

## 2 Revisão da literatura acadêmica

O presente capítulo oferece uma fundamentação teórica respaldada na literatura acadêmica para os conceitos relacionados ao problema de minimização de custos na programação de navios. A seção 2.1 apresenta os conceitos ligados ao transporte marítimo e a programação de navios. A seção 2.2 descreve um breve histórico dos problemas de otimização em programação de navio e as diferentes abordagens do problema. Por fim, a seção 2.3 apresenta a pesquisa Brown *et al.* (1987) e alguns estudos que seguiram sua estrutura.

### 2.1. Transporte marítimo e a programação de navios

O *Council of Supply Chain Management Professionals* (CSCMP) define logística como a parte do gerenciamento de cadeia de suprimento que planeja, implementa e controla com eficiência e eficácia os fluxos e armazenamento de bens e serviços e a informação relacionada a eles entre o ponto de origem e o ponto de consumo, a fim de atender as necessidades do consumidor (CSCMP, 2011).

O transporte é a área operacional da logística que move e aloca, geograficamente, o inventário. Devido à sua importância e ao seu custo visível, o transporte tem, tradicionalmente, recebido considerável atenção gerencial (Bowersox *et al.*, 2002). O transporte normalmente representa o elemento mais importante em termos de custos logísticos para inúmeras empresas. O usuário de transportes tem uma ampla gama de serviços à disposição, girando em torno de cinco modais básicos: hidroviários, ferroviário, rodoviário, aeroviário e dutoviário (Ballou 2004).

O transporte marítimo é o principal meio de transporte no comércio internacional. Embora seja difícil precisar sua participação no mercado, estimativas da Associação Internacional de Economista Marítimos propõe que cerca de 65% a 85% de todo o comércio internacional, em termos de volume,

seja transportado através do transporte marítimo (Christiansen *et al.*, 2007). Ballou (2004) destaca que o transporte marítimo é responsável por mais de 50% do transporte internacional em dólares e 99% do peso total. Barat (2009) define transporte marítimo como sendo aquele que se faz no mar para transporte de carga ou de passageiros. Pode ser de longo curso, quando o porto de chegada e o porto de partida estão em países diferentes, tendo caráter internacional, ou de cabotagem, quando os portos de chegada e de partida estão no mesmo país, tendo assim, caráter nacional.

Existem três tipos básicos de operadores para navios comerciais: os operadores *liners*, *tramp* e *industrial* (Christiansen *et al.*, 2007; e Lawrance, apud Fagerholt e Lindstad, 2007). Os *liners shipping* operam de acordo com a publicação de um itinerário e programação previamente definidos, como se fossem linhas de ônibus. O operador *tramp shipping* programa seus navios seguindo as cargas disponíveis no mercado, de forma similar a um taxi. Normalmente, o operador *tramp shipping* tem uma dada quantidade de cargas para ser transportada em contrato e tenta maximizar o lucro transportando cargas opcionais. Os operadores *intrustrial shipping* normalmente, são os donos das cargas e controlam os navios utilizados para transportá-las. O objetivo do operador *industrial shipping* é transportar todas as cargas a um custo mínimo, enquanto os operadores *liners* e *tramp shipping* buscam maximizar o lucro por unidade de tempo. A PETROBRAS se enquadra na classe de operadores *industrial shipping*, pois possui uma frota de navios controlado para transportar suas próprias cargas.

Christiansen *et al.* (2007) apresentam alguns problemas reais de programação de navios e os divide em classes começando do mais simples até o mais complexo, adicionando restrições e possibilidades em cada um dos problemas, sendo eles:

- *Full shiploads* é o tipo de problema mais simples, onde os navios são carregados no porto de origem com sua capacidade máxima e seguem diretamente para um porto de destino, onde será descarregada toda a carga;
- *Multiple cargos on board* é uma extensão do *full shipload*, onde se permite o carregamento de múltiplas cargas de uma só vez, com tamanhos fixos;

- *Flexible cargo size* é um problema similar ao anterior, porém com quantidade de carga variável;
- *Multiple nonmixable problems* é um problema de programação quando se tem múltiplas cargas, que não podem ser misturadas, mas são transportadas simultaneamente em um mesmo navio, que possui sua capacidade de armazenamento compartimentada;
- *Contracted and Optimal cargoes* é um caso típico da *tramp shipping*, onde o navio deve carregar as cargas contratadas e as opcionais devem ser otimizadas, buscando transportar somente aquelas que aumentem o lucro da companhia; e
- *Spot charter* é o caso em que se discute a contratação de navios de mercado para atendimento de algumas cargas excedentes.

Ainda segundo Christiansen *et al.* (2007), um problema típico do transporte de petróleo é o *full shipload*, e, em muitos casos, a frota de navios controlados não tem capacidade suficiente para atender todas as cargas contratadas. Nesses casos, o transportador tem a opção de contratar navios no mercado, os chamados *spot charter*.

## **2.2. Problemas de otimização em programação de navios**

Problemas de otimização em programação de navios já são estudados há bastante tempo. A literatura oferece trabalhos científicos sobre o tema desde a década de 50. Diante de uma frota militar de navios tanque homogêneos e um programa de movimentação de cargas de petróleo, Flood (1954) entendeu que os custos portuários e os custos operacionais dos navios, enquanto carregados, são os mesmos para todas as possibilidades. Portanto, segundo Flood (1954), os únicos custos que poderiam ser controlados são os custos operacionais durante as viagens em lastro, ou seja, com o navio vazio, pois somente estes trechos dependem da programação definida pelo programador. Flood (1954) utilizou o método *simplex* desenvolvido por Dantzing para minimizar a distância em lastro percorrida pelos navios da frota, entendendo que desta forma estaria minimizando os custos de operação dos navios.

Outro estudo pioneiro sobre otimização em programação de navios é (Appelgren, 1969). Segundo Gronhaug *et al.*(2010), este foi o primeiro estudo a

utilizar a abordagem de geração de colunas em um problema de programação de navios. Gronhaug *et al.* (2010) ressalta que a abordagem por geração de colunas é fundamental para alguns casos descritos na literatura, onde o problema se torna tão grande que gerar todas as programações factíveis o tornaria inviável. Nestes casos, a abordagem de geração de colunas vem sendo utilizada com sucesso para obter as melhores programações de navios.

Ronen *et al.* (1993) elaborou uma pesquisa apresentando o *status* e perspectivas do problema de programação e roteirização de navios àquela época. O estudo foi embasado nas publicações entre os anos de 1984 e 1993 e identificou que o rápido desenvolvimento da computação nesse período facilitou a solução de problemas de otimização em programação de navios, que podem resultar em um impacto econômico significativo. Muito embora, ainda nessa década se observava uma escassez de estudos publicados sobre este tema na literatura, o que indicava o baixo nível de penetração de modelos quantitativos em aplicações industriais.

Christiansen *et al.* (2004) elaboraram uma pesquisa continuando o estudo de Ronen *et al.* (1993), porém para a década seguinte e abordaram problemas sobre planejamento estratégico de frota, planejamento operacional e tático de frota e aplicações militares. Segundo Christiansen *et al.* (2004), o estudo sobre programação de navios recebeu um grande impulso no meio acadêmico a partir da pesquisa feita por Brown *et al.* (1987). Seu modelo aborda problemas restritos, onde é possível utilizar um gerador de colunas que cria um conjunto completo de programações factíveis. Muitos trabalhos sobre otimização em problemas de programação de navios foram desenvolvidos posteriormente utilizando uma abordagem semelhante à apresentada por Brown *et al.* (1987). Dada sua relevância para esta dissertação, seu estudo e outros que seguiram sua estrutura serão detalhados na Seção 2.3.

Gronhaug *et al.* (2010) ressalta que, para alguns casos descritos na literatura e muitos outros casos reais de programação de navios, o problema se torna tão grande, que gerar todas as programações factíveis, conforme sugere Brown *et al.* (1987), poderia inviabilizar a solução do problema. Nestes casos, a abordagem de geração de colunas vem sendo utilizada com sucesso para obter as melhores programações para os navios.

Appelgren (1969) foi o primeiro a utilizar esta abordagem aplicada a um problema de programação de navios. Seu estudo apresenta um algoritmo para programação de navios de uma empresa de navegação sueca, que opera como uma *tramp shipping*. Appelgren (1969) percebeu que utilizando a abordagem de geração de todas as programações factíveis, quando se estende o período de programação para além 60 dias, o número de programações geradas cresce para um número da ordem de  $10^{18}$ , limitando a utilização do modelo. Portanto, o algoritmo apresentado utiliza a decomposição de Dantzig-Wolfe, que é a base para a geração de colunas, para solucionar o problema. A abordagem de geração de colunas propõe decomposição do problema em um problema principal e um subproblema. Christiansen *et al.* (2007) descrevem o procedimento de Dantzig-Wolfe da seguinte maneira. O problema principal trata das cargas e suas restrições. Como nenhuma das restrições inclui interação entre navios, então essas restrições podem ser separadas em um subproblema para cada navio. Primeiro, um problema principal restrito é resolvido. Então, algumas novas colunas são adicionadas ao problema restrito. Essas colunas correspondem a novas programações com o custo reduzido positivo na solução do problema principal. Isto significa que os valores *duais* da solução restrita são transferidos para os subproblemas. Os subproblemas são resolvidos e novas programações são geradas. Em seguida, o problema principal é otimizado novamente, adicionando as novas colunas geradas no subproblema. Este procedimento continua até que não existam novas colunas com custo reduzido positivo e nenhuma solução melhor exista.

Christiansen e Nygreen (1998) utilizam a abordagem da decomposição de Dantzig-Wolfe para solucionar um problema real de programação de navios. O problema é a variação de um *pick-up and delivery problem* com multi veículos combinado com um problema de gestão de inventário, onde é considerado o estoque e a produção no porto de carga, assim como o consumo no porto de descarga.

Outra abordagem para a solução do problema de programação de navios é utilizando uma heurística de busca local para gerar um conjunto inicial de colunas. Fagerholt e Lindstad (2007), Bronmo *et al.* (2007a) e Norstad *et al.* (2010) adotaram esta abordagens em seus estudos.

Fagerholt e Lindstad (2007) propõem um SSD de uso comercial que utiliza uma heurística para a geração de um número considerável de

programações iniciais de forma aleatória. As melhores soluções deste conjunto inicial são então melhoradas por uma heurística de busca local até o ótimo local ser atingido. A heurística de busca local é realizada em múltiplas soluções iniciais para evitar ou reduzir as possibilidades de se chegar a um ótimo local pobre.

Os resultados da heurística de busca local para geração de um conjunto de soluções iniciais foi comparado com uma abordagem baseada em otimização e estudos computacionais mostraram que, consistentemente, a heurística apresentou resultados ótimos ou próximos ao ótimo em casos reais (Bronmo *et al.*, 2007a).

Norstad *et al.* (2010) aprestando um problema de uma companhia de navegação do tipo *tramp shipping*. Seu modelo também utiliza a heurística de busca local para gerar um conjunto de soluções iniciais para a geração de programações. O modelo busca maximização do resultado adicionando cargas *spot* opcionais e otimizando a velocidade dos navios. Neste caso, a velocidade em cada trecho de navegação torna-se mais uma variável do problema.

Estudos de programação de veículos com gestão de estoque raramente eram discutidos no contexto marítimo até o final da década de 90 (Christiansen *et al.*, 2004). Muito embora, a partir de então, tem-se verificado o aumento do interesse neste tema (Agra *et al.*, 2011).

Christiansen (1999) foi um dos pioneiros a tratar do problema de programação de navios associado à gestão de estoque nos pontos de carga e descarga. Christiansen (1999) propõe um modelo para minimizar os custos de transporte de uma frota de navios que transporta um único produto (amônia) entre os pontos de produção e consumo. As quantidades carregadas e descarregadas são determinadas de acordo com a taxa de produção no ponto de carregamento, níveis de estoque e a capacidade do navio que está visitando cada porto. O problema é solucionado utilizando a abordagem de decomposição de Dantzig-Wolfe.

Ronen (2002) estudou um problema de programação de navios enfrentado por produtores de grãos líquidos, onde os pontos produtores e de consumo possuem tancagem com capacidade limitada. As datas, a origem, o destino e o tamanho dos lotes de carregamento devem ser determinados visando minimizar custo, porém sem violar a capacidade de estoque nas duas

pontas, ou seja, deve-se evitar perda de produção por falta de espaço de tancagem e perda de processamento por falta de produto em estoque. Uma heurística baseada em custo é utilizada a fim de garantir que uma solução aceitável gerencialmente seja encontrada rapidamente.

Person e Gothe-Lundgren (2005) estudaram um problema de programação de navios, que são utilizados para transportar produtos de refinarias para depósitos intermediários. Os níveis de estoque, que são afetados tanto pela programação das refinarias, quanto pela demanda nos depósitos intermediários, são levados em consideração neste problema. Uma abordagem baseada em geração de colunas com horizonte de tempo contínuo é proposta para solucionar este problema.

Al-Khayyal and Hwang (2007) estudaram um problema similar aos de Christiansen (1999) e Ronen (2002), porém consideraram uma frota de navios heterogêneos, empregados para transportar diferentes tipos de produtos granéis líquidos, que devem se mantidos em tanques segregados durante o transporte. Os autores propõem um modelo utilizando programação não linear inteira mista para minimizar os custos de transporte respeitando as restrições de estoque nos pontos de carga e descarga.

Gronhaug *et al.* (2010) consideram um problema de programação de navios com gestão de estoque na indústria de gás natural liquefeito (GNL). Seu estudo incluiu alguns aspectos complicadores tais como taxa constante de evaporação da carga, que é utilizada como combustível durante o transporte; produção e consumo de GNL variáveis nos pontos de carga e descarga; e um número variável de tanques a serem descarregados nos terminais de regaseificação. Gronhaug *et al.* (2010) utilizam a abordagem de geração de colunas, onde o problema principal é a gestão de estoques com restrições de capacidade. Neste trabalho os subproblemas geram as colunas que representam as programações dos navios.

Song e Furman (2010) propõem uma abordagem prática para o problema de programação de navios com gestão de estoques. Neste estudo os autores também utilizam a programação linear inteira mista com o horizonte de tempo discreto para gerar as programações factíveis. O objetivo do problema é encontrar a programação ótima, que minimiza custo de transporte de uma frota de navios heterogêneos, que deve transportar um único tipo de produto a granel

entre múltiplos portos, respeitando todas as capacidades e restrições relacionadas com o estoque do produto em cada porto.

Agra *et al.* (2011) propõem em seu estudo abordar o problema da programação de navios com gestão de estoque como um problema de programação linear inteira mista. Para tanto, os autores optaram por trabalhar com o horizonte de tempo discreto, ou seja, o tempo de espera, tempo de carregamento ou descarga e o tempo do navio viajando podem ser medidos por números inteiros múltiplos de um período de tempo, que pode ser de oito horas ou um dia, por exemplo. O movimento dos navios é apresentado em um rede espaço-tempo ilustrado na Figura 1. O navio inicia no ponto inicial O e navega para o Porto 3. Neste porto, o navio opera por dois períodos antes de navegar para o Porto 1, onde espera por um período antes de operar. Após operação, o navio navega para o Porto 2, onde espera e opera antes de finalizar a programação. Cada caminho ao longo do tempo navegado define uma programação para um navio.

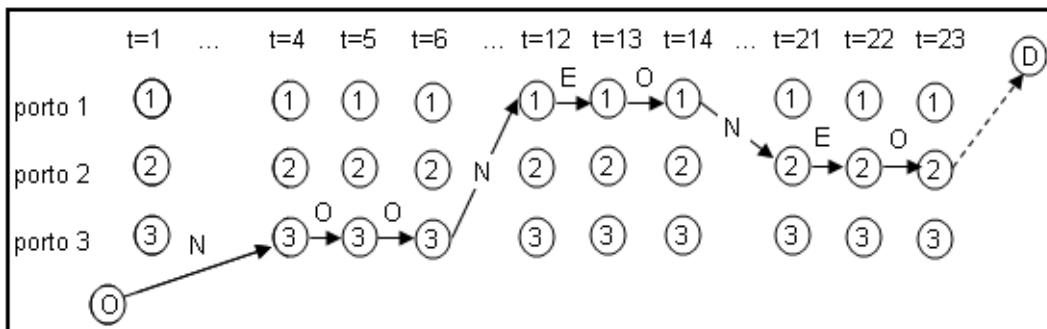


Figura 1: Movimentação de navios no horizonte de tempo discreto.

Os rótulos dos arcos são O para operando, E para esperando e N para navegando.

De acordo com Agra *et al.* (2010), a abordagem que utiliza o horizonte de tempo discreto permite lidar mais facilmente com múltiplas janelas de tempo e com demandas variáveis nos pontos consumidores. Porém, a desvantagem desta abordagem é que ela gera um número muito grande de variáveis o que pode dificultar a resolução do modelo.

A Tabela 1 faz um sumário dos estudos publicados na literatura acadêmica e mencionados nesta dissertação que tratam do problema de programação de navios.



Artigo	Decisões Principais	Objetivo	Carga	Destakes
Flood (1954)	Programação	Minimizar a distância navegada	Petróleo	Pioneiro na programação de navios, busca reduzir custo através de programação dos trechos de navegação em lastro.
Appelgren (1969)	Programação	Maximizar lucro	Não especificada	Pioneiro no uso da decomposição de Dantzig-Wolfe.
Fisher e Rosenwein (1989)	Programação	Minimizar custo/Maximizar lucro	Granel	Interface gráfica colorida.
Perakis e Bremer (1992)	Programação	Minimizar custo	Petróleo	Aplicado a uma companhia <i>Major</i> de petróleo.
Scott (1995)	Programação	Minimizar custo, maximizar lucro ou maximizar satisfação do cliente	Derivados de petróleo	Permite o carregamento de diversos produtos na mesma no mesmo navio em tanques segregados.
Kim e Lee (1997)	Programação	Maximizar lucro	Granel	Apresenta uma nova abordagem para a geração das programações.
Sherali et al. (1999)	Programação	Minimizar custo	Petróleo e derivados	Maximiza a utilização de navios da frota e aplica penalidades às violações das janelas de tempo de acordo com o cliente.
Fagerholt (2004)	Programação	Minimizar custo/Maximizar lucro	Não especificada	Desenvolvido para solucionar problemas de programação de navios diversos; foca mais na interação do usuário com o sistema do que na otimização.
Bronmo et al. (2007a)	Programação	Maximizar lucro	Granel	utiliza uma heurística de busca local para achar soluções iniciais.
Bronmo et al. (2007b)	Programação	Maximizar lucro	Granel	Cargas com lotes flexíveis.
Fagerholt e Lindstad (2007)	Programação	Minimizar custo/Maximizar lucro	Não especificada	Permite a programação manual, semi-manual ou automática (com otimização).

Tabela 1: Sumário da revisão bibliográfica

<b>Artigo</b>	<b>Decisões Principais</b>	<b>Objetivo</b>	<b>Carga</b>	<b>Destaques</b>
Brown et al. (1987)	Programação e velocidade	Minimizar custo	Petróleo	Aplicação em uma companhia Major de petróleo.
Bausch et al. (1998)	Programação e velocidade	Minimizar custo	Derivados de petróleo	Interface em diversos idiomas.
Fagerholt (2001)	Programação e velocidade	Minimizar custo	Granel seco	Janelas de tempo flexíveis. Permite violar janelas de tempo para alguns clientes a algum custo.
Norstad et al. (2010)	Programação e velocidade	Maximizar lucro	Granel	Apresenta uma heurística de busca local para solucionar o problema, onde a velocidade dos navios também é uma variável.
Fagerholt e Christiansen (2000)	Programação e alocação de carga	Minimizar custo	Fertilizantes	Permite alocação das cargas em lotes menores para serem segregados nos tanques dos navios.
Christiansen e Fagerholt (2002)	Programação robusta	Minimizar custo	Granel seco	Programação robusta considerando múltiplas janelas de tempo para operação; penalidade para chegada nos portos em horário de risco.
Christiansen e Nygreen (1998)	Programação e estoque	Minimizar custo	Amônia	Pioneiro no estudo de problemas de programação de navios com gestão de estoque.
Christiansen (1999)	Programação e estoque	Minimizar custo	Amônia	Enfatiza a formulação dos fluxo de arcos.

Tabela 1: Sumário da revisão bibliográfica (Continuação)

<b>Artigo</b>	<b>Decisões Principais</b>	<b>Objetivo</b>	<b>Carga</b>	<b>Destaque</b>
Ronen (2002)	Programação e estoque	Minimizar custo	Granel líquido	Compara a utilização de uma heurística com a otimização utilizando PIM.
Person e Gothe-Lundgren (2005)	Programação e estoque	Minimizar custo	Betume	Integração da programação dos navios e dos processos das refinarias.
Al-Kayyal e Hwang (2007)	Programação e estoque	Minimizar custo	Granel líquido	Frota heterogênea com navios compartimentados.
Gronhaug et al. (2010)	Programação e estoque	Maximizar lucro	GLP	Inclui aspectos como: taxa de evaporação do produto a bordo; taxa de produção e consumo variáveis; e número variável de tanques disponíveis no terminal de descarga.
Song e Furman (2010)	Programação e estoque	Minimizar custo	Granel	Aplica algumas simplificações em relação aos estudos teóricos a fim de abranger aspectos práticos do problema.
Agra et al. (2010)	Programação e estoque	Minimizar custo	Óleo combustível	Distribuição de óleo combustível em um arquipélago em Cabo Verde. Utiliza horizonte de tempo contínuo e discreto.
Agra et al. (2011)	Programação e estoque	Minimizar custo	Granel líquido	Apresenta duas formulações para o problema utilizando o horizonte de tempo discreto.
Furman et al. (2011)	Programação e estoque	Minimizar custo	Gasóleo de vácuo	Frota heterogênea, taxa de produção e consumo variáveis pontos de carga e descarga múltiplos por navio. Aplicado em uma companhia <i>Maior</i> de petróleo.

Tabela 1: Sumário da revisão bibliográfica (Continuação)

## 2.3.

### Abordagem de Brown *et al.* (1987) e estudos subsequentes

Brown *et al.* (1987) estudaram o caso de uma grande empresa de petróleo (uma *major*) que tinha que prover o transporte de petróleo do Oriente Médio para a Europa e Estados Unidos. Para solucionar este problema, Brown *et al.* (1987) utilizaram um modelo denominado *Elastic Set Partitioning*. O modelo, desenvolvido por eles, busca identificar uma programação onde, dentro de um horizonte de tempo de aproximadamente três meses, todas as cargas possam ser transportadas a um custo mínimo. Foram considerados os principais componentes de custos em transporte marítimo, incluindo custo de oportunidade do navio, despesas portuárias e de passagem em canais, custos de combustível e sobrestadia do navio (*demurrage*), além do custo de mercado do navio, quando necessária a contratação destes (*spot charter*) para atender cargas excedentes. O modelo proposto em Brown *et al.* (1987) tem quatro etapas principais, que posteriormente foram utilizadas com algumas variações por outros autores, sendo elas: **1ª etapa:** um gerador de programações que cria um conjunto completo de programações factíveis para atender todas as cargas disponíveis; **2ª etapa:** um avaliador de custos que calcula os custos de todas as programações geradas; **3ª etapa:** um modelo de programação inteira que encontra a combinação de programações que minimiza o custo de operação da frota; e **4ª etapa:** um procedimento de solução eficiente que é aplicado ao problema. Cada uma destas quatro etapas é descrita a seguir.

#### 2.3.1.

##### 1ª Etapa: Geração das programações factíveis

Gronhaug *et al.* (2010) descreve que, de maneira geral, os problemas de programação de navios podem ser estruturados de acordo com modelos baseados em geração de rotas. Frequentemente, os tempos de navegação são muito longos em relação aos tempos gastos nos portos, tanto que as rotas possuem poucos trechos de navegação. As vantagens dos modelos baseados em rotas são que os custos e as restrições mais complexas e não lineares podem ser facilmente incorporadas ao modelo quando da geração das rotas. Alguns problemas de programação de navios são tão restritos, que em alguns casos é possível gerar todas as programações factíveis. Esta abordagem pode ser encontrada nos trabalhos de Brown *et al.* (1987), Kim e Lee (1997), Bausch *et al.* (1998), Christiansen and Fagerholt (2002) e Brønmo *et al.* (2007b). Fisher e Roseiwein (1989) definem uma programação factível como um itinerário

completo para um navio controlado, onde sejam informados em sequência os portos visitados durante um horizonte de tempo, assim como, a data e hora de entrada nestes portos e a quantidade e o tipo de produto carregado e descarregado em cada um deles. Brown *et al.* (1987) chamam de programações factíveis àquelas que atendam aos requisitos operacionais relacionados com cada navio, porto, carga e a relação entre eles. Quanto mais detalhes forem adicionados às restrições operacionais, mais realista será o resultado encontrado, o que reforça a aceitação do modelo.

Brown *et al.* (1987) utilizam um algoritmo gerador de colunas para definir todas programações factíveis dentro de um horizonte de tempo contínuo definido. Uma descrição completa dos dados da frota e das cargas se faz necessária para a geração de programações. Com relação aos navios, são necessárias informações sobre as características físicas, posição geográfica, *status* atual e disponibilidade. Já em relação às cargas, a descrição dos portos de carga e descarga e suas principais características, assim como, a descrição de rotas e passagens de canais são de fundamental importância para a definição de programações factíveis.

Kim e Lee (1997) propõem uma abordagem em duas etapas para gerar todas as programações factíveis. Primeiro, é utilizado um grafo  $G_k(V_k, A_k)$  gerador de programações factíveis, que é definido e construído para cada navio  $k$ . Em seguida, é aplicado um algoritmo específico que modifica os grafos e gera todas as programações factíveis para cada navio  $k$ . As cargas que podem ser transportadas pelo navio  $k$  são definidas considerando todas as restrições operacionais e físicas que envolvem o transporte marítimo.

Em Bausch *et al.* (1998), o gerador de programações gera todas as programações factíveis para cada navio da frota dentro do horizonte de programação. As programações são geradas respeitando a política de despacho, os dados de entrada e as restrições operacionais relacionadas com os navios, portos e cargas. O gerador recebe informações do local e data de abertura de um navio, aloca-o em algum porto de carregamento potencial e estrutura esta programação separadamente. O gerador assume esta programação estruturada e tenta encaixar novas cargas na sequência da anterior. Este processo é repetido até que não haja mais cargas ou o horizonte de tempo seja atingido. O mesmo procedimento é repetido para cada navio, gerando assim todas as programações factíveis. Segundo Bausch *et al.* (1998),

dependendo da instância do problema, um navio pode terminar com apenas algumas programações ou milhares delas.

Christiansen e Fagerholt (2002) geram todas as programações candidatas para cada navio da seguinte forma. A programação inicial consiste na sequência de três nós,  $0 - i - (n+i)$ , a posição inicial do navio, o ponto de carregamento e o ponto de descarga do produto, respectivamente. Novas programações são geradas estendendo a programação inicial para atender as próximas cargas disponíveis, uma de cada vez, de forma que a nova programação não tenha sido considerada anteriormente. O algoritmo para gerar novas programações é repetido até que não haja mais cargas que possam ser adicionadas à programação. Deve ser levado em consideração que cada nó deve ser visitado dentro da sua janela de tempo, sendo que o tempo de operação é calculado com base na chegada do navio e o nó de carregamento deve ser visitado antes do nó de descarga. O estudo de Christiansen e Fagerholt (2002) introduz ao problema restrições de múltiplas janelas de tempo, que representa os horários de funcionamento dos portos. Esta nova restrição junto às incertezas da atividade de transporte marítimo, leva os autores a considerar a programação robusta de navios. O estudo aplica um modelo artificial de custo que penaliza àquelas programações em que o navio chega no porto em um horário de risco.

Fagerholt e Christiansen (2000) apresentam um problema de programação de navios combinado com alocação de produtos nos tanques dos mesmos. A alocação dos produtos nos diferentes tanques é incluída na fase de geração das programações.

Fagerholt (2001) também utiliza um algoritmo que *a priori* gera todas as programações factíveis. Seu estudo introduz o conceito de janelas de tempo flexíveis, permitindo a violação controlada das janelas de tempo de alguns clientes, desde que isto traga reduções significativas no custo de transporte. O modelo proposto também determina a velocidade ótima para os navios em vários trechos de navegação.

A principal contribuição de Bronmo *et al.* (2007b) é a introdução do carregamento de cargas com lotes flexíveis no problema de programação de navio em viagens de curta distância. Porém, seu estudo não dá detalhes sobre a geração das programações.

Quando o número de programações geradas é muito grande e atrapalha o tempo de processamento, pode se lançar mão de heurísticas para viabilizar o modelo. Christiansen *et al.* (2004) afirmam que o método de geração de programações pode ser através de heurística ou por otimização, dependendo da qualidade da solução desejada e do tempo disponível para processamento dos dados. Algumas heurísticas para redução no número de programações geradas podem ser encontradas em Fisher e Rosenwein (1989) e Perakis e Bremer (1992).

De acordo com Fisher e Rosenwein (1989), as restrições físicas, tanto do navio, como velocidade e porte, assim como dos portos, como a profundidade e as janelas de tempo para operação, limitam significativamente o número de programações factíveis ou candidatas. Adicionalmente, Fisher e Rosenwein (1989) utilizam uma heurística para excluir programações que tem pouca probabilidade de serem ótimas, por terem em seu itinerário trechos de navegação em lastro muito longos. Esta exclusão é baseada na definição de um parâmetro chamado de MAXB, que limita o trecho em lastro entre portos de descarga e carga. O trecho em lastro não deve ser maior do que MAXB vezes o trecho de navegação com o navio carregado. Desta forma Fisher e Rosenwein (1989) limitam o número de programações candidatas diminuindo a complexidade do problema. Quanto menor for o valor de MAXB, menor é o número de programações candidatas geradas.

O algoritmo de Perakis e Bremer (1992) elaborado para gerar todas as programações factíveis para os navios da frota é baseado nas restrições das cargas, dos navios, dos portos e restrições de tempo. O programa verifica o porte do navio, o volume do produto a ser carregado, o calado e as restrições de tempo para definir as programações. Caso o número de programações geradas seja muito grande, tornando o problema proibitivo, então é utilizada uma heurística que restringe o número de programações geradas. Para tanto, o tempo máximo de espera de cada navio é restringido, de forma que, o programa elimina programações onde há previsão de uma espera muito longa do navio antes de um carregamento, entendendo que esta programação tem baixa probabilidade de ser uma solução ótima.

Bremer e Perakis (1992) abordam em seu estudo o *trade-off* entre o tamanho do problema e a qualidade da solução encontrada. Segundo eles, quanto menor a restrição quanto ao número de dias de espera de um navio,

menor será o número de programações candidatas gerado. Corre-se o risco de uma programação ótima ser desconsiderada, em função da restrição do tempo de espera do navio. Por outro lado, caso seja adotada uma restrição de tempo de espera muito longa, muitas programações podem ser geradas tornando o problema muito complexo para ser processado.

### 2.3.2.

#### 2ª Etapa: Avaliador de Custo

Segundo Brown *et al.* (1987), o avaliador de custos utiliza os mesmos dados de entrada do gerador de programações para calcular o custo de cada programação para cada navio, embora somente as programações factíveis tenham seus custos calculados. A maior preocupação de um programador de navios é como transportar todas as cargas a um custo mínimo. As decisões de programação, via de regra, incorrem nos seguintes tipos de custos:

- custo diário (*daily value*), que representa o custo de oportunidade do navio;
- consumo de combustível (bunker), que é uma função da velocidade e tempo de navegação de um navio;
- consumo de combustível para sistemas auxiliares;
- custos portuários e de passagens por canais, envolvendo aí custos específicos de quando um navio entra em um porto ou passa por algum canal;
- custo de afretamento de navio de mercado, que varia de acordo com o mercado; e
- custo do navio ocioso ou custo de espera do navio.

Appelgren (1969) destaca que um elemento importante na programação de navios é a oportunidade de aparecimento de novas cargas após o término do horizonte de programação. Isto levou ao conceito de *time value* ou custo diário ou ainda custo de oportunidade, em uma tradução livre. O custo diário premia os navios que terminaram suas programações antecipadamente e estão prontos para serem alocados para novas cargas em novas programações.

Perakis e Bremer (1992) realçam a importância do custo de oportunidade nas decisões de programação. Como o horizonte de tempo de utilização do navio pode variar de acordo com a carga que o navio for alocado, então o custo de oportunidade do navio é utilizado para equalizar os horizontes de tempos de



cada programação, a fim de que se possam comparar alternativas com o mesmo horizonte de tempo. Perakis e Bremer (1992) entendem que uma boa aproximação para o custo de oportunidade do navio poderia ser definido como a receita que um navio pode gerar na sua melhor alternativa de uso menos os custos relativos à operação do mesmo. Segundo eles, a melhor alternativa de uso de um navio seria o fretamento do navio no mercado *spot* (*out-charter*), neste caso o custo de oportunidade seria definido pela eq. (1), que representa a diferença entre a receita do fretamento do navio em viagem única, menos os custos operacionais desta viagem, dividido pelo tempo de ciclo da viagem. O tempo de ciclo é o tempo total da viagem, desde o ponto de carregamento até a descarga e posterior retorno ao ponto inicial de carregamento.

$$V_{Di} = \frac{[R_{Ai} - C_{Ai}]}{t_{Ai}} \quad (1)$$

Onde:

$V_{Di}$  = *Custo de Oportunidade*

$R_{Ai}$  = *Receita da melhor alternativa*

$C_{Ai}$  = *Custo da viagem na melhor alternativa*

$t_{Ai}$  = *tempo de ciclo da viagem na melhor alternativa*

Segundo Ronen (1982), o custo do *bunker* é o principal gasto operacional de um navio. Tanto Ronen (1982), quanto Norstad *et al.* (2010) dizem que a relação entre a velocidade do navio e o consumo de combustível pode ser aproximada por uma função cúbica. Portanto, se a velocidade de um navio for reduzida em cerca de 20%, então a redução no consumo de *bunker* pode chegar a 50%. Perakis e Bremer (1992) simplificam o entendimento de custo de bunker, considerando apenas duas taxas de consumo diferentes em seu problema, o consumo com o navio no porto e o consumo do navio navegando.

De acordo com Brown *et al.* (1987) os combustíveis auxiliares não são consumidos enquanto o navio navega, mas quando estão nos portos, durante as manobras e nos sistemas de geração de energia.

Christiansen e Fagerholt (2002) afirmam que o custo portuário depende principalmente do porte do navio que chega ao porto, quantidade de produto que

será carregada ou descarregada e do porto que está sendo visitado. É importante ressaltar que o custo portuário não tem relação com o tempo em que o navio está no porto, mas com o tempo de operação efetiva do navio carregando ou descarregando um determinado produto. Perakis e Bremer (1992) assumem que o custo portuário em um determinado porto é diretamente proporcional ao porte bruto do navio e definem uma fórmula para custo portuário.

$$\text{Custo Portuário} = C_{pj} \left( \frac{DWT_k}{150} \right) \quad (2)$$

Onde:

$C_{pj}$  = Custo portuário para um navio de 150.000 DWT (*dead weight tonnage*)

$DWT_k$  = Tonelada de porte bruto do navio  $k$

De acordo com Fisher e Rosewein (1989), os principais custos decorrentes da decisão de programação são o custo de afretamento de um navio do mercado *spot* e os custos para operar um navio da frota controlada (*daily value*, custo de combustível e custos portuários). Os custos de operação dos navios da frota controlada já foram mencionados anteriormente, já o custo de afretamento se refere a um valor determinado para o transporte de uma carga específica, ou seja, o custo de um navio contratado no mercado *spot* para transportar uma única carga durante uma única viagem. Na indústria se diz que, neste caso, este navio é contratado para um única viagem ou *voyage charter party* (VCP). De acordo com Kavussanos e Alizadeh (2002) os preços e taxas de afretamento na indústria de navegação para navios tanques variam bastante, mesmo em períodos curtos. Uma parte desta flutuação é devida a atividade econômica mundial, mas outra parte tem caráter sazonal.

De acordo com Wakamatsu (2008), a taxa de frete para navios tanques no mercado *spot* é expressa por um índice chamado de *Worldwide Tanker Nominal Freight Scale* ou apenas *worldscale*. Ainda segundo Wakamatsu (2008), o *worldscale* é amplamente utilizado no mercado de navios petroleiros ao redor do mundo. De acordo com o Kumar (2004), o termo *worldscale* (WS) é utilizado para referenciar a taxa de frete no mercado *spot* de navios tanques, onde o WS100 representa a taxa nominal em dólares por tonelada métrica para uma viagem padrão com um navio de 75.000 toneladas métricas e um *hire* de US\$ 12.000. *Hire* é a expressão em inglês utilizada na indústria que significa a taxa

de aluguel diária do navio pago aos armadores. Anualmente, a *Worldscale Association* publica uma lista de taxas WS100 ou *flat* para todas as potenciais rotas comerciais do mundo. O *flat* representa o custo do frete em uma determinada rota expresso em dólar por tonelada transportada (Wakamatsu, 2008). Ainda segundo Kumar (2004), o princípio do WS é que, independente da viagem e da rota, navios idênticos vão receber a mesma receita bruta por dia de viagem. Como a receita total é proporcional à quantidade de carga carregada, então o sistema WS serve de ponto de partida para negociações no mercado spot. O cálculo do frete utilizando a metodologia do WS ocorre da seguinte forma: o lote da carga, medido em toneladas, multiplicado pelo *flat*, expresso em dólares por tonelada e, finalmente, multiplicado pela cotação do WS no mercado *spot* dividido por 100. Desta forma, se obtém o custo de afretamento de um navio tanque medido em dólares (Worldscale Association, 2011). Por exemplo, se uma empresa precisa afretar um navio para transportar uma carga de 130.000 toneladas de petróleo do porto de Angra dos Reis, no Brasil, até o porto de Nova Iorque, nos Estados Unidos, sabe-se que o *flat* desta rota em 2011 é de 18,03 dólares/tonelada e supondo que o WS nesta rota seja de 80 pontos o custo deste frete seria calculado da seguinte forma:

$$\text{Custo de Frete} = 130.000 * 18,03 * \frac{80}{100} = \text{US\$ } 1.875.120$$

Para Brown *et al.* (1987) o custo do navio ocioso ou do navio em espera é uma combinação do custo de oportunidade, do custo dos combustíveis auxiliares e dos custos portuários. Em navios afretados por viagem, esta é a quantia paga aos armadores pelos atrasos nos portos, também chamados de *demurrage cost*.

Christiansen *et al.* (2007) enfatizam que os custos fixos da frota são desconsiderados, uma vez que não têm influência sobre a programação dos navios, ou seja, independente da decisão de programação que for tomada o custo fixo vai continuar o mesmo. Perakis e Bremer (1992) concordam que apenas os custos que são afetados pelas decisões de programação devem ser levados em consideração na análise, por esta razão todos os custos fixos relacionados com o transporte marítimo são desconsiderados em sua análise, corroborando assim com Christiansen *et al.* (2007).

### 2.3.3.

#### 3ª Etapa: Modelo de Programação Inteira

Esta subseção apresenta modelos de minimização de custos encontrados na literatura para um problema típico de *industrial shipping*, considerando o caso do *full shipload* e a possibilidade de se contratar navios no mercado *spot*.

Christiansen *et al.* (2004) buscam minimizar o custo de operação de todos os navios de uma frota, enquanto assegura que todas as cargas serão transportadas do porto de origem ao porto de destino.

A formulação do problema de programação de navios apresentado por Christiansen *et al.* (2004) é dada a seguir:

$$\min \sum_{v \in V} \sum_{r \in Rv} C_{vr} \cdot x_{vr} \quad (3)$$

Sujeito à:

$$\sum_{v \in V} \sum_{r \in Rv} a_{ivr} \cdot x_{vr} = 1, \quad \forall i \in N, \quad (4)$$

$$\sum_{r \in Rv} x_{vr} = 1, \quad \forall v \in V, \quad (5)$$

$$x_{vr} \in \{0,1\}, \quad \forall v \in V, \quad r \in Rv \quad (6)$$

Em seu estudo, Christiansen *et al.* (2004) utiliza a seguinte notação:

V: conjunto de navios

v: índice do navio

N: conjunto de cargas

i: índice da carga

R<sub>v</sub>: conjunto de rotas candidatas para cada navio v

r: índice da programação

C<sub>vr</sub>: custo da programação r com o navio v

$a_{ivr}$ : constante é igual a 1 se a programação  $r$  é feita pelo navio  $v$  para servir a carga  $i$ ; senão é 0, em caso contrário

$x_{vr}$ : variável binária igual a 1 se o navio  $v$  faz a programação  $r$ , senão é 0 em caso contrário

Ainda segundo Christiansen *et al.* (2004), caso a frota disponível pelo dono da carga não seja suficiente para atender todas as cargas, pode ser necessário a contratação de navios *spot charter*. Estes navios são contratados para única viagem e após término da descarga são devolvidos aos armadores.

Neste caso, (3) e (4) se transformam em:

$$\min \sum_{v \in V} \sum_{r \in Rv} c_{vr} \cdot x_{vr} + \sum_{i \in N} C_{spot_i} \cdot s_i, \quad (3')$$

$$\sum_{v \in V} \sum_{r \in Rv} a_{ivr} \cdot x_{vr} + s_i = 1, \quad \forall i \in N, \quad (4')$$

$$s_i \in \{0,1\}, \quad \forall i \in N \quad (7)$$

(3) e (3') minimizam os custos

(4) e (4') garantem que todas as cargas serão carregadas

(5) garante que cada navio da frota será alocado em apenas uma programação

(6) e (7) identificam e restringem às variáveis do sistema como binárias, onde:

$s_i$ : variável binária igual a 1, se a carga for atendida por um navio spot, senão 0

$C_{spot_i}$ : custo associado à contratação de navio spot

Bausch *et al.* (1998) incluíram uma variável binária no seu modelo que informa se o navio está ocioso ou não e uma constante que informa o custo diário do navio ocioso. Segue notação:

$Idle_v$  é igual 1 se o navio estiver em sobrestadia ou ocioso durante todo o horizonte de programação, senão igual a 0.

$ICost_v$  é o custo do navio  $v$  enquanto ocioso.

Guardada a diferença de notação, no problema apresentado por Bausch *et al.* (1998), a função objetivo seria alterada da seguinte forma:

$$\min \sum_{v \in V} \sum_{r \in Rv} c_{vr} \cdot x_{vr} + \sum_{i \in N} c_{spot} \cdot s_i + \sum_{v \in V} ICost_v \cdot Idle_v \quad (3'')$$

A eq. (5), referente à restrição que garante que todo navio seria programado, seria então alterada da seguinte forma:

$$\sum_{r \in Rv} x_{vr} + idle_v = 1, \quad \forall v \in N, \quad (5')$$

$$idle_v \in \{0,1\}, \quad \forall v \in V \quad (8)$$

(3'') garante que o custo de sobrestadia será incluído na função objetivo

(5') permite e garante que todo navio esteja alocado em uma única rota ou esteja em sobrestadia.

(8) identifica e restringe à variável *idle* como binária

#### 2.3.4.

#### 4ª Etapa: Aplicação de um procedimento de solução eficiente

A última etapa dentro da estrutura do modelo de Brown *et al.* (1987) é a aplicação de um *software* de otimização eficiente. Tanto Brown *et al.* (1987) quanto Bausch *et al.* (1998) utilizam um sistema chamado de X-System para otimizar seus modelos matemáticos. O algoritmo do X-System é ambivalente alternando entre operações primal e dual até encontrar, no máximo, uma programação por navio e garantir que todas as cargas sejam transportadas. Bausch *et al.* (1998) utilizaram o *software* Microsoft Excel® como interface para o usuário, o que permite que a interface seja disponibilizada em diversos idiomas, além de ter uma visualização familiar para a maioria dos usuários. Kim e Lee (1997) utilizam o LINDO para otimizar seu modelo. A solução é disponibilizada de forma amigável, com utilização de gráficos e tabelas. Christiansen e Fagerholt (2002) escrevem os algoritmos para geração das programações em *Borland Pascal 7.0*. O problema é implementado e otimizado utilizando *GAMS/CPLEX* versão 5.0. Em uma versão inicial, Furman *et al.* (2011) utilizaram o Microsoft Excel® como interface com o usuário e para entrada de dados, o *GAMS* foi utilizado para formulação do problema e o *ILOG-CPLEX* como otimizador do modelo. Posteriormente, Furman *et al.* (2011) continuaram utilizando o Microsoft Excel® como interface com o usuário, porém os dados exportados para o AIMMS®, uma plataforma de modelagem, no qual o problema

foi implementado. O *software* de otimização utilizado pelo AIMMS® também é o *CPLEX*.