FORMULAÇÕES PARA O PROBLEMA DO TRANSPORTE DE DERIVADOS DE PETRÓLEO

Luiz Aizemberg, Artur Alves Pessoa, Eduardo Uchoa, Hugo Harry Kramer
Departamento de Engenharia de Produção - Universidade Federal Fluminense
Rua Passo da Pátria, 156, Bloco E, 4º andar, São Domingos, 24210-240, Niterói, RJ
luizaizemberg@gmail.com, {artur, uchoa}@producao.uff.br, hugoharry@gmail.com

Rafaelli Coutinho, Ubiratam de Paula
Instituto de Computação - Universidade Federal Fluminense
Rua Passo da Pátria, 156, Bloco E, 3º andar, São Domingos, 24210-240, Niterói, RJ
{rcoutinho, upaula}@ic.uff.br

Roger Rocha
CENPES - Petrobrás
Av. Horácio Macedo, 950, Cidade Universitária, Rio de Janeiro, RJ
rogerocha@petrobras.com.br

RESUMO

O presente artigo faz um estudo de modelagem por programação matemática para o problema de transporte de derivados de petróleo. Este estudo gerou várias formulações que foram testadas e comparadas utilizando um conjunto de 50 instâncias encontradas na literatura. Bons resultados foram obtidos, de forma que uma das formulações foi capaz de resolver 46 destas instâncias em baixo tempo computacional. Com o objetivo de melhor avaliar o potencial desta formulação, foram geradas 25 novas instâncias com maior grau de dificuldade. Para os testes com as novas instâncias, duas suposições foram consideradas. Tais suposições influenciam o desempenho da formulação sem prejudicar os requisitos de solução desejados para a situação real do problema.
PALAVRAS–CHAVE. Programação Matemática, Logística, Petróleo.

Áreas Principais: Otimização, Gerenciamento da Cadeia de Suprimentos.

ABSTRACT

This paper presents a mathematical programming study for the crude oil transportation problem. Such study results in a bunch of formulations which were tested using a set of 50 instances from literature. Good results were obtained, since one of the formulations was able to solve 46 of those instances to optimality in a reasonable computational time. For a better assesment of such formulation, 25 harder instances were created. Two assumptions are considered for the tests with these new instances. These assumptions influence the performance of the formulation keeping the requirements for a solution in real world situation.
KEY WORDS. Mathematical Programming, Logistics, Petroleum.

Main areas: Optimization, Supply Chain Management.

1 Introdução

A quantidade de recursos humanos e financeiros investida pela Petrobras no desenvolvimento de sistemas de otimização é significativa e visa aumentar a eficiência de sua cadeia de suprimentos. Dentre as técnicas de otimização aplicadas no desenvolvimento desses sistemas, encontra-se a programação matemática, devido à facilidade na construção dos modelos e à disponibilidade de pacotes comerciais altamente especializados para sua resolução.

O problema da cadeia de suprimentos da Petrobras (Aizemberg *et al.*, 2011) é de extrema complexidade. Uma boa forma de abordagem de solução de problemas complexos é através de relaxações do mesmo, ou até mesmo utilizando um problema similar, porém bem mais simples. Neste artigo, a segunda opção foi utilizada. O problema similar tratado aqui consiste em escalonar as diversas rotas disponíveis entre as plataformas (pontos de produção marítimos) e os terminais (pontos de demanda terrestres) para escoar a produção dos primeiros, provendo os produtos para os segundos. Os estoques de todos os lugares têm que ser respeitados, tanto no limite superior quanto no inferior, não sendo permitida a falta ou sobra de produto. Todos os modais de transporte são marítimos, e assume-se ilimitada a capacidade global da frota. Esta suposição não foge à realidade, dado que a Petrobras pode alugar mais navios, caso seja necessário.

Com a complexidade do problema drasticamente reduzida, um estudo de modelagem matemática torna-se viável. Este estudo começa com uma formulação intuitiva, onde um conjunto de variáveis contínuas representa os estoques em cada período, e um conjunto de variáveis binárias representa os lotes sendo enviados pelas rotas. Passa por formulações mais elaboradas, que requerem mais conhecimento do problema, como é o caso das formulações em que os estoques só podem assumir valores múltiplos do lote mínimo, ou quando os estoques são arredondados, não permitindo que os mesmos possam ter valores que deixariam a solução inviável. Termina com uma transformação total no modo de pensar a modelagem, que leva em conta o desempenho da mesma ao ser executada pelo algoritmo de *branch-and-bound* do resolvedor utilizado.

As instâncias utilizadas para testar o desempenho das diversas formulações foram as mesmas utilizadas por Rocha (2010). Os resultados mostram que houve um avanço importante de desempenho à medida que as formulações foram sendo criadas e aperfeiçoadas, o que levou à criação de novas e mais difíceis instâncias, baseadas nas existentes.

Este mesmo problema também é tratado através de abordagens diferentes da apresentada neste trabalho, bem como com considerações que aumentam sua complexidade. Banaszewski *et al.* (2010) tratam do planejamento de médio prazo da transferência de produtos derivados de petróleo na rede multimodal da cadeia de suprimentos da indústria petrolífera do Brasil utilizando modelos de sistemas multiagentes baseados em leilões. Magatão *et al.* (2004) e Boschetto *et al.* (2010) apresentam abordagens baseadas em Programação Linear Inteira Mista para otimizar o escalonamento de atividades em operações de um oleoduto real.

O restante do artigo está organizado da seguinte forma: a Seção 2 apresenta o problema estudado e descreve detalhadamente todas as formulações desenvolvidas e testadas; a Seção 3 mostra e compara os resultados obtidos através de valores médios; por fim, a Seção 4 lista as conclusões e os trabalhos futuros.

2 Formulações matemáticas

Nesta seção, serão apresentadas todas as formulações matemáticas desenvolvidas e testadas neste trabalho. Será visto que a formulação Cascading Arredondado Acumulado obteve os melhores resultados, sendo muito superior às outras nos testes com as instâncias geradas por Rocha (2010).

O problema de transporte de derivados de petróleo tratado neste artigo pode ser descrito da seguinte forma: sejam P o conjunto de plataformas e T o conjunto de terminais. O horizonte de tempo considerado para o planejamento é dado por D em dias. Um único produto deve ser transportado das plataformas para os terminais de modo que as demandas destes últimos sejam satisfeitas. Os níveis de estoque em todos os elementos de P e T devem ser respeitados. O único tipo de modal utilizado é o marítimo. Os navios utilizados são classificados de acordo com sua capacidade volumétrica em classes, e o conjunto formado por todas as classes de navios é chamado de C. O objetivo é a minimização dos custos totais de transporte do produto. Além disso, as formulações descritas nas próximas subseções seguem a seguinte notação adicional:

Índices:

- p: Plataforma, $p \in P = \{1, ..., |P|\}$;
- t: Terminal, $t \in T = \{|P| + 1, ..., |P| + |T|\}$;
- c: Classe de navio, $c \in C = \{1, ..., |C|\}$;
- d: Dia, $d = \{1, ..., D\}$.

Subconjuntos:

- $T(p) \subseteq T$: Terminais que podem receber navios de uma plataforma $p \in P$;
- $P(t) \subseteq P$: Plataformas que podem enviar navios para um terminal $t \in T$;
- $C(p) \subseteq C$: Classes de navios que podem aportar em uma plataforma $p \in P$.

Todos os navios são permitidos nos terminais.

Dados:

- $e_{p,0}$: Estoque inicial de uma plataforma $p \in P$;
- $e_{t,0}$: Estoque inicial de um terminal $t \in T$;
- $P_{p,d} \geq 0$: Produção de uma plataforma $p \in P$ em um dia d;
- $C_{t,d} \leq 0$: Consumo de um terminal $t \in T$ em um dia d;
- $CAP_{p,d}$: Capacidade máxima de estoque de uma plataforma $p \in P$ em um dia d;
- $CAP_{t,d}$: Capacidade máxima de estoque de um terminal $t \in T$ em um dia d;
- V_c: Volume de um navio de classe $c \in C$;
- F_c: Custo de transporte por dia de um navio de classe $c \in C$;
- $D_{p,t}$: Tempo de trânsito entre uma plataforma $p \in P$ e um terminal $t \in T$ em dias.

Ademais, outras definições serão acrescentadas à medida que sejam necessárias para cada formulação.

2.1 Formulação Natural

Este modelo foi construído baseado no apresentado por Rocha (2010). Para esta formulação, consideram-se as variáveis de decisão da seguinte forma: $z_{p,t}^{c,d} \in \{0, 1\}$ é uma variável binária que indica se um navio de classe $c \in C$ foi alocado para transportar o produto da plataforma $p \in P$ para o terminal $t \in T$ saindo no dia d, e as variáveis $e_{p,d} \in \mathbb{R}_+$ e $e_{t,d} \in \mathbb{R}_+$ indicam o estoque na plataforma $p \in P$ no dia d e o estoque no terminal $t \in T$ no dia d, respectivamente. Isto posto, a formulação natural $(\mathcal{FN})$ está disposta a seguir:

$$(\mathcal{FN})\ z = Min \sum_{p\in P}\sum_{t\in T(p)}\sum_{c\in C(p)}\sum_{d=1}^{D} 2F_c D_{p,t} z_{p,t}^{c,d} \tag{1}$$

sujeito a

$$e_{p,d} = e_{p,d-1} + P_{p,d} - \sum_{t\in T(p)}\sum_{c\in C(p)} V_c z_{p,t}^{c,d} \qquad \forall p \in P, d \in \{1,...,D\} \tag{2}$$

$$e_{t,d} = e_{t,d-1} + C_{t,d} + \sum_{p\in P(t)}\sum_{c\in C(p)} V_c z_{p,t}^{c,d} \qquad \forall t \in T, d \in \{1,...,D\} \tag{3}$$

$$0 \le e_{p,d} \le CAP_{p,d} \qquad \forall p \in P, d \in \{1,...,D\} \tag{4}$$

$$0 \le e_{t,d} \le CAP_{t,d} \qquad \forall t \in T, d \in \{1,...,D\} \tag{5}$$

$$z_{p,t}^{c,d} \in \{0,1\} \qquad \forall p \in P, t \in T(p), c \in C(p), d \in \{1,...,D\}. \tag{6}$$

A função objetivo (1) minimiza o custo total de transporte. O custo é multiplicado por 2 porque os navios retornam à plataforma de onde saíram. As restrições (2) calculam o balanço do estoque em cada plataforma, enquanto as restrições (3) calculam o balanço do estoque em cada terminal. As restrições (4) e (5) garantem que o estoque a cada dia, em cada plataforma e em cada terminal, não será nem menor que a capacidade mínima e nem maior que a capacidade máxima de estoque.

2.2 Formulação Básica

Para esta segunda formulação são consideradas algumas definições que serão listadas a seguir. Seja $k_{min}(p,d)$ o número mínimo de lotes que precisam ser enviados até o dia d pela plataforma p para que a capacidade não seja excedida. Esse número é dado por:

$$k_{min}(p,d) = \left\lceil \frac{AC_{p,d} - CAP_{p,d}}{L_p} \right\rceil.$$

Seja também $k_{max}(p,d)$ o número máximo de lotes que podem ser enviados até o dia d pela plataforma p sem que o estoque desta fique negativo. Tal quantidade pode ser calculada pela seguinte expressão:

$$k_{max}(p,d) = \left\lfloor \frac{AC_{p,d}}{L_p} \right\rfloor,$$

onde L_p é o MDC das capacidades de armazenamento de todas as classes de navios que podem atracar na plataforma $p \in P$, e $AC_{p,d}$ é a quantidade do produto acumulada na plataforma $p \in P$ do dia 0 até o dia d e é dada por:

$$AC_{p,d} = e_{p,0} + \sum_{\tau=1}^{d} P_{p,\tau}.$$

Com estas definições, a restrição (4) pode ser substituída por:

$$AC_{p,d} - k_{max}(p,d)L_p \le e_{p,d} \le AC_{p,d} - k_{min}(p,d)L_p \qquad \forall p \in P, d \in \{1,...,D\}. \tag{7}$$

Agora, seja $k_{min}(t,d)$ o número mínimo de lotes que precisam ser recebidos pelo terminal $t \in T$ até o dia d para que o estoque deste não fique negativo. Tal quantidade é dada por:

$$k_{min}(t,d) = \left\lfloor \frac{AC_{t,d}}{L_t} \right\rfloor.$$

Seja $k_{max}(t,d)$ o número máximo de lotes que podem ser recebidos pelo terminal $t \in T$ até o dia d sem que a capacidade de estoque deste seja excedida. Esse número é calculado pela expressão que segue:

$$k_{max}(t,d) = \left\lceil \frac{AC_{t,d} - CAP_{t,d}}{L_t} \right\rceil,$$

onde L_t é o MDC das capacidades de armazenamento de todas as classes de navios que podem atracar no terminal $t \in T$, e $AC_{t,d}$ é a quantidade do produto acumulada no terminal $t \in T$ do dia 0 até o dia d e é dada por:

$$AC_{t,d} = e_{t,0} + \sum_{\tau=1}^{d} C_{t,\tau}.$$

Isto posto, a restrição (5) pode ser substituída por:

$$AC_{t,d} - k_{min}(t,d)L_t \leq e_{t,d} \leq AC_{t,d} - k_{max}(t,d)L_t \qquad \forall t \in T, d \in \{1,...,D\}. \tag{8}$$

Desta forma, a formulação básica ($\mathcal{FB}$) será dada pela função objetivo (1) sujeita às restrições (2), (3), (7), (8) e (6).

2.3 Knapsack Cascading

Para esta formulação, adaptada de Rocha (2010), são consideradas algumas definições anteriores, além das que seguem:

$$DA_{p,d} = AC_{p,d} - CAP_{p,d}$$

$$DA_{t,d} = AC_{t,d} - CAP_{t,d}$$

Assim, a formulação Knapsack Cascading ($\mathcal{KC}$) é mostrada a seguir:

$$(\mathcal{KC})\ z = Min \sum_{p \in P} \sum_{t \in T(p)} \sum_{c \in C(p)} \sum_{d=1}^{D} 2F_c D_{p,t} z_{p,t}^{c,d} \tag{9}$$

sujeito a

$$\sum_{t \in T(p)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau} \leq AC_{p,d} \qquad \forall p \in P, d \in \{1,...,D\} \tag{10}$$

$$\sum_{t \in T(p)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau} \geq DA_{p,d} \qquad \forall p \in P, d \in \{1,...,D\} \tag{11}$$

$$\sum_{p \in P(t)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau-D_{p,t}} \geq -AC_{t,d} \qquad \forall t \in T, d \in \{1,...,D\} \tag{12}$$

$$\sum_{p \in P(t)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau-D_{p,t}} \leq -DA_{t,d} \qquad \forall t \in T, d \in \{1,...,D\} \tag{13}$$

$$z_{p,t}^{c,d} \in \{0,1\} \qquad \forall p \in P, t \in T(p), c \in C(p), d \in \{1,...,D\}. \tag{14}$$

A função objetivo (9) minimiza o custo total de transporte. As restrições (10) garantem que o estoque em cada plataforma nunca será menor que a capacidade mínima. As restrições (11) garantem que o estoque em cada plataforma nunca ultrapassará a capacidade máxima. As restrições

(12) garantem que o estoque em cada terminal nunca será menor que a capacidade mínima. As restrições (13) garantem que o estoque em cada terminal nunca ultrapassará a capacidade máxima. As restrições (14) declaram as variáveis z como binárias.

A partir da formulação $\mathcal{KC}$ é possível obter uma nova formulação ao substituir as restrições (10) e (11) pelas restrições (15) e (16) a seguir

$$\sum_{t \in T(p)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau} \leq k_{max}(p,d) L_p \qquad \forall p \in P, d \in \{1,...,D\} \tag{15}$$

$$\sum_{t \in T(p)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau} \geq k_{min}(p,d) L_p \qquad \forall p \in P, d \in \{1,...,D\} \tag{16}$$

e ao substituir as restrições (12) e (13) pelas restrições (17) e (18) a seguir

$$\sum_{p \in P(t)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau - D_{p,t}} \leq -k_{max}(t,d) L_t \qquad \forall t \in T, d \in \{1,...,D\} \tag{17}$$

$$\sum_{p \in P(t)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} V_c z_{p,t}^{c,\tau - D_{p,t}} \geq -k_{min}(t,d) L_t \qquad \forall t \in T, d \in \{1,...,D\}. \tag{18}$$

A formulação resultante destas substituições é chamada de Knapsack Cascading Arredondado ($\mathcal{KCA}$).

2.4 Cascading Acumulado Arredondado

Nesta formulação, as variáveis de decisão são definidas como segue: $x_{p,t}^{c,d}$ é uma variável inteira que representa a quantidade de navios de classe $c \in C(p)$ que saem da plataforma $p \in P$ para o terminal $t \in T(p)$ do início do horizonte de tempo até o dia d. Também são consideradas as variáveis $z_{p,t}^{c,d}$ já definidas anteriormente. A formulação Cascading Acumulado Arredondado ($\mathcal{CAA}$) está disposta a seguir.

$$(\mathcal{CAA}) \; z = Min \sum_{p \in P} \sum_{t \in T(p)} \sum_{c \in C(p)} 2 F_c D_{p,t} x_{p,t}^{c,D} \tag{19}$$

sujeito a

$$x_{p,t}^{c,d} = x_{p,t}^{c,d-1} + z_{p,t}^{c,d} \qquad \forall p \in P, t \in T(p), c \in C(p), d \in \{1,...,D\} \tag{20}$$

$$\sum_{t \in T(p)} \sum_{c \in C(p)} V_c x_{p,t}^{c,d} \leq k_{max}(p,d) L_p \qquad \forall p \in P, d \in \{1,...,D\} \tag{21}$$

$$\sum_{t \in T(p)} \sum_{c \in C(p)} V_c x_{p,t}^{c,d} \geq k_{min}(p,d) L_p \qquad \forall p \in P, d \in \{1,...,D\} \tag{22}$$

$$\sum_{p \in P(t)} \sum_{c \in C(p)} V_c x_{p,t}^{c,d-D_{p,t}} \leq -k_{max}(t,d) L_t \qquad \forall t \in T, d \in \{1,...,D\} \tag{23}$$

$$\sum_{p \in P(t)} \sum_{c \in C(p)} V_c x_{p,t}^{c,d-D_{p,t}} \geq -k_{min}(t,d) L_t \qquad \forall t \in T, d \in \{1,...,D\} \tag{24}$$

$$x_{p,t}^{c,d} \in \mathbb{Z}_+ \qquad \forall p \in P, t \in T(p), c \in C(p), d \in \{1,...,D\} \tag{25}$$

$$z_{p,t}^{c,d} \in \{0,1\} \text{ ou } [0,1] \qquad \forall p \in P, t \in T(p), c \in C(p), d \in \{1,...,D\}. \tag{26}$$

A função objetivo (19) minimiza o custo total de transporte. As restrições (20) transformam x em variáveis acumuladas, e limitam em uma unidade a quantidade máxima de lotes enviados a cada dia, por cada plataforma. As restrições (21) e (22) garantem que o estoque em cada plataforma, a cada dia, não será menor que a capacidade mínima e não ultrapassará a capacidade máxima. As restrições (23) e (24) garantem que o estoque em cada terminal, a cada dia, não ultrapassará a capacidade máxima e não será menor que a capacidade mínima. Por fim, as restrições (25) declaram as variáveis x como inteiras não-negativas e as restrições (26) declaram as variáveis z como binárias ou contínuas entre zero e um.

2.5 Formulação para geração de colunas

A formulação que será apresentada a seguir é utilizada apenas com o objetivo de obter limites inferiores para o problema.

Para a formulação para geração de colunas considera-se a seguinte notação. Seja x uma solução, $x \in X_p = \{1, ..., |X_p|\}$, onde X_p é o conjunto das soluções válidas para uma plataforma $p \in P$. Sejam também $V_{p,x}^{t,d}$ o volume enviado para o terminal t na solução x para a plataforma p, e $C_{p,x}$ é o custo da solução x para a plataforma p. A variável $\lambda_{p,x}$ é binária e indica se a solução x para a plataforma p está sendo usada. A formulação para geração de colunas ($\mathcal{FGC}$) é da seguinte forma:

$$(\mathcal{FGC})\ z = Min \sum_{p \in P} \sum_{x \in X_p} C_{p,x} \lambda_{p,x} \tag{27}$$

sujeito a

$$\sum_{x \in X_p} \lambda_{p,x} = 1 \qquad \forall p \in P \tag{28}$$

$$\sum_{p \in P} \sum_{x \in X_p} \sum_{\tau=1}^{d} V_{p,x}^{t,\tau - D_{p,t}} \lambda_{p,x} \geq -AC_{t,d} \qquad \forall t \in T, d \in \{1, ..., D\} \tag{29}$$

$$\sum_{p \in P} \sum_{x \in X_p} \sum_{\tau=1}^{d} V_{p,x}^{t,\tau - D_{p,t}} \lambda_{p,x} \leq -DA_{t,d} \qquad \forall t \in T, d \in \{1, ..., D\} \tag{30}$$

$$\lambda_{p,x} \in \{0, 1\} \qquad \forall p \in P, x \in X_p. \tag{31}$$

A função objetivo (27) minimiza o custo total de transporte. As restrições (28) garantem que exatamente uma variável λ será escolhida para cada plataforma. As restrições (29) garantem que o estoque em cada terminal nunca será menor que a capacidade mínima. As restrições (30) garantem que o estoque em cada terminal nunca ultrapassará a capacidade máxima. Por fim, as restrições (31) declaram as variáveis λ como binárias.

2.5.1 Subproblema de *pricing*

Como o número de variáveis da formulação $\mathcal{FGC}$ é muito grande, apenas um subconjunto destas é considerado inicialmente para a resolução da relaxação linear. Este problema com uma parte das variáveis é chamado de Problema Mestre Restrito (PMR). A cada iteração do procedimento de geração de colunas, o PMR é resolvido e suas variáveis duais são utilizadas como entrada para o subproblema de *pricing* para gerar novas variáveis λ. Uma nova variável λ é adicionada ao PMR se seu custo reduzido, dado pelo valor da função objetivo do subproblema de *pricing*, for negativo. Caso nenhuma variável de custo reduzido negativo seja encontrada, a solução corrente do PMR é ótima para o Problema Mestre Completo.

Sejam π_p o vetor das variáveis duais associadas às restrições (28), $\theta_{t,d}$ o vetor das variáveis duais das restrições (29), e $\sigma_{t,d}$ o vetor das variáveis duais associadas às restições (30). A variável binária $z_{p,t}^{c,d}$ indica se a plataforma p envia para o terminal t um navio de classe c no dia d. Desta forma, para cada plataforma $p \in P$ é definido um subproblema de *pricing* como segue.

$$(\mathcal{SP})\ z = Min \sum_{t \in T(p)} \sum_{c \in C(p)} \sum_{d=1}^{D} (2F_c D_{p,t} - V_c(\theta_{t,d+D_{p,t}} + \sigma_{t,d+D_{p,t}})) z_{p,t}^{c,d} - \pi_p \tag{32}$$

Sujeito a

$$\sum_{t \in T(p)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} z_{p,t}^{c,\tau} V_c \leq AC_{p,d} \qquad \forall d \in \{1, ..., D\} \tag{33}$$

$$\sum_{t \in T(p)} \sum_{\tau=1}^{d} \sum_{c \in C(p)} z_{p,t}^{c,\tau} V_c \geq DA_{p,d} \qquad \forall d \in \{1, ..., D\} \tag{34}$$

$$z_{p,t}^{c,d} \in \{0, 1\} \qquad \forall t \in T(p), c \in C(p), d \in \{1, ..., D\}. \tag{35}$$

A função do subproblema de *pricing* é gerar uma variável λ para o PMR com o menor custo reduzido, que é representado pela função objetivo (32). As restrições (33) garantem que o estoque em cada plataforma nunca será menor que a capacidade mínima. As restrições (34) garantem que o estoque em cada plataforma nunca ultrapassará a capacidade máxima. As restrições (35) declaram as variáveis z como binárias. Apesar de ter sido apresentado como um problema de Programação Inteira, o subproblema de *pricing* pode ser melhor resolvido por programação dinâmica.

3 Resultados computacionais

Esta seção contém os resultados obtidos por cada formulação nas instâncias de Rocha (2010) e em uma nova classe de instâncias mais difíceis geradas neste trabalho. Todos os testes, incluindo o algoritmo de melhor desempenho de Rocha (2010), *HullRel*, foram rodados em um computador com processador Intel Core2 Quad CPU Q8300 2.50GHz, memória de 2 GB, na plataforma Windows Seven, com o resolvedor Cplex 12.1. As instâncias possuem as seguintes características gerais: um único produto (óleo cru), um único modal (navio), 9 locais de produção e 2 locais de consumo. Outras características, relacionadas às classes de instâncias são:

- um único tamanho de lote (fácil);
- um tamanho de lote por rota (médio);
- dois tamanhos de lote por rota (difícil);
- cinco tamanhos de lote por rota e todas as plataformas podem enviar para todos os terminais (novas instâncias, mais difíceis).

Neste artigo não serão reportados os resultados obtidos para a primeira classe de instâncias (fáceis), dado que estas instâncias podem ser facilmente resolvidas pela formulação $\mathcal{FN}$ utilizando um resolvedor comercial. Os resultados para as classes média e difícil serão mostrados separadamente dos resultados da nova classe de instâncias. Para cada formulação, as tabelas contém os resultados médios obtidos. Dentre estes resultados, encontram-se dois tipos de diferença relativa percentual (*gap*), dados por:

$$gap_{LB} = 100 \times \frac{UB_m - LB_f}{UB_m}, \tag{36}$$

$$gap_{UB} = 100 \times \frac{UB_f - UB_m}{UB_m}, \tag{37}$$

onde UB_m é a melhor solução viável conhecida, LB_f é o melhor *lower bound* para uma dada formulação, e UB_f é a melhor solução viável para uma dada formulação.

Os resultados para as instâncias das classes média e difícil são mostrados nas Tabelas 1 e 2. Os resultados das formulações $\mathcal{FN}$ e $\mathcal{KC}$ não são mostrados pelo fato de que a formulação $\mathcal{FB}$ domina a primeira e a formulação $\mathcal{KCA}$ domina a segunda. Nestas tabelas, a coluna **Tempo (s)** mostra o tempo de execução médio em segundos para cada classe de instâncias. Salienta-se que, em todos os testes realizados, o tempo máximo de execução permitido para cada instância é de 720 segundos. A coluna $\mathbf{gap_{LB}}$ **raiz (%)** mostra o gap_{LB} médio na raiz, calculado utilizando o gap_{LB} de cada instância dado por (36) considerando LB_f como o limite inferior do nó raiz. A coluna $\mathbf{gap_{LB}}$ **final (%)** contém a média dos gap_{LB} ao final da execução de cada instância e também é calculado utilizando os valores de (36) obtidos para cada instância, com LB_f sendo o limite inferior ao final da execução de cada instância. A coluna $\mathbf{gap_{UB}}$ **(%)** mostra o gap_{UB} médio obtido ao final das execuções calculado utilizando os valores obtidos por (37) para cada instância. A coluna **Nº de nós** contém o número médio de nós resolvidos no algoritmo de *branch-and-bound*. Por fim, a coluna **Instâncias resolvidas** mostra o número de instâncias de cada classe que foram resolvidas até a otimalidade.

Tabela 1: Resultados médios e número de instâncias resolvidas para as formulações matemáticas e resultados de Rocha (2010) para as instâncias da classe média

Formulação	Tempo (s)	gap_{LB} raiz (%)	gap_{LB} final (%)	gap_{UB} (%)	Nº de nós	Instâncias resolvidas
$\mathcal{FB}$	685,47	0,30	0,24	0,01	3944016	2
$\mathcal{KCA}$	422,47	0,25	0,13	0,00	672843	10
$\mathcal{CAA}_{bin}$ [1]	31,76	0,20	0,00	0,00	123197	24
$\mathcal{CAA}_{cont}$ [2]	30,61	0,18	0,00	0,00	146637	24
$\mathcal{FGC}$	–	0,35	–	–	–	–
Rocha (2010)	351,53	–	–	0,04	–	13

[1]: Formulação $\mathcal{CAA}$ com variáveis z binárias.

[2]: Formulação $\mathcal{CAA}$ com variáveis z contínuas entre 0 e 1.

Tabela 2: Resultados médios e número de instâncias resolvidas para as formulações matemáticas e resultados de Rocha (2010) para as instâncias da classe difícil

Formulação	Tempo (s)	gap_{LB} raiz (%)	gap_{LB} final (%)	gap_{UB} (%)	Nº de nós	Instâncias resolvidas
$\mathcal{FB}$	720,00	2,62	1,95	0,29	1326605	0
$\mathcal{KCA}$	662,75	1,96	1,13	0,27	133742	2
$\mathcal{CAA}_{bin}$ [1]	111,13	1,48	0,81	0,01	82975	22
$\mathcal{CAA}_{cont}$ [2]	102,94	1,47	0,79	0,03	53669	22
$\mathcal{FGC}$	–	1,19	–	–	–	–
Rocha (2010)	691,40	–	–	1,24	–	1

[1]: Formulação $\mathcal{CAA}$ com variáveis z binárias.

[2]: Formulação $\mathcal{CAA}$ com variáveis z contínuas entre 0 e 1.

Nas Tabelas 1 e 2 é possível observar que a formulação $\mathcal{FB}$ teve o pior desempenho em comparação com as demais. Embora o gap_{UB} médio tenha sido baixo, esta formulação só foi capaz de comprovar a otimalidade em duas instâncias, ambas de dificuldade média. Este fato também influencia no tempo médio de execução nesta mesma classe de instâncias, sendo próximo do tempo máximo permitido para cada instância. Para as instâncias da classe difícil, o tempo médio de execução é igual ao tempo máximo permitido de execução pelo fato de que esta formulação não foi capaz de comprovar a otimalidade em nenhuma instância. Por ser uma formulação leve, o número médio de nós resolvidos no *branch-and-bound* é consideravelmente alto. Em relação a esta primeira, a formulação $\mathcal{KCA}$ obteve um desempenho bem melhor, sendo capaz de resolver 12 das 50 instâncias, sendo 10 instâncias de dificuldade média e 2 instâncias difíceis. Isto se dá pelo fato de que os estoques são arredondados nesta formulação. Assim, permite-se que as variáveis de estoque somente assumam valores viáveis, diminuindo o espaço de soluções e resultando em uma boa melhora de desempenho. A formulação $\mathcal{CAA}$, pelos resultados relatados, foi a que obteve o melhor desempenho, bem superior ao da literatura (Rocha, 2010). Com esta formulação, foram resolvidas 24 instâncias médias e 22 difíceis com tempo médio de execução significativamente baixo. Das 4 instâncias não resolvidas por esta formulação, uma solução viável melhor foi conseguida com a formulação de Rocha (2010) para uma instância da classe difícil. Como nas 49 instâncias restantes o gap_{UB} foi igual a zero, na média o gap_{UB} também foi zero para a classe média e próximo de zero para a classe difícil. Vale notar que a formulação $\mathcal{CAA}$ foi testada de duas formas, com a variável de envio de lote z binária e com a mesma variável contínua entre 0 e 1. Comparando os dois resultados, nota-se uma pequena melhora no gap_{LB} médio do nó raiz. Embora seja possível que esta melhora seja aleatória, será considerado que a formulação $\mathcal{CAA}$ com variáveis de envio de lote contínua foi a que obteve o melhor desempenho entre todas as testadas nas instâncias das classes média e difícil. Com relação à formulação para geração de colunas, observa-se que esta encontrou o melhor gap_{LB} médio do nó raiz para as instâncias difíceis. No entanto, por ser uma formulação muito pesada, os resultados para soluções viáveis estão aquém daqueles das outras formulações. A intenção de estudar a formulação $\mathcal{FGC}$ é a de analisar como os limites inferiores podem ser melhorados dependendo da formulação utilizada.

Tabela 3: Resultados médios e número de instâncias resolvidas para a formulação $\mathcal{CAA}$ e suas relaxações para a classe de instâncias mais difíceis criadas

Formulação	Tempo (s)	gap_{LB} raiz (%)	gap_{LB} final (%)	N^o de nós	Instâncias resolvidas
$\mathcal{CAA}_{cont}$ [1]	637,64	8,79	7,16	14044	3
$\mathcal{CAA}_{frac}$ [2]	552,07	5,74	5,15	119432	8
$\mathcal{CAA}_{est}$ [3]	625,46	6,11	5,43	107039	4
$\mathcal{CAA}_{frac/est}$ [4]	525,00	4,80	4,15	103730	9

[1]: Formulação $\mathcal{CAA}$ com variáveis z contínuas entre 0 e 1.

[2]: Formulação $\mathcal{CAA}$ com variáveis z contínuas entre 0 e 1, e variáveis dos 4 últimos períodos fracionárias.

[3]: Formulação $\mathcal{CAA}$ com variáveis z contínuas entre 0 e 1, e possibilidade de estouro de estoque nos pontos de oferta.

[4]: Formulação $\mathcal{CAA}$ com variáveis z contínuas entre 0 e 1, variáveis dos 4 últimos períodos fracionárias, e possibilidade de estouro de estoque nos pontos de oferta.

Para os testes com a classe de instâncias mais difíceis criadas, optou-se por trabalhar apenas com a formulação que obteve o melhor desempenho nas instâncias anteriores, a formulação $\mathcal{CAA}$ com variáveis z de envio de lote contínuas. Além disso, nesta classe de instâncias, um novo teste foi realizado com a formulação $\mathcal{CAA}$, onde permite-se que as variáveis dos 4 últimos períodos sejam fracionárias. Esta é uma suposição razoável, visto que, na realidade, os valores de oferta e demanda são estimativas e, no final do horizonte de tempo tendem a ser menos precisos. Outra forma de tentar melhorar o desempenho da formulação consiste em permitir o estouro de estoque nos pontos de demanda, tanto para baixo quanto para cima. Esta suposição também é razoável, dado que, na prática, o ponto crítico é o estouro para baixo dos estoques dos pontos de oferta, já que não é possível criar produtos do nada. A vantagem disto é a possibilidade de que instâncias reais da Petrobras que seriam inviáveis pela ocorrência de excesso ou falta de produtos sejam resolvidas com a permissão de estouro, dando uma informação importante ao decisor de onde estas situações ocorrem. Por fim, as relaxações de variáveis fracionárias nos 4 últimos períodos e de estouro de estoques foram testadas simultaneamente. Os resultados para as instâncias mais difíceis são mostrados na Tabela 3.

Os resultados mostram que a formulação $\mathcal{CAA}_{cont}$ resolveu apenas 3 das novas instâncias. Com os 4 últimos períodos fracionários, o desempenho foi muito superior, tanto em termos de gap_{LB} da raiz e do gap_{LB} final quanto em quantidade de instâncias resolvidas. A permissão de estouro obteve melhora nos $gaps_{LB}$ raiz e final, mas resolveu poucas instâncias. Por fim, a formulação $\mathcal{CAA}_{frac/est}$ foi a que mostrou o melhor desempenho, sendo capaz de resolver 9 instâncias e obtendo os melhores $gaps_{LB}$ raiz e final. No geral, os gaps médios não ficaram bons e a melhor formulação resolveu apenas 36% das instâncias da classe nova criada.

4 Conclusões e trabalhos futuros

O presente artigo desenvolveu um estudo de modelagem para o problema de abastecimento da cadeia de suprimentos da Petrobras. Para facilitar o estudo, um problema parecido com o original, porém bem menos complexo, foi considerado. Três classes de instâncias retiradas da literatura foram usadas nos testes. Com o excelente desempenho obtido, uma nova classe foi criada, com instâncias de maior grau de dificuldade.

Foram implementadas e testadas cinco formulações: Formulação Natural ($\mathcal{FN}$), Formulação Básica ($\mathcal{FB}$), Knapsack Cascading ($\mathcal{KC}$), Cascading Acumulado Arredondado ($\mathcal{CAA}$) e Formulação para Geração de Colunas ($\mathcal{FGC}$), sendo a formulação $\mathcal{CAA}$ a que obteve os melhores resultados.

Comparando os resultados obtidos por Rocha (2010) com os deste trabalho, foi alcançada uma excelente melhora. Tal melhora foi conseguida apenas mudando a formulação matemática, o que pode diminuir os esforços de implementação no problema completo, que considera várias das restrições presentes na cadeia de suprimentos da Petrobras.

Como trabalhos futuros, técnicas de corte e heurísticas serão estudadas para o problema. Espera-se conseguir resolver bem a nova classe de instâncias para certificar a consistência dos métodos investigados.

Referências

Aizemberg, L., Uchoa, E., Pessoa, A., Rocha, R., Coutinho, R. e Paula Junior, U. C. Modelo de otimização para o problema do transporte de derivados de petróleo com busca local por MIP e simulação. *Anais do XLIII SBPO 2011*, Ubatuba–SP, Brasil, 2011.

Banaszewski, R. F., Tacla, C. A., Pereira, F. R., Arruda, L. V. e Enembreck, F. Planning transport of crude oil derivatives with simultaneous auctions. *2010 IEEE International Conference on Systems Man and Cybernetics (SMC)*, p. 2820–2827. IEEE, 2010.

Boschetto, S. N., Magatão, L., Brondani, W. M., Neves-Jr, F., Arruda, L. V. R., Barbosa-Póvoa, A. P. F. D. e Relvas, S. (2010), An operational scheduling model to product distribution through a pipeline network. *Industrial & Engineering Chemistry Research*, v. 49, n. 12, p. 5661–5682.

Magatão, L., Arruda, L. V. R., Neves, F. et al. (2004), A mixed integer programming approach for scheduling commodities in a pipeline. *Computers & chemical engineering*, v. 28, n. 1-2, p. 171–185.

Rocha, R. *Petroleum Supply Planning: Models, Reformulations and Algorithms*. Tese de doutorado, Departamento de Informática, PUC-Rio, 2010.