros referentes às embarcações, como por exemplo a classe, o custo de locação e a velocidade de deslocamento das mesmas.

Carregadas as instâncias do problema executa-se a ordenação das tarefas a serem programadas. Neste estudo foram elaboradas duas regras de ordenação das tarefas, sendo a primeira por penalidade, em que as tarefas são ordenadas na ordem decrescente, de acordo com o valor da penalidade por atraso na sua execução, e a segunda por disponibilidade para início, ou seja, cada tarefa possui uma janela de tempo para a sua execução, sendo a_j^S o limite inferior da janela soft, que é equivalente ao instante de liberação da tarefa para execução. Portanto as tarefas são ordenadas na ordem crescente, de acordo com a sua disponibilidade para início, em que as primeiras da fila são as que possuem menor valor de a_j^S e as últimas, que possuem instante de início mais próximo ao final do horizonte de planejamento.

Na sequência inicia-se a fase construtiva da heurística em que é criada, de maneira iterativa uma primeira solução viável para o problema. Nesta etapa do programa são selecionadas, de maneira aleatória, tarefa por tarefa e, as mesmas, incluídas nas rotas das embarcações através do modelo matemático de inserção, até que todas as tarefas sejam programadas e, com isso, a solução inicial seja criada.

Obtida a solução inicial, inicia-se a fase de busca local da heurística, que possui como objetivo refinar e melhorar a primeira solução obtida na fase construtiva do programa. Este mecanismo de busca local desconstrói a solução inicial através da

remoção (segundo um critério pré-determinado) de um determinado número de tarefas das rotas das embarcações e posteriormente as reprograma uma a uma, retornando as tarefas excluídas para a solução inicial, criando-se assim uma nova solução.

Após o refino da solução por meio da heurística de busca local inicia-se a etapa de comparação das soluções obtidas nas iterações. Caso se esteja executando a primeira iteração global, a solução obtida nesta primeira iteração é gravada como a melhor solução obtida no programa $S^* \leftarrow S \; e \; f^* \leftarrow f(S)$. Caso não seja mais a primeira, é feita uma comparação entre a solução obtida na iteração e a melhor solução global obtida. Se houve melhoria na solução, ou seja $f(S) < f(S^*)$, então $S^* \leftarrow S \; e \; f^* \leftarrow f(S)$, e no caso em que não houve melhoria na solução, a solução é descartada e a melhor solução global obtida permanece a mesma.

Por fim, é verificado se a quantidade de iterações já executadas é menor que a quantidade de total de iterações globais definida (critério de parada). Caso positivo, retorna-se à fase construtiva da solução e executam-se todas as etapas novamente para se obter uma nova solução até que a desigualdade deixe de ser verdadeira. Se a desigualdade não é verdadeira, o programa chega ao final, e a melhor solução global registrada é a solução final do programa. Vide Figura 4-3 – Fluxograma da heurística – Visão global para uma ilustração do diagrama de blocos da heurística, na visão global do método de solução.

Esta estrutura global da metodologia de solução é aplicada às duas variações de heurísticas desenvolvidas, em que cada método possui características distintas nas etapas de construção e refino das soluções (fase construtiva e fase de busca local). No Método 1 utilizou-se mecanismos de seleção aleatória para o desenvolvimento das fases de construção e refino das soluções, e no Método 2, utilizou-se o método de seleção aleatória na fase constritiva da heurística e análise de valores de penalidades por atraso das tarefas para a fase de busca local.

Com o entendimento da estrutura global da metodologia de solução desenvolvida, a qual é aplicada às duas variações de heurísticas, na seção seguinte

serão explicados com maiores detalhes cada um dos métodos estudados para este problema.

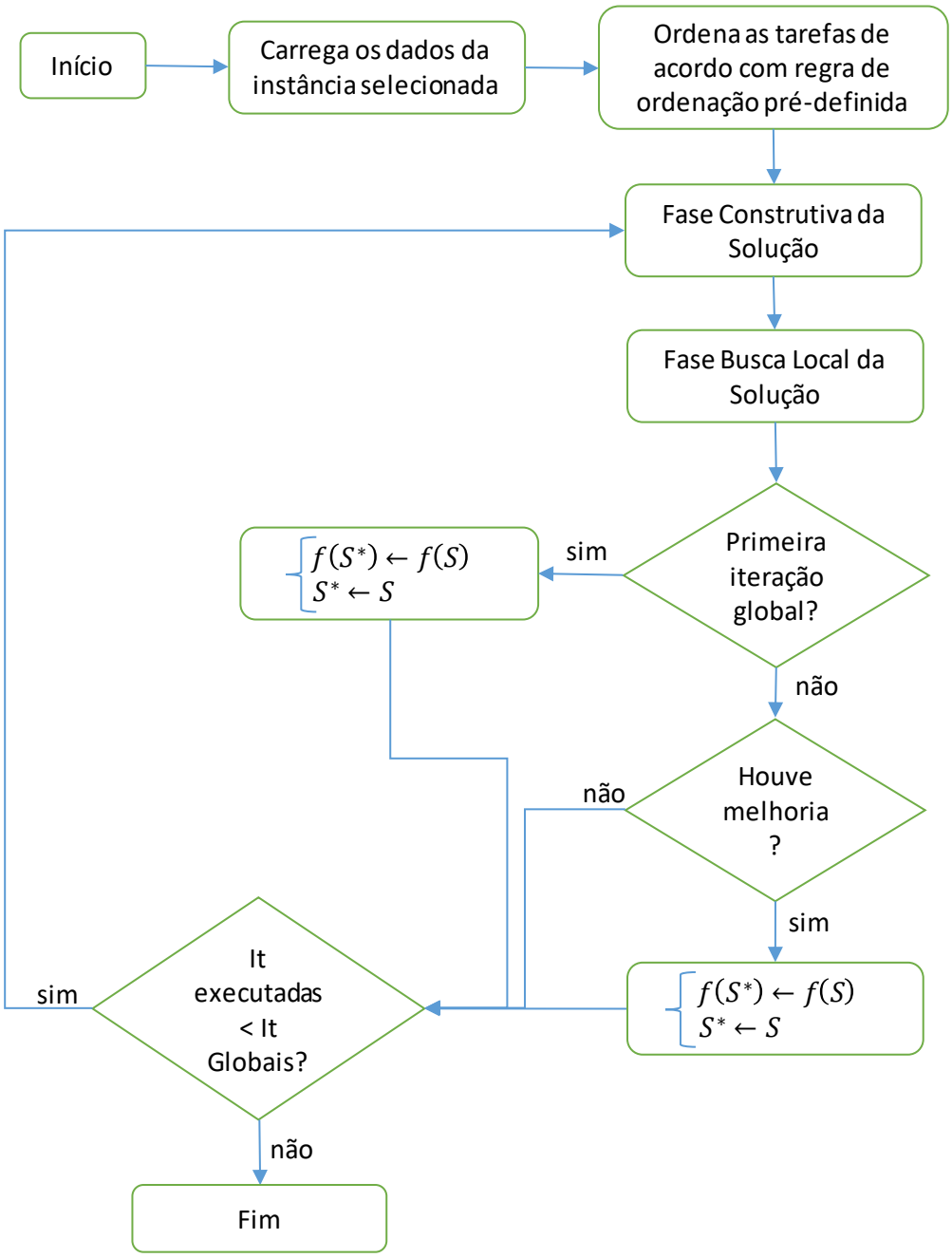

Figura 4-3 – Fluxograma da heurística – Visão global

Fonte: Elaborado pelo autor

4.2.1 Método 1 – Seleção Aleatória

O Método 1 consiste em utilizar mecanismos de geração de números aleatórios no processo de construção e refino das soluções do problema estudado. O algoritmo de geração de números aleatórios utilizado no programa foi o Mersenne Twister (MT), desenvolvido por Makoto Matsumoto e Takuji Nishimura em 1996/1997. Este algoritmo de geração de números aleatórios foi escolhido para utilização na heurística devido à sua rapidez na geração dos números e seu uso eficiente da memória, características que vão de encontro ao objetivo do presente trabalho, que é reduzir o tempo e o custo operacional na busca de soluções viáveis para problemas de alta complexidade, como é o caso do problema de programação de frota com múltiplos recursos alocados às tarefas.

4.2.1.1 Heurística construtiva

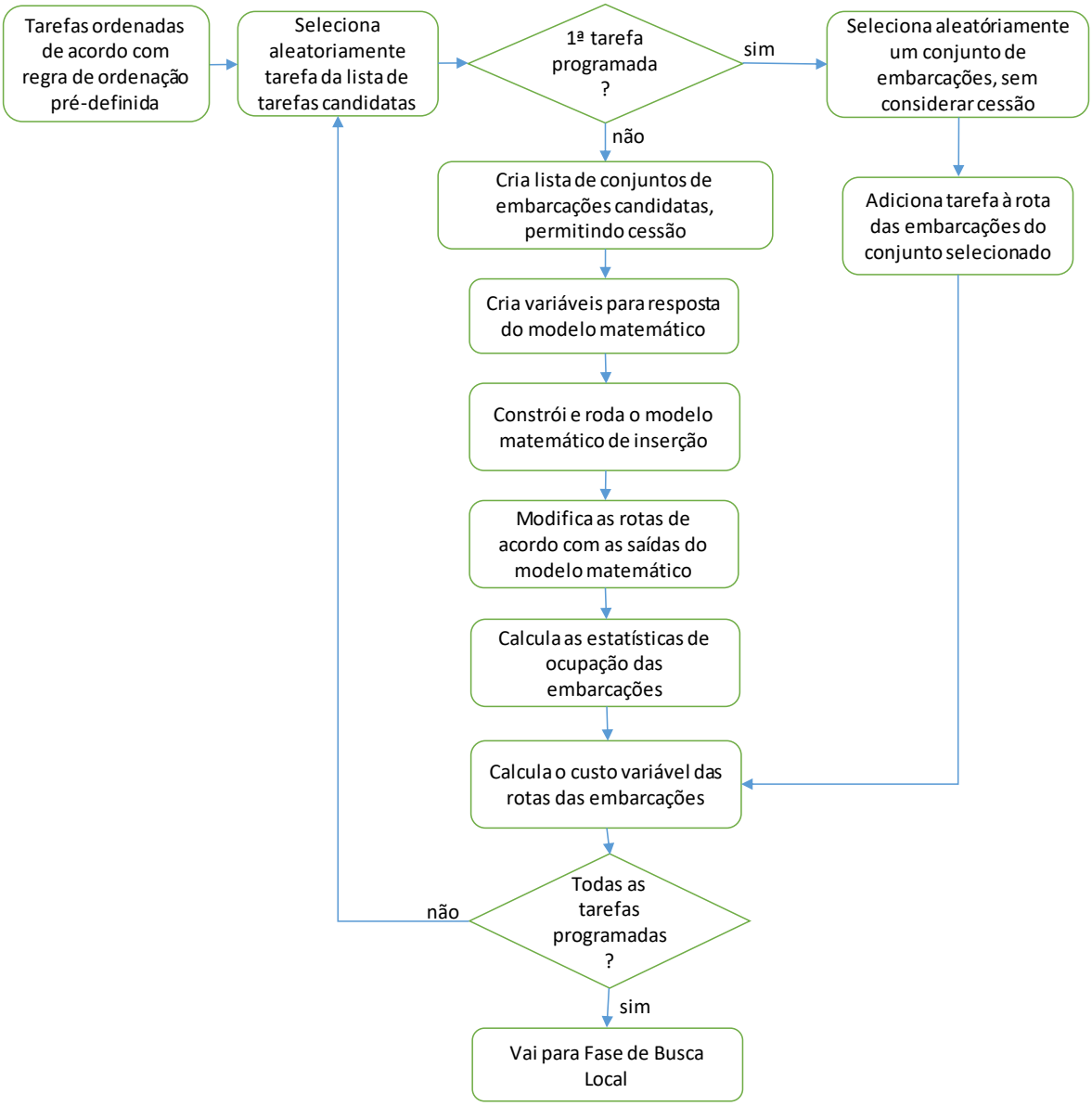

Figura 4-4 – Fluxograma – Heurística fase construtiva

Fonte: Elaborado pelo autor

A partir da estrutura global das heurísticas, uma vez definida e executada a regra de ordenação das tarefas, se inicia a Fase Construtiva da heurística. Como primeiro passo se elabora uma lista das tarefas candidatas que poderão ser selecionadas para serem programadas. A quantidade de tarefas presentes nesta lista de tarefas candidatas é um dado de entrada do programa ("*sampleSize*"), e pode

variar de 1 até n, que é a quantidade total de tarefas a serem programadas. Esta lista é formada pelas primeiras tarefas da lista ordenada de tarefas conforme critério de ordenação, ou seja, se, por exemplo, o valor do dado de entrada "*sampleSize*" for 3, a lista de tarefas candidatas será formada pelas 3 primeiras tarefas da lista. A rotina de elaboração de uma lista com um número determinado de tarefas candidatas na qual uma tarefa é selecionada de maneira aleatória para ser programada e incluída nas rotas das embarcações é semelhante à rotina utilizada por Villegas et al. (2013) na resolução do problema de roteirização de cavalos mecânicos e carretas.

Portanto, como as tarefas estão dispostas de acordo com uma ordenação pré-definida, que pode ser por valor da penalidade ou data de início, quanto menor o tamanho da lista de tarefas candidatas maior a probabilidade de a programação das tarefas seguir a ordenação da lista, e quanto maior o valor da entrada "*sampleSize*", menor a probabilidade da programação das tarefas seguir a ordem da lista, o que faz com que a ordem de programação das tarefas seja mais aleatória. Elaborada a lista de tarefas candidatas, se escolhe de maneira aleatória, através do gerador de números aleatórios MT, a tarefa a ser programada.

Com a tarefa escolhida, o algoritmo verifica se é a primeira a ser programada ou não. Caso seja a primeira, uma vez que todas as embarcações se encontram disponíveis e não há nada programado em suas rotas, adotou-se que não é necessário de se otimizar a inserção da tarefa nas rotas das embarcações. Portanto, para a primeira tarefa a ser programada se verifica a demanda por número e tipo de embarcações para a realização da tarefa. Na sequência, se seleciona de maneira aleatória, um conjunto de embarcações capazes de atender a demanda da tarefa escolhida, não considerando a cessão de embarcações de uma classe para outra.

Os conjuntos de embarcações citados anteriormente são gerados logo no início do programa, após a leitura dos dados de entrada da instância a ser simulada. O processo de criação dos conjuntos se inicia com o levantamento das demandas de todas as tarefas a serem programadas no horizonte de simulação. De posse dos dados de demanda, são criadas todas as possíveis combinações de embarcações capazes de atender a demanda de cada tarefa, com e sem a possibilidade de cessão de embarcações de uma classe para outra.

Por exemplo, uma tarefa, para ser executada, demanda duas embarcações, sendo uma da classe 1 e outra da classe 2. A frota de embarcações disponível é composta de 4 embarcações, sendo 2 da classe 1 e duas da classe 2, ou seja, as embarcações 1 e 2 são da classe 1 e as embarcações 3 e 4 são da classe 2. Portanto, para atender a demanda da tarefa, são criados os seguintes conjuntos de embarcações:

a) conjuntos que não consideram a cessão entre classes: (1;3), (1;4), (2;3), (2;4);

b) conjuntos que consideram a cessão entre classes: (1;3), (1;4), (2;3), (2;4), (3;4);

Com a tarefa a ser executada e o conjunto de embarcações selecionados, se adiciona a tarefa à rota das embarcações que compõem o conjunto e, na sequência, se calcula o custo variável das rotas das embarcações (custos de deslocamento das embarcações até o local de execução da tarefa) através da função “calculateRoutesTotalVariableCost”. Este custo é calculado através das seguintes etapas:

a) cálculo da distância a ser percorrida por cada embarcação, partindo-se do local em que se encontra até o local de execução da tarefa a ser executada.

b) cálculo do tempo de deslocamento de cada embarcação.

c) cálculo do custo variável do deslocamento de cada embarcação do local atual até o local de execução da tarefa.

Após estes passos, é verificado se ainda há tarefas pendentes, a serem programadas, e, caso positivo, a rotina retorna à etapa de seleção aleatória da tarefa a ser executada, da lista de tarefas candidatas.

Uma vez selecionada a nova tarefa a ser programada, é feita a verificação se a tarefa selecionada é a primeira a ser programada. Caso negativo o programa cria uma lista de conjuntos de embarcações candidatas, porém, desta vez, permitindo-se a cessão de classes de embarcações “listOfVesselSet”. Para se elaborar esta lista de conjuntos de embarcações, foi criada uma função “selectVesselSets”, cujos dados de entrada são:

a) dados da instância selecionada;

b) tarefa selecionada a ser incluída na rota das embarcações “selectedJob”;

c) quantidade de conjuntos de embarcações que irá compor a lista “numberOfSets”;

d) verificação se a tarefa a ser programada é a primeira ou não;

e) vetor de saída, contendo a lista dos conjuntos de embarcações candidatas “listOfVesselSet”;

f) Vetor de rotas das embarcações “routes”.

Esta função irá buscar no banco de dados de todos os conjuntos de embarcações, aqueles capazes de atender a demanda da tarefa selecionada. Selecionados os conjuntos capazes de atender a demanda da tarefa, é feito o cálculo do tempo total de operação de cada conjunto através da soma dos os tempos de operação de cada embarcação que compõe o conjunto. Na sequência os conjuntos são ordenados na ordem crescente, de acordo com tempo total de operação.

Então, é verificado se a quantidade total de conjuntos capazes de executar a tarefa é maior do que a quantidade determinada para compor a lista de embarcações candidatas “numberOfSets”. Caso a quantidade total de conjuntos seja menor, então todos os conjuntos são carregados no vetor “listOfVesselSet”, e caso a quantidade total de conjuntos seja maior do que o “numberOfSets”, são carregados apenas os primeiros conjuntos, com menor tempo total de operação, até que a lista de embarcações candidatas seja preenchida.

A quantidade de elementos presentes nesta lista de conjuntos “numberOfSets” foi definida em 100 conjuntos, ou seja, do total de conjuntos capazes de se executar uma determinada tarefa, são selecionados 100 conjuntos para compor a lista de conjuntos de embarcações candidatas. Este número de conjuntos que compõem a lista de conjuntos candidatos é um número fixo, definido manualmente, podendo ser alterado com a finalidade de se estudar o a mudança no comportamento dos resultados.

Quanto maior o número de conjuntos selecionados para compor a lista de embarcações candidatas, aumenta-se a região para a busca de uma solução ideal,

porém também se aumenta o custo de processamento computacional. Como o objetivo do trabalho é o desenvolvimento de uma rotina com baixo custo computacional, optou-se por reduzir o número de conjuntos selecionados para compor a lista e utilizar a rotina de ordenação dos conjuntos de acordo com o tempo total de utilização, que serve como uma ferramenta de pré-seleção dos conjuntos que irão compor a lista de conjuntos candidatos que alimentará o modelo matemático de otimização.

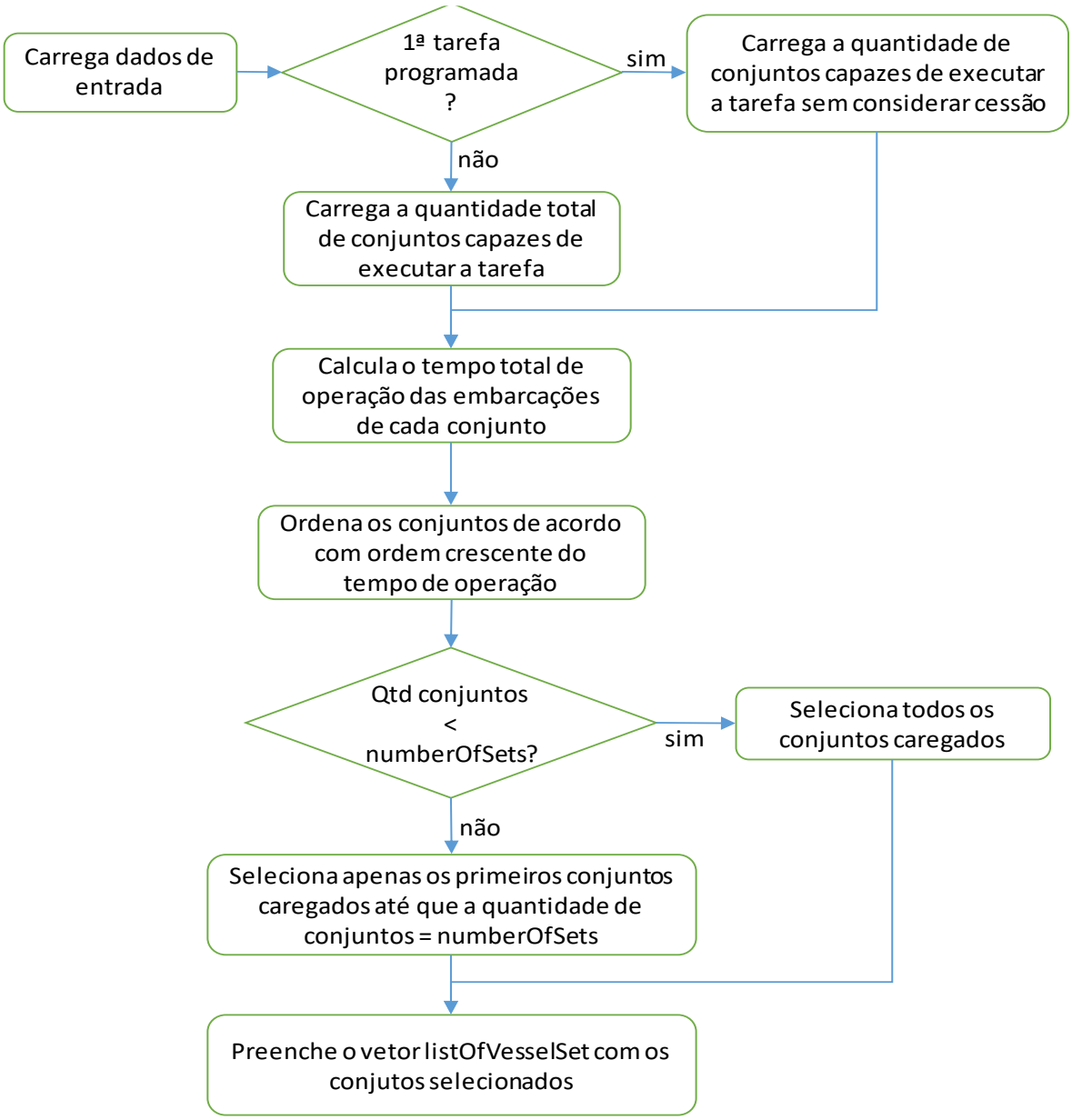

Figura 4-5 – Fluxograma função "selectVesselSets"

Fonte: Elaborado pelo autor

Na sequência, são criadas as variáveis que irão gravar as saídas do modelo matemático, como o vetor de rotas das embarcações, o vetor de instante de término das tarefas e a variável de penalidade total, para a solução encontrada. Então, o modelo matemático de inserção é construído e executado para se buscar uma solução ótima de inclusão da tarefa selecionada à rota das embarcações. Os dados de entrada para a função que carrega o modelo são (função “buildModel”):

a) dados da instância selecionada;

b) tarefa selecionada a ser incluída na rota das embarcações “selectedJob”;

c) lista dos conjuntos de embarcações candidatos para a execução da tarefa “listOfVesselSet”;

d) vetor contendo as rotas das embarcações no momento anterior à inclusão da nova tarefa “routes”;

e) vetor contendo as novas rotas das embarcações, após a inclusão da nova tarefa “modelOutputVesselRoutes”;

f) vetor contendo o instante de término de cada uma das tarefas programadas após a inclusão da nova tarefa “modelOutputJobCompletionTime”;

g) variável contendo o valor total de penalidade por atraso na execução das tarefas após a inclusão da nova tarefa “totalPenalty”.

Com as saídas do modelo, o vetor de rotas das embarcações “routes” é atualizado de acordo com o vetor de rotas após a inclusão da nova tarefa “modelOutputVesselRoutes”. Isto é feito para que, nas próximas iterações para inclusão de tarefas às rotas das embarcações, o vetor “routes” esteja atualizado com todas as tarefas já programadas até o momento.

Atualizadas as rotas, são calculadas as estatísticas das embarcações através da função “updateVesselStatistics”. Os dados de entrada para esta função são:

a) dados da instância selecionada;

b) vetor contendo as rotas das embarcações “routes”;

c) vetor contendo o instante de término de cada uma das tarefas programadas após a inclusão da nova tarefa “modelOutputJobCompletionTime”;

d) módulo “stats” que realiza os cálculos das estatísticas das embarcações e tarefas.

Esta função calcula:

a) o tempo em operação, ou seja, executando cada uma das tarefas programadas em sua rota, que é a somatória do tempo de execução de todas as tarefas executadas pela embarcação;

b) o tempo total no qual a embarcação gasta no mar sem estar executando tarefas: $Instante\ de\ término\ da\ última\ tarefa - tempo\ em\ operação$;

c) tempo de atraso total sujeito a incidência de penalidade:

$$\sum_{i=0}^{n} Instante\ de\ término\ da\ tarefa - instante\ de\ início\ programado - tempo\ de\ execução\ da\ tarefa \tag{4.1}$$

d) custo total de penalidade por atraso, que é a multiplicação do tempo de atraso pelo custo de penalidade para cada uma das tarefas.

Na sequência, são calculados os custos variáveis de deslocamento das embarcações conforme já demonstrado para a inclusão da primeira tarefa selecionada. Então é feita a verificação se ainda existem tarefas pendentes de serem programadas. Caso positivo, se retorna à fase de seleção aleatória de tarefas, e caso todas as tarefas já tenham sido programadas, se finaliza a fase construtiva.

4.2.1.2 Busca local – exclusão aleatória

Finalizada a fase construtiva da heurística e de posse de uma solução inicial viável para o problema de programação de frotas, se inicia a fase de Busca Local para refino da solução inicial. Esta fase de Busca Local consiste em, dada uma solução inicial viável, se remove um determinado número de tarefas das rotas das embarcações. Para a reinserção das tarefas na solução do problema, são criadas todas as possíveis combinações de ordenação das tarefas e são testadas uma a uma para se verificar a melhor solução encontrada. Como este é um algoritmo iterativo, a rotina de se remover as tarefas de uma solução inicial e as reinserir, testando todas

as combinações de reinserção possíveis, é feita repetidas vezes até que se atinja um número pré-definido de iterações em que não são encontradas soluções que apresentaram melhoria no resultado (iterações sem melhoria de resultado). O critério de parada para encerrar a fase de Busca Local é se o número de iterações executadas que não apresentaram melhoria seja maior ou igual ao número pré-definido de iterações sem melhoria.

Nesta etapa de Busca Local serão nomeadas alguns dos principais vetores e variáveis utilizados no código com a finalidade de facilitar o entendimento das diversas etapas do programa. Como esta heurística possui como principal característica métodos iterativos para a obtenção da solução, a definição dos vetores que armazenam os valores de solução de uma etapa para outra facilita a explicação do método.

O algoritmo da Busca Local se inicia com a definição do número de tarefas a serem removidas da solução inicial do problema. Este é um número definido manualmente e foi definido em 3 ou 4 tarefas inicialmente. Na sequência, é criado o vetor “initialRoutes”, que irá armazenar todas as rotas referentes à solução inicial de uma dada iteração da Busca Local. Na primeira iteração o vetor “initialRoutes” é carregado com a solução obtida na Fase Construtiva, que está armazenada no vetor “routes”.

Feito isto, se inicia a rotina iterativa da fase de Busca Local. O primeiro passo é criar o vetor “routesMinusExcludedJobs” que armazenará as rotas das embarcações após a exclusão das tarefas, e, na sequência, se carrega este vetor com as rotas contidas no vetor de solução inicial “initialRoutes”.

Outro vetor criado para armazenar as diversas soluções obtidas em cada permutação, que irá permitir a futura comparação dos resultados, na busca da melhor solução é o vetor “routesComp”. Este vetor irá armazenar as rotas de cada uma das embarcações em cada permutação de reinserção executada. Também, é criado um último vetor de rotas que irá armazenar as rotas das embarcações em cada permutação de reinserção das tarefas “shortRoutes”. Este vetor é possui função similar ao vetor “routes” na fase construtiva.

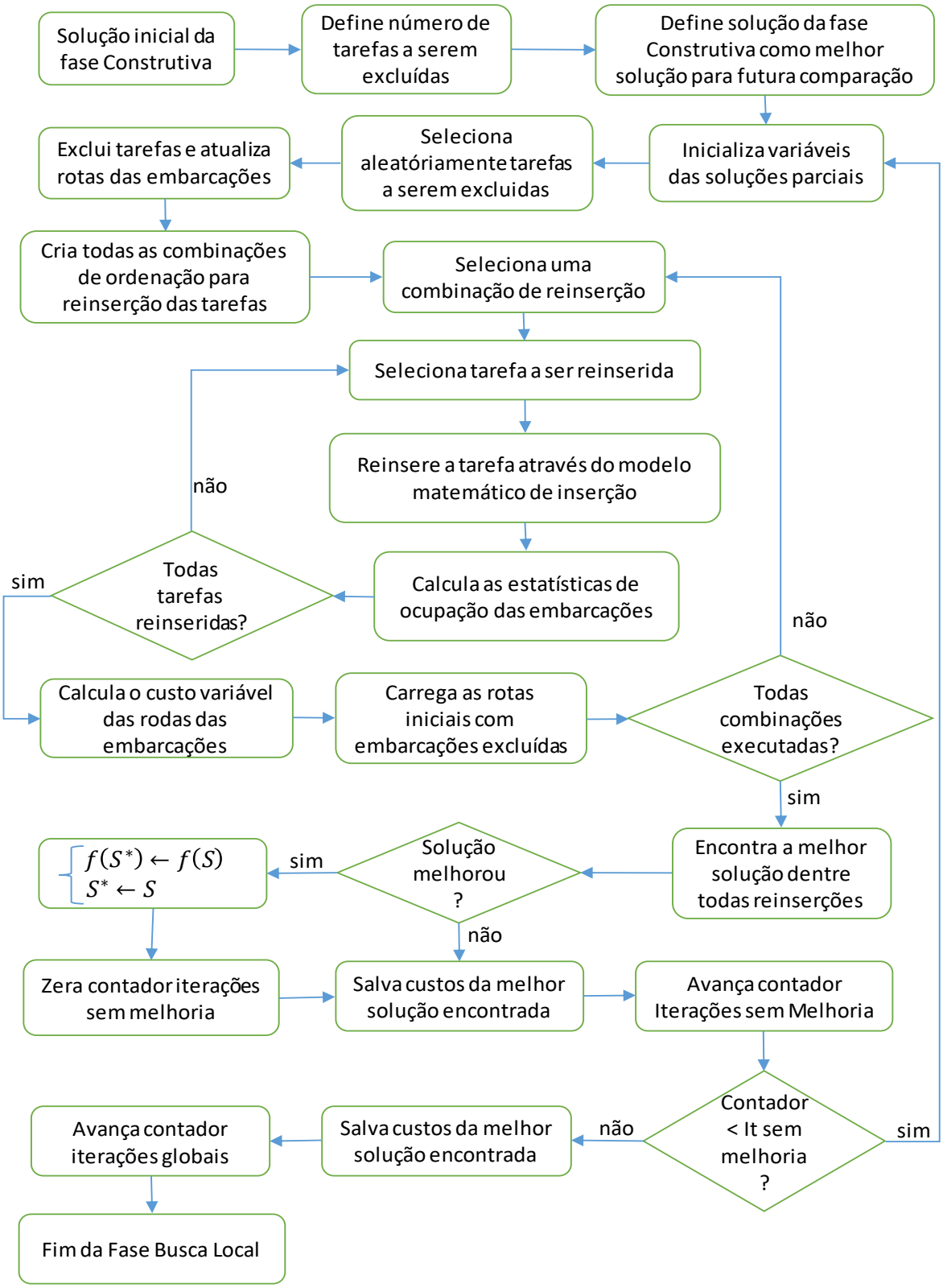

Figura 4-6 - Fluxograma – Heurística Fase Busca Local – Método 1

Fonte: Elaborado pelo autor

Depois disso, é utilizado o algoritmo Mersenne Twister para se selecionar de maneira aleatória as tarefas a serem excluídas da solução base. Escolhidas as tarefas, as mesmas são localizadas e excluídas das rotas das embarcações e é feito um backup das rotas com as tarefas excluídas no vetor “routesMinusExcludedJobs”. Este backup será utilizado ao final da permutação de inserção para carregar as rotas iniciais sem as tarefas excluídas para a próxima permutação. Depois de gravadas as rotas com as tarefas excluídas, as mesmas são carregadas no vetor “shortRoutes”, que irá receber as tarefas reinseridas.

Excluídas as tarefas são criadas as combinações de ordenação das tarefas para reinserção nas rotas das embarcações. Nesta etapa é criado um vetor “removedTaskvector” contendo cada uma das combinações de reinserção das tarefas, ou seja, caso sejam excluídas quatro tarefas, o vetor conterá 4! (4 x 3 x 2 x 1 = 24) combinações de ordenação de reinserção das tarefas.

Criadas as combinações, se entra na rotina iterativa das permutações de inserção das tarefas. Esta rotina se inicia com a partir da primeira combinação de reinserção com a seleção da primeira tarefa desta combinação a ser reinserida. Definida a tarefa se elabora a lista de conjuntos candidatos (função “selectVesselSets”), onde foi mantida a configuração de seleção de, no máximo 1.000 conjuntos, capazes de realizar a tarefa. Na sequência são criados os vetores que armazenarão as respostas do modelo matemático, como “modelOutputVesselRoutes” e “modelOutputJobCompletionTime”.

Para a reinserção da tarefa na rota das embarcações, foi utilizado o modelo matemático do gráfico de inserção (o mesmo usado na fase construtiva), porém, desta vez, o vetor de input das rotas no modelo é o “shortRoutes”, e, desta maneira, a função que chama o modelo matemático fica da seguinte maneira:
buildModel (instance, selectedJob, listOfVesselSet, shortRoutes, modelOutputVesselRoutes, modelOutputJobCompletionTime, totalPenalty)

Então, o modelo matemático de inserção é construído e executado para se buscar uma solução para a inclusão da tarefa selecionada à rota das embarcações.

Após a execução do modelo, o vetor de rotas das embarcações “shortRoutes” é atualizado de acordo com o vetor de rotas de resposta do modelo “modelOutputVesselRoutes”. Atualizadas as rotas, são calculadas as estatísticas de utilização e penalidades por atraso das embarcações através da função “updateVesselStatistics”.

```
Entra no for - até que todas as permutações de ordenação das tarefas sejam
executadas
    Seleciona uma combinação de reinserção
    Entra no for - Até que todas as tarefas removidas sejam reinseridas nas
    rotas das embarcações
        Define a tarefa a ser reinserida na solução
        Elabora lista dos conjuntos de embarcações candidatas, permitindo a
        seção de embarcações
        Cria vetor de rotas temporário para output do modelo matemático
        “modelOutputVesselRoutes”
        Cria vetores de respostas do modelo matemático:
        “modelOutputJobCompletionTime” e "totalPenalty"
        Constrói e roda o modelo matemático de inserção
        Modifica as rotas de acordo com as saídas do modelo:
        “modelOutputVesselRoutes” para “shortRoutes”
        Calcula as estatíticas de ocupação dos navios: função
        “updateVesselStatistics”
    Sai do for
    Calcula o custo variável das rotas das embarcações
    Faz backup das soluções obtidas: "shortRoutes" para "routesComp"
    Carrega as rotas originais, sem tarefas excluídas:
    “routesMinusExcludedJobs” para "shortRoutes"
Sai do for
```

Figura 4-7 - Pseudo código do algoritmo da rotina iterativa de permutações de inserção

Fonte: Elaborado pelo autor

Calculadas as estatísticas, é verificado se todas as tarefas do conjunto selecionado já foram reinseridas nas rotas das embarcações. Caso ainda haja tarefas a serem programadas, o programa retorna ao ponto de seleção da tarefa a ser reinserida e executa todo o processo de reinserção através do modelo matemático. Caso todas as tarefas já tenham sido reinseridas nas rotas das embarcações, e já se tenha uma nova solução para o problema, é feito o cálculo do custo variável das rotas das embarcações, através da função “calculateRoutesTotalVariableCost”.

Então, é feito um backup da solução obtida na permutação e os resultados contidos no vetor “shortRoutes” são carregados no vetor “routesComp”. Com a solução gravada no vetor de comparação, a solução contida no vetor “shortRoutes” é apagada, e o vetor é carregado com as rotas contidas em “routesMinusExcludedJobs”, que são as rotas com as tarefas excluídas, para que o vetor “shortRoutes” fique pronto para a próxima iteração, caso necessário.

Com isto é feita a verificação se todas as permutações de reinserção de tarefas já foram executadas. Caso negativo, o programa retorna à etapa de seleção da combinação de tarefas a ser executada e segue todo o processo de reinserção das tarefas e cálculos de custos variáveis e de penalidades. Caso todas as permutações já tenham sido executadas, o programa sai da rotina de permutações e é iniciada a fase de comparação dos resultados obtidos em cada uma das permutações.

Terminada a rotina de permutações, são comparados todos os resultados obtidos para cada uma das combinações. O melhor resultado é definido como melhor solução obtida na iteração da fase de Busca Local. Na sequência, se compara este resultado com o melhor resultado já obtido nas iterações anteriores (no caso da primeira iteração global, o melhor resultado obtido é a solução da Fase Construtiva da Heurística). Caso seja verificada melhoria na solução da presente iteração, o resultado é gravado como melhor solução obtida e o vetor “initialRoutes” recebe os resultados da melhor solução obtida “routesComp” para que, caso haja uma próxima iteração, a solução base seja correspondente à melhor solução encontrada no momento. Também, o contador de iterações sem melhoria é zerado e são salvos os custos da melhor solução encontrada. Caso não seja verificada melhoria na solução, apenas

são salvos os custos da melhor solução anterior já encontrada, e se avança com o contador de iterações sem melhoria.

Então é feita a verificação se o contador de iterações sem melhorias é menor que número pré-definido de iterações sem melhoria. Caso positivo se retorna à etapa de inicialização de variáveis das soluções parciais para, na sequência, se selecionar de maneira aleatória as tarefas a serem excluídas. Caso o número de iterações sem melhoria seja maior ou igual ao número pré-definido de iterações sem melhoria, são salvos os custos da melhor solução encontrada, o contador de iterações globais é avançado e a fase de Busca Local é finalizada.

4.2.2 Método 2 – Seleção por Penalidade Aplicada à Tarefa

O Método 2 consiste em utilizar mecanismos de geração de números aleatórios no processo de construção da solução inicial do problema e em analisar a penalidade por atraso aplicada a cada tarefa na solução inicial como critério de seleção de tarefas para a exclusão no algoritmo de busca local.

4.2.2.1 Heurística construtiva

A fase construtiva da heurística do Método 2 é igual ao do Método 1 em que se constrói uma solução inicial através da seleção aleatória das tarefas de uma lista de tarefas candidatas e as inclui, uma a uma, nas rotas das embarcações através do modelo de otimização matemática de inserção.

4.2.2.2 Busca local – exclusão por penalidade aplicada à tarefa

Finalizada a fase construtiva da heurística e de posse de uma solução inicial viável para o problema de programação de frotas, se inicia a fase de Busca Local para refino da solução inicial. Esta fase de Busca Local consiste em, dada uma solução inicial viável, se faz uma análise dos valores de penalidade por atraso relativos a cada uma das tarefas e, de acordo com o valor da penalidade, remove um determinado número de tarefas das rotas das embarcações.

Dada uma solução inicial da fase construtiva, se define a quantidade de tarefas a serem excluídas da solução e se utiliza a solução da fase construtiva como a melhor solução encontrada para o problema até o momento. Então inicializa-se as variáveis das soluções parciais e se ordena as tarefas (programadas na fase construtiva) em ordem decrescente, conforme o valor da penalidade por atraso incidida sobre cada tarefa.

O próximo passo, é verificar se houve melhoria na solução. Para o caso da primeira iteração, é definido que houve melhoria na solução. Neste caso, é utilizada a lista de tarefas ordenadas conforme valor da penalidade por atraso para a exclusão das tarefas. Antes de se excluir a tarefa da lista ordenada, é feita a verificação se a tarefa possui penalidade por atraso e, caso positivo, se exclui a tarefa conforme ordenação da lista. Caso todas as tarefas que possuem penalidade por atraso já tenham sido selecionadas e excluídas da lista, as próximas tarefas as serem excluídas são selecionadas de maneira aleatória, dentre as que sobraram na lista, e que não possuem penalidade por atraso.

Durante o processo de desenvolvimento deste método de solução, foi verificado que, para o caso da exclusão apenas das tarefas que apresentaram penalidade por atraso, após a reinserção destas tarefas nas rotas das embarcações e a obtenção de uma nova solução, as tarefas que apresentam penalidade por atraso são as mesmas que as já excluídas anteriormente. Portanto, o algoritmo de busca local iria sempre excluir as mesmas tarefas, o que reduz o espaço amostral de soluções e diminui a eficácia do método.

Com isto, definiu-se que, nos casos em que, através da remoção e reinserção das tarefas que apresentaram maior penalidade por atraso, existe a melhoria no resultado final do problema e a solução base é atualizada, o critério de exclusão das tarefas conforme a penalidade aplicada é mantido. Porém, para o caso em que, após a remoção e reinserção das tarefas conforme penalidade por atraso, não houve melhoria da solução, e a solução base permanece a mesma, o critério de exclusão das tarefas é alterado, e se exclui as tarefas de maneira aleatória para que se aumente o espaço amostral das soluções e o algoritmo possa buscar novas soluções através da exclusão de outras tarefas não excluídas anteriormente.

Após a exclusão das tarefas conforme metodologias de ordenação e seleção indicadas, se atualiza a rota das embarcações, se criam todos os conjuntos de combinações de ordenação das tarefas excluídas para a reinserção, e se entra na rotina iterativa de seleção de um conjunto de ordenação das tarefas e reinserção nas rotas das embarcações através do modelo matemático de otimização. Deste ponto para o final, o Método 2 é igual ao Método 1, já descrito anteriormente.

Na sequência será feita a descrição do Modelo Matemático de Inserção, utilizado nas etapas de construção da solução inicial e de reinserção das tarefas excluídas nas rotas das embarcações.

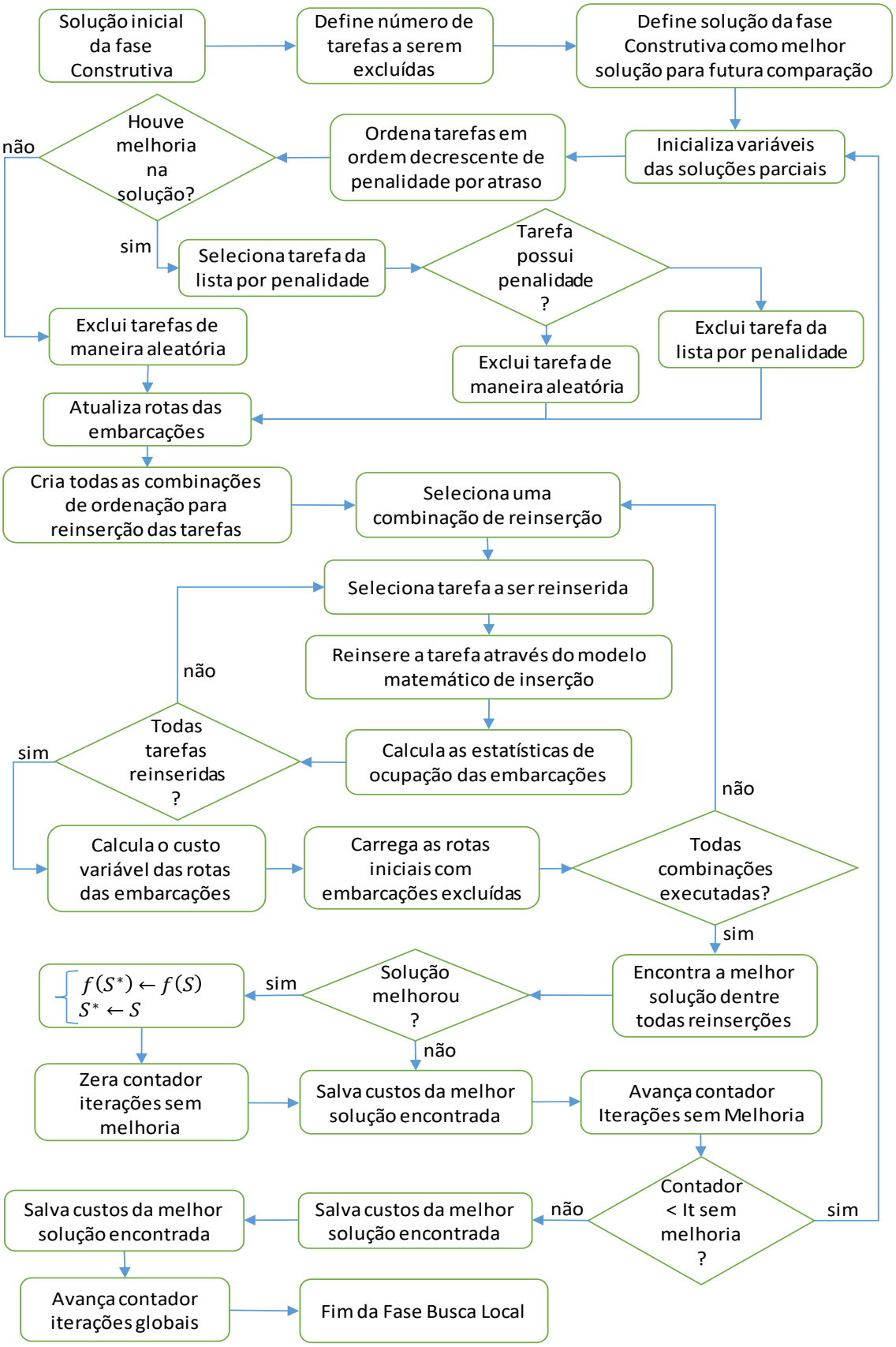

Figura 4-8 - Fluxograma – Heurística Fase Busca Local – Método 2

Fonte: Elaborado pelo autor

4.3 Modelo Matemático de Inserção

Nesta seção será apresentado o modelo matemático de otimização utilizado para as etapas de inserção de tarefas nas rotas das embarcações. Este modelo matemático, embora focado na obtenção da inserção de menor custo, tem como saída a sequência de atividades, similar à saída de um gráfico de Gantt, que é uma ferramenta muito utilizada em gerenciamento de projetos, cujo objetivo é mostrar de maneira bastante visual e de fácil compreensão como se desenvolve o cronograma completo das atividades. Também esse recurso permite que se tenha uma visão do tempo investido em cada tarefa e do prazo total do projeto, identificando interdependências entre as atividades no tempo.

Como o problema estudado se baseia na programação de tarefas nas rotas da frota de embarcações disponíveis em uma determinada janela de tempo, buscou-se inspiração no gráfico de Gantt para o desenvolvimento do modelo matemático. Isso se dá devido ao fato de a função objetivo do modelo retratar a necessidade de se incluir novas tarefas nas rotas das embarcações, identificando a melhor posição para incluir uma nova tarefa nas rotas das mesmas, e minimizando o custo da inserção. A função objetivo visa minimizar o custo da inserção no que tange à incidência de penalidade por atraso da tarefa a ser inserida e das demais tarefas já programadas que poderão ter seu instante de início alterado com a inclusão da nova tarefa na rota, como também, minimizar o custo variável de deslocamento das embarcações de suas posições atuais até o local de execução da tarefa inserida. A seguir, são apresentados os conjuntos e índices, parâmetros e variáveis referentes ao modelo matemático de inserção de tarefas.

4.3.1 Conjuntos e Índices

O conjunto J é formado por todas as tarefas a serem programadas no horizonte de planejamento do problema, indexadas por i e j, sendo o número total de tarefas,

representado por n. O conjunto das tarefas já programadas é representado por $\bar{J}$, sendo $\bar{J} \subseteq J$.

O conjunto de embarcações disponíveis para a realização das tarefas é representado por M, indexado por k, sendo o número total de embarcações representado por m. Conforme já apresentado anteriormente, o problema estudado é um problema envolvendo múltiplos recursos, em que são necessárias múltiplas embarcações simultaneamente para a execução das tarefas. O conjunto das embarcações designadas para a execução de uma determinada tarefa $j \in \bar{J}$ é representado por $M_j \subseteq M$.

Ainda com respeito aos conjuntos das embarcações, o conjunto de embarcações candidatas, capazes de executar a tarefa $\boldsymbol{u} \in J \backslash \bar{J}$ é representado por $M_{\boldsymbol{u}} \subseteq M$. Com o intuito de se reduzir o processamento computacional e tornar o método de solução mais ágil, ao invés de se analisar todos os conjuntos de embarcações capazes de executar uma determinada tarefa, foi criada uma lista que contém apenas os conjuntos de embarcações candidatas, restritos a um número pré-determinado de conjuntos. Esta lista de combinações de embarcações candidatas para a realização da tarefa u é representado por S, indexado por l, cujo número total de conjuntos candidatos é representado pela letra s. A lista das embarcações $k \in M_{\boldsymbol{u}}$ associadas ao elemento l do conjunto S, é representada por S_l. Estas são as embarcações que formam um conjunto de embarcações candidatas, capazes de executar a tarefa u.

Fazendo-se um paralelo com o gráfico de Gantt, as embarcações são alocadas na primeira coluna do gráfico e as linhas correspondentes a cada embarcação representam as suas rotas. O intuito do modelo matemático é incluir as tarefas nas rotas das embarcações de acordo com sua disponibilidade e característica para atendimento aos requisitos de cada tarefa.

Sendo assim, dada uma embarcação $k \in M$, e sua rota com tamanho $|R_k|$, o índice de posições das tarefas na rota da embarcação k é representado por p, sendo $p: 0 \ldots |R_k| - 1$. Considerando a rota da embarcação $k \in M$, o conjunto de posições

possíveis de inserção de uma tarefa na rota da embarcação k é determinado por P_k, sendo $P_k = \{1 \dots |R_k| - 1\}$.

De forma resumida, os conjuntos são:

J	Conjunto de tarefas: $J = \{1,2, \dots, n\}$ (índice i e j)
$\bar{J} \subseteq J$	Conjunto de tarefas já programadas
M	Conjunto de embarcações: $M = \{1,2, \dots, m\}$ (índice k)
$M_j \subseteq M$	Conjunto de embarcações designadas para a tarefa $j \in \bar{J}$
$M_{\boldsymbol{u}} \subseteq M$	Conjunto de embarcações candidatas a receber a tarefa $\boldsymbol{u}$
S	Conjunto de combinações de embarcações candidatas a realizarem a tarefa $\boldsymbol{u}$: $S = \{1,2, \dots, s\}$ (índice l)
S_l	Lista de embarcações $k \in M_{\boldsymbol{u}}$ associadas ao elemento l do conjunto S
p	Índice de posições das tarefas em uma rota (p: $0 \dots |R_k| - 1$)
P_k	Conjunto de posições possíveis de inserção na rota da embarcação k: $P_k = \{1 \dots |R_k| - 1\}$
$\boldsymbol{u} \in J \backslash \bar{J}$	Tarefa a ser inserida (adicionada)
R_k	Rota da embarcação $k \in M$
$|R_k|$	Tamanho da rota da embarcação $k \in M$

4.3.2 Parâmetros

Os parâmetros deste problema são referentes ao binômio tarefa-embarcação. O parâmetro $j(p)$ representa uma tarefa j que ocupa a posição p da rota de uma embarcação e o parâmetro c_{kp} representa o custo de inserção da tarefa $\boldsymbol{u} \in J \backslash \bar{J}$ na posição p da rota da embarcação k.

$j(p)$	Tarefa ocupando a posição p em uma rota
c_{kp}	Custo de inserção da tarefa $\boldsymbol{u}$ na posição p da rota da embarcação k

4.3.3 Variáveis

As variáveis de decisão são divididas em dois grupos, em que o primeiro se refere aos conjuntos de embarcações a serem selecionados z_l, e o segundo grupo se refere à posição de inserção da tarefa $\boldsymbol{u} \in J \backslash \bar{J}$ a ser inserida na rota das embarcações x_{kp}, w_j^0, w_j^1.

z_l	Variável binária que será igual a 1 se a combinação de embarcações $l \in S$ for escolhida, e zero em caso contrário
x_{kp}	Variável binária que será igual a 1 se a inserção da tarefa $\boldsymbol{u}$ ocorrer na posição $p \in P_k$ da rota da embarcação k, e zero em caso contrário
w_j^0	Variável binária que será igual a 1 se a tarefa $\boldsymbol{u}$ for inserida imediatamente antes da tarefa j, e zero em caso contrário
w_j^1	Variável binária que será igual a 1 se a tarefa $\boldsymbol{u}$ for inserida imediatamente após a tarefa j, e zero em caso contrário

As variáveis auxiliares armazenam os instantes de término de execução das tarefas nas rotas das embarcações e os tempos de atraso para a conclusão das mesmas.

C_{jk}	Instante de término da tarefa j pelo navio k
$\bar{C}_j$	Instante de término da tarefa j
T_j	Atraso para conclusão da tarefa j

4.3.4 Modelo

A função objetivo do modelo matemático de inserção de tarefas nas rotas das embarcações é dada por:

$$\min C(u) = \sum_{j \in \bar{J} \cup \boldsymbol{u}} T_j w_j + \sum_{k \in M_{\boldsymbol{u}}} \sum_{p \in P_k} x_{kp} c_{kp} \tag{4.2}$$

A função objetivo apresentada na equação (4.2) refere-se ao custo da inserção, que deverá ser minimizado, e contempla, na primeira parcela o atraso para início das tarefas ponderado pelas respectivas penalidades diárias, e, na segunda parcela, o custo de inserção da tarefa $\boldsymbol{u}$ nas rotas das embarcações selecionadas.

$$\sum_{l \in S} z_l = 1 \tag{4.3}$$

A restrição (4.3) força que apenas uma combinação de embarcações seja necessariamente escolhida para realizar a tarefa $\boldsymbol{u}$.

$$\sum_{p \in P_k} x_{kp} \geq z_l \qquad \forall l \in S, \forall k \in S_l \tag{4.4}$$

A restrição (4.4) faz a ligação entre as variáveis x_{kp} e z_l, e impõe que na rota de cada embarcação pertencente à combinação l pelo menos 1 posição de inserção seja escolhida.

$$C_{j(p)k} \geq \max\left(\bar{a}^k + t^k_{0^+_k j(p)}, a^S_j\right) + \tilde{t}_{j(p)} \qquad \forall k \in M \backslash M_{\boldsymbol{u}} \quad p = \{1\} \tag{4.5a}$$

$$C_{j(p)k} \geq \bar{C}_{j(p-1)} + t^k_{j(p-1)j(p)} + \tilde{t}_{j(p)} \qquad \forall k \in M \backslash M_{\boldsymbol{u}} \quad p \in P_k\{1\} \tag{4.5b}$$

As restrições (4.5a) e (4.5b) referem-se às rotas das embarcações que não são candidatas a receber a tarefa $\boldsymbol{u}$, independentemente de qual combinação l for escolhida. Apesar disto, as tarefas nelas contidas poderão ter os seus instantes de início modificados.

Na restrição (4.5a) é considerado o instante de término da primeira tarefa da rota da embarcação $\forall k \in M \backslash M_{\boldsymbol{u}}$ (tarefa $j(1)$), sendo este maior ou igual do que, ou o instante de liberação da embarcação k somado ao tempo de deslocamento da embarcação até o local de realização da tarefa, ou o instante de liberação da tarefa para a execução, somados ao tempo de execução da tarefa.

Na restrição (4.5b), o término de cada tarefa $j(p)$ pela embarcação k é dependente do instante de conclusão da tarefa na posição anterior $j(p-1)$ somado ao tempo de deslocamento entre os locais de execução das tarefas e a duração da tarefa $j(p)$.

$$C_{j(p)k} \geq \bar{a}^k + t^k_{0^+_k j(p)} + \tilde{t}_{j(p)} - x_{kp}T \qquad \forall k \in M_{\boldsymbol{u}} \quad p = 1 \qquad (4.6a)$$

$$C_{j(p)k} \geq \bar{C}_{\boldsymbol{u}} + t^k_{\boldsymbol{u}j(p)} + \tilde{t}_{j(p)} - (1 - x_{kp})T \qquad \forall k \in M_{\boldsymbol{u}} \quad p = 1 \qquad (4.6b)$$

$$C_{\boldsymbol{u}k} \geq \bar{a}^k + t^k_{0^+_k \boldsymbol{u}} + \tilde{t}_{\boldsymbol{u}} - (1 - x_{kp})T \qquad \forall k \in M_{\boldsymbol{u}} \quad p = 1 \qquad (4.6c)$$

$$C_{j(p)k} \geq \bar{C}_{j(p-1)} + t^k_{j(p-1)j(p)} + \tilde{t}_{j(p)} - x_{kp}T \qquad \forall k \in M_{\boldsymbol{u}} \quad p \in P_k \backslash 1 \qquad (4.6d)$$

$$C_{j(p)k} \geq \bar{C}_{\boldsymbol{u}} + t^k_{\boldsymbol{u}j(p)} + \tilde{t}_{j(p)} - (1 - x_{kp})T \qquad \forall k \in M_{\boldsymbol{u}} \quad p \in P_k \backslash 1 \qquad (4.6e)$$

$$C_{\boldsymbol{u}k} \geq \bar{C}_{j(p-1)} + t^k_{j(p-1)\boldsymbol{u}} + \tilde{t}_{\boldsymbol{u}} - (1 - x_{kp})T \qquad \forall k \in M_{\boldsymbol{u}} \quad p \in P_k \backslash 1 \qquad (4.6f)$$

As restrições (4.6a) a (4.6f) modelam a inserção da tarefa $\boldsymbol{u}$ na rota das embarcações candidatas, e permitem reprogramar o instante de início das tarefas subsequentes à posição p de inserção.

Na restrição (4.6a), se a inserção de $\boldsymbol{u}$ ocorrer na posição 1 da rota da embarcação k, então o instante de término da tarefa que ocupava a posição 1 ficará desvinculado do instante de liberação da embarcação k. Em caso contrário, a vinculação estará mantida.

Na restrição (4.6b), havendo a inserção da tarefa $\boldsymbol{u}$ na posição 1, a tarefa que ocupava esta posição terá o seu instante de término recalculado, e será dependente do instante de término da tarefa $\boldsymbol{u}$ somado ao tempo de deslocamento entre os locais das tarefas $\boldsymbol{u}$ e $j(p)$ e a duração da tarefa $j(p)$. Se a inserção não ocorrer, a vinculação entre $j(p)$ e $\boldsymbol{u}$ ficará desativada.

Por último, se a inserção da tarefa $\boldsymbol{u}$ ocorrer na posição 1, então o instante de término da tarefa $\boldsymbol{u}$ dependerá do instante de liberação da embarcação k somado ao tempo de posicionamento da embarcação entre o local de início e o local de execução da tarefa. As restrições (4.6d) a (4.6f) fazem o papel das restrições (4.6a) a (4.6c), só que considerando as demais posições possíveis de inserção.

$$\bar{C}_j \geq C_{jk} \qquad \forall j \in \bar{J}, \forall k \in M_j \qquad (4.7)$$

A restrição (4.7) garante que o instante de término da tarefa j seja maior que o último instante de término dentre todas as embarcações que realizaram a tarefa, e força a sincronização entre as embarcações.

$$\bar{C}_{\boldsymbol{u}} \geq C_{\boldsymbol{u}k} \qquad \forall k \in M_{\boldsymbol{u}} \qquad (4.8)$$

A restrição (4.8) força a sincronização do instante de término entre as embarcações que realizaram a tarefa $\boldsymbol{u}$.

$$T_j \geq \bar{C}_j - \tilde{t}_j - a_j^H \qquad \forall j \in \bar{J} \cup \boldsymbol{u} \qquad (4.9)$$

A restrição (4.9) calcula o atraso de cada tarefa programada através da comparação entre o instante de término da tarefa e a soma do tempo de execução e o limite inferior da janela de tempo hard (que corresponde ao instante máximo no qual a tarefa pode ser iniciada, sem a incidência de penalidade por atraso).

$$C_{jk} \geq a_j^S + \tilde{t}_j \qquad \forall j \in \bar{J}, \forall k \in M_j \qquad (4.10)$$

A restrição (4.10) força que o instante de término de uma tarefa por uma embarcação leve em consideração o instante de liberação da tarefa e o seu tempo de execução.

$$\sum_{k \in M_{\boldsymbol{u}}} \sum_{p \in P_k} x_{kp} = \sum_{c} d_{uc} \qquad (4.11)$$

A restrição (4.11) indica que o número de inserções possíveis nas diversas rotas será igual à demanda total da tarefa $\boldsymbol{u}$**,** por embarcações de todas as classes.

$$w_j^0 + w_j^1 \leq 1 \qquad \forall j \in \bar{J} \qquad (4.12)$$

A restrição (4.12) indica que a tarefa $\boldsymbol{u}$ será eventualmente inserida imediatamente antes ou imediatamente após a tarefa j, ou não estará diretamente ligada a ela.

$$x_{kp} \leq w_{j(p)}^0 \qquad \forall k \in M_{\boldsymbol{u}} \quad p = 1 \qquad (4.13a)$$

$$x_{kp} \leq w_{j(p-1)}^1 \qquad \forall k \in M_{\boldsymbol{u}} \quad \forall p \in P_k \backslash \{1 \cup |R_k| - 1\} \qquad (4.13b)$$

$$x_{kp} \leq w_{j(p)}^0 \qquad \forall k \in M_{\boldsymbol{u}} \qquad (4.13c)$$

$$\forall p \in P_k \backslash \{1 \cup |R_k| - 1\}$$

$$x_{kp} \leq w^1_{j(p-1)} \qquad \forall k \in M_{\boldsymbol{u}} \qquad p = |R_k| - 1 \tag{4.13d}$$

No caso da tarefa $\boldsymbol{u}$ ser inserida na primeira posição, a restrição (4.13a) forçará a variável $w^0_{j(p)}$ ser igual a 1, indicando que a tarefa $\boldsymbol{u}$ foi inserida antes da tarefa que ocupava a posição 1. No caso da tarefa $\boldsymbol{u}$ ser inserida em qualquer posição p que não seja nem a posição 1 e nem a última da rota, então $\boldsymbol{u}$ será sucessor imediato da tarefa que ocupa a posição $p - 1$ e antecessor imediato da tarefa que ocupava a posição p, forçando ambas as variáveis, respectivamente $w^1_{j(p-1)}$ e $w^0_{j(p)}$, serem iguais a 1, conforme as restrições (4.13b) e (4.13c). Por fim, se a inserção da tarefa $\boldsymbol{u}$ ocorrer na última posição da rota, então só será tratado o caso de vinculá-la como sucessor imediato da tarefa que ocupa a posição $p - 1$, já que a tarefa final de cada rota sempre será a tarefa fictícia $n + 1$, conforme indicado na restrição (4.13d).

$$C_{jk} \geq 0, \bar{C}_j \geq 0, T_j \geq 0 \qquad \forall j \in \bar{J} \cup \boldsymbol{u} \qquad \forall k \in M \tag{4.14a}$$

$$z_l \in \{0,1\} \qquad \forall l \in S \tag{4.14b}$$

$$x_{kp} \in \{0,1\} \qquad \forall k \in M_{\boldsymbol{u}}, \forall p \in P_k \tag{4.14c}$$

$$w^0_j \in \{0,1\}, w^1_j \in \{0,1\} \qquad \forall j \in \bar{J} \tag{4.14d}$$

As restrições (4.14a) a (4.14d) tratam do domínio das variáveis de decisão.

5 APLICAÇÃO DOS MÉTODOS DE SOLUÇÃO

Neste capítulo serão apresentados os resultados obtidos pelas *Heurísticas de programação linear inteira para resolução de problemas de programação de frota com restrições de sincronização*. Inicialmente será apresentada a base de dados dos problemas utilizada e, em seguida, os resultados obtidos pela aplicação de cada método aos problemas serão apresentados e discutidos.

A característica principal dos métodos desenvolvidos neste trabalho é a de solucionar um problema complexo, com um tempo de processamento admissível no contexto do processo de tomada de decisão, para instâncias de pequeno e médio porte. Para controlar os tempos limites de tempo de processamento, alguns parâmetros dos métodos precisam ser calibrados. Tanto para o Método 1 - Seleção Aleatória, como para o Método 2 - Seleção por Penalidade das Tarefas, os principais parâmetros e regras são:

- Regra de ordenação das tarefas na fase na construtiva da heurística: i) penalidade diária decrescente; ii) instante de liberação crescente;
- Quantidade de iterações globais;
- Quantidade de tarefas para compor a lista restrita de candidatas;
- Quantidade de iterações sem melhoria;
- Quantidade de conjuntos de embarcações candidatas que serão consideradas no modelo de otimização para inserção de tarefas nas rotas das embarcações;
- Quantidade de tarefas excluídas da solução inicial que, posteriormente, serão reinseridas nas rotas das embarcações.

Para analisar o impacto da variação de cada um dos parâmetros de entrada do algoritmo, cada parâmetro foi variado mantido fixos os demais valores. Os resultados obtidos em cada rodada são analisados tanto no que se refere ao valor da função objetivo, como no tempo de processamento. Como o critério de parada dos métodos não está relacionado ao tempo de processamento, mas sim à quantidade de iterações de remoção e reinserção de tarefas nas rotas das embarcações, os valores dos parâmetros utilizados foram definidos de maneira que, preferencialmente, não se tenha períodos de processamento muito longos.

5.1 Base de Dados

A base de dados das instâncias utilizadas para a aplicação dos métodos de solução foi semelhante à base de dados utilizada na pesquisa de Mendes (2007), porém com uma redução da frota usada em algumas das instâncias, visando elevar a ocupação média da frota.

As instâncias utilizadas nos testes computacionais, possuem os seguintes itens como parâmetros de entrada:

- *Número de tarefas a serem programadas* – representa a quantidade total de tarefas a serem programadas no horizonte de programação do problema;
- *Demanda por embarcações* – representa a demanda por embarcações e por classe de embarcações, para cada tarefa;
- *Horizonte de planejamento* – representa a extensão ou período total de planejamento [dias];
- *Duração das tarefas* – representa a duração das tarefas [dias];
- *Instante de liberação da tarefa* – representa o instante a partir do qual a tarefa pode ser programada;
- *Janela de tempo soft da tarefa* – representa o período no qual a tarefa pode ter seu início sem que haja incidência de penalidade por atraso, ou seja, quando o início ocorrer entre o instante de liberação e o limite superior da janela de tempo *soft*;
- *Penalidade por atraso* – representa o valor da penalidade por atraso no início de uma tarefa [$/dia];
- *Coordenadas* – cada tarefa possui coordenadas referentes aos locais de início e término da atividade que podem, eventualmente, ser distintas. A representação do problema requer a definição da faixa de coordenadas válidas para geração dos diversos locais onde ocorrerão as tarefas;
- *Número de embarcações* – representa a quantidade de embarcações disponíveis em cada instância para a realização das tarefas, segmentadas por classe;
- *Custo variável por dia de utilização da embarcação* – representa o custo por dia de utilização de cada embarcação [$/dia];
- *Velocidade de navegação* – representa a velocidade de deslocamento de cada embarcação;

A base de dados utilizada como *input* dos métodos é composta por 24 instâncias de teste sendo as primeiras 12 instâncias de 10 tarefas e as últimas 12, de 20 tarefas. A Tabela 5-1 indica os parâmetros e os respectivos valores empregados para cada uma das instâncias. As colunas da tabela representam: número da instância de teste; número de tarefas por instância (n); número de embarcações da instância (m); a extensão do horizonte de planejamento (H); o tempo de duração médio das tarefas (Dt); o valor da demanda média de embarcações por tarefa (Net); e a ocupação média das embarcações. Os demais parâmetros utilizados, porém não apresentados na Tabela 5-1 foram: o número de classes de embarcações, fixo em 2; a faixa de variação da duração das tarefas, variando entre 2 e 22 dias; a faixa de variação das penalidades diárias, variando entre R$ 0,00 e R$ 320.000,00; a probabilidade de um local de início ser o mesmo local do fim da tarefa, igual a 80%; e a faixa de variação das coordenadas x e y, dos locais das tarefas, variando entre 0 e 300 milhas náuticas.

5.2 Calibração dos Parâmetros

Com o intuito de calibrar os parâmetros do algoritmo foram elaborados diversos cenários, nos quais se variou apenas um parâmetro, enquanto os demais permaneceram fixos. Para conseguir verificar com maior facilidade as variações nos resultados obtidos através da variação dos parâmetros do algoritmo, foram utilizados valores de parâmetros baixos, o que prejudica o resultado final, porém reduz o esforço computacional e o tempo de processamento além de evidenciar mais as diferenças entre resultados. Os valores definidos para os parâmetros fixos estão indicados na sequência:

- Quantidade de iterações globais = 2;
- Quantidade de tarefas candidatas = 3;
- Quantidade de iterações sem melhoria = 4;
- Quantidade de conjuntos de embarcações candidatas: 100;
- Quantidade de tarefas excluídas da solução inicial: 2.

Tabela 5-1 – Parâmetros dos cenários

Cenários	N° Tarefas	N° Embarcações	Horizonte Planej. [dias]	Duração Méd. Tarefas [dias]	Demanda Méd. por Tarefa	Ocupação Méd. Emb.
Cenário 1	10	12	30	9,9	2,4	66%
Cenário 2	10	12	30	9,9	2,6	72%
Cenário 3	10	13	30	10,1	2,8	73%
Cenário 4	10	13	30	11,9	2,4	73%
Cenário 5	10	12	30	12,1	2,6	87%
Cenário 6	10	16	30	12	2,8	70%
Cenário 7	10	11	30	10,1	2,4	73%
Cenário 8	10	12	30	9,9	2,6	72%
Cenário 9	10	13	30	10,1	2,8	73%
Cenário 10	10	12	30	12	2,4	80%
Cenário 11	10	14	30	12	2,6	74%
Cenário 12	10	15	30	12	2,8	75%
Cenário 13	20	22	30	9,9	2,4	72%
Cenário 14	20	20	30	9,9	2,6	86%
Cenário 15	20	23	30	9,9	2,8	80%
Cenário 16	20	27	30	11,95	2,4	71%
Cenário 17	20	29	30	11,95	2,6	71%
Cenário 18	20	27	30	12,05	2,8	83%
Cenário 19	20	17	30	10	2,4	94%
Cenário 20	20	16	30	9,5	2,6	103%
Cenário 21	20	27	30	9,1	2,8	63%
Cenário 22	20	21	30	12	2,4	91%
Cenário 23	20	21	30	12,05	2,6	99%
Cenário 24	20	26	30	12	2,8	86%

A partir dos valores fixos dos parâmetros do algoritmo, primeiramente foram estudados os cenários de variação da quantidade de tarefas candidatas, que compõem o conjunto das n primeiras tarefas da lista de tarefas ordenadas que poderão ser selecionadas para serem incluídas nas rotas das embarcações.

Este primeiro teste está atrelado à regra de ordenação das tarefas, uma vez que a construção da solução inicial pela heurística construtiva é feita através de escolhas aleatórias de tarefas que compõem a lista de tarefas candidatas. A ordenação das tarefas pode ser de acordo com o valor da penalidade (decrescente) ou pelo instante de liberação da tarefa para execução (crescente). Desta maneira, quando se define que a quantidade de tarefas candidatas será igual a 1 (um), significa que a heurística construtiva irá construir a solução inicial seguindo exatamente a ordem de ordenação das tarefas.

Em contrapartida, quanto maior for o valor da quantidade de tarefas candidatas, mais variabilidade haverá na escolha das tarefas para a construção da solução inicial. A Tabela 5-2 apresenta o Cenário 1 de calibração, com rodadas que variaram apenas a quantidade de tarefas candidatas, em que se partiu de uma solução que segue exatamente a ordenação das tarefas, conforme ordenação pré-definida até uma solução com seleção de tarefas totalmente aleatória.

Tabela 5-2 – Cenário 1: Tarefas candidatas

Parâmetro	Valor	Valor	Valor	Valor	Valor
Iterações Globais	2	2	2	2	2
Tarefas candidatas	**1**	**3**	**5**	**7**	**n**
Iterações sem melhoria	4	4	4	4	4
Conjuntos Embarcações	100	100	100	100	100
Tarefas excluídas	2	2	2	2	2

No Cenário 2 de calibração foi avaliada a variação na quantidade de conjuntos de embarcações candidatas que serão utilizados no modelo matemático de inserção de uma nova tarefa na rota das embarcações. Para este estudo, além do impacto no resultado final, também se analisou o impacto no tempo total de processamento, uma vez que o aumento da quantidade de conjuntos candidatos no modelo matemático aumenta a quantidade de equações de restrições. Neste cenário variou-se a quantidade de conjuntos candidatos de 25 conjuntos até 100 conjuntos. Vide Tabela 5-3.

Tabela 5-3 – Cenário 2: Conjuntos de embarcações candidatas

Parâmetro	Valor	Valor	Valor	Valor
Iterações Globais	2	2	2	2
Tarefas candidatas	3	3	3	3
Iterações sem melhoria	4	4	4	4
Conjuntos Embarcações	**25**	**50**	**75**	**100**
Tarefas excluídas	2	2	2	2

No Cenário 3 de calibração foi estudado o impacto da variação da quantidade de iterações sem melhoria do algoritmo, que representa o critério de parada do algoritmo. A variação deste parâmetro aumenta a varredura do espaço de solução, uma vez que ele faz com que mais iterações de remoção e reinserção de tarefas sejam executadas até que o processamento atinja o valor de iterações sem melhoria pré-definido. Como este parâmetro do algoritmo pode aumentar bastante a quantidade de iterações, além da análise do impacto no resultado final, também foi analisado o impacto no tempo total de processamento. Neste cenário variou-se a quantidade de iterações sem melhoria de 2 até 6 iterações. Vide Tabela 5-4.

Tabela 5-4 – Cenário 3: Iterações sem melhoria

Parâmetro	Valor	Valor	Valor	Valor
Iterações Globais	2	2	2	2
Tarefas candidatas	3	3	3	3
Iterações sem melhoria	**2**	**4**	**5**	**6**
Conjuntos Embarcações	100	100	100	100
Tarefas excluídas	2	2	2	2

O Cenário 4 de calibração analisou a variação da quantidade de tarefas excluídas da solução inicial. A variação da quantidade de tarefas excluídas é o parâmetro que causa maior impacto no tempo total de processamento, uma vez que, quando se aumenta esta quantidade, o algoritmo avalia todas as possíveis combinações de reinserção das tarefas excluídas nas rotas das embarcações, ou seja, se forem excluídas 3 tarefas da solução inicial, a heurística irá executar todas as

combinações possíveis de ordenação das tarefas no momento da reinserção, que é igual a 3! (3 x 2 x 1 = 6) combinações. Apesar do aumento tempo de processamento, este parâmetro também é responsável pelo aumento da varredura do espaço de soluções, possibilitando, eventualmente, alcançar melhores resultados. Como o intuito desta etapa de calibração do trabalho é o estudo do impacto da variação dos parâmetros nos resultados finais e a obtenção de soluções viáveis com baixo custo de processamento e curtos períodos de tempo, limitou-se a quantidade máxima de tarefas excluídas em 4. Portanto a variação da quantidade de tarefas excluídas neste cenário foi de 1 a 4 tarefas. Vide Tabela 5-5.

Tabela 5-5 – Cenário 4: Tarefas excluídas

Parâmetro	Valor	Valor	Valor	Valor
Iterações Globais	2	2	2	2
Tarefas candidatas	3	3	3	3
Iterações sem melhoria	4	4	4	4
Conjuntos Embarcações	100	100	100	100
Tarefas excluídas	**1**	**2**	**3**	**4**

5.3 Resultados Modelo Matemático Exato – Base de Comparação

A seguir serão apresentados os resultados obtidos através da resolução do modelo matemático do problema desta pesquisa usando o método *branch-and-bound* do pacote Gurobi. Como base de comparação da efetividade e eficiência do método proposto, as 24 instâncias de teste foram executadas por períodos de 1 hora e de 12 horas. Os resultados obtidos são apresentados na Tabela 5-6.

Tabela 5-6 – Resultados do Processamento do Modelo Matemático

Cenários	Tarefas	Embarcações	Resultado 1h		Resultado 12h	
			Resultado	Gap	Resultado	Gap
Cenário 1	10	12	2,208E+05	0,0%	2,208E+05	0,0%
Cenário 2	10	12	3,209E+05	0,0%	3,209E+05	0,0%
Cenário 3	10	13	1,223E+05	0,0%	1,223E+05	0,0%
Cenário 4	10	13	7,653E+04	0,0%	7,653E+04	0,0%
Cenário 5	10	12	1,352E+05	0,0%	1,352E+05	0,0%
Cenário 6	10	16	1,832E+05	0,0%	1,832E+05	0,0%
Cenário 7	10	11	4,013E+05	0,0%	4,013E+05	0,0%
Cenário 8	10	12	6,296E+05	0,0%	6,296E+05	0,0%
Cenário 9	10	13	2,373E+05	0,0%	2,373E+05	0,0%
Cenário 10	10	12	2,094E+05	0,0%	2,094E+05	0,0%
Cenário 11	10	14	1,010E+06	0,0%	1,010E+06	0,0%
Cenário 12	10	15	2,618E+05	0,0%	2,618E+05	0,0%
Cenário 13	20	22	3,029E+05	0,0%	3,029E+05	0,0%
Cenário 14	20	20	2,261E+06	94,2%	1,872E+06	92,3%
Cenário 15	20	23	1,570E+06	93,2%	4,660E+05	71,3%
Cenário 16	20	27	3,820E+06	95,9%	1,830E+06	89,7%
Cenário 17	20	29	6,284E+05	0,0%	6,284E+05	0,0%
Cenário 18	20	27	1,437E+06	90,1%	1,006E+06	52,6%
Cenário 19	20	17			2,378E+06	95,4%
Cenário 20	20	16	9,357E+06	98,9%	4,241E+06	97,5%
Cenário 21	20	27			2,185E+05	0,0%
Cenário 22	20	21	1,140E+06	90,5%	7,148E+05	52,2%
Cenário 23	20	21	7,230E+06	97,9%	4,134E+06	96,1%
Cenário 24	20	26	1,531E+06	94,8%	1,347E+06	89,3%

Analisando-se os dados obtidos através da resolução do modelo matemático exato, é possível verificar que para os processamentos executados pelo período de 1 hora, foi possível obter a solução ótima para as instâncias contendo 10 tarefas (instâncias 1 a 12), porém para as instâncias de 20 tarefas os “gaps” foram maiores que 90%, com exceção das instâncias 13 e 17 que obtiveram a solução ótima e das

instâncias 19 e 21, para as quais, não foram obtidas nenhuma solução viável para o tempo imposto. Para as rodadas executadas pelo período de 12 horas, as instâncias contendo 10 tarefas continuaram produzindo as soluções ótimas, e para as instâncias com 20 tarefas, foram obtidas as soluções ótimas apenas para as instâncias 13, 17 e 21, e, para as demais instâncias que não obtiveram a solução ótima, o gap médio ficou em 81,8% (excluindo-se o valor dos gaps das instâncias 13, 17 e 21, que obtiveram a solução ótima).

5.4 Calibração de Parâmetros - Método 1 – Seleção Aleatória

O Método 1 - Seleção Aleatória foi aplicado ao conjunto de 24 instâncias descritas na seção 5.1, visando a calibração dos parâmetros. Nesta fase de calibração, os valores dos parâmetros foram definidos de maneira que, preferencialmente, não se tenha períodos de processamento muito longos. As rodadas foram executadas em um computador com processador Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz, 2401 Mhz, 2 Núcleo(s), 4 Processador(es) Lógico(s) e 8 Gb de RAM, e os softwares utilizados para a programação e compilação do problema foram o Visual Studio 2015 e o Gurobi 7.0.2 para o processamento dos modelos de otimização. Nesta etapa de calibração, as rodadas foram executadas com o compilador no modo “debug”, o que faz com que os tempos de processamento sejam ligeiramente maiores.

5.4.1 Resultados Cenário 1 – Tarefas Candidatas

As rodadas executadas para o Cenário 1 de calibração foram rodadas tanto para a regra de ordenação das tarefas por penalidade, como por instante de liberação das tarefas para execução, e os resultados obtidos para o processamento com a ordenação das tarefas por penalidade estão apresentados na Tabela 5-7 e Figura 5-1, para as instâncias de 10 tarefas eTabela 5-8 e Figura 5-2 para as instâncias de 20

tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre as soluções obtidas nas rodadas do algoritmo de Seleção Aleatória, e os obtidos no processamento do modelo matemático.

Neste cenário 1, referente à quantidade de tarefas candidatas, foram avaliadas as variações com 1, 3, 5, 7 e n (que corresponde a quantidade total de tarefas para cada instância = 10 ou 20), que significa que foram testadas desde a configuração em que a primeira solução da fase construtiva segue exatamente a regra de ordenação das tarefas (quantidade = 1), até a configuração em que a seleção de tarefas candidatas é feita de maneira totalmente aleatória (quantidade = n).

Tabela 5-7 – Resultados Cenário 1 - 10 tarefas - Penalidade

Instâncias	1 tarefa	3 tarefas	5 tarefas	7 tarefas	N tarefas
	Gap	Gap	Gap	Gap	Gap
Instância 1	58%	25%	58%	58%	6%
Instância 2	1%	0%	0%	31%	1%
Instância 3	362%	321%	351%	348%	57%
Instância 4	2%	111%	74%	75%	1%
Instância 5	147%	0%	0%	59%	2%
Instância 6	58%	2%	2%	2%	3%
Instância 7	238%	2%	1%	0%	0%
Instância 8	16%	0%	51%	127%	154%
Instância 9	40%	143%	43%	42%	43%
Instância 10	13%	10%	2%	16%	24%
Instância 11	58%	22%	58%	29%	0%
Instância 12	279%	282%	226%	226%	254%
Média para 10 tarefas	106%	77%	72%	84%	45%

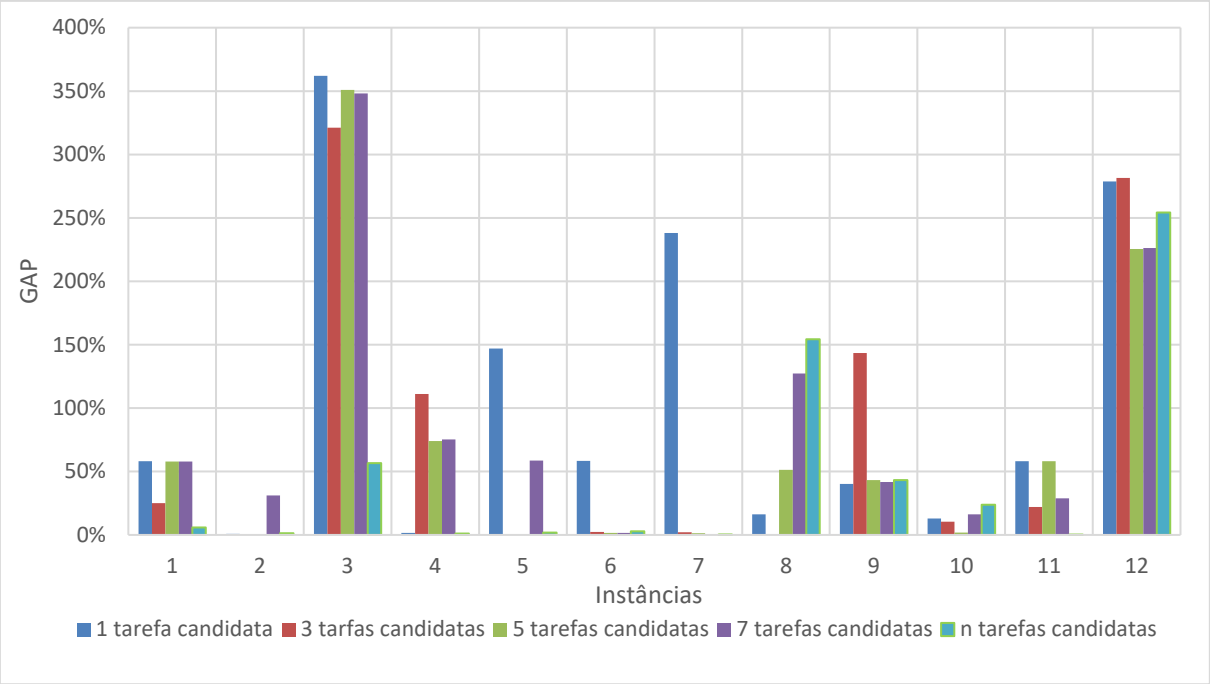

Figura 5-1 – Gráfico resultados Cenário 1 - 10 tarefas - Penalidade

Fonte: Elaborado pelo autor

Tabela 5-8 – Resultados Cenário 1 - 20 tarefas - Penalidade

Instâncias	1 tarefa	3 tarefas	5 tarefas	7 tarefas	N tarefas
	Gap	Gap	Gap	Gap	Gap
Instância 13	914%	525%	702%	282%	499%
Instância 14	60%	87%	40%	96%	88%
Instância 15	253%	458%	349%	531%	493%
Instância 16	51%	28%	26%	41%	65%
Instância 17	240%	224%	177%	229%	397%
Instância 18	282%	93%	180%	151%	107%
Instância 19	-9%	-20%	62%	-9%	19%
Instância 20	-1%	4%	-12%	-10%	48%
Instância 21	211%	297%	9%	109%	149%
Instância 22	76%	200%	285%	191%	337%
Instância 23	25%	15%	6%	-2%	10%
Instância 24	44%	50%	37%	36%	64%
Média para 20 tarefas	179%	164%	155%	137%	190%

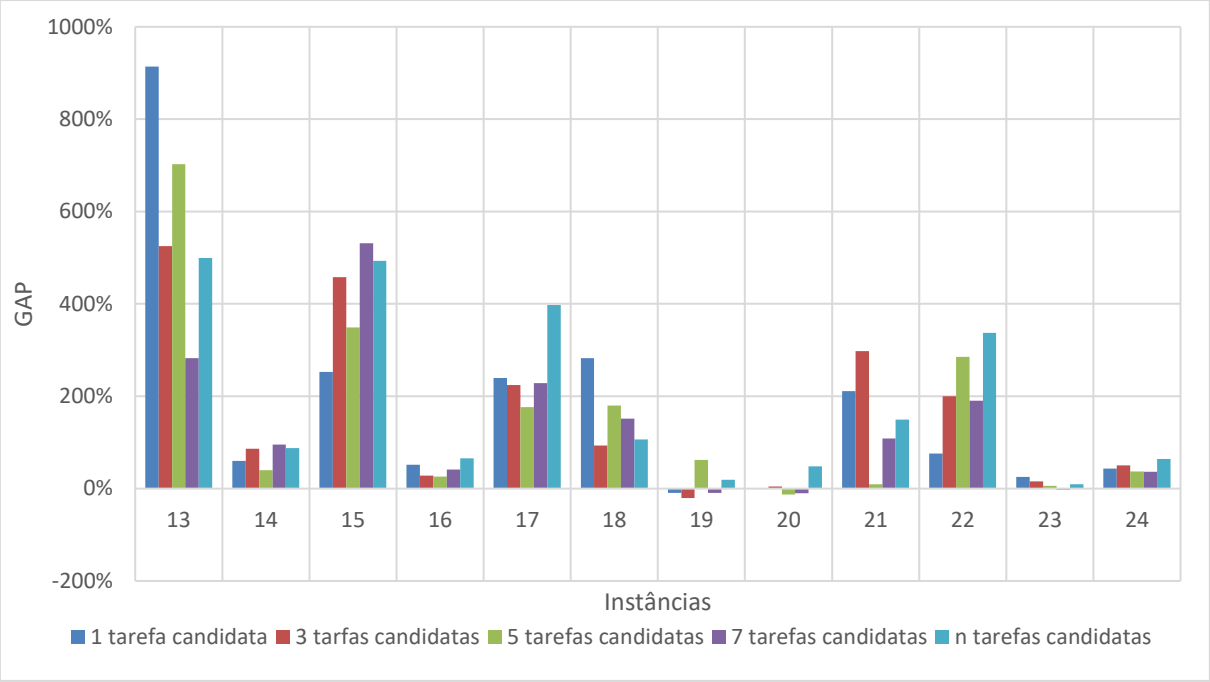

Figura 5-2 – Gráfico resultados Cenário 1 – 20 tarefas – Penalidade

Fonte: Elaborado pelo autor

Avaliando-se os resultados obtidos para o cenário 1 com regra de ordenação por penalidade das tarefas, e que varia a quantidade de tarefas candidatas, ou seja, a maneira como será construída a primeira solução: i) se é construída respeitando-se a ordenação por penalidades; ii) respeitando-se a ordenação por instante de libração das tarefas para execução; ou iii) de maneira aleatória; é possível se concluir que a regra de ordenação das tarefas na ordem decrescente do valor da penalidade diária por atraso não é benéfica para a construção da primeira solução, e, por consequência, prejudica o resultado final. Isso pode ser concluído através dos gráficos apresentados nas Figura 5-1 e Figura 5-2 em que, conforme a quantidade de tarefas candidatas aumenta (o que torna a construção da solução inicial cada vez mais aleatória) os resultados finais obtidos melhoram.

Os resultados obtidos para a calibração com ordenação das tarefas por instante de liberação das tarefas estão apresentados na Tabela 5-9 e Figura 5-3 para as instâncias de 10 tarefas, e na Tabela 5-10 e Figura 5-4 para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução

obtida pelo processamento do algoritmo de Seleção Aleatória e os obtidos no processamento do modelo matemático.

Tabela 5-9 – Resultados Cenário 1 – 10 tarefas – Instante de Liberação

instâncias	1 tarefa	3 tarefas	5 tarefas	7 tarefas
	Gap	Gap	Gap	Gap
Instância 1	58%	8%	8%	1%
Instância 2	1%	135%	84%	0%
Instância 3	99%	195%	409%	412%
Instância 4	3%	2%	3%	7%
Instância 5	0%	1%	0%	0%
Instância 6	3%	4%	34%	148%
Instância 7	67%	59%	174%	270%
Instância 8	49%	129%	45%	137%
Instância 9	43%	41%	41%	330%
Instância 10	12%	21%	2%	9%
Instância 11	59%	8%	9%	17%
Instância 12	252%	15%	192%	793%
Média para 10 tarefas	54%	52%	83%	177%

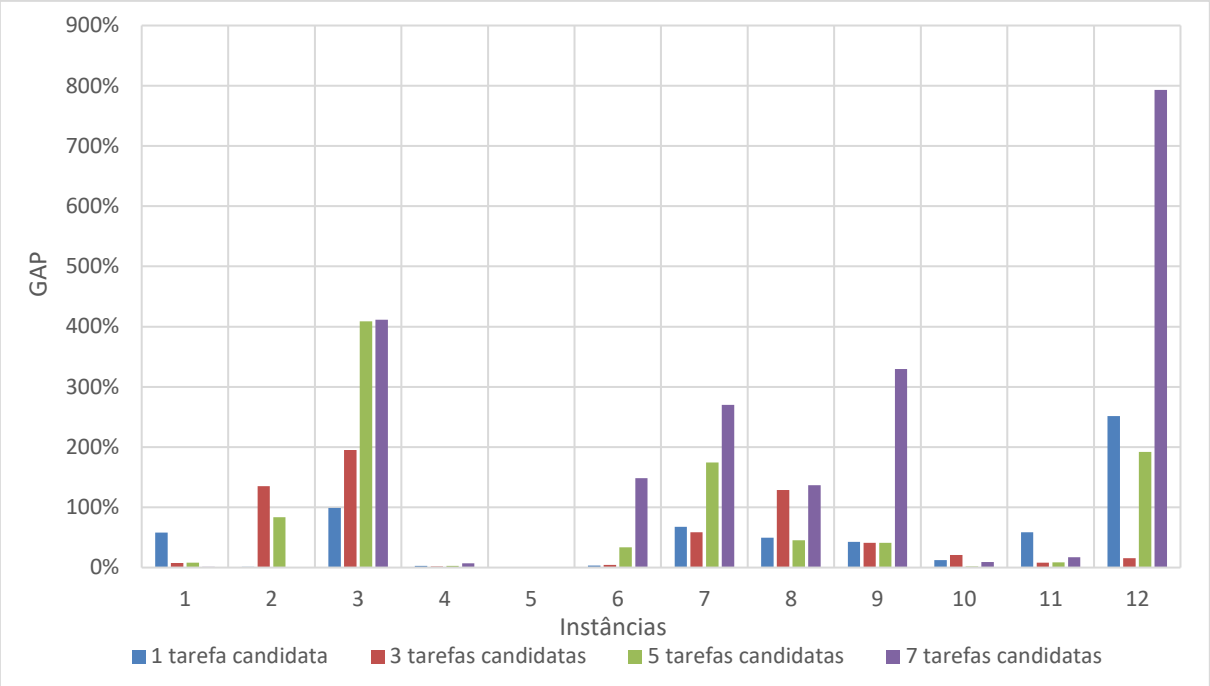

Figura 5-3 – Gráfico resultados Cenário 1 – 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-10 – Resultados Cenário 1 – 20 tarefas – Instante de Liberação

Instâncias	1 tarefa	3 tarefas	5 tarefas	7 tarefas
	Gap	Gap	Gap	Gap
Instância 13	491%	145%	667%	613%
Instância 14	49%	12%	73%	89%
Instância 15	160%	90%	277%	230%
Instância 16	4%	83%	47%	38%
Instância 17	140%	292%	397%	393%
Instância 18	152%	118%	229%	116%
Instância 19	-17%	-22%	-4%	-5%
Instância 20	-11%	2%	-7%	5%
Instância 21	106%	74%	65%	177%
Instância 22	381%	544%	256%	130%
Instância 23	21%	9%	25%	15%
Instância 24	64%	63%	4%	50%
Média para 20 tarefas	128%	118%	169%	154%

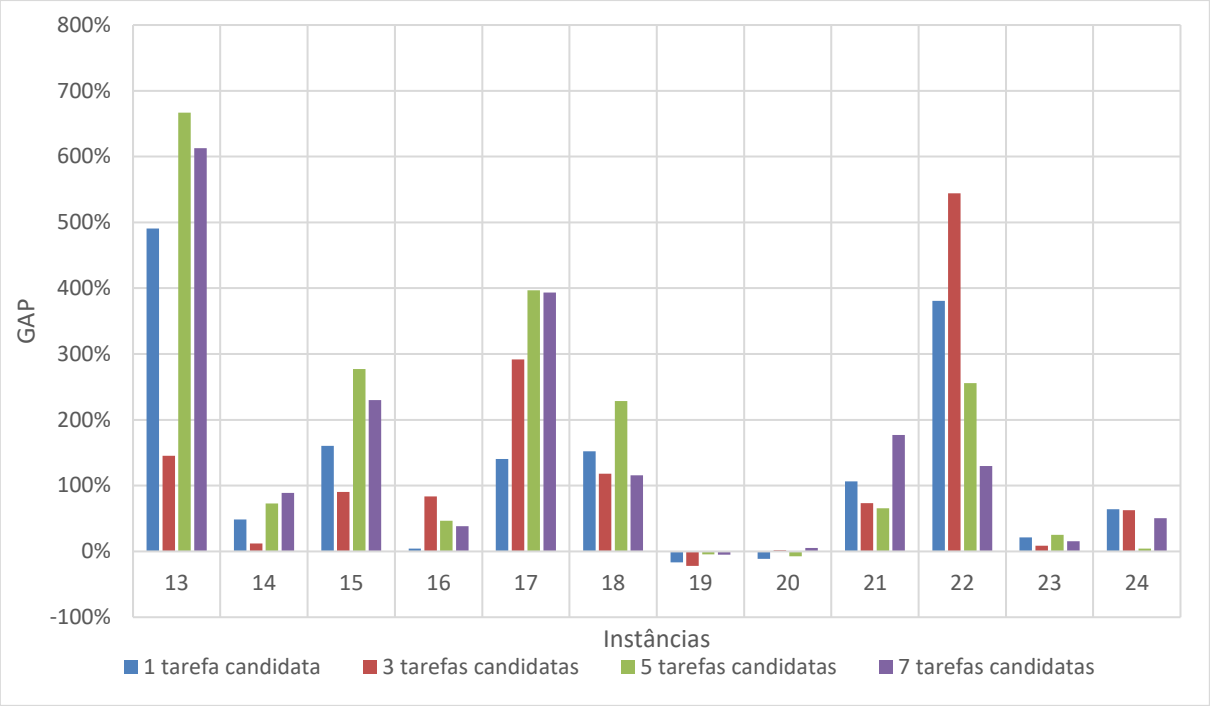

Figura 5-4 – Gráfico resultados Cenário 1 - 20 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Diferentemente da ordenação por penalidade, os resultados obtidos para o caso em que as tarefas foram ordenadas conforme instante de liberação para início da tarefa apresentaram um bom desempenho, mais acentuado para as instâncias com 10 tarefas, em que a média dos valores de gap das 12 instâncias testadas saltaram de 54% (para 1 tarefa candidata) para 177% (para 7 tarefas candidatas). Para as instâncias de 20 tarefas foi verificado um desempenho menos acentuado, nas quais a média dos valores de gap das 12 instâncias saíram de 128% (para 1 tarefa candidata) para 154% (para 7 tarefas candidatas), e a melhor média verificada foi para a rodada com 3 tarefas candidatas, em que a média ficou em 118%. Estas variações nos resultados, para os processamentos com ordenação das tarefas por instante de liberação, podem ser verificadas nos gráficos das Figura 5-3 e Figura 5-4.

Ainda seguindo na análise dos resultados obtidos no cenário 1, a rodada com a ordenação das tarefas de maneira totalmente aleatória, expressa pela simulação “2-n-4-100-2” também apresentou bons resultados, principalmente para as instâncias de 10 tarefas, em que a média dos valores de gap obtidos para as 12 instâncias ficou em

45%, enquanto que o a média dos valores de gap para as 12 instâncias de 20 tarefas ficou em 190%, mostrando ser um resultado não muito satisfatório. Este resultado é interessante, porque com a seleção das tarefas feita de maneira aleatória, os resultados podem ser tanto muito bons, como foi o caso das instâncias de 10 tarefas, como também, podem ser ruins, como foi o caso das instâncias de 20 tarefas. Portanto, é possível se concluir que, tanto a regra de ordenação das tarefas por instante de liberação para início como a de ordenação aleatória apresentam uma tendência de conseguirem melhores resultados, quando comparados com a regra de ordenação por penalidade.

Uma última observação importante neste cenário de calibração é que em algumas das rodadas, para as instâncias 19, 20 e 23, foram verificados valores de gaps negativos, indicando que a heurística, mesmo que com poucas iterações e tarefas excluídas, já encontrou soluções melhores do que as encontradas pelo método exato rodando por um período de 12 horas.

5.4.2 Resultados Cenário 2 – Conjuntos de Embarcações Candidatas

Assim como as rodadas feitas para o Cenário 1, os processamentos do Cenário 2 foram feitos tanto para a regra de ordenação das tarefas por penalidade, como por instante de liberação para início das tarefas. Para o Cenário 2, relativo à quantidade de conjuntos de embarcações candidatas para serem considerados no modelo matemático, foram testadas as configurações com 25, 50, 75 e 100 conjuntos candidatos. Como a variação da quantidade de conjuntos candidatos impacta diretamente no porte do modelo matemático de inserção, neste cenário de calibração também foi analisado o tempo total de processamento.

Os resultados obtidos para o processamento com a ordenação das tarefas por penalidade estão apresentados na

Tabela 5-11 e Figura 5-5, para as instâncias de 10 tarefas, e na Tabela 5-12 e Figura 5-6, para as instâncias de 20 tarefas. Os valores dos resultados indicados nas

tabelas expressam os gaps entre a solução obtida pelo processamento do algoritmo, e os obtidos no processamento do modelo matemático. Os tempos de processamento estão indicados em minutos. Os resultados obtidos para o processamento com a ordenação das tarefas por instante de liberação para início estão apresentados na Tabela 5-13 e Figura 5-7 para as instâncias de 10 tarefas e na Tabela 5-14 e Figura 5-8 para as instâncias de 20 tarefas.

Tabela 5-11 – Resultados Cenário 2 - 10 tarefas – Penalidade

Instâncias	25 Conjuntos		50 Conjuntos		75 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 1	1,8	59%	1,9	71%	1,4	70%	2,7	25%
Instância 2	0,9	83%	1,4	30%	0,9	2%	1,6	0%
Instância 3	1,3	278%	1,4	286%	1,5	313%	3,1	321%
Instância 4	1,4	76%	1,5	83%	2,2	3%	3,0	111%
Instância 5	0,8	16%	1,5	15%	1,6	236%	2,6	0%
Instância 6	1,2	118%	1,2	25%	2,7	193%	2,2	2%
Instância 7	0,5	239%	0,6	250%	1,2	239%	1,1	2%
Instância 8	1,0	143%	0,9	86%	2,3	68%	5,5	0%
Instância 9	1,0	256%	0,7	43%	3,1	43%	2,2	143%
Instância 10	0,8	69%	0,9	13%	1,4	2%	4,6	10%
Instância 11	0,8	23%	1,3	17%	1,8	30%	2,6	22%
Instância 12	0,8	313%	1,1	246%	9,9	253%	3,5	282%
Média 10 tarefas	1,0	139%	1,2	97%	2,5	121%	2,9	77%

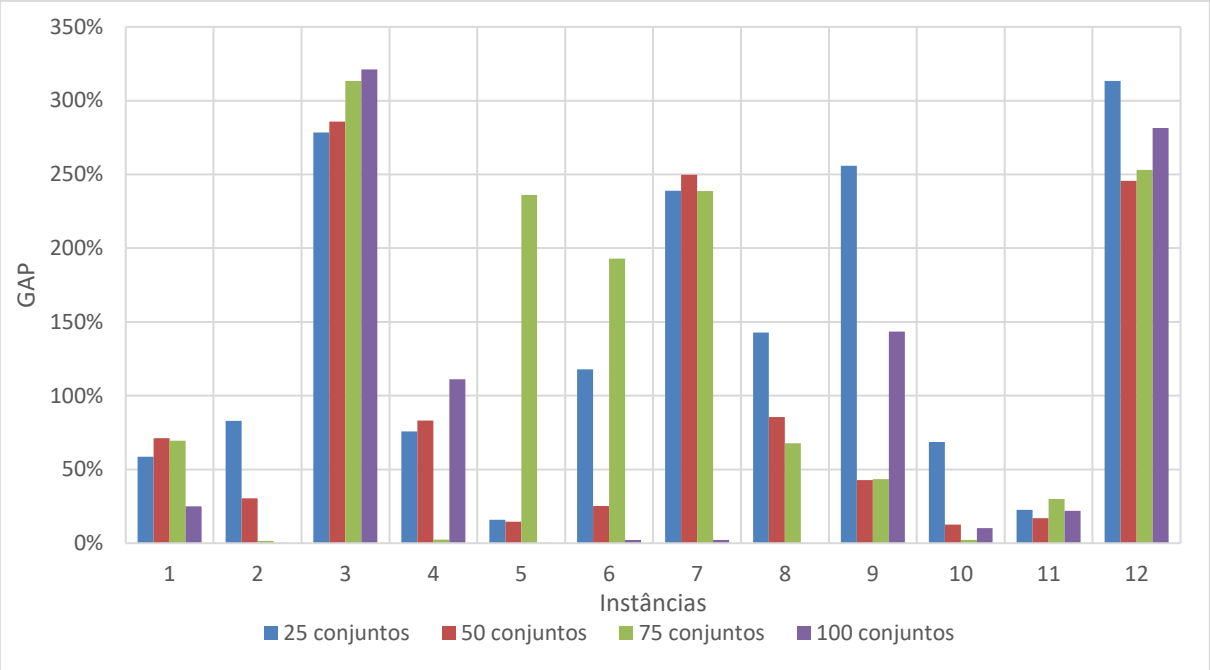

Figura 5-5 – Gráfico resultados Cenário 2 - 10 tarefas – Penalidade

Fonte: Elaborado pelo autor

Tabela 5-12 – Resultados Cenário 2 - 20 tarefas – Penalidade

Instâncias	25 Conjuntos		50 Conjuntos		75 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 13	2,4	869%	3,2	638%	10,0	781%	11,0	525%
Instância 14	3,6	88%	3,6	86%	2,6	74%	4,8	87%
Instância 15	5,8	472%	3,3	406%	11,1	218%	10,0	458%
Instância 16	3,0	95%	3,0	78%	5,9	39%	10,3	28%
Instância 17	4,3	335%	6,4	311%	4,1	282%	15,2	224%
Instância 18	3,2	202%	3,2	244%	3,8	157%	9,8	93%
Instância 19	1,5	-2%	1,4	-10%	1,9	-36%	3,4	-20%
Instância 20	1,4	2%	2,0	18%	2,5	10%	7,2	4%
Instância 21	3,2	275%	4,7	149%	7,8	29%	11,2	297%
Instância 22	1,7	289%	1,7	222%	2,0	168%	5,0	200%
Instância 23	1,8	22%	2,2	10%	1,8	22%	6,2	15%
Instância 24	5,8	97%	3,3	112%	3,2	121%	11,3	50%
Média 20 tarefas	3,1	229%	3,2	189%	4,7	155%	8,8	164%

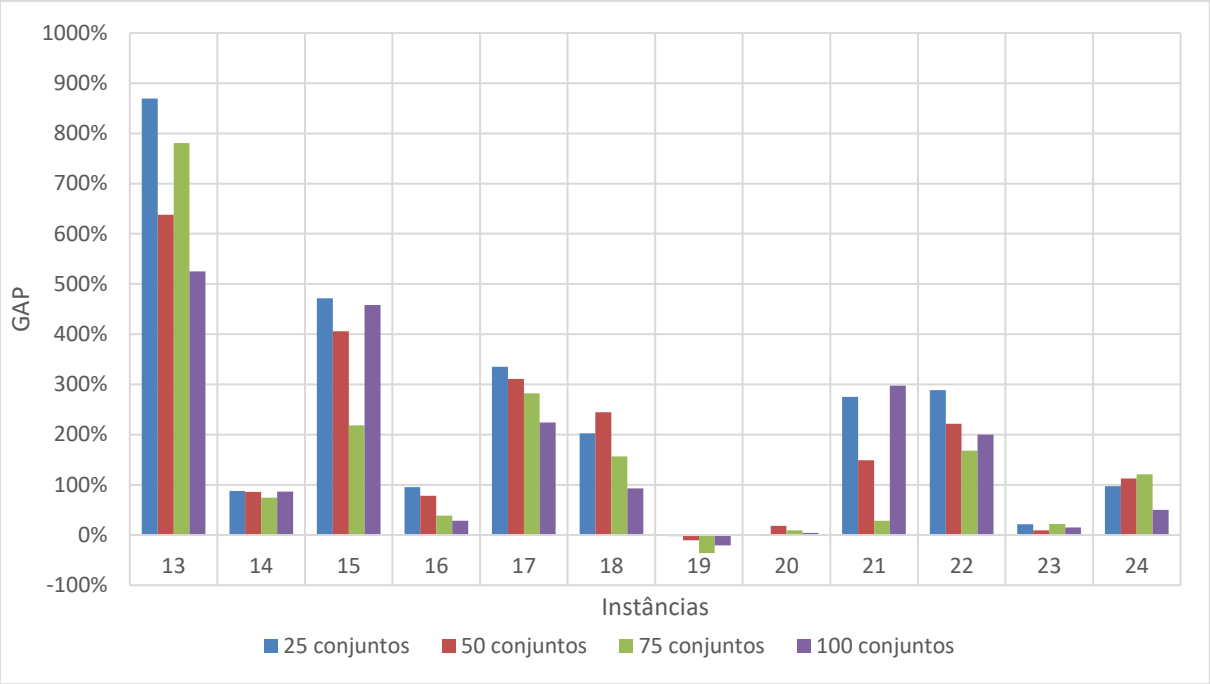

Figura 5-6 – Gráfico resultados Cenário 2 - 20 tarefas – Penalidade

Fonte: Elaborado pelo autor

Tabela 5-13 – Resultados Cenário 2 - 10 tarefas – Instante de Liberação

Instâncias	25 Conjuntos		50 Conjuntos		75 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 1	1,6	58%	1,6	71%	2,0	8%	2,4	8%
Instância 2	1,2	262%	0,8	29%	1,1	30%	3,0	135%
Instância 3	0,7	195%	1,1	147%	0,8	181%	3,4	195%
Instância 4	1,0	2%	1,6	1%	1,0	0%	2,2	2%
Instância 5	1,1	21%	1,1	1%	1,0	14%	1,5	1%
Instância 6	1,5	10%	1,6	187%	1,5	333%	3,5	4%
Instância 7	0,9	21%	0,9	290%	1,0	59%	1,8	59%
Instância 8	0,6	87%	0,6	75%	1,2	99%	2,5	129%
Instância 9	1,0	43%	1,2	43%	1,4	41%	4,6	41%
Instância 10	1,0	22%	0,7	22%	1,8	8%	2,2	21%
Instância 11	0,7	58%	0,8	9%	1,7	8%	2,0	8%
Instância 12	0,8	326%	3,5	282%	0,9	15%	1,1	15%
Média 10 tarefas	1,0	92%	1,3	96%	1,3	66%	2,5	52%

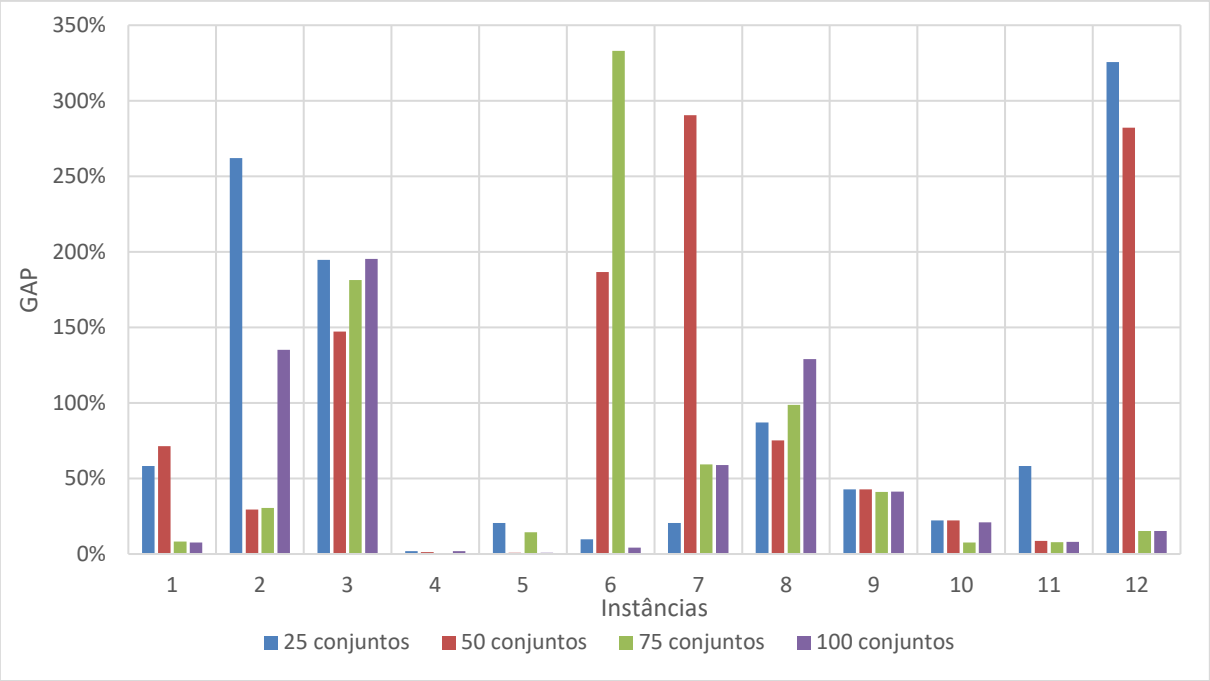

Figura 5-7 – Gráfico resultados Cenário 2 - 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-14 – Resultados Cenário 2 - 20 tarefas – Instante de Liberação

Instâncias	25 Conjuntos		50 Conjuntos		75 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 13	2,8	405%	2,9	664%	3,7	363%	8,6	145%
Instância 14	1,8	78%	2,6	58%	2,0	83%	4,8	12%
Instância 15	3,8	159%	2,8	292%	3,0	323%	11,2	90%
Instância 16	4,6	64%	3,6	49%	3,4	23%	8,8	83%
Instância 17	3,9	387%	4,3	119%	4,4	71%	15,5	292%
Instância 18	3,2	438%	3,3	249%	4,9	284%	19,8	118%
Instância 19	1,5	-29%	2,2	-18%	2,9	-42%	5,1	-22%
Instância 20	1,9	69%	2,1	-10%	4,4	-18%	6,5	2%
Instância 21	4,5	178%	2,3	185%	3,3	236%	8,6	74%
Instância 22	2,9	505%	3,6	362%	2,8	560%	4,9	544%
Instância 23	3,1	122%	2,9	57%	4,1	0%	16,2	9%
Instância 24	4,9	67%	7,1	34%	3,8	108%	14,4	63%
Média 20 tarefas	3,2	204%	3,3	170%	3,6	166%	10,4	118%

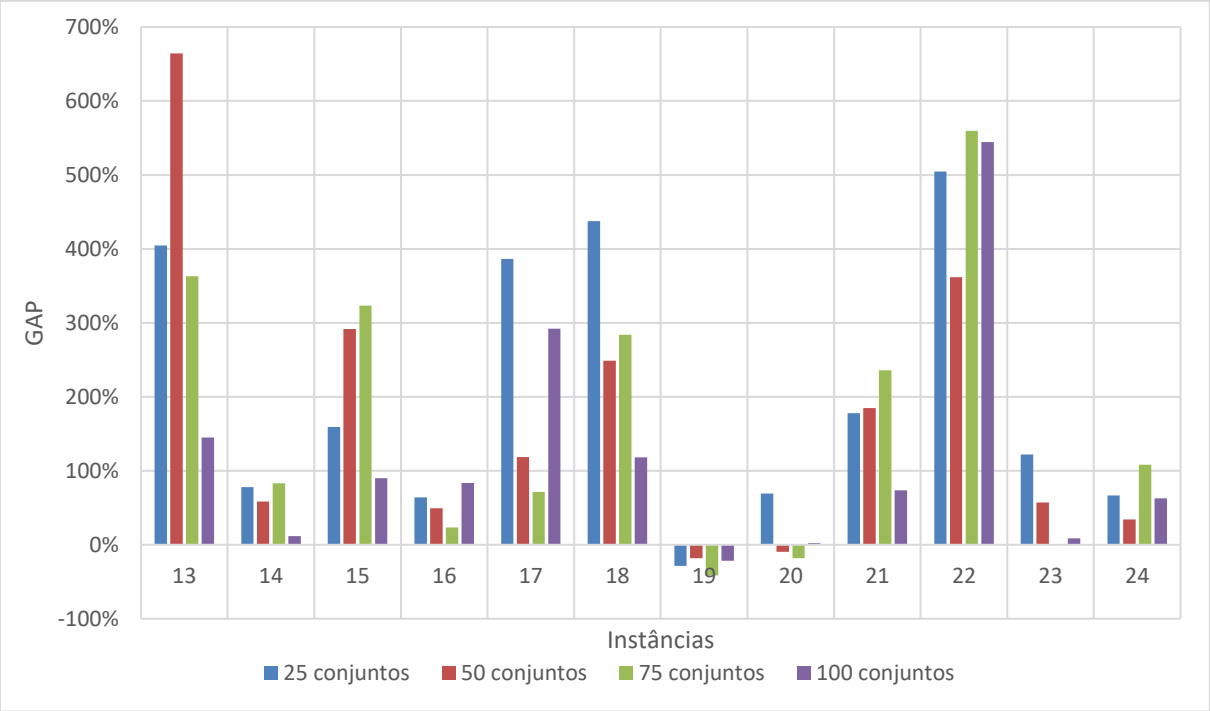

Figura 5-8 – Gráfico resultados Cenário 2 - 20 tarefas –Instante de Liberação

Fonte: Elaborado pelo autor

Avaliando-se os resultados obtidos para o cenário 2 de calibração, que varia a quantidade de conjuntos de embarcações candidatas, é possível concluir que o aumento da quantidade de conjuntos de embarcações nos parâmetros da heurística melhora os resultados finais obtidos. Este comportamento pode ser verificado tanto para a regra de ordenação das tarefas por penalidade, como por instante de liberação para início da execução da tarefa. Por meio dos gráficos apresentados nas figuras Figura 5-5, Figura 5-6, Figura 5-7 e Figura 5-8, é possível constatar que na maioria das instâncias, a redução do valor da solução final diminui conforme se aumenta a quantidade de conjuntos de embarcações candidatas.

Assim como verificado no cenário 1 de calibração, os resultados obtidos para os processamentos com a regra de ordenação das tarefas de acordo com o instante de liberação para início apresentaram valores melhores do que com a regra de ordenação por penalidade. Conforme apresentado nas tabelas de resultados, a melhor média dos gaps, que corresponde a 100 conjuntos de embarcações, para o caso da ordenação por penalidade ficou em 77% para instâncias de 10 tarefas e 164%

para instâncias de 20 tarefas, e para o caso de ordenação por liberação para início, ficou em 52% para instâncias de 10 tarefas e 118% para instâncias de 20 tarefas.

Em relação ao tempo de processamento, verificou-se que na variação de 25 conjuntos para 100 conjuntos de fato houve aumento do tempo, que, percentualmente apresentou um aumento de 220%, e de forma absoluta apresentou um aumento de 7,1 minutos para o caso de 20 tarefas e ordenação por liberação para início (caso mais crítico). Apesar de percentualmente o aumento ser expressivo, em valores, um aumento de 3,2 minutos para 10,4 minutos não apresenta grandes impactos. Porém, é importante ressaltar que quando se combinam diversos parâmetros com valores mais altos, essa diferença de 220% no tempo de processamento pode ser bastante impactante no tempo final.

5.4.3 Resultados Cenário 3 – Iterações Sem Melhoria

Neste cenário de calibração foram avaliadas a quantidade de iterações em melhoria como critério de interrupção do algoritmo, sendo testados casos com 2, 4, 5 e 6 iterações sem melhoria. Este cenário 3 de calibração foi processado tanto para a regra de ordenação das tarefas por penalidade, como por instante de liberação para início das tarefas. Como a variação da quantidade de iterações sem melhoria causa impacto direto no tempo total de processamento, também foi analisada a variação do tempo total de processamento.

Os resultados obtidos para o processamento com a ordenação das tarefas por penalidade estão apresentados na Tabela 5-15 e Figura 5-9, para as instâncias de 10 tarefas, e na Tabela 5-16 e Figura 5-10 para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no processamento do algoritmo e os obtidos no processamento do modelo matemático exato. Os tempos de processamento estão indicados em minutos. Os resultados obtidos para a simulação com a ordenação das tarefas por instante de liberação para

início estão apresentados na Tabela 5-17 e Figura 5-11 para as instâncias de 10 tarefas e na Tabela 5-18 e Figura 5-12 para as instâncias de 20 tarefas.

Tabela 5-15 – Resultados Cenário 3 - 10 tarefas – Penalidade

Instâncias	2 Iterações		4 Iterações		5 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 1	1,2	69%	2,7	25%	4,4	14%	7,6	11%
Instância 2	0,7	31%	1,6	0%	2,9	1%	3,8	0%
Instância 3	0,9	260%	3,1	321%	6,3	46%	5,0	33%
Instância 4	0,8	135%	3,0	111%	5,0	116%	6,6	1%
Instância 5	1,1	38%	2,6	0%	6,4	25%	3,3	25%
Instância 6	1,3	4%	2,2	2%	6,3	38%	7,1	2%
Instância 7	0,7	2%	1,1	2%	6,2	25%	2,6	14%
Instância 8	0,9	49%	5,5	0%	7,9	90%	4,8	90%
Instância 9	1,0	43%	2,2	143%	3,5	43%	9,3	42%
Instância 10	0,8	8%	4,6	10%	5,4	3%	7,0	1%
Instância 11	1,1	22%	2,6	22%	5,2	8%	5,7	17%
Instância 12	2,6	388%	3,5	282%	7,0	224%	5,0	224%
Média para 10 tarefas	1,1	88%	2,9	77%	5,5	53%	5,7	38%

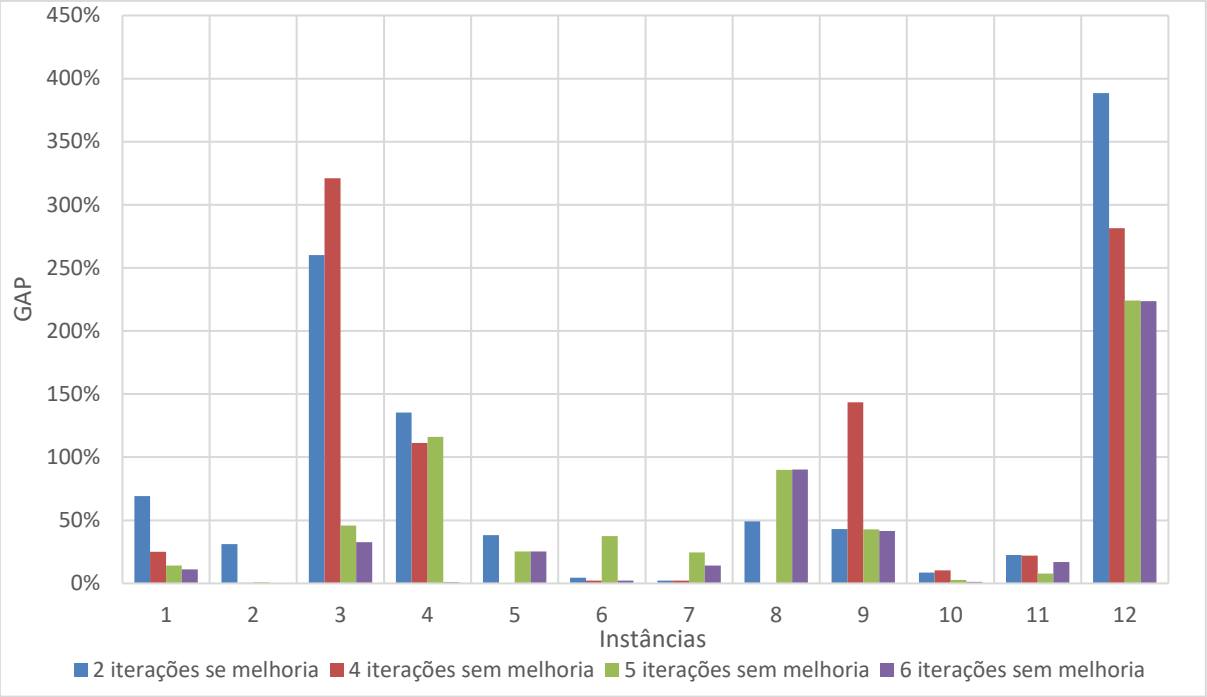

Figura 5-9 – Gráfico resultados Cenário 3 - 10 tarefas – Penalidade

Fonte: Elaborado pelo autor

Tabela 5-16 – Resultados Cenário 3 - 20 tarefas – Penalidade

Instâncias	2 Iterações		4 Iterações		5 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 13	2,5	473%	11,0	525%	12,2	786%	8,4	625%
Instância 14	2,0	108%	4,8	87%	3,9	72%	13,7	53%
Instância 15	2,3	347%	10,0	458%	4,9	249%	6,1	278%
Instância 16	3,4	66%	10,3	28%	10,7	66%	15,3	40%
Instância 17	3,9	319%	15,2	224%	16,2	136%	14,3	211%
Instância 18	2,5	179%	9,8	93%	6,5	191%	16,9	142%
Instância 19	1,4	-15%	3,4	-20%	4,1	-22%	9,5	-34%
Instância 20	2,3	12%	7,2	4%	7,4	-6%	6,2	2%
Instância 21	2,9	362%	11,2	297%	13,7	117%	19,8	159%
Instância 22	1,6	166%	5,0	200%	4,7	94%	10,5	83%
Instância 23	3,8	7%	6,2	15%	10,9	-3%	5,0	0%
Instância 24	4,8	48%	11,3	50%	7,3	106%	43,4	-11%
Média para 20 tarefas	2,8	173%	8,8	164%	8,5	149%	14,1	129%

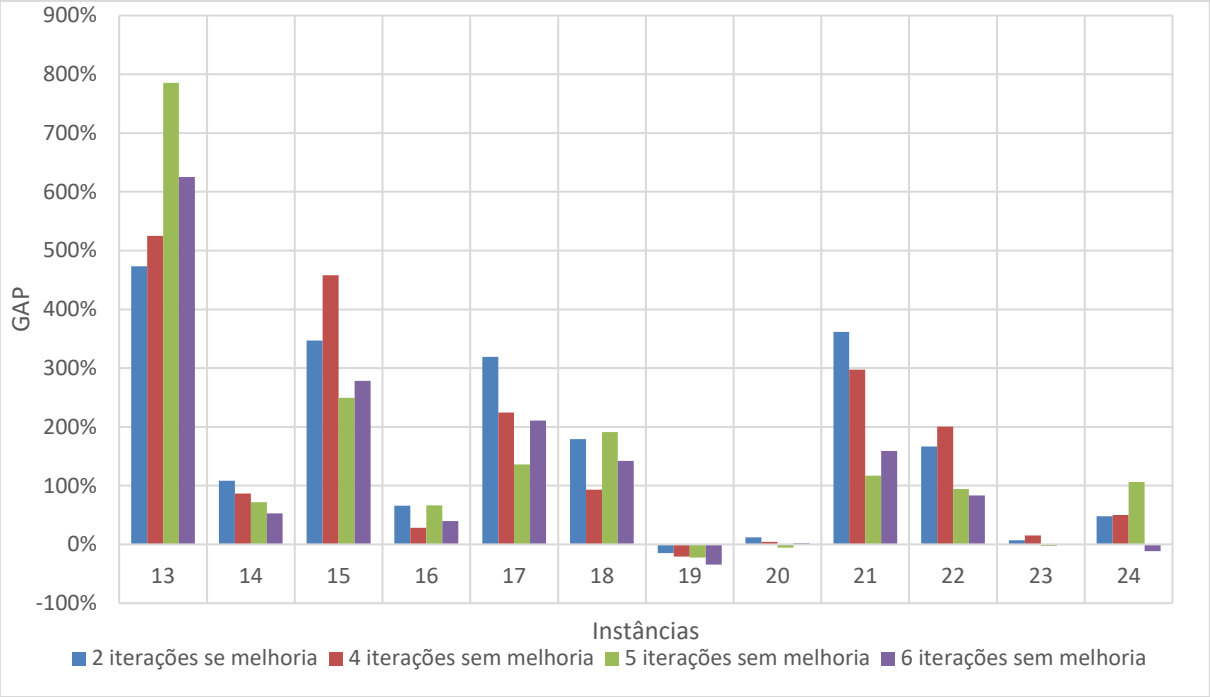

Figura 5-10 – Gráfico resultados Cenário 3 - 20 tarefas – Penalidade

Fonte: Elaborado pelo autor

Tabela 5-17 – Resultados Cenário 3 - 10 tarefas – Instante de Liberação

Instâncias	2 Iterações		4 Iterações		5 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 1	1,2	9%	2,4	8%	3,1	9%	5,7	50%
Instância 2	1,0	30%	3,0	135%	2,8	2%	4,1	31%
Instância 3	0,7	181%	3,4	195%	2,5	65%	7,4	59%
Instância 4	0,7	14%	2,2	2%	1,8	111%	3,5	112%
Instância 5	0,9	1%	1,5	1%	2,7	0%	4,0	1%
Instância 6	1,0	42%	3,5	4%	3,7	1%	8,9	1%
Instância 7	0,5	59%	1,8	59%	12,4	2%	1,9	60%
Instância 8	0,9	49%	2,5	129%	3,2	0%	3,7	85%
Instância 9	0,7	42%	4,6	41%	3,6	67%	3,7	41%
Instância 10	0,7	3%	2,2	21%	2,1	8%	2,7	13%
Instância 11	0,6	17%	2,0	8%	4,4	8%	4,7	8%
Instância 12	0,8	278%	1,1	15%	3,9	284%	3,5	244%
Média para 10 tarefas	0,8	60%	2,5	52%	3,8	46%	4,5	59%

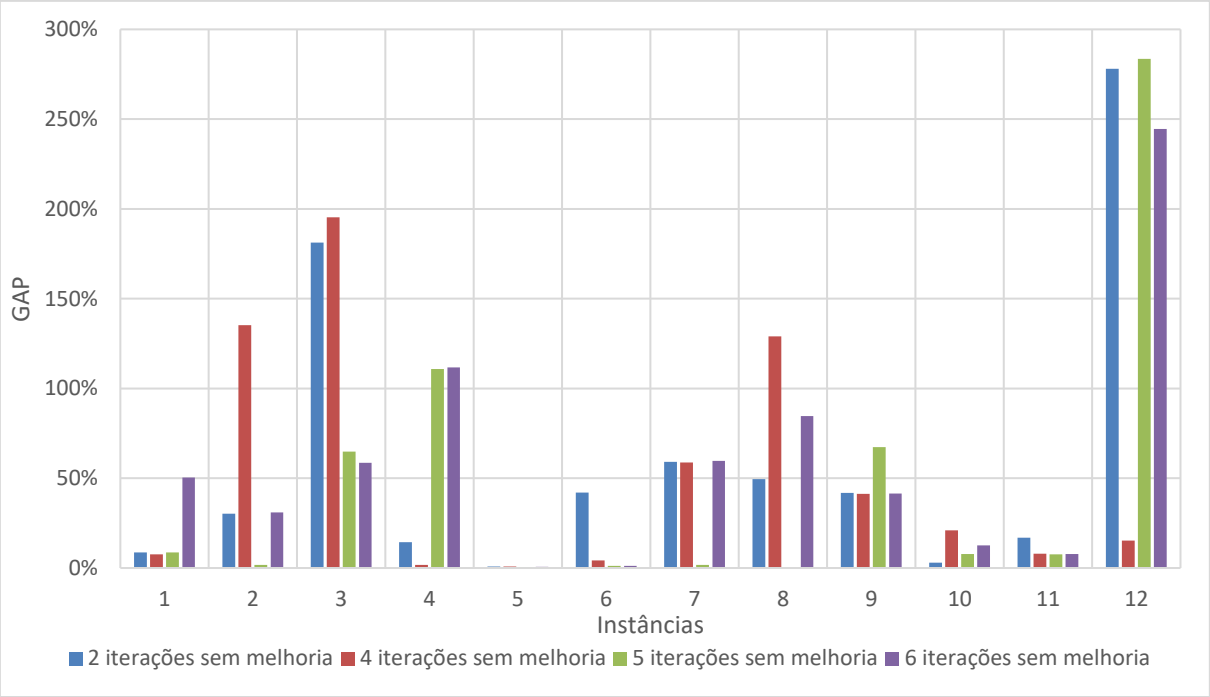

Figura 5-11 – Gráfico resultados Cenário 3 - 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-18 – Resultados Cenário 3 - 20 tarefas – Instante de Liberação

Instâncias	2 Iterações		4 Iterações		5 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 13	2,3	594%	8,6	145%	8,3	430%	9,1	226%
Instância 14	1,6	43%	4,8	12%	3,3	47%	9,2	52%
Instância 15	2,4	235%	11,2	90%	4,8	53%	10,4	185%
Instância 16	3,1	15%	8,8	83%	11,9	26%	12,8	22%
Instância 17	4,3	71%	15,5	292%	9,3	152%	18,6	140%
Instância 18	2,9	286%	19,8	118%	12,0	208%	31,3	89%
Instância 19	1,3	56%	5,1	-22%	10,4	-10%	6,1	-37%
Instância 20	1,5	-17%	6,5	2%	8,5	0%	4,0	14%
Instância 21	2,4	264%	8,6	74%	9,9	24%	22,1	66%
Instância 22	1,9	304%	4,9	544%	10,7	83%	11,5	342%
Instância 23	2,2	80%	16,2	9%	12,6	-1%	12,1	19%
Instância 24	2,6	45%	14,4	63%	21,4	39%	22,0	22%
Média para 20 tarefas	2,4	165%	10,4	118%	10,2	88%	14,1	95%

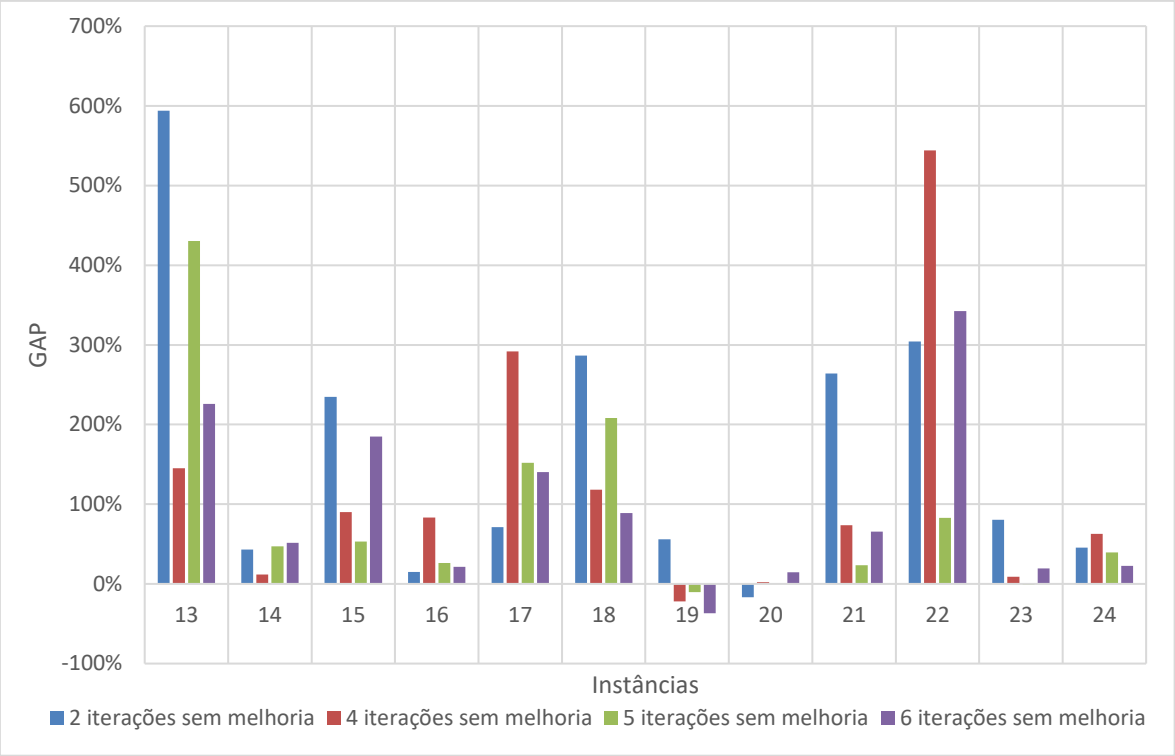

Figura 5-12 – Gráfico resultados Cenário 3 - 20 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Avaliando-se os resultados obtidos para o cenário 3 de calibração, que varia a quantidade de iterações sem melhoria, é possível se concluir que o aumento da quantidade de iterações do algoritmo melhora os resultados finais obtidos. Este comportamento pode ser verificado tanto para a regra de ordenação das tarefas por penalidade, como por instante de liberação para início da execução da tarefa. Através dos gráficos contidos nas Figura 5-9, Figura 5-10, Figura 5-11 e Figura 5-12, é possível se verificar que na maioria das instâncias este comportamento de redução do valor da solução final diminui, conforme se aumenta a quantidade de iterações sem melhoria.

Neste cenário, a média dos gaps obtidos para as instâncias de 10 tarefas foi menor para as rodadas com as tarefas ordenadas por penalidade, cujo menor valor obtido foi o de 38% para 6 iterações sem melhoria, enquanto que o melhor valor obtido para o caso da ordenação das tarefas por liberação para início foi de 46%, para 5 iterações sem melhoria. Já para as instâncias de 20 tarefas, a situação se inverte, sendo o melhor resultado obtido para a ordenação das tarefas por liberação para início foi de 88% para 5 iterações sem melhoria, e no caso da ordenação por penalidade foi

de 129% para 6 iterações sem melhoria. Em relação ao tempo de processamento, verificou-se que na variação de 2 iterações para 6 iterações houve aumento do tempo considerável, em que, percentualmente, apresentou um aumento de 494% e numericamente apresentou um aumento de 11,7 minutos para o caso de 20 tarefas e ordenação por liberação para início.

5.4.4 Resultados Cenário 4 – Tarefas Excluídas

Neste cenário 4 de calibração foi avaliada a quantidade de tarefas excluídas, sendo testadas variações com 1, 2, 3 e 4 tarefas excluídas. Como a variação da quantidade de tarefas excluídas impacta diretamente na quantidade iterações de reinserção das tarefas nas rotas das embarcações, também foi estudado o impacto na variação do tempo de processamento total para este cenário. Esta calibração foi processada tanto para a regra de ordenação das tarefas por penalidade, como por instante de liberação para início das tarefas.

Os resultados obtidos para o processamento com a ordenação das tarefas por penalidade estão apresentados na Tabela 5-19 e Figura 5-13Tabela 5-15, para as instâncias de 10 tarefas, e na Tabela 5-20 e Figura 5-14 para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no processamento do algoritmo e os obtidos no processamento do modelo matemático. Os tempos de processamento estão indicados em minutos. Os resultados obtidos para o processamento com a ordenação das tarefas por instante de liberação para início estão apresentados na Tabela 5-21 e Figura 5-15, para as instâncias de 10 tarefas, e na Tabela 5-22 e Figura 5-16 para as instâncias de 20 tarefas.

Avaliando-se os resultados obtidos para o cenário 4, que varia a quantidade de tarefas removidas, é possível concluir que o aumento da quantidade de tarefas removidas da solução inicial nos parâmetros da heurística melhora bastante os resultados finais obtidos. Este comportamento pode ser verificado tanto para a regra

de ordenação das tarefas por penalidade, como por instante de liberação para início da execução da tarefa. Através dos gráficos contidos nas figuras Figura 5-13, Figura 5-14, Figura 5-15 e Figura 5-16, é possível verificar que na maioria das instâncias este comportamento de redução do valor da solução final diminui, conforme se aumenta a quantidade de tarefas removidas.

Neste cenário, a diferença de desempenho entre os processamentos executados com a ordenação das tarefas por penalidade e por instante de liberação para início ficou mais evidente. Para as instâncias de 10 tarefas, as médias dos gaps dos cenários com a exclusão de 1 e 4 tarefas ficaram respectivamente em: 129% e 32% para a ordenação por penalidade e 57% e 8% para a ordenação por instante de liberação. A diferença de 24 pontos percentuais entre uma regra de ordenação e outra no cenário com a exclusão de 4 tarefas é bastante expressivo.

Nas instâncias com 20 tarefas, as médias dos gaps dos cenários com a exclusão de 1 e 4 tarefas ficaram respectivamente em: 226% e 72% para a ordenação por penalidade e 121% e 47% para a ordenação por instante de liberação. Um ponto importante para se ressaltar, para as instâncias de 20 tarefas é que, excluindo-se os resultados obtidos para as instâncias 13 e 15, que são pontos que ficaram fora da curva, a média dos gaps para as instâncias restantes ficou em 10%, indicando um resultado muito bom. Também, para as instâncias 19, 20 e 23, foram obtidos gaps de -36%, -30% e -19% respectivamente, indicando que a heurística conseguiu encontrar resultados até 36% melhores em pouco mais de 2 horas de processamento que é um desempenho excelente.

Analisando-se o impacto da variação dos parâmetros no tempo, pode-se notar que, conforme esperado, tempo médio para a resolução da simulação com a exclusão de 1 tarefa para as instâncias de 20 tarefas foi de 2,6 minutos, e conforme a quantidade de tarefas excluídas vai aumentando, o tempo também sofre grandes impactos, chegando a 193 minutos para 4 tarefas excluídas, que representa aproximadamente 74 vezes o tempo para a exclusão de 1 tarefa.

Tabela 5-19 – Resultados Cenário 4 - 10 tarefas – Penalidade

Instâncias	1 Tarefa		2 Tarefas		3 Tarefas		4 Tarefas	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 1	1,2	59%	2,7	25%	5,8	1%	74,8	6%
Instância 2	1,0	32%	1,6	0%	12,0	0%	59,8	0%
Instância 3	0,9	362%	3,1	321%	7,3	29%	69,4	276%
Instância 4	1,3	114%	3,0	111%	12,0	0%	64,4	0%
Instância 5	0,8	224%	2,6	0%	10,6	0%	57,4	0%
Instância 6	1,5	3%	2,2	2%	9,6	1%	80,9	2%
Instância 7	0,9	237%	1,1	2%	5,8	0%	23,9	0%
Instância 8	1,1	123%	5,5	0%	11,2	0%	34,5	41%
Instância 9	1,3	43%	2,2	143%	6,0	42%	58,7	42%
Instância 10	1,4	15%	4,6	10%	13,1	1%	49,2	1%
Instância 11	1,7	58%	2,6	22%	9,3	42%	47,2	0%
Instância 12	0,9	275%	3,5	282%	14,8	225%	90,8	13%
Média 10 tarefas	1,2	129%	2,9	77%	9,8	28%	59,3	32%

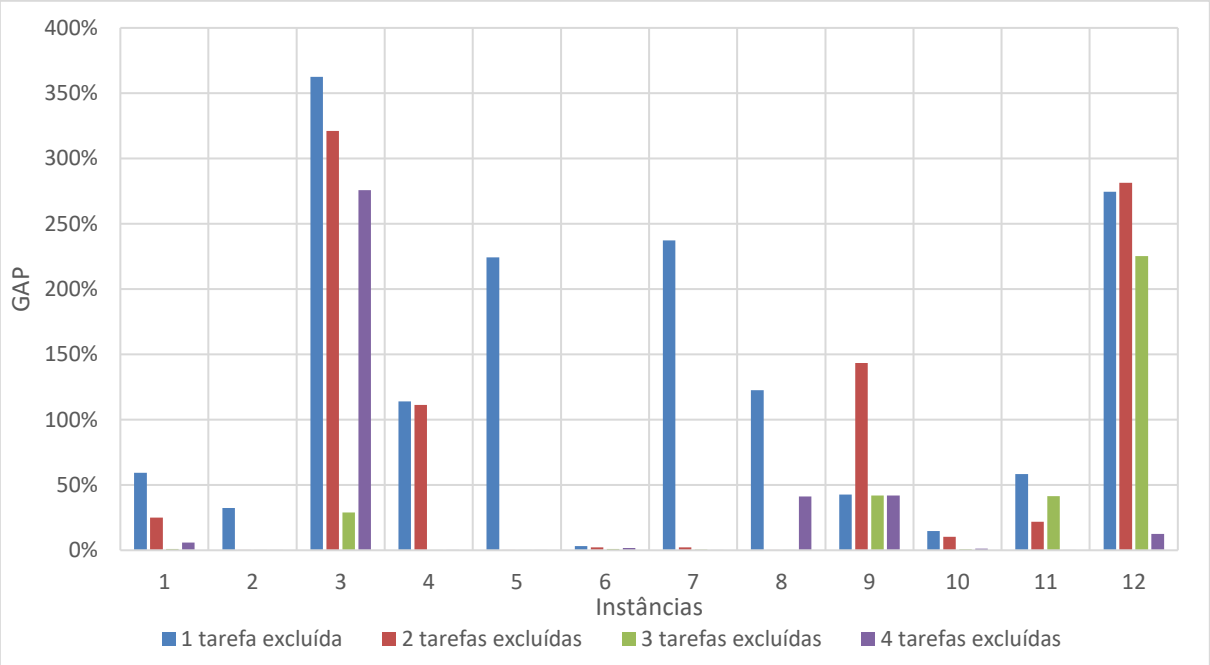

Figura 5-13 – Gráfico resultados Cenário 4 - 10 tarefas – Penalidade

Fonte: Elaborado pelo autor

Tabela 5-20 – Resultados Cenário 4 - 20 tarefas – Penalidade

Instâncias	1 Tarefa		2 Tarefas		3 Tarefas		4 Tarefas	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 13	2,3	1057%	11,0	525%	18,6	335%	278,0	369%
Instância 14	2,0	53%	4,8	87%	32,3	39%	180,0	-3%
Instância 15	3,5	271%	10,0	458%	34,1	222%	255,7	157%
Instância 16	2,7	93%	10,3	28%	14,0	73%	203,9	10%
Instância 17	4,4	246%	15,2	224%	38,8	89%	270,6	150%
Instância 18	2,8	214%	9,8	93%	35,2	42%	172,9	142%
Instância 19	1,9	-12%	3,4	-20%	12,9	-35%	67,6	-36%
Instância 20	1,8	-5%	7,2	4%	43,1	-20%	140,6	-23%
Instância 21	2,8	486%	11,2	297%	67,1	35%	237,0	9%
Instância 22	2,0	202%	5,0	200%	14,8	75%	173,9	86%
Instância 23	2,0	24%	6,2	15%	13,0	-3%	122,7	-22%
Instância 24	3,5	81%	11,3	50%	63,8	36%	222,5	23%
Média 20 tarefas	2,6	226%	8,8	164%	32,3	74%	193,8	72%

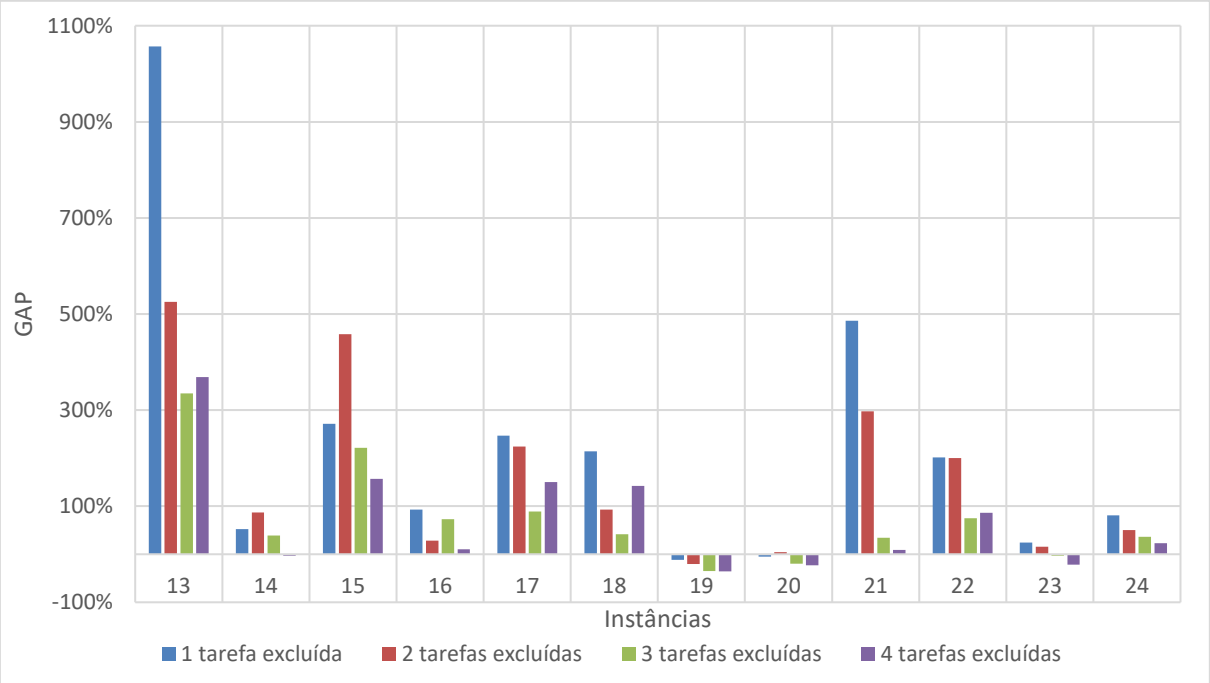

Figura 5-14 – Gráfico resultados Cenário 4 - 20 tarefas – Penalidade

Fonte: Elaborado pelo autor

Tabela 5-21 – Resultados Cenário 4 - 10 tarefas – Instante de Liberação

Instâncias	1 Tarefa		2 Tarefas		3 Tarefas		4 Tarefas	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 1	1,0	2%	2,4	8%	15,9	3%	49,9	1%
Instância 2	0,8	41%	3,0	135%	7,1	1%	81,7	0%
Instância 3	1,2	201%	3,4	195%	10,1	56%	56,6	24%
Instância 4	0,9	2%	2,2	2%	11,5	0%	60,4	1%
Instância 5	1,0	41%	1,5	1%	10,2	0%	33,9	0%
Instância 6	1,3	3%	3,5	4%	13,4	1%	80,9	1%
Instância 7	0,6	59%	1,8	59%	9,7	13%	31,6	0%
Instância 8	0,6	33%	2,5	129%	14,2	18%	60,0	1%
Instância 9	0,7	24%	4,6	41%	11,2	42%	85,3	40%
Instância 10	1,0	19%	2,2	21%	6,4	14%	47,1	6%
Instância 11	1,0	0%	2,0	8%	11,3	0%	36,7	0%
Instância 12	0,9	265%	1,1	15%	17,2	266%	31,5	19%
Média 10 tarefas	0,9	57%	2,5	52%	11,5	35%	54,6	8%

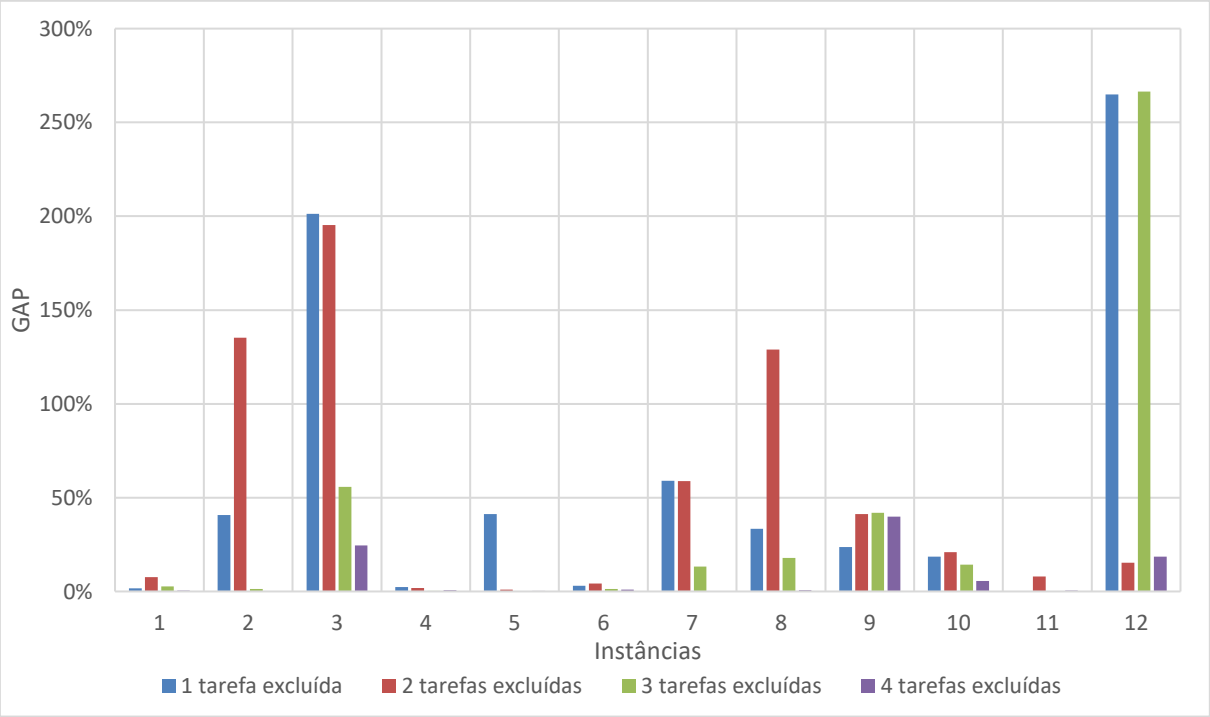

Figura 5-15 – Gráfico resultados Cenário 4 - 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-22 – Resultados Cenário 4 - 20 tarefas – Instante de Liberação

Instâncias	1 Tarefa		2 Tarefas		3 Tarefas		4 Tarefas	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
Instância 13	2,4	224%	8,6	145%	28,7	135%	108,0	309%
Instância 14	1,7	36%	4,8	12%	16,4	40%	201,8	10%
Instância 15	2,5	129%	11,2	90%	76,2	11%	68,2	150%
Instância 16	3,5	28%	8,8	83%	50,9	15%	251,0	14%
Instância 17	3,7	109%	15,5	292%	20,5	90%	211,6	10%
Instância 18	3,2	327%	19,8	118%	75,6	80%	235,4	42%
Instância 19	2,4	-29%	5,1	-22%	15,7	-29%	136,6	-36%
Instância 20	1,7	-18%	6,5	2%	17,5	24%	56,3	-30%
Instância 21	2,5	33%	8,6	74%	23,5	132%	108,3	11%
Instância 22	1,8	486%	4,9	544%	24,6	194%	127,8	92%
Instância 23	2,3	49%	16,2	9%	35,3	20%	65,3	-19%
Instância 24	4,5	73%	14,4	63%	28,6	33%	241,0	4%
Média 20 tarefas	2,7	121%	10,4	118%	34,5	62%	150,9	47%

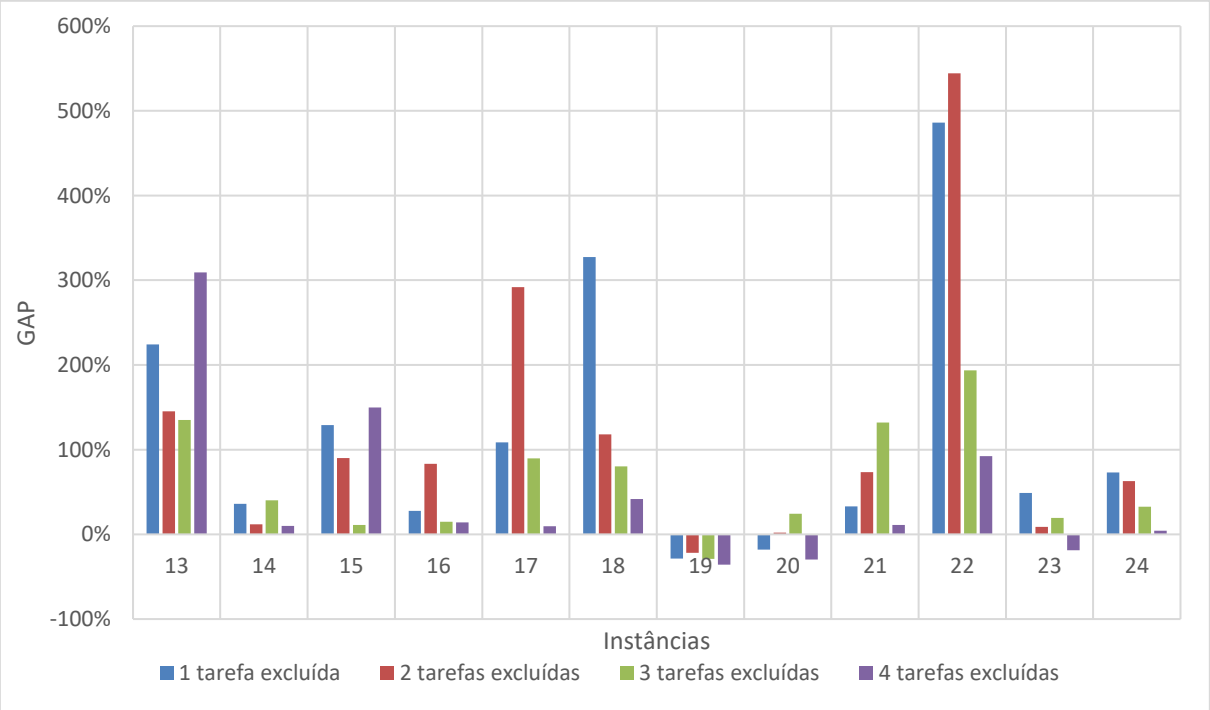

Figura 5-16 – Gráfico resultados Cenário 4 - 20 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

5.4.5 Considerações Finais – Método Seleção Aleatória

Analisando-se os resultados obtidos nos 4 cenários de calibração aplicados para o Método de Seleção Aleatória verificou-se que a ordenação das tarefas de acordo com os instantes de liberação para início de execução das tarefas e a ordenação aleatória foram mais efetivos do que a ordenação por penalidade. Também, que o aumento da quantidade de conjuntos de embarcações candidatas, da quantidade de iterações sem melhoria e da quantidade de tarefas removidas da solução base melhoram o desempenho do algoritmo no que diz respeito ao resultado final obtido porém penalizam na questão do tempo de processamento, que, para o caso da exclusão de tarefas, no teste com 4 tarefas excluídas, chegou-se a um aumento de 74 vezes do tempo para 1 tarefa excluída.

As tabelas: Tabela 5-23, Tabela 5-24, Tabela 5-25 e Tabela 5-26 contém um resumo dos resultados obtidos nesta fase de calibração dos parâmetros de entrada do algoritmo, e apresentam os valores médios de GAP e Tempo para as instâncias de 10 e 20 tarefas e para as regras de ordenação por penalidade e por instante de liberação para início.

Tabela 5-23 – Cenário 1 – Tarefas Candidatas – Valores Médios

Instâncias	1 tarefa	3 tarefas	5 tarefas	7 tarefas	N tarefas
	Gap	Gap	Gap	Gap	Gap
10 tarefas Penalidade	106%	77%	72%	84%	45%
20 tarefas Penalidade	179%	164%	155%	137%	190%
10 tarefas Início	54%	52%	83%	177%	-
20 tarefas Início	128%	118%	169%	154%	-

Tabela 5-24 – Cenário 2 – Conjuntos de Embarcações – Valores Médios

Instâncias	25 Conjuntos		50 Conjuntos		75 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
10 tarefas Penalidade	1,0	139%	1,2	97%	2,5	121%	2,9	77%
20 tarefas Penalidade	3,1	229%	3,2	189%	4,7	155%	8,8	164%
10 tarefas Início	1,0	92%	1,3	96%	1,3	66%	2,5	52%
20 tarefas Início	3,2	204%	3,3	170%	3,6	166%	10,4	118%

Tabela 5-25 – Cenário 3 – Iterações sem Melhoria – Valores Médios

Instâncias	2 Iterações		4 Iterações		5 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
10 tarefas Penalidade	1,1	88%	2,9	77%	5,5	53%	5,7	38%
20 tarefas Penalidade	2,8	173%	8,8	164%	8,5	149%	14,1	129%
10 tarefas Início	0,8	60%	2,5	52%	3,8	46%	4,5	59%
20 tarefas Início	2,4	165%	10,4	118%	10,2	88%	14,1	95%

Tabela 5-26 – Cenário 4 – Tarefas Excluídas – Valores Médios

Instâncias	1 Tarefa		2 Tarefas		3 Tarefas		4 Tarefas	
	Minutos	Gap	Minutos	Gap	Minutos	Gap	Minutos	Gap
10 tarefas Penalidade	1,2	129%	2,9	77%	9,8	28%	59,3	32%
20 tarefas Penalidade	2,6	226%	8,8	164%	32,3	74%	193,8	72%
10 tarefas Início	0,9	57%	2,5	52%	11,5	35%	54,6	8%
20 tarefas Início	2,7	121%	10,4	118%	34,5	62%	150,9	47%

Analisando-se os resultados obtidos para cada uma das instâncias de testes processadas, pode-se verificar que as instâncias que, repetidas vezes, apresentaram um resultado abaixo de 10% de gap foram: 2; 6; 10; 19; 23. Já as instâncias que apresentaram os piores desempenhos, com gaps acima dos 100% foram: 3; 12; 13; 15; 17; 18; 22.

5.5 Calibração de Parâmetros - Método 2 – Seleção por Penalidade das Tarefas

O Método 2 - Seleção por Penalidades também foi aplicado ao mesmo conjunto de 24 instâncias elencadas no subitem 5.1, em que foram processados testes para calibração, visando compreender como o algoritmo se comporta conforme alguns dos seus parâmetros são alterados, vide subitem 5.2. Nesta fase de calibração, os valores dos parâmetros foram definidos de maneira que, preferencialmente, não se tenha períodos de processamento muito longos. As rodadas foram executadas no mesmo hardware e nas mesmas condições da seção anterior.

5.5.1 Resultados Cenário 1 – Tarefas Candidatas

Como os resultados obtidos para o Método 1 - Seleção Aleatória indicaram que a ordenação por instante de liberação para início da execução das tarefas e a ordenação aleatória obtém melhores resultados, quando comparados com a regra de ordenação por penalidade diária de cada tarefa, para o Método 2, foram apenas testados alguns dos cenários com a ordenação das tarefas por instante de liberação e por ordenação aleatória. Os resultados obtidos para o processamento com a ordenação das tarefas por penalidade estão apresentados na Tabela 5-27 e Figura 5-17, para as instâncias de 10 tarefas, e na Tabela 5-28 e Figura 5-18, para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no processamento do algoritmo de Seleção por Penalidades, e os obtidos no processamento do modelo matemático.

Neste cenário 1 de calibração relativa à quantidade de tarefas candidatas, foram testadas apenas para as variações com 1, 7 e n (que corresponde a quantidade total de tarefas para cada instância = 10 ou 20). Isto significa que foram testadas as seguintes configurações: i) configuração em que a primeira solução da fase construtiva segue exatamente a regra de ordenação das tarefas (quantidade = 1); ii) configuração intermediária, que mescla a seleção conforme regra de ordenação e a seleção aleatória; iii) configuração em que a seleção de tarefas candidatas é feita de maneira totalmente aleatória (quantidade = n).

Semelhante aos resultados obtidos no Método 1 de seleção aleatória, a análise dos resultados indica que, para as instâncias de 10 tarefas, com o aumento da aleatoriedade da seleção das tarefas os resultados melhoram, saindo de uma média de 88% para uma tarefa candidata para 39% para seleção totalmente aleatória. Já para as instâncias de 20 tarefas, os resultados se invertem, apontando que, com a regra de ordenação das tarefas de acordo com o instante de liberação para início, os resultados são melhores do que os obtidos com a seleção aleatória. As médias dos gaps obtidos nestas instâncias de 20 tarefas foram de 136% para seleção aleatória e

109% para seleção por instante de liberação para início. Nos gráficos presentes nas Figura 5-17 e Figura 5-18 é possível se notar estas tendências.

Tabela 5-27 - Resultados Cenário 1 - 10 tarefas – Instante de Liberação

Instâncias	1 tarefa	7 tarefas	n tarefas
	Gap	Gap	Gap
Instância 1	19%	7%	17%
Instância 2	21%	2%	12%
Instância 3	44%	81%	81%
Instância 4	114%	1%	6%
Instância 5	0%	14%	0%
Instância 6	133%	70%	4%
Instância 7	207%	17%	1%
Instância 8	81%	79%	18%
Instância 9	43%	1%	41%
Instância 10	12%	21%	2%
Instância 11	58%	0%	58%
Instância 12	325%	238%	227%
Média para 10 tarefas	**88%**	**44%**	**39%**

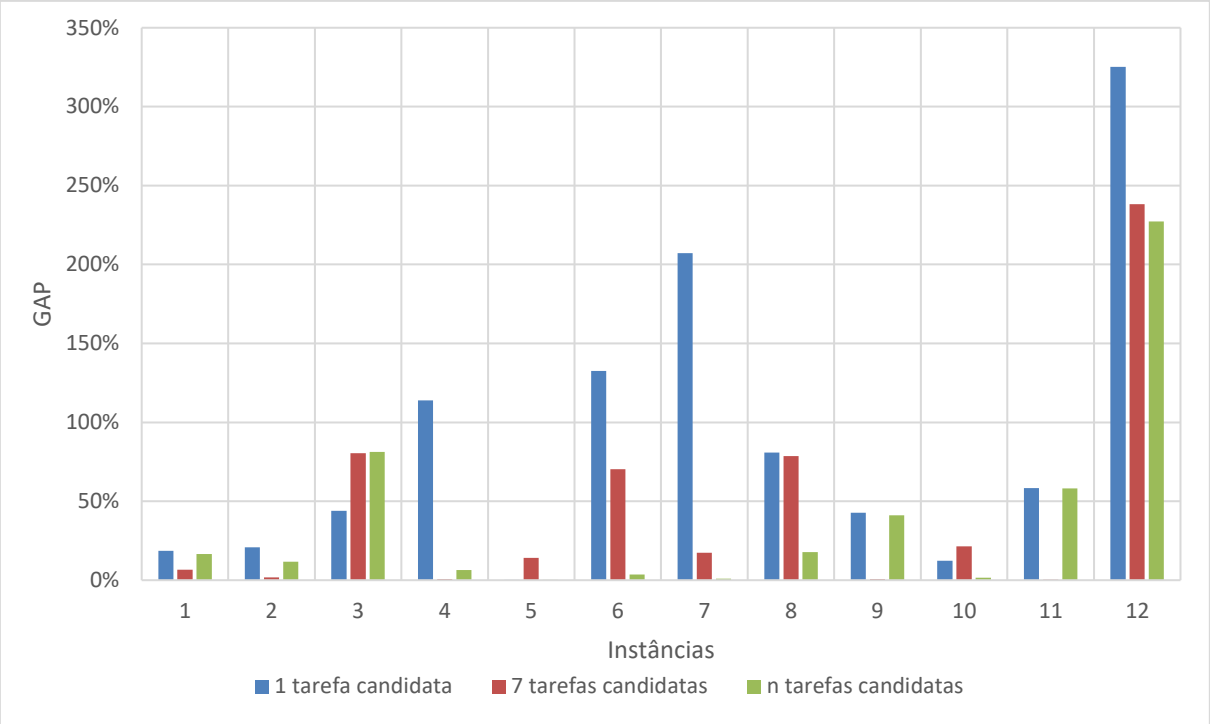

Figura 5-17 – Gráfico resultados Cenário 1 - 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-28 – Resultados Cenário 1 – 20 tarefas – Instante de Liberação

Instâncias	1 tarefa	7 tarefas	n tarefas
	Gap	Gap	Gap
Instância 13	265%	610%	349%
Instância 14	49%	50%	74%
Instância 15	109%	64%	326%
Instância 16	5%	36%	37%
Instância 17	238%	161%	133%
Instância 18	173%	100%	112%
Instância 19	-17%	-24%	-4%
Instância 20	9%	-15%	-9%
Instância 21	106%	332%	218%
Instância 22	333%	155%	338%
Instância 23	31%	30%	10%
Instância 24	11%	30%	43%
Média para 20 tarefas	**109%**	**128%**	**136%**

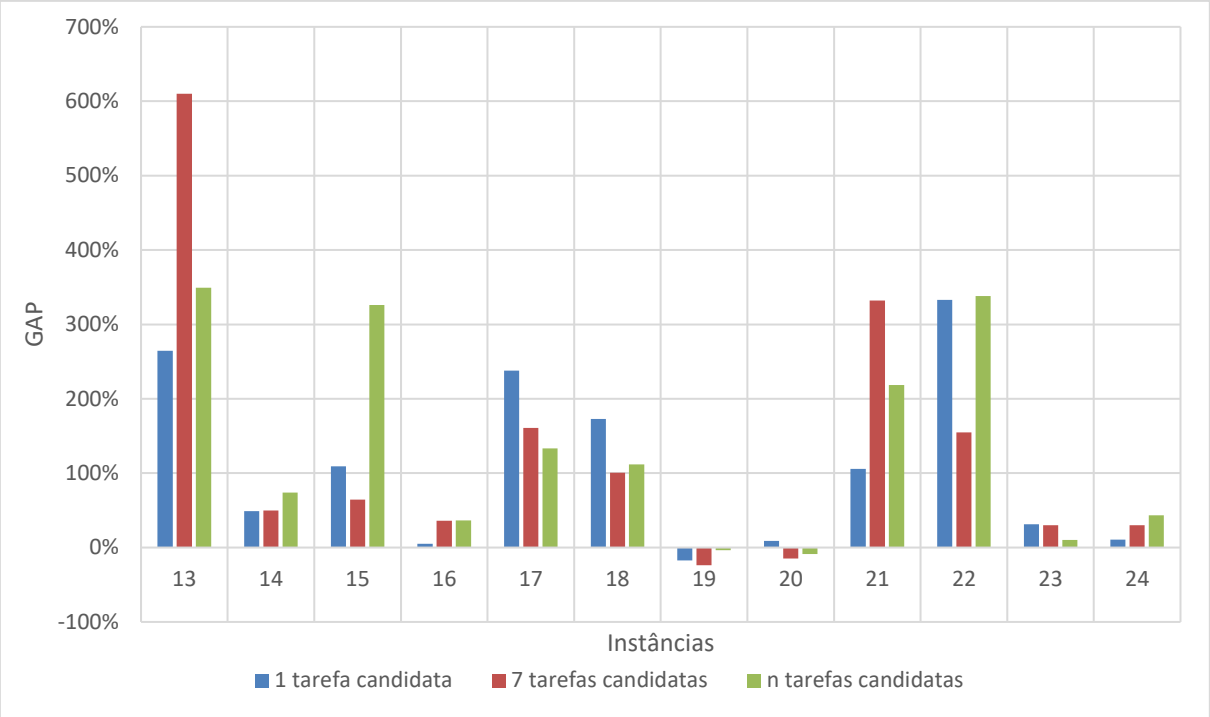

Figura 5-18 – Gráfico resultados Cenário 1 – 20 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

5.5.2 Resultados Cenário 2 – Conjuntos de Embarcações Candidatas

Assim como os testes feitos para o cenário 1, os processamentos do cenário 2 de calibração foram rodados apenas para a regra de ordenação das tarefas por instante de liberação para início das tarefas.

Para o cenário 2, referente à quantidade de conjuntos de embarcações candidatas, foram testados apenas os casos mais extremos com 25 e 100 conjuntos candidatos. Os resultados obtidos para a rodada com a ordenação das tarefas por instante de liberação para início estão apresentados na Tabela 5-29 e Figura 5-19, para as instâncias de 10 tarefas, e na Tabela 5-30 e Figura 5-20 para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no processamento do algoritmo e os obtidos no processamento do modelo matemático. Os tempos de processamento estão indicados em minutos.

Tabela 5-29 – Resultados Cenário 2 - 10 tarefas – Liberação para Início

Instâncias	25 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap
Instância 1	3,3	8%	3,4	1%
Instância 2	2,4	41%	6,0	2%
Instância 3	2,2	167%	2,8	53%
Instância 4	2,7	1%	4,6	0%
Instância 5	2,7	3%	3,5	0%
Instância 6	2,5	59%	3,5	4%
Instância 7	1,9	66%	2,2	4%
Instância 8	2,0	79%	3,7	0%
Instância 9	1,7	43%	2,3	43%
Instância 10	2,1	13%	4,9	11%
Instância 11	2,2	17%	4,6	16%
Instância 12	2,2	39%	2,9	190%
Média 10 tarefas	**2,3**	**45%**	**3,7**	**27%**

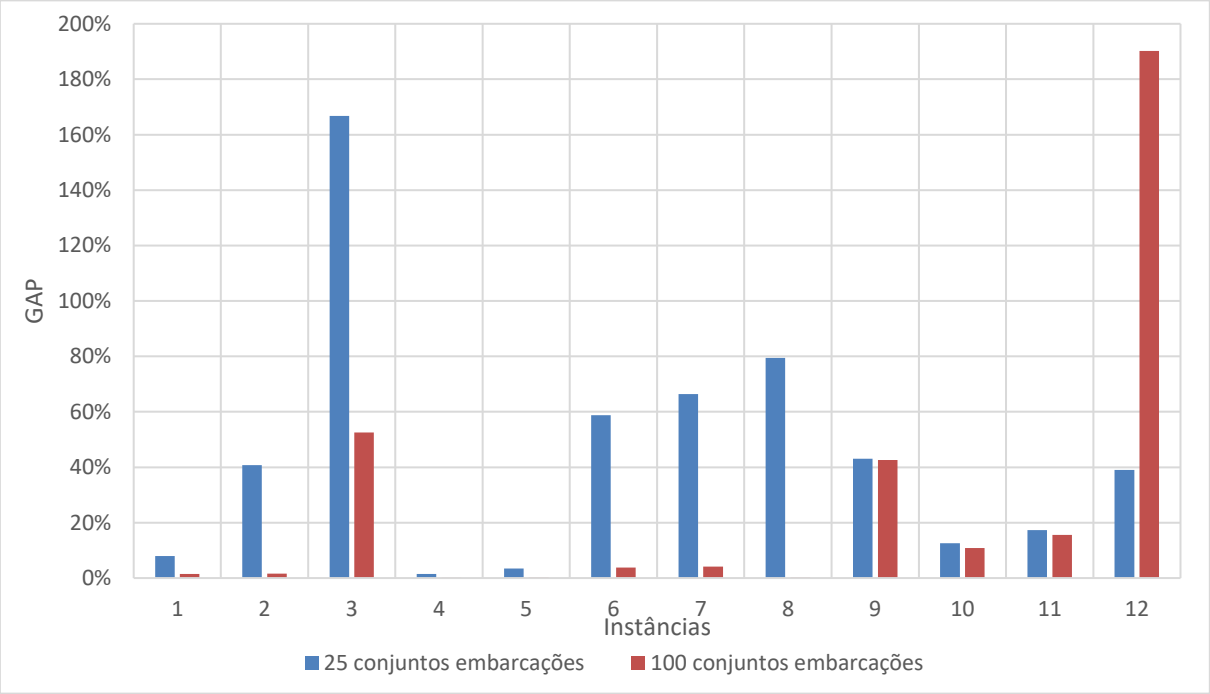

Figura 5-19 – Gráfico resultados Cenário 2 - 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-30 – Resultados Cenário 2 - 20 tarefas – Instante de Liberação

Instâncias	25 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap
Instância 13	22,1	274%	6,8	396%
Instância 14	6,8	86%	4,3	69%
Instância 15	4,2	530%	12,8	178%
Instância 16	18,1	25%	13,2	14%
Instância 17	18,2	26%	18,1	89%
Instância 18	12,6	215%	18,0	104%
Instância 19	8,6	13%	8,4	-10%
Instância 20	4,0	13%	8,3	-19%
Instância 21	26,6	11%	8,3	4%
Instância 22	15,4	270%	9,5	356%
Instância 23	9,6	24%	13,5	52%
Instância 24	24,9	37%	13,3	8%
Média 20 tarefas	**14,2**	**127%**	**11,2**	**103%**

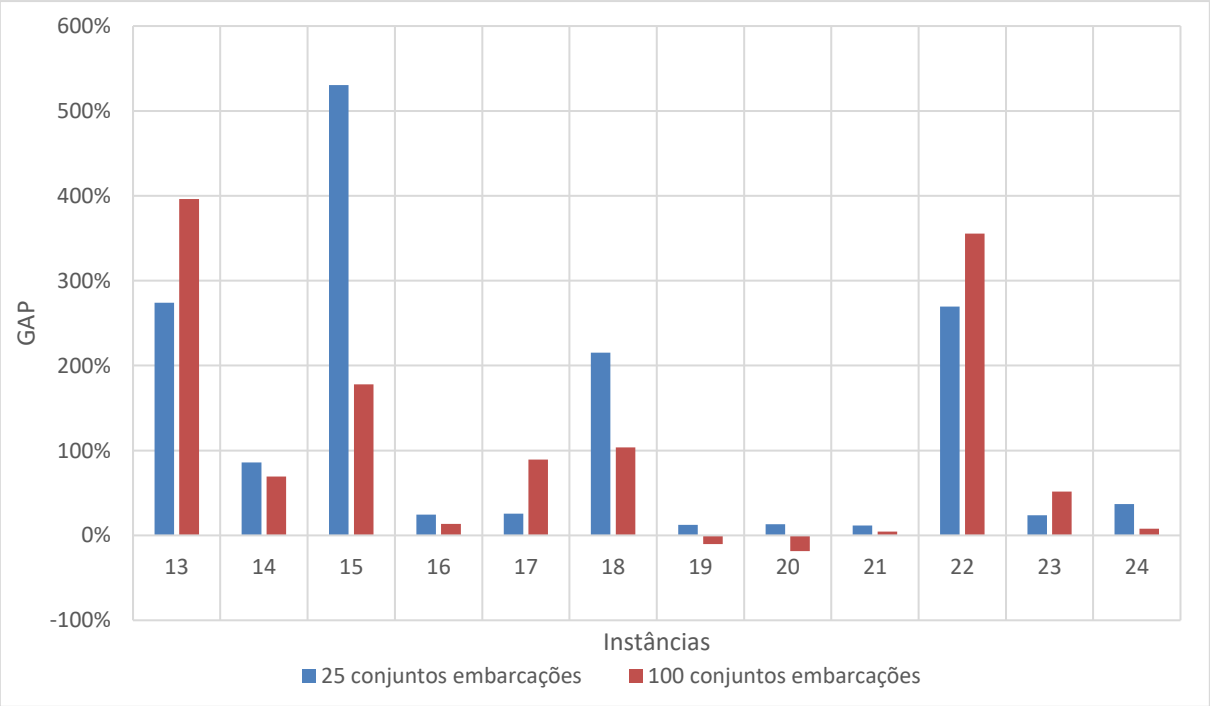

Figura 5-20 – Gráfico resultados Cenário 2 - 20 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Os resultados obtidos nos processamentos do cenário 2 indicam uma melhoria no valor da solução final, conforme se aumenta a quantidade de conjuntos de embarcações. A média dos gaps obtidos para as instâncias com 10 tarefas foi de 45% para 25 conjuntos e 27% para 100 conjuntos, e para as instâncias de 20 tarefas, se obtiveram médias de 127% para 25 conjuntos e 103% para 100 conjuntos. Nos gráficos apresentados nas Figura 5-19 e Figura 5-20 pode-se verificar a melhoria do desempenho do algoritmo quando se aumenta a quantidade de conjuntos de embarcações candidatas.

Os tempos médios de processamento tiveram uma pequena variação entre os processamentos com 25 e 100 conjuntos, indicando que o impacto no tempo, do aumento da quantidade de conjuntos, foi pequeno. Neste caso em particular, a média dos tempos para as simulações com 25 conjuntos foi maior do que a média dos tempos para as simulações com 100 conjuntos. Isto se deve ao desempenho do computador que processou este cenário estar rodando outras tarefas em paralelo ao programa que simula o algoritmo, o que pode variar o tempo de processamento das soluções.

5.5.3 Resultados Cenário 3 – Iterações Sem Melhoria

Os testes do cenário 3 de calibração foram rodados apenas para a regra de ordenação das tarefas por instante de liberação para início das tarefas. Neste cenário 3, referente à quantidade de iterações sem melhoria, apenas foram consideradas as variações com 2 e 6 iterações sem melhoria. Como a variação da quantidade de iterações sem melhoria causa impacto direto no tempo total de processamento, também foi estudada a variação do tempo de processamento total para este cenário. Os resultados obtidos para o processamento com a ordenação das tarefas por penalidade estão apresentados na Tabela 5-31 e Figura 5-21, para as instâncias de 10 tarefas, e na Tabela 5-32 e Figura 5-22 para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no

processamento do algoritmo e os obtidos no processamento do modelo matemático. Os tempos de processamento estão indicados em minutos.

Tabela 5-31 – Resultados Cenário 3 - 10 tarefas – Instante de Liberação

Instâncias	2 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap
Instância 1	1,3	30%	3,4	1%
Instância 2	1,4	30%	6,5	1%
Instância 3	1,3	512%	4,2	38%
Instância 4	16,3	2%	7,9	1%
Instância 5	1,5	2%	3,3	30%
Instância 6	1,6	147%	14,3	2%
Instância 7	1,3	1%	3,1	3%
Instância 8	1,3	65%	4,9	0%
Instância 9	1,4	43%	5,8	41%
Instância 10	1,4	13%	5,5	12%
Instância 11	1,3	58%	3,7	58%
Instância 12	1,7	226%	6,6	252%
Média para 10 tarefas	**2,6**	**94%**	**5,8**	**37%**

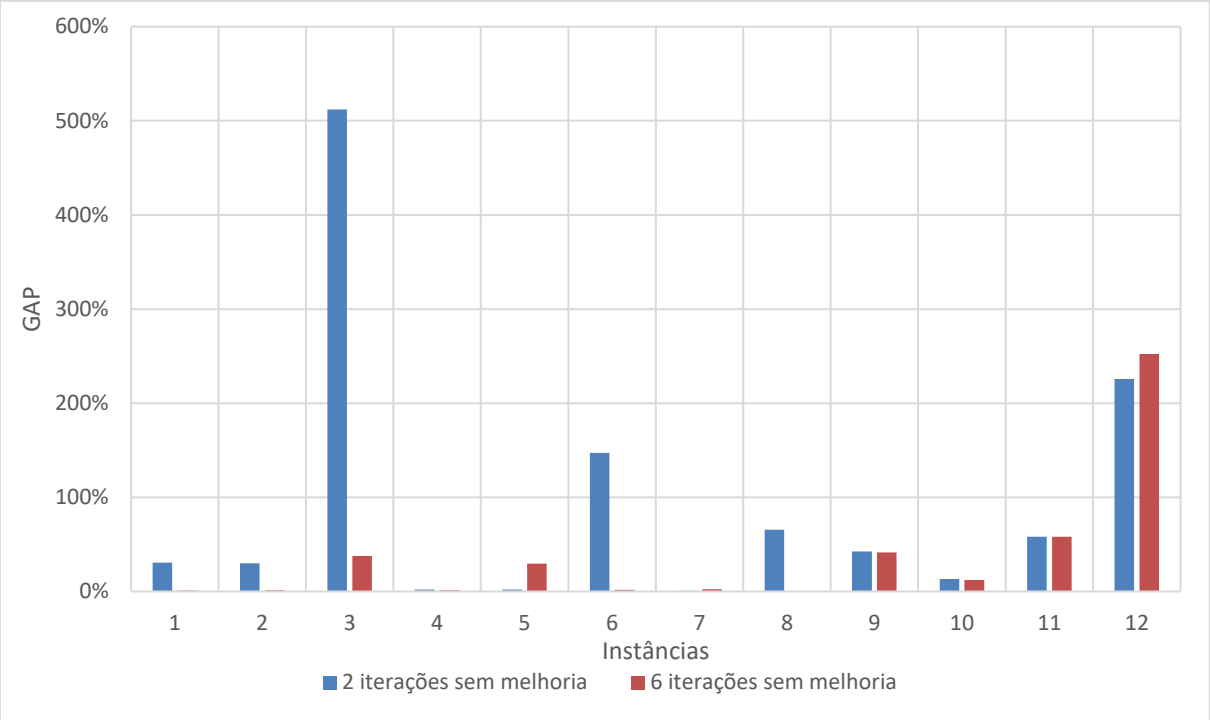

Figura 5-21 – Gráfico resultados Cenário 3 - 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-32 – Resultados Cenário 3 - 20 tarefas – Instante de Liberação

Instâncias	2 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap
Instância 13	6,0	393%	31,1	238%
Instância 14	2,7	92%	6,2	73%
Instância 15	5,7	205%	17,0	32%
Instância 16	5,5	21%	11,1	24%
Instância 17	6,3	173%	40,5	179%
Instância 18	9,0	232%	28,0	192%
Instância 19	2,7	-34%	24,8	-34%
Instância 20	2,9	-19%	16,2	2%
Instância 21	4,4	46%	15,9	3%
Instância 22	4,1	470%	16,3	92%
Instância 23	4,1	14%	13,7	42%
Instância 24	5,4	52%	48,7	13%
Média para 20 tarefas	**4,9**	**137%**	**22,4**	**71%**

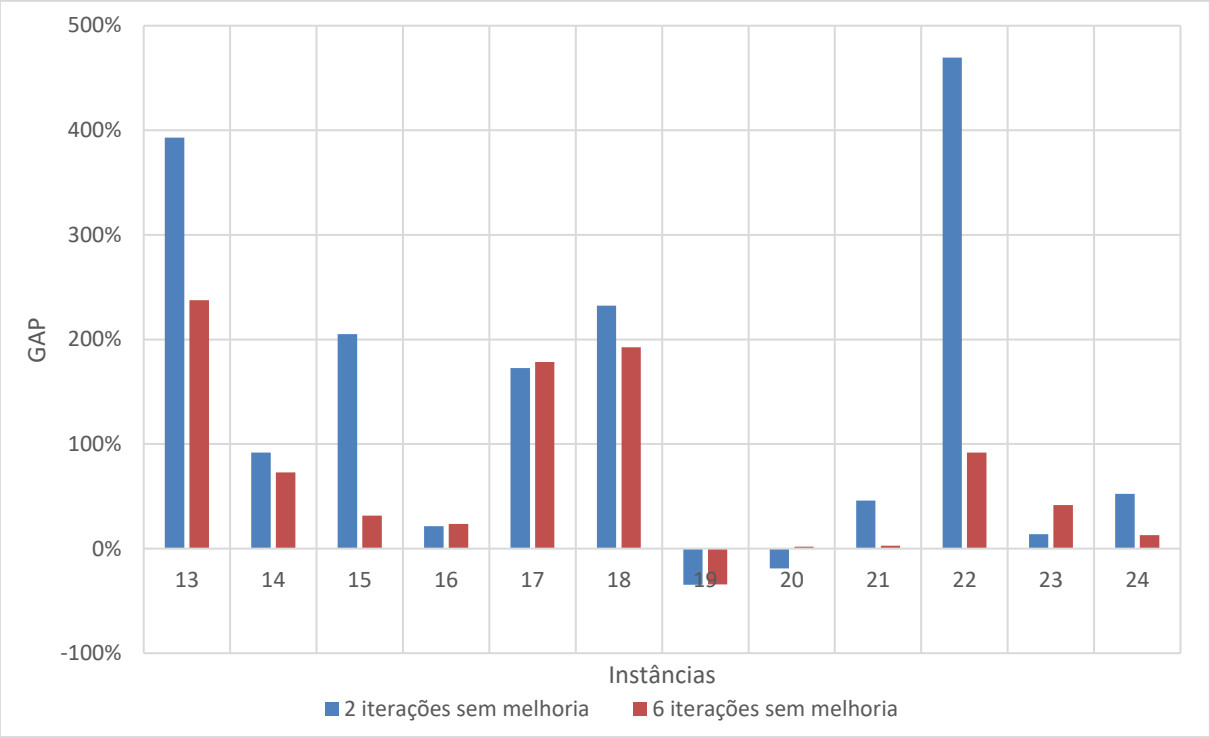

Figura 5-22 – Gráfico resultados Cenário 3 - 20 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Os resultados obtidos no Cenário 3 podem ser visualizados nos gráficos presentes nas Figura 5-21 e Figura 5-22. Nestes gráficos é possível se notar que, conforme verificado no Método 1, o aumento da quantidade de iterações sem melhoria apresenta uma boa melhora de resultado, em que nas rodadas das instâncias de 10 tarefas, o gap médio salta de 94% para 2 iterações sem melhoria para 37% para 6 iterações sem melhoria, o que representa uma redução de 61%. Já para as instâncias de 20 tarefas, o gap médio varia de 137% para 2 iterações sem melhoria para 71% para 6 iterações sem melhoria, representando uma melhoria de 48% nos resultados finais.

O tempo de processamento médio obtido para as instâncias de 10 tarefas variou de 2,6 minutos para 5,8 minutos para 2 iterações sem melhoria e 6 iterações sem melhoria, respectivamente. Para as instâncias de 20 tarefas, o tempo médio de processamento ficou em 4,9 minutos e 22,4 minutos, para as simulações com 2 e 6 iterações sem melhoria.

5.5.1 Resultados Cenário 4 – Tarefas Excluídas

Os testes do cenário 4 de calibração foram feitos para a regra de ordenação das tarefas por instante de liberação. Neste cenário 4, que estuda os impactos da quantidade de tarefas excluídas, foram testadas apenas as variações com 1 e 4 tarefas excluídas. Como a variação da quantidade de tarefas excluídas causa impacto direto na quantidade iterações de reinserção das tarefas nas rotas das embarcações, também foi estudado o impacto na variação do tempo de processamento total para este cenário.

Os resultados obtidos para o processamento com a ordenação das tarefas por instante de liberação para início estão apresentados na Tabela 5-33 e Figura 5-23 para as instâncias de 10 tarefas, e na Tabela 5-34 e Figura 5-24 para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no processamento do algoritmo e os obtidos no processamento do modelo matemático. Os tempos de processamento estão indicados em minutos.

Avaliando-se os resultados obtidos para o cenário 4, que varia a quantidade de tarefas removidas, é possível se concluir que o aumento da quantidade de tarefas removidas da solução inicial nos parâmetros da heurística melhora bastante os resultados finais obtidos. Através dos gráficos contidos nas figuras: Figura 5-23 e Figura 5-24, é possível se verificar que em todas as instâncias, este comportamento de redução do valor da solução final diminuiu, conforme se aumentou a quantidade de tarefas removidas.

Para as instâncias de 10 tarefas, as médias dos gaps dos cenários com a exclusão de 1 e 4 tarefas ficaram respectivamente em: 107% e 5%. A diferença entre a média dos resultados obtidos para os testes com exclusão de 1 e 4 tarefas é bastante expressiva, e a média de 5% de variação para as instâncias com 10 tarefas é um resultado muito bom.

Tabela 5-33 – Resultados Cenário 4 - 10 tarefas – Instante de Liberação

Instâncias	1 Tarefa		4 Tarefas	
	Minutos	Gap	Minutos	Gap
Instância 1	1,4	6%	52,2	2%
Instância 2	0,9	56%	50,8	0%
Instância 3	1,3	222%	97,3	0%
Instância 4	1,2	115%	77,6	0%
Instância 5	1,3	1%	60,0	0%
Instância 6	1,7	147%	100,4	2%
Instância 7	0,7	203%	32,6	1%
Instância 8	1,3	152%	57,9	18%
Instância 9	1,2	42%	73,4	1%
Instância 10	0,9	124%	77,3	3%
Instância 11	1,2	9%	51,9	0%
Instância 12	1,1	214%	163,0	27%
Média 10 tarefas	**1,2**	**107%**	**74,5**	**5%**

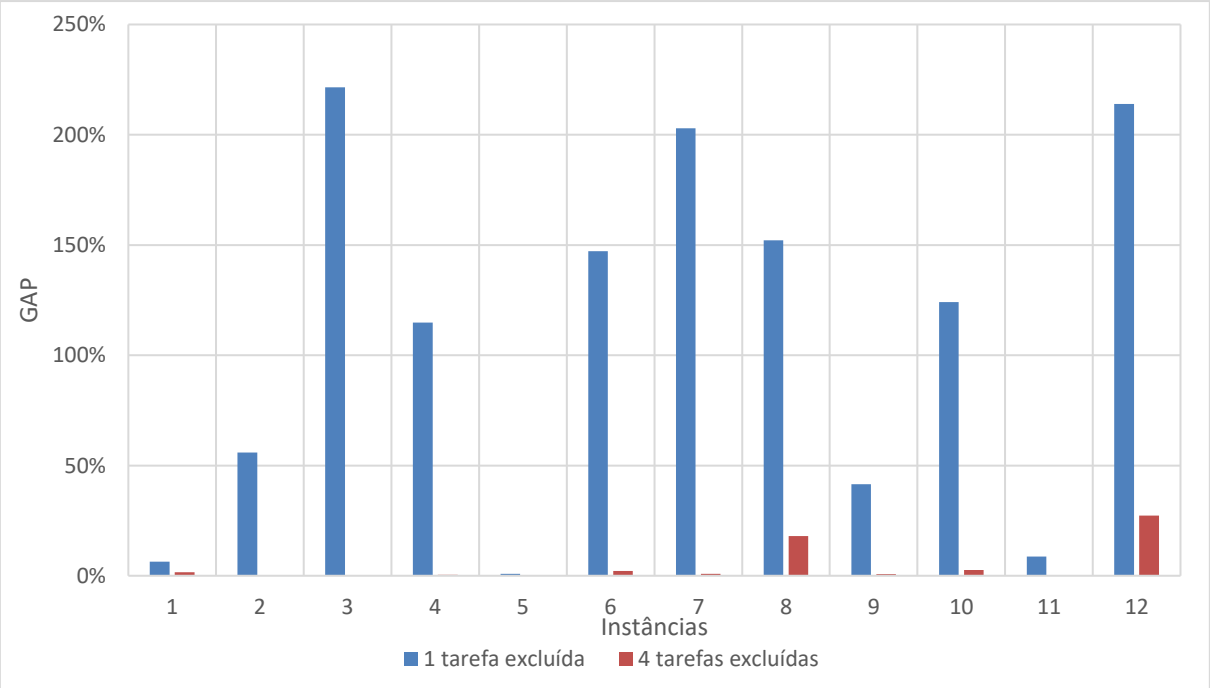

Figura 5-23 – Gráfico resultados Cenário 4 - 10 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Tabela 5-34 – Resultados Cenário 4 - 20 tarefas – Instante de Liberação

Instâncias	1 Tarefa		4 Tarefas	
	Minutos	Gap	Minutos	Gap
Instância 13	4,5	297%	245,9	158%
Instância 14	2,6	23%	111,5	17%
Instância 15	4,1	183%	241,2	13%
Instância 16	4,2	49%	420,5	0%
Instância 17	4,1	328%	253,6	128%
Instância 18	4,5	184%	419,5	67%
Instância 19	2,3	-11%	134,0	-17%
Instância 20	3,0	19%	228,7	-16%
Instância 21	3,2	131%	157,7	5%
Instância 22	2,5	379%	139,2	130%
Instância 23	3,1	27%	195,8	-16%
Instância 24	6,1	94%	628,7	-13%
Média 20 tarefas	**3,7**	**142%**	**264,7**	**38%**

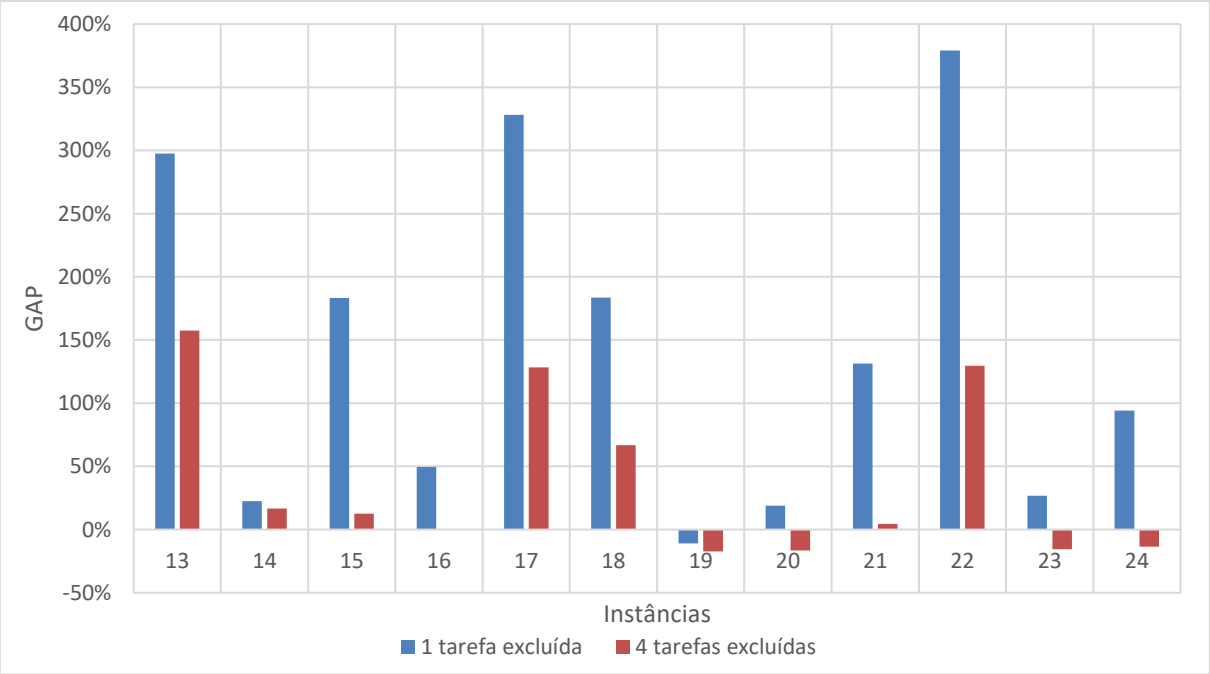

Figura 5-24 – Gráfico resultados Cenário 4 - 20 tarefas – Instante de Liberação

Fonte: Elaborado pelo autor

Nas instâncias com 20 tarefas, as médias dos gaps dos cenários com a exclusão de 1 e 4 tarefas ficaram respectivamente em: 142% e 38%. Um ponto importante para se ressaltar para as instâncias de 20 tarefas é que, excluindo-se os resultados obtidos para as instâncias 13 e 22, que são os dois maiores pontos e que ficaram fora da curva, a média dos gaps para as instâncias restantes ficou em 17%, que indica um resultado muito bom. Também, para as instâncias 19, 20, 23 e 24 foram obtidos gaps de -17%, -16%, -16% e -13% respectivamente, indicando que a heurística conseguiu encontrar resultados até 17% melhores em pouco mais de 4 horas de processamento.

Analisando-se o impacto da variação dos parâmetros no tempo de processamento, pode-se notar que, conforme esperado, o tempo médio de processamento com a exclusão de 1 tarefa para as instâncias de 20 tarefas foi de 3,7 minutos, e conforme a quantidade de tarefas excluídas vai aumentando, o tempo também sofre grandes impactos, chegando a 264 minutos para 4 tarefas excluídas, que representa aproximadamente 71 vezes o tempo necessário para a exclusão de 1 tarefa.

5.5.2 Considerações Finais – Método Seleção Por Penalidade das Tarefas

Analisando os resultados obtidos nos 4 cenários de calibração testados para o Método de Seleção por Penalidades verificou-se que o aumento da quantidade de conjuntos de embarcações candidatas, da quantidade de iterações sem melhoria e da quantidade de tarefas removidas da solução base melhoram o desempenho do algoritmo no que diz respeito ao resultado final obtido porém penalizam o tempo de processamento que, para o caso da exclusão de tarefas, o processamento com 4 tarefas excluídas, chegou-se a um aumento de 71 vezes do tempo quando comparado com a exclusão de uma única tarefa. Em relação à regra de ordenação das tarefas, foi possível verificar que, assim como ocorrido no Método 1, a seleção aleatória apresenta os melhores resultados para as instâncias com 10 tarefas, e a ordenação por instante de liberação para início, para as instâncias com 20 tarefas.

As tabelas: Tabela 5-35, Tabela 5-36, Tabela 5-37 e Tabela 5-38 contém um resumo dos resultados obtidos nesta fase de calibração dos parâmetros de entrada do algoritmo, e apresentam os valores médios de gap e tempo para as instâncias de 10 e 20 tarefas para a regra de ordenação por instante de liberação, testadas para o Método 2 de Seleção por Penalidade das Tarefas.

Tabela 5-35 – Cenário 1 – Tarefas Candidatas - Valores Médios

Instâncias	1 tarefa	7 tarefas	n tarefas
	Gap	Gap	Gap
Média para 10 tarefas	**88%**	**44%**	**39%**
Média para 20 tarefas	**109%**	**128%**	**136%**

Tabela 5-36 – Cenário 2 – Conjuntos de Embarcações - Valores Médios

Instâncias	25 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap
Média 10 tarefas	**2,3**	**45%**	**3,7**	**27%**
Média 20 tarefas	**14,2**	**127%**	**11,2**	**103%**

Tabela 5-37 – Cenário 3 – Iterações sem Melhoria - Valores Médios

Instâncias	2 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap
Média para 10 tarefas	**2,6**	**94%**	**5,8**	**37%**
Média para 20 tarefas	**4,9**	**137%**	**22,4**	**71%**

Tabela 5-38 – Cenário 4 – Tarefas Excluídas - Valores Médios

Instâncias	1 Tarefa		4 Tarefas	
	Minutos	Gap	Minutos	Gap
Média 10 tarefas	**1,2**	**107%**	**74,5**	**5%**
Média 20 tarefas	**3,7**	**142%**	**264,7**	**38%**

Analisando-se os resultados obtidos para cada um dos parâmetros de entrada simulados, pode-se verificar que as instâncias que, repetidas vezes, apresentaram um

resultado abaixo de 10% de gap foram: 4; 5; 7; 19; 20. Já as instâncias que apresentaram os piores desempenhos, com gaps acima dos 100% foram: 12; 13; 17; 18; 22.

5.6 Resultados Finais

Com os resultados obtidos na fase de calibração dos parâmetros de entrada, foi elaborada uma configuração final com o objetivo de obter melhores resultados em um curto período de tempo de processamento, para se fazer uma comparação com a base de resultados do modelo matemático exato. Assim como na seção de calibração de parâmetros, os testes computacionais foram executados em um computador com processador Intel(R) Core(TM) i5-4210U CPU @ 1.70GHz, 2401 Mhz, 2 Núcleo(s), 4 Processador(es) Lógico(s) e 8 Gb de RAM, e os softwares utilizados para a programação e compilação do problema foram o Visual Studio 2015 e o Gurobi 7.0.2 para o processamento dos modelos de otimização. Nesta etapa de rodada dos resultados finais, os processamentos foram executados com o compilador no modo “release”, que é o modo de execução do compilador, o que faz com que os tempos de processamento sejam menores.

5.6.1 Definição dos Parâmetros para as Simulações dos Resultados Finais

Os parâmetros definidos para a obtenção dos resultados finais foram:

- Regra de ordenação das tarefas: por instante de liberação para início da execução;
- Quantidade de iterações globais = 20;
- Quantidade de tarefas candidatas = 5;
- Quantidade de iterações sem melhoria = 5;
- Quantidade de conjuntos de embarcações candidatas: 200;
- Quantidade de tarefas excluídas da solução inicial: 3.

A seleção de parâmetros desta simulação final foi definida com base nos resultados obtidos nos tópicos de calibração. Para o parâmetro de quantidade de tarefas candidatas, em alguns casos a seleção seguindo uma determinada ordenação colaborou com a melhoria dos resultados, porém, em outros casos não. Devido a isto, optou-se por definir um número médio, como 5 tarefas candidatas com a regra de ordenação por instante de liberação para início, que se mostrou mais eficiente do que a regra de ordenação por penalidade.

Para o parâmetro de quantidade de iterações sem melhoria, constatou-se que o aumento do valor deste parâmetro de fato contribui para a melhoria dos resultados, porém, os resultados obtidos para 5 e 6 iterações sem melhoria foram bastantes próximos. Devido a isso, optou-se por definir o valor deste parâmetro em 5, o que reduz um pouto o tempo de processamento de cada iteração global, não se penalizando muito o resultado final.

Para a quantidade de conjuntos de embarcações candidatas, verificou-se que, de fato, o aumento da quantidade de conjuntos colabora com a melhoria dos resultados, e o impacto do aumento da quantidade de conjuntos no tempo total de processamento foi pequeno. Devido a isto, optou-se por definir a quantidade de conjuntos em 200, para se aumentar ainda mais o espaço de busca das soluções, sem penalizar o tempo de processamento.

O parâmetro de quantidade de tarefas excluídas também foi outro que contribuiu bastante para a melhoria dos resultados finais. Analisando-se os resultados obtidos, pode-se notar que os resultados finais tiveram uma grande melhoria quando se passou de 2 para 3 tarefas excluídas. De 3 para 4 tarefas excluídas também se verificou uma melhoria nos resultados finais, porém a exclusão de 4 tarefas apresentou uma grande penalidade no tempo de processamento. Portanto, para ser possível efetuar diversas iterações globais, sem penalizar muito o tempo total de processamento e o resultado final, optou-se por definir o valor deste parâmetro em 3 tarefas excluídas.

Por fim, a definição da quantidade total de iterações globais foi definida em 20 iterações, para que se tenha uma boa quantidade de soluções e o tempo de processamento não seja muito impactado.

5.6.2 Método 1 – Seleção Aleatória

Os resultados obtidos para o processamento final do Método 1 estão apresentados na Tabela 5-39 e Figura 5-25 para as instâncias de 10 tarefas, e na Tabela 5-40 e Figura 5-26 para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no processamento do algoritmo e os obtidos no processamento do modelo matemático exato. Os tempos de processamento estão indicados em horas, para se facilitar a comparação com os resultados de referência, os quais foram obtidos por um processamento de 12h.

Tabela 5-39 – Resultados Finais – 10 tarefas – Método 1

Instâncias	1 Tarefa	
	Horas	Gap
Instância 1	0,9	1%
Instância 2	0,8	0%
Instância 3	2,7	8%
Instância 4	0,7	0%
Instância 5	0,9	0%
Instância 6	1,3	1%
Instância 7	0,5	0%
Instância 8	1,3	0%
Instância 9	1,3	0%
Instância 10	1,0	0%
Instância 11	1,3	0%
Instância 12	0,4	2%
Média 10 tarefas	**1,1**	**1%**

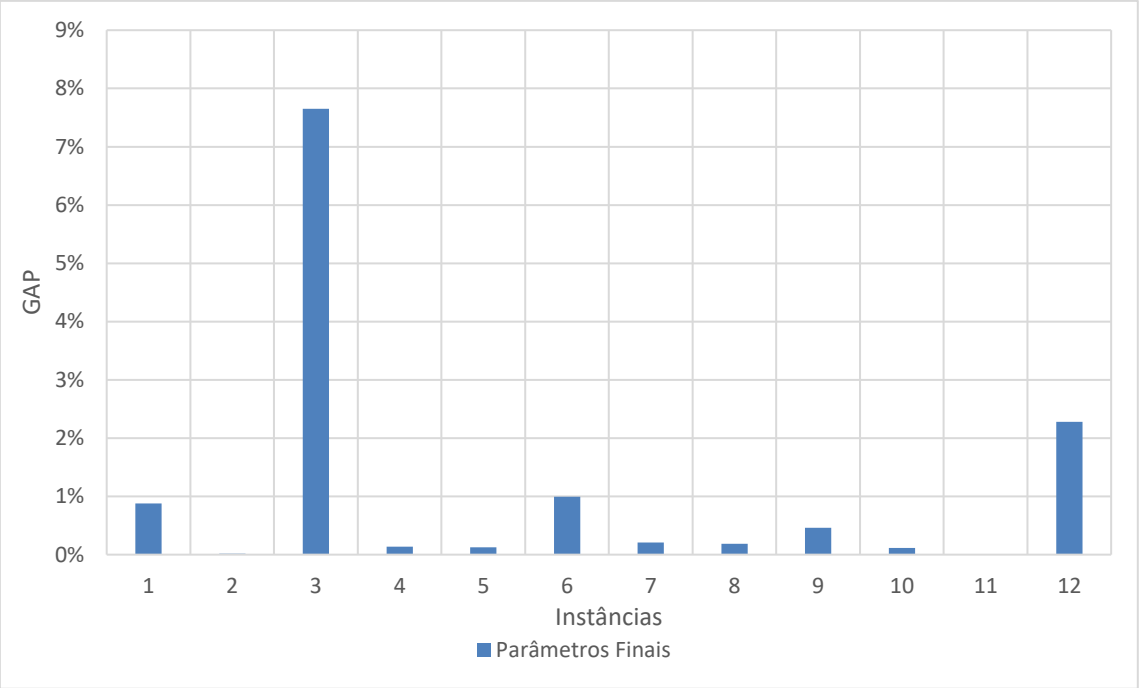

Figura 5-25 – Gráfico resultados Finais – 10 tarefas – Método 1

Fonte: Elaborado pelo autor

Tabela 5-40 – Resultados Finais – 20 tarefas – Método 1

Instâncias	1 Tarefa	
	Horas	Gap
Instância 13	0,9	23%
Instância 14	0,8	3%
Instância 15	1,6	8%
Instância 16	2,3	-1%
Instância 17	0,8	1%
Instância 18	1,6	5%
Instância 19	1,9	-51%
Instância 20	5,1	-38%
Instância 21	4,5	4%
Instância 22	4,2	3%
Instância 23	6,3	-13%
Instância 24	2,4	-15%
Média 20 tarefas	**2,7**	**-6%**

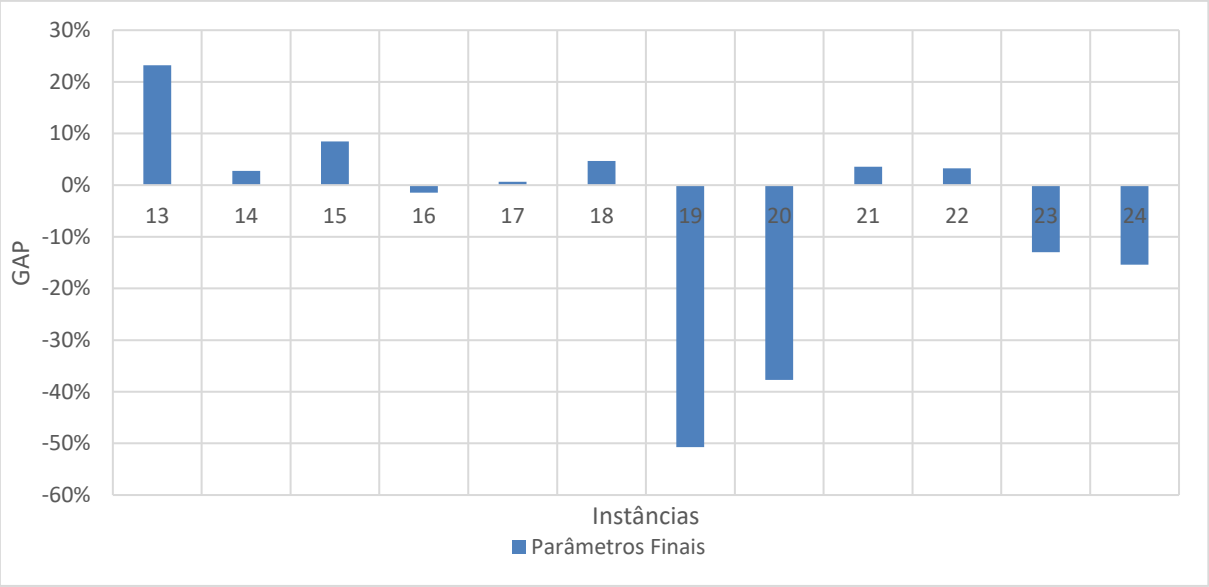

Figura 5-26 – Gráfico resultados Finais – 20 tarefas – Método 1

Fonte: Elaborado pelo autor

Os resultados obtidos para as instâncias com 10 tarefas apresentaram um gap médio de 1% para os resultados obtidos no modelo matemático exato e um tempo médio de processamento de cada instância de 1,1 horas. Este resultado verificado é muito bom uma vez que, pelo modelo matemático exato, foram obtidas as soluções ótimas destas instâncias de 10 tarefas com o tempo de processamento de 1 hora. Com isto, o desvio de 1% obtido para 1,1 hora de processamento é um desempenho muito próximo ao do modelo matemático exato. Este desvio de 1% se deve, principalmente, aos resultados obtidos para as instâncias 3 e 12, as quais apresentaram um gap de 8% e 2% respectivamente. A apresentação gráfica dos resultados obtidos para as instâncias de 10 tarefas pode ser visualizada na Figura 5-25.

Para as instâncias de 20 tarefas o gap médio obtido através do processamento do cenário com os parâmetros finais foi de –6%, com um tempo de processamento de 2,7 horas. Este resultado mostra que o desempenho do algoritmo foi muito bom, sendo superior ao desempenho do modelo matemático, obtendo melhores resultados e com um tempo de processamento de apenas 2,7 horas, frente às 12h de processamento do modelo matemático exato. Isto mostra que a eficácia do método estudado é muito boa para as instâncias com maior número de tarefas. Os melhores resultados

observados foram para as instâncias 16, 19, 20, 23 e 24, nos quais se obteve um resultado melhor do que o modelo matemático, sendo que para as instâncias 19 e 20, foram obtidos resultados da ordem de -51% e -38% respectivamente. As instâncias que apresentaram resultados não muito satisfatórios foram 13, 15 e 18, com gaps de 23%, 8% e 5% respectivamente, que foram responsáveis por elevar o valor da média das instâncias de 20 tarefas. A apresentação gráfica dos resultados obtidos para as instâncias de 20 tarefas pode ser visualizada na Figura 5-26Figura 5-25.

5.6.3 Método 2 – Seleção por Penalidade das Tarefas

Os resultados obtidos para o processamento final do Método 2 estão apresentados na Tabela 5-41 e Figura 5-27Tabela 5-33 para as instâncias de 10 tarefas, e na Tabela 5-42 e Figura 5-28 Tabela 5-34para as instâncias de 20 tarefas. Os valores dos resultados indicados nas tabelas expressam os gaps entre a solução obtida no processamento do algoritmo e os obtidos no processamento do modelo matemático. Os tempos de processamento estão indicados em horas, para facilitar a comparação com os resultados de referência, os quais foram obtidos através de um processamento de 12h.

Os resultados obtidos para as instâncias com 10 tarefas apresentaram um gap médio de 1% em relação aos resultados obtidos no modelo matemático exato e um tempo médio de processamento de cada instância de 0,7 horas. Este resultado é semelhante ao resultado obtido para o Método 1, o que é muito bom uma vez que pelo modelo matemático exato foram obtidas as soluções ótimas destas instâncias de 10 tarefas com o tempo de processamento de 1 hora. Este desvio de 1% se deve aos resultados obtidos para as instâncias 1, 3 e 12, as quais apresentaram um gap de 1%, 2% e 6% respectivamente. Em comparação ao Método 1, o Método 2 obteve melhor resultado para a instância 3, que foi de 8% no Método 1 e 2% no Método 2, e para a instância 12, o cenário se inverteu, sendo que o Método 1 obteve 2% e o Método 2, 6%. A apresentação gráfica dos resultados obtidos para as instâncias de 10 tarefas pode ser visualizada na Figura 5-27.

Tabela 5-41– Resultados Finais – 10 tarefas – Método 2

Instâncias	1 Tarefa	
	Horas	Gap
Instância 1	0,2	1%
Instância 2	0,4	0%
Instância 3	0,3	2%
Instância 4	0,6	0%
Instância 5	0,7	0%
Instância 6	1,0	0%
Instância 7	0,5	0%
Instância 8	0,8	0%
Instância 9	0,8	0%
Instância 10	1,0	0%
Instância 11	1,2	0%
Instância 12	1,1	6%
Média 10 tarefas	**0,7**	**1%**

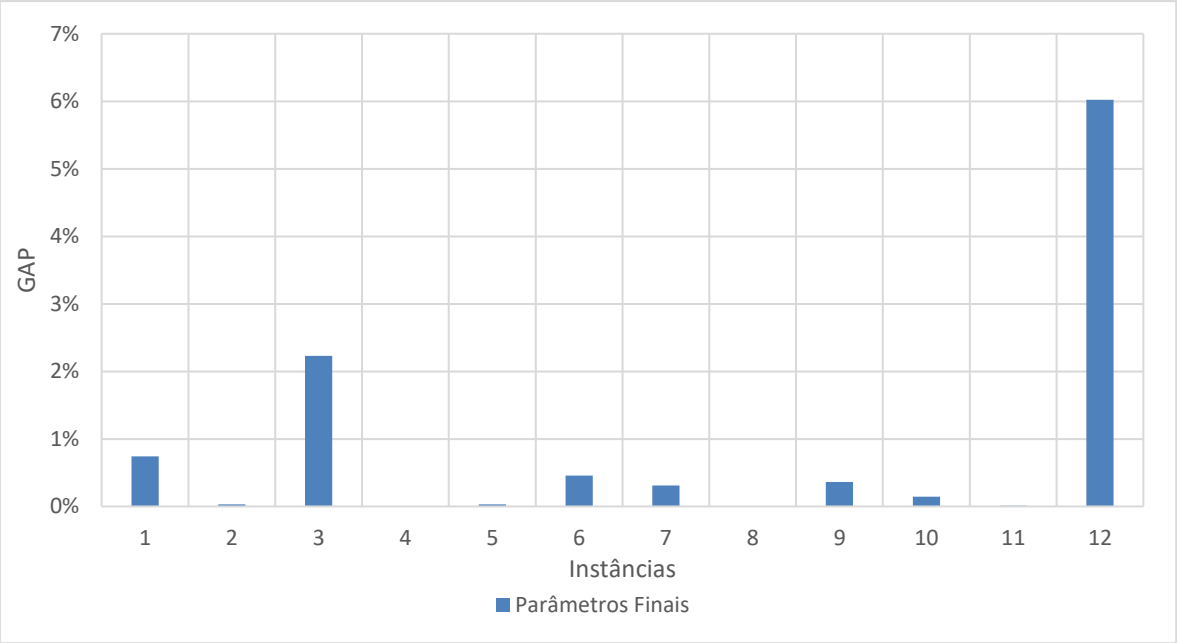

Figura 5-27 – Gráfico Resultados Finais – 10 tarefas – Método 2

Fonte: Elaborado pelo autor

Tabela 5-42 – Resultados Finais – 20 tarefas – Método 2

Instâncias	1 Tarefa	
	Horas	Gap
Instância 13	1,0	44%
Instância 14	0,7	-1%
Instância 15	1,8	3%
Instância 16	0,9	1%
Instância 17	11,4	1%
Instância 18	1,0	13%
Instância 19	1,6	-54%
Instância 20	2,1	-34%
Instância 21	1,1	4%
Instância 22	0,9	6%
Instância 23	0,9	-25%
Instância 24	1,2	-14%
Média 20 tarefas	**2,1**	**-5%**

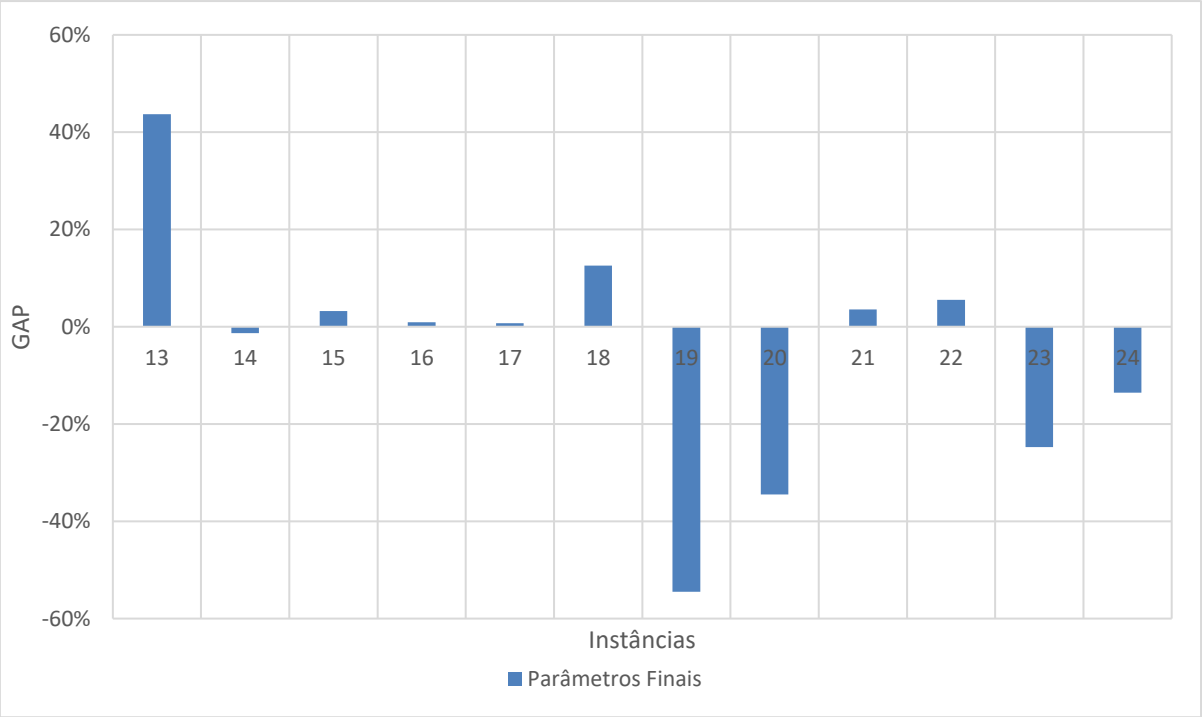

Figura 5-28 – Gráfico Resultados Finais – 20 tarefas – Método 2

Fonte: Elaborado pelo autor

Para as instâncias de 20 tarefas o gap médio obtido através do processamento do cenário com os parâmetros finais foi de –5%, com um tempo de processamento de 2,1 horas. Assim como observado para o Método 1, o desempenho do algoritmo foi muito bom, sendo superior ao desempenho do modelo matemático, obtendo melhores resultados e com um tempo de processamento de apenas 2,1 horas, frente às 12h de processamento do modelo matemático exato. Os melhores resultados observados foram para as instâncias 14, 19, 20, 23 e 24, nos quais se obteve um resultado melhor do que o modelo matemático, sendo que para as instâncias 19 e 20, foram obtidos resultados da ordem de -54% e -34% respectivamente. As instâncias que apresentaram resultados não muito satisfatórios foram 13, 18, 21 e 22, com gaps de 44%, 13%, 4% e 6% respectivamente, que foram responsáveis por elevar o valor da média das instâncias de 20 tarefas. A apresentação gráfica dos resultados obtidos para as instâncias de 20 tarefas pode ser visualizada na Figura 5-28Figura 5-25.

5.7 Discussão dos Resultados

Nas seções anteriores, foram apresentados os resultados obtidos para a aplicação dos métodos desenvolvidos, que combinam as metodologias de múltiplos recomeços associado a uma heurística híbrida, aplicada ao problema de programação de frota com restrições de sincronização. Nesta seção apresenta-se uma discussão dos resultados obtidos.

O intuito principal dos processamentos foi o de analisar a qualidade dos resultados obtidos através da comparação com resultados do modelo matemático exato e o tempo total de processamento. Para isto os processamentos iniciaram-se pela calibração dos parâmetros de entrada dos algoritmos para se entender o impacto da variação de cada um dos parâmetros de entrada no resultado final e no tempo de processamento.

A comparação dos resultados obtidos na fase de calibração dos parâmetros de entrada, do Método 1 – Seleção Aleatória e do Método 2 – Seleção por Penalidade da

Tarefa está apresentada nas tabelas: Tabela 5-43, Tabela 5-44, Tabela 5-45 e Tabela 5-46, as quais apresentam os valores médios de GAP e Tempo para as instâncias de 10 e 20 tarefas e para as regras de ordenação por instante de liberação para início. Os resultados obtidos pela rodada dos cenários através do Método 1 estão apresentados na cor preta e os resultados do Método 2 na cor azul.

Tabela 5-43 – Cenário 1 – Tarefas Candidatas – Valores Médios

Instâncias	1 tarefa	7 tarefas	N tarefas
	Gap	Gap	Gap
10 tarefas Método 1	54%	177%	45%
20 tarefas Método 1	128%	154%	190%
10 tarefas Método 2	88%	44%	39%
20 tarefas Método 2	109%	128%	136%

Tabela 5-44 – Cenário 2 – Conjuntos de Embarcações – Valores Médios

Instâncias	25 Conjuntos		100 Conjuntos	
	Minutos	Gap	Minutos	Gap
10 tarefas Método 1	1,0	92%	2,5	52%
20 tarefas Método 1	3,2	204%	10,4	118%
10 tarefas Método 2	2,3	45%	3,7	27%
20 tarefas Método 2	14,2	127%	11,2	103%

Tabela 5-45 – Cenário 3 – Iterações sem Melhoria – Valores Médios

Instâncias	2 Iterações		6 Iterações	
	Minutos	Gap	Minutos	Gap
10 tarefas Método 1	0,8	60%	4,5	59%
20 tarefas Método 1	2,4	165%	14,1	95%
10 tarefas Método 2	2,6	94%	5,8	37%
20 tarefas Método 2	4,9	137%	22,4	71%

Tabela 5-46 – Cenário 4 – Tarefas Excluídas – Valores Médios

Instâncias	1 Tarefa		4 Tarefas	
	Minutos	Gap	Minutos	Gap
10 tarefas Método 1	0,9	57%	54,6	8%
20 tarefas Método 1	2,7	121%	150,9	47%
10 tarefas Método 2	1,2	107%	74,5	5%
20 tarefas Método 2	3,7	142%	264,7	38%

Na fase de calibração dos parâmetros pode-se notar que os resultados obtidos pelo Método 2 foram ligeiramente melhores do que os resultados obtidos pelo Método 1. Analisando-se quantitativamente os resultados obtidos para os Cenários 2, 3 e 4 das instâncias de 10 tarefas, a diferença entre os resultados obtidos pelo Método 1 e pelo Método 2 (resultados Método 1 – resultados Método 2) são:

- 25% para o Cenário 2 (100 conjuntos de embarcações);
- 22% para o Cenário 3 (6 iterações sem melhoria);
- 3% para o Cenário 4 (4 tarefas excluídas).

Nas instâncias com 20 tarefas, fazendo-se uma análise quantitativa dos resultados, o valor da diferença entre os resultados obtidos pelo Método 1 e pelo Método 2 (resultados Método 1 – resultados Método 2) foi de:

- 15% para o Cenário 2 (100 conjuntos de embarcações);
- 24% para o Cenário 3 (6 iterações sem melhoria);
- 9% para o Cenário 4 (4 tarefas excluídas).

Importante ressaltar que estas reduções verificadas para o Método 2 vieram acompanhadas de aumento nos tempos de processamento médio, o que está coerente, uma vez que o processo construtivo da primeira solução é o mesmo para os dois métodos, e o que diferencia os dois métodos é a parte de busca local. No Método 2, foi verificado um maior número de iterações com melhoria de resultado, antes de atingir o critério de parada. Analisando-se o tempo de processamento total para as rodadas dos cenários 2, 3 e 4, a diferença entre os tempos obtidos pelos processamentos das instâncias de 10 tarefas através do Método 2 e através do

Método 1 (tempo processamento Método 2 – tempo processamento Método 1) foram de:

- 1,2 minutos para o Cenário 2 (100 conjuntos de embarcações);
- 1,3 minutos para o Cenário 3 (6 iterações sem melhoria);
- 19,9 minutos para o Cenário 4 (4 tarefas excluídas).

Para as instâncias de 20 tarefas a diferença entre o tempo de processamento observado no Método 2 e o tempo observado no Método 1 (tempo processamento Método 2 – tempo processamento Método 1) foi de:

- 0,8 minutos para o Cenário 2 (100 conjuntos de embarcações);
- 8,3 minutos para o Cenário 3 (6 iterações sem melhoria);
- 113,8 minutos para o Cenário 4 (4 tarefas excluídas).

Na fase de calibração, foram consideradas apenas 2 iterações globais e os parâmetros de entrada tinham valores reduzidos, para se ressaltar o impacto da variação de cada parâmetro e reduzir o tempo total de processamento. Conforme mencionado anteriormente, os resultados obtidos para a etapa de calibração dos parâmetros de entrada foram compilados no modo “debug”, uma vez que ainda se estavam executando-se testes para entender o comportamento dos métodos. Esta compilação em modo “debug” gerou um aumento proporcional no tempo de processamento de todas as rodadas de calibragem, porém, pelo fato do impacto ser igual para todos os casos testados, não houve prejuízo nos resultados obtidos.

Já para a etapa de resultados finais, priorizou-se a qualidade dos resultados obtidos, aumentando-se a quantidade de iterações globais para 20 iterações e os parâmetros de entrada para valores mais altos, o que aumenta o custo e o tempo de processamento. Com estas alterações nas configurações dos parâmetros, foi possível observar que o comportamento dos resultados obtidos através das simulações das instâncias pelo Método 1 e pelo Método 2 se alterou. A comparação dos resultados obtidos nas simulações finais estão apresentadas na Tabela 5-47, para as instâncias de 10 tarefas e na Tabela 5-48, para as instâncias de 20 tarefas.

Tabela 5-47 – Resultados Finais – Comparação Método 1 e Método 2 – 10 tarefas

Instâncias	Método 1		Método 2	
	Horas	Gap	Horas	Gap
Instância 1	0,9	1%	0,2	1%
Instância 2	0,8	0%	0,4	0%
Instância 3	2,7	8%	0,3	2%
Instância 4	0,7	0%	0,6	0%
Instância 5	0,9	0%	0,7	0%
Instância 6	1,3	1%	1,0	0%
Instância 7	0,5	0%	0,5	0%
Instância 8	1,3	0%	0,8	0%
Instância 9	1,3	0%	0,8	0%
Instância 10	1,0	0%	1,0	0%
Instância 11	1,3	0%	1,2	0%
Instância 12	0,4	2%	1,1	6%
Média 10 tarefas	**1,1**	**1%**	**0,7**	**1%**

Como primeiro ponto, foi feita uma análise comparativa dos resultados observados para o Método 1 e para o Método 2.

Para as instâncias de 10 tarefas a média dos gaps obtidos para ambos os métodos foi de 1%. Este desvio de 1% em relação ao resultado ótimo obtido através do modelo matemático exato se deu principalmente pelos resultados das instâncias 3 e 12, que, em ambos os métodos, apresentaram gaps superiores a 2%. Em relação ao tempo de processamento, para o Método 1 foi observado um tempo médio de 1,1 horas e para o Método 2, um tempo de 07 horas. Esta diferença no tempo de processamento pode ser reflexo do desempenho do computador utilizado para executar a compilação, pois durante o processamento das instâncias, se estavam executando outras atividades em paralelo.

Para as instâncias de 20 tarefas, a média dos gaps obtidos para o Método 1 foi de -6% e para o Método 2 foi de -5%. Este resultado final é interessante porque o desempenho do Método 1 e Método 2 se inverteu em relação aos resultados obtidos

na fase de calibração dos parâmetros de entrada, em que o Método 2 apresentou melhores resultados que o Método 1. Isto demonstra que para um número pequeno de iterações e parâmetros de entrada menores, a metodologia desenvolvida para o Método 2 direciona melhor a busca pela solução ótima, enquanto que a metodologia do Método 1 faz buscas mais aleatórias, as quais, podem obter bons resultados ou não. Porém, quando se aumenta bastante o número de iterações globais e os parâmetros de entrada, pode-se verificar que o desempenho apresentado pelo Método 2 continua sendo efetivo e, para o Método 1 que é totalmente aleatório, aumentam-se as chances de se obter bons resultados, uma vez que este método faz buscas locais mais amplas do que o Método 2. E devido a isto, o Método 1 apresentou bons resultados, e em algumas instâncias até melhores do que o Método 2.

Tabela 5-48 – Resultados Finais – Comparação Método 1 e Método 2 – 20 tarefas

Instâncias	Método 1		Método 2	
	Horas	Gap	Horas	Gap
Instância 13	0,9	23%	1,0	44%
Instância 14	0,8	3%	0,7	-1%
Instância 15	1,6	8%	1,8	3%
Instância 16	2,3	-1%	0,9	1%
Instância 17	0,8	1%	11,4	1%
Instância 18	1,6	5%	1,0	13%
Instância 19	1,9	-51%	1,6	-54%
Instância 20	5,1	-38%	2,1	-34%
Instância 21	4,5	4%	1,1	4%
Instância 22	4,2	3%	0,9	6%
Instância 23	6,3	-13%	0,9	-25%
Instância 24	2,4	-15%	1,2	-14%
Média 20 tarefas	**2,7**	**-6%**	**2,1**	**-5%**

Em relação aos tempos de processamento obtidos para as instâncias de 20 tarefas, a quantidade de iterações executadas pelo Método 2 continuou sendo maior do que a quantidade de iterações executadas no Método 1, porém, o desempenho do computador durante a execução das simulações pode ter impactado no tempo de

processamento, uma vez que estavam sendo rodados diversas atividades em paralelo às simulações. Na prática, os resultados observados entre os Métodos 1 e 2 são iguais.

Como segundo ponto, foi feita uma análise global dos resultados obtidos pelos métodos desenvolvidos, comparados com os resultados obtidos no modelo matemático exato.

Fazendo-se uma análise global dos resultados obtidos, e possível se afirmar que os métodos desenvolvidos no presente trabalho apresentaram um desempenho razoável para as instâncias com 10 tarefas e um desempenho muito bom para as instâncias com 20 tarefas.

Para as instâncias com 10 tarefas, o desvio médio dos resultados obtidos foi de 1% para os dois métodos estudados e o tempo médio de processamento das instâncias foi de 1,1 horas para o Método 1 e 0,7 horas para o Método 2. Comparando-se com os resultados obtidos no modelo matemático exato, em que, com 1 hora de processamento, atingiu-se a solução ótima para todas as instâncias de 10 tarefas, o desempenho do dos métodos estudados no presente trabalho que foi de 1% de desvio no valor do resultado final com o mesmo tempo de processamento, pode se dizer que o desempenho observado foi razoável.

Para as instâncias com 20 tarefas, o desvio médio dos resultados obtidos foi de -6% para o Método 1 e -5% para o Método 2. Este valor do desvio médio para as instâncias de 20 tarefas indica que os métodos estudados no presente trabalho apresentaram um desempenho superior ao desempenho observado no modelo matemático exato. Isto só foi possível pois o modelo matemático exato não obteve as soluções ótimas para todas as instâncias simuladas, permitindo que melhorias nos resultados finais fossem observadas. Em relação ao tempo total de processamento, o tempo médio observado para as instâncias de 20 tarefas foi de 2,7 horas para o Método 1 e 2,1 horas para o Método2. Estes tempos médios de processamento representam apenas 22% do tempo médio de processamento considerado no modelo matemático exato, que foi de 12 horas. Conforme já mencionado anteriormente, as simulações finais foram compiladas no modo “release” o que fez com que os tempos

de processamento diminuíssem, quando comparados aos tempos observados na fase de calibragem dos parâmetros de entrada.

Combinando-se o desvio médio obtidos para os resultados finais com o tempo médio de processamento das instâncias de 20 tarefas, pode-se se concluir que, para instâncias com maior número de tarefas, o desempenho dos métodos estudados no presente trabalho foi muito bom, uma vez que os resultados finais foram melhores e o tempo de processamento foi menor, o que significa que os métodos desenvolvidos e estudados neste trabalho possuem capacidade de obter melhores resultados em um pequeno tempo de processamento.

6 CONCLUSÕES

Este capítulo apresenta as principais conclusões a respeito da pesquisa relativa ao problema de programação de frota com restrições de sincronização.

O problema estudado nesta dissertação abordou a programação da frota de rebocadores que realizam serviços de apoio marítimo “offshore”, incluindo o reboque e a ancoragem de sondas de perfuração e de plataformas de produção de petróleo no mar. Este problema é complexo por impor a requisição de múltiplas embarcações, que operam de forma sincronizada, entre outras restrições.

O foco da pesquisa foi no desenvolvimento de um método de resolução para o problema estudado, visando a minimização do custo total da operação, dentro de um tempo de processamento aceitável, considerando o contexto do processo decisório. Para alcançar este objetivo, foi elaborada uma heurística híbrida, em que as tarefas eram ordenadas segundo diferentes critérios, e sorteadas a partir de uma lista restrita de candidatos. Cada tarefa selecionada foi inserida nas rotas de forma otimizada, através de um modelo matemático de programação linear inteira. Adicionalmente, foi efetuada uma busca local com remoção aleatória de tarefas, seguida de reinserção das mesas. Este procedimento foi submetido a uma rotina de múltiplos recomeços, para explorar a aleatoriedade nos sorteios das tarefas.

O Método 1 - Seleção Aleatória, utilizou a heurística construtiva para gerar uma solução inicial viável, e a fase de busca local considerou a seleção aleatória das tarefas a serem excluídas das rotas das embarcações. O Método 2, de Seleção por Penalidade Aplicada à Tarefa também utilizou a heurística construtiva para gerar uma solução inicial viável, e a fase de busca local considerou a seleção das tarefas a serem excluídas de acordo com a penalidade aplicada a cada tarefa.

A primeira fase da aplicação dos métodos consistiu em fazer uma calibragem dos parâmetros de entrada e entender o impacto de cada um dos parâmetros no

resultado final e no tempo de processamento para se definir os valores de parâmetros ideais para as simulações dos resultados finais.

Os parâmetros de entrada que se estudou nesta fase de calibragem foram: quantidade de tarefas candidatas para compor a lista de tarefas que podem ser selecionadas na fase de construção da solução inicial, que está diretamente relacionada com as regras de ordenação das tarefas; quantidade de iterações sem melhoria que é o critério de parada para uma iteração global; quantidade de conjuntos de embarcações candidatas que define o número de conjuntos de embarcações capazes de executar a tarefa selecionada que serão considerados no modelo matemático de inserção; quantidade de tarefas excluídas da solução inicial na fase de busca local.

Desta etapa de calibragem dos parâmetros de entrada dos modelos foram definidos os parâmetros a serem utilizados para a simulação dos resultados finais. O processamento final, após a fase de calibração, apresentou um desempenho razoável para as instâncias com 10 tarefas e, muito bom para as instâncias com 20 tarefas. Estes resultados indicam que os métodos propostos possuem melhor performance (função objetivo e tempo) para instâncias com maior número de tarefas, alcançando resultados melhores do que o modelo matemático exato, em um tempo de processamento menor, o que está de acordo com a proposta inicial deste trabalho.

Alguns desdobramentos e linhas de pesquisa que poderiam ser desenvolvidos a partir deste trabalho são:

- Calibração adicional do algoritmo visando encontrar melhores resultados para as instâncias de 20 tarefas;
- Aperfeiçoamento da fase construtiva da heurística visando reduzir a quantidade de iterações de melhoria, e obter melhores resultados. Sugere-se trabalhar no desenvolvimento de novos critérios de seleção das tarefas e dos conjuntos de embarcações candidatas para que o modelo matemático de inserção possa convergir mais rapidamente;
- Aperfeiçoamento da fase de busca local da heurística, desenvolvendo-se novos critérios de seleção das tarefas a serem excluídas.

7 REFERÊNCIAS

AKBARI, V.; SALMAN, F. S. Mutli-vehicle synchronized arc routing problem to restore post-disaster network connectivity. European Journal of Operational Research 257: 625-640, 2017.

BERTRAND, J. W. M.; FRANSOO, J. C. Operations management research methodologies using quantitative modeling. International Journal of Operations & Production Management, 22:2, p.241-254, 2002.

CLARKSONS RESEARCH. Disponível em: < https://clarksonsresearch.wordpress.com/tag/drilling-rigs/>. Acesso em 01/06/2019.

DREXL, M. Synchronization in vehicle routing - a survey of VRPs with multiple synchronization constraints. Transportation Science, 46(3): 297-316, 2012.

EKSIOGLU, B.; VURAL, A. V.; REISMAN, A. The vehicle routing problem: A taxonomic review. Computers & Industrial Engineering, 57(4): 1472-1483, 2009.

FAGERHOLT, K.; LINDSTAD, H. Optimal policies for maintaining a supply service in the Norwegian sea. Omega 28, 269–275, 2000.

FIKAR, C.; HIRSCH, P. A matheuristic for routing real-world home service transport systems facilitating walking. Journal of Cleaner Production 105: 300-310, 2015.

HALVORSEN-WEARE, E.E.; FAGERHOLT, K. Robust supply vessel planning. In: Network Optimization. In: Pahl, J., Reiners, T., Vob, S. (Eds.). Lecture Notes in Computer Science, vol. 6701. Springer, Berlin/Heidelberg, pp. 559–573, 2011.

HALVORSEN-WEARE, E.E.; FAGERHOLT, K.; NONÅS, L.M.; ASBJØRNSLETT, B.E. Optimal fleet composition and periodic routing of offshore supply vessels. European Journal of Operational Research 223, 508–517, 2012.

HEMPSCH, C.; IRNICH, S. Vehicle routing problems with intertour resource constraints. B. Golden, S. Raghavan, E. Wasil, eds. The Vehicle Routing Problem2 Latest Advances and New Challenges. Springer, New York, 421–444, 2008.

KRAMER, R.; SUBRAMANIAN, A.; VIDAL, T.; CABRAL, L. A. F. A matheuristic approach for the Pollution-Routing Problem. European Journal of Operational Research 243: 523-539, 2015.

LAHYANI, R.; KHEMAKHEN, M.; SEMET, F. Rich vehicle routing problems: From a taxonomy to a definition. European Journal of Operational Research 241(1): 1-14, 2015.

LAPORTE, G.; OSMAN, I.H. Routing problems: a bibliography. Annals of Operations Research, 61, p.227-262, 1995.

MANCINI, S; GIUSEPPE S. A large neighborhood search based matheuristic for the tourist cruises itinerary planning. Computers & Industrial Engineering 122: 140-148, 2018.

MARTÍ, R.; RESENDE, M. G. C.; RIBEIRO, C. C. Multi-start methods for combinatorial optimization. European Journal of Operational Research 226: 1-8, 2013

MENDES, A. B. Programação de frota de apoio a operações “offshore” sujeita à requisição de múltiplas embarcações para uma mesma tarefa. 2007. 240 f. Tese (Doutorado em Engenharia Naval e Oceânica) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

PERBOLI, G., R. TADEI, D. VIGO. The two-echelon capacitated vehicle routing problem: Models and math-based heuristics. Technical report CIRRELT-2008-55, Centre interuniversitaire de recherche sur les réseaux d’entreprise, la logistique et letransport, Montréal, 2008.

PRESCOTT-GAGNON, E., G. DESAULNIERS, L.-M. ROUSSEAU. Heuristics for an oil delivery vehicle routing problem. Technical report G-2010-63, Les Cahiers du GERAD, Montréal, 2010.

RITCHIE, G. Practical introduction to anchor handling and supply vessel operations. Houston, Texas: Oilfield Publications Limited, 2004. 186p.

SOBRAPO. Sociedade Brasileira de Pesquisa Operacional. Disponível em: <http://www.sobrapo.org.br/o_que_e_po.php>. Acesso em 10/02/2018.

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 203: 230–240, 2010.

TIAGO, L. L. Dimensionamento de Frota de Navios Rebocadores de Apoio Marítimo Offshore. 2018. 142 f. Dissertação (Mestrado em Engenharia Naval e Oceânica) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2018.

VILLEGAS, J. G.; PRINS, C.; PRODHON, C.; MEDAGLIA, A. L.; VELASCO, N. A matheuristic for the truck and trailer routing problem. European Journal of Operational Research 203: 231-244, 2013.