

Dimensionamento e sequenciamento de sondas usando a formulação *bucket-indexed*

Luana Carrilho¹

luanamc@tecgraf.puc-rio.br

Tiago Andrade¹

tiago.andrade@tecgraf.puc-rio.b

Gabriela Ribas¹

gabriela.ribas@tecgraf.puc-rio.br

Silvio Hamacher¹

hamacher@puc-rio.br

¹ Pontifícia Universidade Católica do Rio de Janeiro (Puc-Rio)
R. Marquês de São Vicente, 225 – Gávea, Rio de Janeiro – RJ, 22451-900
Departamento de Engenharia Industrial (DEI)

RESUMO

Este trabalho apresenta 2 formulações para o problema de dimensionamento de sondas heterogêneas para perfuração e completação de poços *offshore* com janela de tempo e relações de precedência. As duas formulações são baseadas no problema de sequenciamento de máquinas. A primeira é inspirada na formulação clássica de *time-indexed*, na qual o tempo é discretizado em dias e a segunda é um modelo *bucket-indexed*, uma nova abordagem para problemas de sequenciamento de máquinas. As formulações são testadas usando instâncias da indústria e os resultados sugerem que a formulação *bucket-indexed* é vantajosa comparada à formulação *time-indexed* nos casos onde os tempos de processamento são grandes.

PALAVRAS CHAVE. *Bucket-indexed*, Dimensionamento de Sondas, Sequenciamento de Máquinas.

PO na Área de Petróleo e Gás, Programação Matemática

ABSTRACT

This work presents 2 formulations for the heterogeneous fleet sizing for well drilling and completion with time-window and precedence relations. The two formulations are based on machine scheduling problem. The first one is inspired in a classical time-indexed formulation where time is discretized in days and the second one is a bucket-indexed model, a new approach to machine scheduling problems. The formulations are tested using industry instances and the results suggest that the bucket-indexed formulation is advantageous compared to the time-indexed formulation for the instances where the time-processing is large.

KEYWORDS. *Bucket-indexed*, Rig Sizing, Machine Scheduling.

OR in Oil & Gas, Mathematical Programming

1. Introdução

O elevado investimento nas operações da área de óleo e gás leva empresas do setor a investigarem métodos para melhorar a utilização de seus recursos e diminuir seus custos. Em particular, na área de exploração de poços de petróleo, a diária das sondas usadas para perfuração custam centenas de milhares de dólares [Kaiser e Snyder, 2013; Osmundsen et al., 2010; Markit, 2016].

Irgens et al. [2008] aponta que o planejamento e programação de uma frota de sondas são uma das principais atividades da indústria do setor de petróleo, devido aos altos custos associados a aquisição e operação das sondas, além da complexidade que envolve o sequenciamento de poços.

Por isso, o Problema de Sequenciamento de Sondas (PSS) atraiu a atenção dos pesquisadores desde a década de 60, quando Aronofsky e Williams [1962] abordaram o problema de forma simplificada, desenvolvendo dois modelos lineares para tratar o PSS com o objetivo de maximizar o fluxo de caixa acumulado. Porém, nesta época não havia aplicações práticas do problema, visto que a capacidade computacional era limitada.

Apenas nos anos 90, devido ao avanço computacional, os estudos foram retomados. Neste período alguns artigos foram publicados, dentre eles Eagle [1996] e Iyer et al. [1998] apresentaram métodos para solucionar o PSS multi-período, com o objetivo de maximizar o Valor Presente Líquido. O primeiro utilizou a metaheurística *Simulated Annealing* (SA) e o segundo um modelo de programação linear inteira mista (PLIM) com técnicas de *Branch & Bound*.

Com o aumento do número de publicações nos anos subsequentes, pode-se observar uma tendência de estudos dedicados ao Problema de Roteamento e Sequenciamento de Sondas *Workover* (PRSSW). As tarefas *workover* são essenciais para a manutenção da produção de óleo e gás pelos poços, podendo influenciar diretamente a curva de óleo estimada. Contudo, essas tarefas não possuem relação de precedência, o que simplifica o problema. Majoritadamente, o PRSSW envolve rotear e sequenciar poços terrestres em uma frota de sondas homogêneas com o objetivo de minimizar a perda de produção [Paiva et al., 2000; Gouvêa et al., 2002; Aloise et al., 2002; Costa e Ferreira Filho, 2004; da Costa e Ferreira Filho, 2005; Alves e Ferreira Filho, 2006; Neves e Ochi, 2007; Ribeiro et al., 2011; Pacheco et al., 2012; Ribeiro et al., 2012aa; Ribeiro et al., 2012b; Duhamel et al., 2012]. Outros autores como Aloise et al. [2006] e Ribeiro et al. [2014] apresentaram o mesmo problema usando frota heterogênea. Vale frisar que, apesar de ser um problema simplificado, todos os estudos citados abordaram o PRSSW usando heurísticas e metaheurísticas para a entrega de boas soluções em tempos computacionais razoáveis.

Entretanto poucos estudos têm tratado do Problema de Sequenciamento de Sondas (PSS) para construção de poços, principalmente sondas *offshore*, mesmo sendo as que mais impactam o custo total da operação. Irgens e Lavenue [2007] e Irgens et al. [2008] abordaram o PSS *offshore*, considerando um conjunto de tarefas de perfuração, através de um algoritmo Busca Local Estocástica com o objetivo de maximizar a produção de óleo e minimizar os custos de transporte. O mesmo problema também foi tratado por Gupta e Grossmann [2012] usando um modelo de programação inteira não-linear e não-convexa multiperíodo (PINLNC) para maximizar o Valor Presente Líquido. Outros artigos trabalharam com tarefas de perfuração, completação e *workover* de poços marítimos usando frota de sondas heterogêneas como é o caso de Accioly e Kobayashi [2002], Serra et al. [2012] e Serra et al. [2012b] usaram a técnica *Constrain Programing* para maximizar a produção de óleo.

Com base na literatura, identificamos que, apesar do assunto ter grande importância prática, há uma escassez de publicações e estudos relacionados ao PSS *offshore* para construção de poços. Com o objetivo de preencher essa lacuna, este artigo visa solucionar o Problema do Dimensionamento e Sequenciamento de Sondas (PDSS) *offshore* para tarefas perfuração e completação de poços marítimos. O problema de dimensionamento envolve avaliar a demanda planejada de poços e com base em premissas de rede de precedências e restrições técnicas definir qual a quantidade mínima de sondas para atender a demanda, em que período elas devem ser contratadas e suas especificações técnicas. As sondas dimensionadas neste problema são heterogêneas e podem ser

classificadas em dois tipos: as especializadas, que apresentam características técnicas que permitem executar as tarefas de perfuração e completação, e as não-especializadas, que executam apenas tarefas de completação.

As formulações apresentadas para o PDSS podem ser divididas em duas categorias. A primeira trata o problema como um roteamento de tarefas, no qual o deslocamento das sondas entre os poços é um fator a ser considerado; e a segunda trata o problema como um sequenciamento de máquinas paralelas, onde o deslocamento é desprezível. Para esse estudo, a segunda abordagem é mais adequada, visto que o tempo de deslocamento de sondas *offshore* é ínfimo em relação ao tempo para realizar as tarefas de perfuração e completação dos poços.

Uma das formulações mais usuais para problemas de sequenciamento de máquinas é a formulação *time-indexed*. Nesta formulação, o tempo é discretizado em períodos, por exemplo, em dias ou horas, e a principal variável de decisão indica em que período e máquina uma tarefa tem sua execução iniciada. Se os dados do problema forem inteiros e múltiplos do tamanho da discretização, não há perdas com aproximações e arredondamentos. Não obstante, não se pode afirmar o mesmo quando a discretização agrupa muitos períodos e o dado não é múltiplo da discretização escolhida. Dessarte, se a discretização for muito pequena, o porte do modelo de programação matemática será grande. Por outro lado, caso a discretização seja pouco refinada, haverá problemas de aproximação, que podem retirar soluções viáveis do espaço de busca ou acrescentar soluções que na prática não podem acontecer.

Uma alternativa à formulação *time-indexed* é a formulação *bucket-indexed*, desenvolvida por Boland et al. [2016] para problemas de sequenciamento de máquinas. Esta nova técnica consiste em discretizar o tempo de maneira menos refinada comparado com a que seria necessária no *time-indexed*, sendo esses períodos chamados de *buckets*. Para corrigir o problema de aproximação, as tarefas não precisam mais começar a ser executadas no início de cada período, e sim, em qualquer instante do *bucket*. Mais detalhes sobre ambas as formulações são mostrados na Seção 2. Formulações do tipo *bucket-indexed* surgiram nas aplicações de cronograma e planejamento de produção e dimensionamento de lotes [Pochet e Wolsey, 2006] [Belvaux e Wolsey, 2000] [Belvaux e Wolsey, 2001]. Posteriormente, a ideia por trás de formulações de *bucket* foram adaptadas para o problema de sequenciamento de tarefas em uma única máquina por Boland et al. [2016]. Desde então, este tipo de formulação tem atraído atenção [Clement et al., 2013], [Riedler et al., 2017] por gerar formulações equivalentes à da *time-indexed*, porém com porte reduzido.

Este artigo propõe a elaboração de uma formulação *bucket-indexed* multi-critério para sondas paralelas heterogênea e com janela de tempo e rede de precedência nas tarefas para a solução do PDSS, o que difere das aplicações de *bucket-indexed* encontradas na literatura para sequenciamento de máquina única com funções objetivos clássicas. Esta formulação é comparada com a formulação *time-indexed* usando instâncias da indústria.

2. Modelos Matemáticos

Nesta seção, duas formulações são apresentadas para o PDSS, a formulação *time-indexed* e a formulação *bucket-indexed*. Primeiro, é importante ressaltar a distinção de nomenclatura e notação para tempo discreto e tempo contínuo. Os períodos discretos são definidos como intervalos semiabertos (abertos na direita e fechados na esquerda) de instantes de tempo como mostrado na Figura 1. Instantes de tempo serão referenciados pela letra grega τ e períodos de tempo pela letra t . Será suposto que o dado de entrada é expresso em tempo contínuo.



Figura 1: Notação para o tempo contínuo e discreto

Os modelos usados neste estudo pertencem a classe de sequenciamento de máquinas paralelas heterôgeneas. Os conjuntos, parâmetros usados nesta formulação estão apresentados na Tabela 1. Cada tarefa i corresponde a uma sequência de tarefas que ocorrem no poço. As máquinas representam as sondas s a serem contratadas e são diferenciadas em 2 tipos: especializadas (atendem a qualquer tipo de tarefa) e não-especializadas (atendem somente a tarefas de completação). As tarefas possuem janelas r_i e d_i que representam o primeiro e último instante de tempo que podem ser iniciadas. Cada tarefa tem uma duração p_i . Caso uma tarefa inicie sua execução no instante τ , ela será executada até o instante $\tau + p_i$. Uma rede de precedência deve ser cumprida. Caso um par de tarefas i, j tenha precedência identificada pelo parâmetro $prec_{i,j} = 1$, devem ter seus inícios, τ_i, τ_j espaçados no tempo, ou seja, $\tau_i - \tau_j \geq dif_{i,j}$ caso $prec_{i,j} = 1$, onde O parâmetro $dif_{i,j}$ representa um intervalo mínimo e pode assumir qualquer valor inteiro maior que p_i .

Tabela 1: Conjuntos e Parâmetros da formulação TI

Conjuntos	
Sondas	s : sonda a ser contratada
Tarefas	i, j : tarefa do poço a ser executada
Períodos	t, a : período contínuo discretizado em dias
Parâmetros	
p_i	Duração da tarefa i (dias)
r_i	Primeiro período em que a tarefa i pode ser iniciada
d_i	Último período em que a tarefa i pode ser iniciada
$prec_{i,j}$	1, tarefa i precede tarefa j ; 0, caso contrário
$dif_{i,j}$	Distância entre as tarefas i e j (dias)
C	Coefficiente suficientemente grande da função objetivo para representar relação lexicográfica entre objetivos

O problema é bi-objetivo lexicográfico. O primeiro critério é minimizar o número total de sondas e o segundo critério, o número de sondas especializadas. Os dois objetivos foram modelados usando apenas uma função objetivo com coeficientes calculados para obter este comportamento hierárquico entre os termos.

2.1. Time-indexed

Nesta seção, o modelo *time-indexed* (TI) será apresentado nas equações (1)-(14) e as suas variáveis na Tabela 2.

Tabela 2: Variáveis da formulação TI

Variáveis	
$x_{s,i,t}$	1 caso a tarefa i está sendo executada na sonda s e se iniciou no período t
$inicio_i$	indica qual período a tarefa i foi executada; 0, caso contrário
usa_s	1 caso a sonda s esteja sendo usada; 0, caso contrário
$usaEsp_s$	1 caso a sonda especializada s esteja sendo usada; 0, caso contrário

$$\min \sum_s (usaEsp_s + C * usa_s) \quad (1)$$

s.a.:

$$\sum_{s,t} x_{s,i,t} = 1 \quad \forall i \quad (2)$$

$$usa_s \geq \sum_t x_{s,i,t} \quad \forall i, s \quad (3)$$

$$usaEsp_s \geq \sum_t x_{s,i,t} \quad \forall i \in P, s \quad (4)$$

$$usa_s \leq usa_{s-1} \quad \forall s > 1 \quad (5)$$

$$usaEsp_s \leq usaEsp_{s-1} \quad \forall s > 1 \quad (6)$$

$$\sum_{t-p_i+1 \leq a \leq t, i} x_{s,i,a} \leq 1 \quad \forall s, t \quad (7)$$

$$inicio_i = \sum_{s,t} t * x_{s,i,t} \quad \forall i \quad (8)$$

$$inicio_i - inicio_j \geq dif_{i,j} \quad \forall i, j | prec_{i,j} = 1 \quad (9)$$

$$x_{s,i,t} = 0 \quad \forall s, i, t | t < r_i + 1 \vee t > d_i + 1 \quad (10)$$

$$x_{s,i,t} \in \{0, 1\} \quad \forall s, i, t \quad (11)$$

$$inicio_i \in \mathbb{Z}_+ \quad (12)$$

$$usa_s \in \{0, 1\} \quad \forall s \quad (13)$$

$$usaEsp_s \in \{0, 1\} \quad \forall s \quad (14)$$

A variável $x_{s,i,t} = 1$ indica que a tarefa i está sendo executada na sonda s e se iniciou no período t , $inicio_i$ é uma variável auxiliar que mostra em que instante τ_i a tarefa i foi executada e as variáveis usa_s e $usaEsp_s$ valem 1 caso a sonda s esteja sendo usada. O objetivo de minimizar o número total de sondas e o número de sondas especializadas representado em (1). A restrição (2) limita que cada tarefa i só pode ser executada exatamente uma vez. As restrições (3) e (4) limitam o uso da sonda s apenas para o caso em que forem contratadas, isto é, pelo menos uma tarefa i sendo executada na sonda s . As restrições (5) e (6) são de quebra de simetria e eliminam soluções equivalentes. A restrição (7) implica que cada sonda só atenda no máximo uma tarefa por vez. A restrição (8) define a variável auxiliar $inicio_i$, que pode ser eliminada sem perda de consistência. A restrição (9) define a rede de precedência usando inícios relativos. A restrição (10) limita as tarefas a serem atendidas dentro da janela de tempo designada. Essa restrição pode ser implementada limitando a geração das variáveis. As restrições (11)-(14) representam os domínios das variáveis.

2.2. Bucket-indexed

A ideia chave do *bucket-indexed* (BI) é agregar períodos de tempo em conjuntos chamados de *buckets* com Δ períodos. Esta propriedade é alçada pois variáveis contínuas no intervalo $[0, 1]$ são adicionadas ao modelo, permitindo o controle de quando a tarefa vai começar a ser executada dentro do *bucket*, não sendo necessário que ocorra no início do período como ocorre no TI. Algumas hipóteses adicionais são necessárias: i) Apenas uma tarefa pode iniciar em cada *bucket*; ii) Apenas duas tarefas podem ser executadas no mesmo *bucket*, porém não no mesmo instante de tempo; iii) O tamanho do *bucket* Δ deve ser pelo menos igual a menor duração das tarefas, isto é $\Delta \geq \min_i p_i$.

Novos conjuntos, parâmetros e variáveis, descritos na Tabela 3, são incorporados a formulação do BI proposta por Boland et al. [2016]. Para cada tarefa i , π_i representa a fração não ocupada do último *bucket* que a tarefa ocupar caso ela tivesse começado a ser executada no início do primeiro *bucket* que a mesma ocupa. A duração em *buckets* é representada pelo parâmetro $P_i = \lfloor p_i / \Delta \rfloor + 1$. Uma tarefa que comece a ser executada no *bucket* b , será executada até o *bucket* $b + P_i + k$, onde k é um elemento do conjunto binário $K = \{0, 1\}$. A tarefa não ocupa um número fixo de *buckets* dado que este valor depende de quando a mesma começou a ser executada no *bucket* b . Se a tarefa i começar a ser executada entre o início do *bucket* b até a fração $\pi_i = P_i - p_i / \Delta$ do *bucket* b , então ela terminará de ser executada no *bucket* $b + P_i - 1$ e ocupará P_i

Tabela 3: Conjuntos, parâmetros e variáveis da formulação BI

Conjuntos	
Buckets	b, h : bucket de tamanho Δ
K	k : 0,1
Parâmetros	
Δ	Quantidade de períodos em um <i>bucket</i>
π_i	Fração que não seria ocupada do último <i>bucket</i> que a tarefa i ocupar, caso ela seja executada no início do primeiro <i>bucket</i> que ela ocupa
P_i	Duração da tarefa i (<i>bucket</i>)
R_i	Primeiro <i>bucket</i> em que a tarefa i pode ser iniciada
D_i	Último <i>bucket</i> em que a tarefa i pode ser iniciada
ρ_i	Fração máxima do <i>bucket</i> que pode ser usada processando o início da tarefa i
δ_i	Fração mínima que do <i>bucket</i> que pode ser usada processando o início da tarefa i
Variáveis	
$z_{s,i,b,k}$	1 caso a tarefa i está sendo executada na sonda s e se iniciou no <i>bucket</i> b e se a tarefa ocupar $P_i + k$ <i>buckets</i>
$inicio_i$	indica qual período a tarefa i foi executada
$u_{s,i,b,k}$	Fração ocupada no primeiro <i>bucket</i> b ocupado pela tarefa i se ela tiver ocupado $P_i + k$ <i>buckets</i> e tenha sido executada na sonda s
$v_{s,i,b,k}$	Fração ocupada no último <i>bucket</i> b ocupado pela tarefa i se ela tiver ocupado $P_i + k$ <i>buckets</i> e tenha sido executada na sonda s

buckets, caso comece a sua execução após a fração π_i , terminará em $b + P_i$ e ocupará $P_i + 1$ *buckets*. Os parâmetros de janelas também são expressas em *buckets*, onde $R_i = \lfloor r_i/\Delta \rfloor + 1$ representa o *bucket* mais cedo em que a tarefa i pode iniciar, $\rho_i = R_i - r_i/\Delta$ indica a fração máxima do *bucket* que pode ser usada processando o início da tarefa i . Analogamente, $D_i = \lfloor d_i/\Delta \rfloor + 1$ representa o *bucket* mais tarde que a tarefa i pode iniciar, $\delta_i = D_i - d_i/\Delta$ indica a fração mínima do *bucket* que pode ser usada processando o início da tarefa i . Novas variáveis são introduzidas ao modelo, $z_{s,i,b,k}$ vale 1 caso a tarefa i comece a ser executada no *bucket* b e ocupe $P_i + k$ *buckets*. A variável $u_{s,i,b,k}$ representa a fração ocupada pela tarefa i do primeiro *bucket* b ocupado e que ocupe $P_i + k + 1$ *buckets*. Da mesma forma, $v_{s,i,b,k}$ representa a fração ocupada pela tarefa i do último *bucket* b ocupado e que ocupe $P_i + k + 1$ *buckets*. Essas novas notações estão e ilustradas na Figura 2. O modelo BI é está representado nas equações (15)-(37).

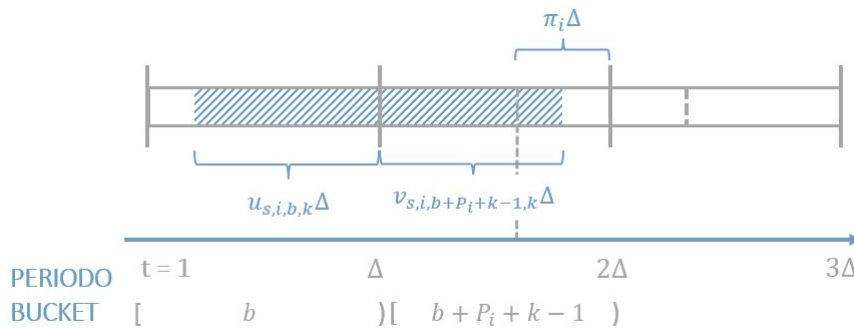


Figura 2: Exemplo da notação BI

$$\min \sum_s (usaEsp_s + C * usa_s) \quad (15)$$

s.a.:

$$\sum_{s,b,k} z_{s,i,b,k} = 1 \quad \forall i \quad (16)$$

$$\sum_{i,k,b-P_i+2 \leq h \leq b} z_{s,i,h,k} \leq 1 \quad \forall s, b \quad (17)$$

$$\sum_{i,k} (u_{s,i,b,k} + v_{s,i,b,k}) + \sum_{i,k,b-P_i+2 \leq h \leq b-1} z_{s,i,h,k} \leq 1 \quad \forall s, b \quad (18)$$

$$usa_s \geq \sum_{b,k} z_{s,i,b,k} \quad \forall i, s \quad (19)$$

$$usaEsp_s \geq \sum_{b,k} z_{s,i,b,k} \quad \forall i \in P, s \quad (20)$$

$$usa_s \leq usa_{s-1} \quad \forall s > 1 \quad (21)$$

$$usaEsp_s \leq usaEsp_{s-1} \quad \forall s > 1 \quad (22)$$

$$u_{s,i,b,k} + v_{s,i,b+P_i+k-1,k} = (2 - k - \pi) z_{s,i,b,k} \quad \forall s, i, b, k \quad (23)$$

$$((1 - k)(1 - \pi_i) + 1/\Delta) z_{s,i,b,k} \leq u_{s,i,b,k} \quad \forall s, i, b, k \quad (24)$$

$$u_{s,i,b,k} \leq (1 - k\pi_i) z_{s,i,b,k} \quad \forall s, i, b, k \quad (25)$$

$$inicio_i = \left(\sum_{s,b,k} b * z_{s,i,b,k} - u_{s,i,b,k} \right) \Delta \quad \forall i \quad (26)$$

$$inicio_i - inicio_j \geq dif_{i,j} \quad \forall i, j | prec_{i,j} = 1 \quad (27)$$

$$u_{s,i,R_i,0} \leq \rho_i z_{s,i,R_i,0} \quad \forall s, i | 1 - \pi_i < \rho_i < 1 \quad (28)$$

$$u_{s,i,R_i,1} \leq \rho_i z_{s,i,R_i,1} \quad \forall s, i | \rho_i < 1 - \pi_i \quad (29)$$

$$\delta_i z_{s,i,D_i,0} \leq u_{s,i,D_i,0} \quad \forall s, i | 1 - \pi_i < \delta_i < 1 \quad (30)$$

$$\delta_i z_{s,i,D_i,1} \leq u_{s,i,D_i,1} \quad \forall s, i | \delta_i < 1 - \pi_i \quad (31)$$

$$z_{s,i,R_i,0} = 0 \quad \forall s, i | \rho_i < 1 - \pi_i \quad (32)$$

$$z_{s,i,D_i,1} = 0 \quad \forall s, i | 1 - \pi_i < \delta_i < 1 \quad (33)$$

$$z_{s,i,b,k} \in \{0, 1\} \quad \forall s, i, b, k \quad (34)$$

$$u_{s,i,b,k} \in [0, 1] \quad \forall s, i, b, k \quad (35)$$

$$v_{s,i,b,k} \in [0, 1] \quad \forall s, i, b, k \quad (36)$$

$$inicio_i \in \mathbb{Z}_+ \quad (37)$$

As restrições (17) e (18) limitam que uma máquina ou sonda execute mais que uma tarefa no mesmo instante de tempo. Foram necessárias 2 restrições ao invés de apenas 1 como nos modelos TI, pois uma tarefa pode ocupar parcialmente um *bucket* nesta formulação, em contraposição ao comportamento de ocupar ou não ocupar um determinado período.

As frações usadas no último e no primeiro *bucket* são relacionadas usando o parâmetro π_i e o número de *buckets* usados pela restrição (23). Esta restrição pode ser usada para eliminar uma das variáveis u ou v através da substituição nas outras restrições de uma das mesmas pela sua definição dada por isolar elas nessa restrição.

O número de *buckets* usados pela tarefa i limita o valor que a variável u pode assumir inferiormente e superiormente como estipulado nas restrições (24) e (25). De forma análoga, limites poderiam ser impostos à variável v , mas estes seriam redundantes e podem ser obtidos através das restrições (23)-(25).

As restrições (28) a (33) representam as janelas de tempo a serem atendidas, equivalente a restrição (10) do modelo TI. As restrições (28) e (29) representam um limite superior para variável u , enquanto as restrições (30) e (31) são um limite inferior.

As outras restrições são iguais às restrições da formulação TI, e suas explicações serão omitidas. Um ponto de atenção da formulação BI é que caso $\Delta = 1$, gerará o dobro de variáveis binárias que a

formulação TI, além das variáveis contínuas u e v adicionais. Caso $\Delta = 2$, o número de variáveis binárias é o mesmo. Este resultado sugere que para a formulação BI ser vantajosa, Δ deve ser maior que 2. Quanto maior o valor do Δ , menor será o porte. Por outro lado, quanto maior o Δ , pior será a relaxação linear. Além disso, o valor de Δ é limitado superiormente pelo menor tempo de duração $\min_i p_i$.

3. Experimentos Computacionais

Nesta seção, serão apresentados os resultados e análises das soluções obtidas pelas duas formulações apresentadas. Analisamos três instâncias da indústria de óleo e gás que são compostas por um conjunto de tarefas dos tipos perfuração e completação, as janelas de execução, a rede de precedências e a distância em dias que as tarefas devem estar separadas entre si. As durações das tarefas são longas, devido ao grau de complexidade que envolve a construção de poços marítimos do pré-sal e as janelas de execução dessas tarefas são bem restritas, impedindo a antecipação de capital investido e o atraso na produção de óleo. O número máximo de sondas que podem ser usadas é definido pela interseção das janelas de tempo das tarefas. A quantidade de tarefas, o número de períodos em dias e *buckets* e o valor assumido por $\Delta = \min_i p_i$ para cada instância estão expostos na Tabela 4.

Tabela 4: Instâncias				
Instância	# Tarefas	# Períodos	Δ	# Buckets
1	18	2191	73	31
2	42	1484	46	33
3	44	1792	56	33

Os experimentos computacionais foram executados no computador com Intel® Core™ i5-4440 CPU 3.10 GHz com 16.00 GB RAM de memória usando o *solver* Cplex 12.6 solver e o *software* Aimms® 3.14. Para fins de comparação entre as formulações TI e BI limitamos o tempo de execução dos modelos em uma hora (3600 s). Os resultados estão apresentados na Tabela 5 onde, para cada instância, o número de variáveis e variáveis inteiras, o número de restrições, o tempo de execução em segundos e o *gap* (%) para cada um dos modelos.

Tabela 5: Resultados								
Instância	TI				BI			
	# Variáveis	# Restrições	Tempo (s)	Gap (%)	# Variáveis	# Restrições	Tempo (s)	Gap (%)
1	29359 (29358)	15065	194,97	0	6868 (2322)	7820	10,30	0
2	136915 (136914)	23790	3600	58,00	23029 (7728)	25968	235,16	0
3	231059 (231058)	46745	3600	85,27	47953 (16052)	52806	3600	11,27

Como esperado, depois de executar os dois modelos concluímos que o modelo BI leva menos tempo do que o TI. Este comportamento acontece pois para todas as instâncias usadas $\Delta > 2$ e com isso este modelo gera um número de variáveis menor, especialmente de variáveis inteiras, do que o modelo TI correspondente. Em contrapartida há um aumento no número de restrições no modelo BI, mas que de forma alguma impactam no bom desempenho.

Na primeira instância ambos modelos chegam ao ótimo rapidamente. O modelo TI demora aproximadamente 3 minutos para encontrar a solução enquanto BI leva alguns segundos. Já na instância 3, nenhum dos modelos alcança a otimalidade em uma hora, porém o modelo BI chega muito próximo com um *gap* de 11,27%. O resultado da instância 2 é o que chama mais atenção, visto que o modelo TI não consegue chegar nem a 50% do ótimo em 3600 segundos diferente do modelo BI que atinge a solução ótima em menos de 4 minutos.

4. Conclusões

A exploração de poços de petróleo é uma atividade que demanda um alto investimento das empresas do setor, principalmente na aquisição de recursos necessários para realizarem as operações de perfuração e completação. Por isso, o bom dimensionamento de sondas e uma programação otimizada é essencial para esse problema. A maioria dos trabalhos encontrados na literatura, tratam do problema de roteamento

e programação de sondas *workover* em poços terrestres usando heurísticas e metaheurística, mas ainda há uma escassez de trabalhos que tratam do problema de sequenciamento de sondas *offshore* para tarefas de perfuração e completação que mantém uma relação de precedência e, principalmente, trabalhos que abordam o problema de dimensionamento de recursos a serem contratados. Portanto, este trabalho teve como objetivo comparar as formulações *time-indexed* e *bucket-indexed* para o Problema de Dimensionamento e Sequenciamento de Sondas (PDSS) heterogêneas e *offshore*.

Foram analisadas três instâncias reais da indústria de óleo e gás e dos resultados obtidos todas observamos que o modelo BI alcançou resultados mais rápidos do que o modelo TI e chegando no ótimo (ou muito próximo dele) em situações que em uma hora o modelo TI ainda está com um *gap* muito alto. O comportamento era esperado, pois a nova formulação gera menos variáveis inteiras quando $\Delta > 2$, condição que foi respeitada em todas as instâncias.

Por fim, para trabalhos futuros aplicaremos essa a formulação *bucket-indexed* em modelos mais complexos que representam melhor casos da realidade, como sondas com janela de tempo ou novas categorias de sondas e outras funções objetivos que envolvam métricas de orçamento. Além disso, é necessário um estudo sobre o efeito do tamanho do *bucket* (Δ) para avaliar como se comporta o modelo em relação ao tempo computacional, qualidade da solução e da relaxação linear.

Referências

- Accioly, F. J. M., R.; Marcellino e Kobayashi, H. (2002). Uma aplicação da programação por restrições no escalonamento de atividades em poços de petróleo.
- Aloise, D., Noronha, T., Maia, R., Bittencourt, V. G., e Aloise, D. J. (2002). Heurísticas de colônia de formigas com path-relinking para o problema de otimização da alocação de sondas de produção terrestre–spt.
- Aloise, D. J., Aloise, D., Rocha, C. T., Ribeiro, C. C., Ribeiro Filho, J. C., e Moura, L. S. (2006). Scheduling workover rigs for onshore oil production. *Discrete Applied Mathematics*, 154(5): 695–702.
- Alves, V. e Ferreira Filho, V. J. M. (2006). Proposta de algoritmo genético para a solução do problema de roteamento e sequenciamento de sondas de manutenção. volume 1848.
- Aronofsky, J. e Williams, A. (1962). The use of linear programming and mathematical models in under-ground oil production. *Management Science*, 8(4):394–407.
- Belvaux, G. e Wolsey, L. A. (2000). bc—prod: A specialized branch-and-cut system for lot-sizing problems. *Management Science*, 46(5):724–738.
- Belvaux, G. e Wolsey, L. A. (2001). Modelling practical lot-sizing problems as mixed-integer programs. *Management Science*, 47(7):993–1007.
- Boland, N., Clement, R., e Waterer, H. (2016). A bucket indexed formulation for nonpreemptive single machine scheduling problems. *INFORMS Journal on Computing*, 28(1):14–30.
- Clement, R., Boland, N., e Waterer, H. (2013). A variable sized bucket indexed formulation for nonpreemptive single machine scheduling problems.
- Costa, L. e Ferreira Filho, V. J. M. (2004). Uma heurística para o problema do planejamento de itinerários de sondas em intervenções de poços de petróleo. *Anais do XXXVI Simpósio Brasileiro de Pesquisa Operacional*, 1853.
- da Costa, L. R. e Ferreira Filho, V. J. M. (2005). Uma heurística de montagem dinâmica para o problema de otimização de itinerários de sondas.
- Duhamel, C., Santos, A. C., e Guedes, L. M. (2012). Models and hybrid methods for the onshore wells maintenance problem. *Computers & Operations Research*, 39(12):2944–2953.

- Eagle, K. (1996). Using simulated annealing to schedule oil field drilling rigs. *Interfaces*, 26(6): 35–43.
- Gouvêa, E. F., Goldberg, M. C., e Costa, W. E. (2002). Algoritmos evolucionários na solução do problema de otimização do emprego de sondas de produção em poços de petróleo.
- Gupta, V. e Grossmann, I. E. (2012). An efficient multiperiod minlp model for optimal planning of offshore oil and gas field infrastructure. *Industrial & Engineering Chemistry Research*, 51(19): 6823–6840.
- Irgens, M. e Lavenue, W. (2007). Use of advanced optimization techniques to manage a complex drilling schedule. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.
- Irgens, M., Guzman, R. P., Stamatopoulos, J. G., Jackson, K., et al. (2008). Optimization for operational decision support: The rig management case. In *SPE Annual Technical Conference and Exhibition*. Society of Petroleum Engineers.
- Iyer, R., Grossmann, I., Vasantharajan, S., e Cullick, A. (1998). Optimal planning and scheduling of offshore oil field infrastructure investment and operations. *Industrial & Engineering Chemistry Research*, 37(4):1380–1397.
- Kaiser, M. J. e Snyder, B. F. (2013). Modeling offshore wind installation costs on the us outer continental shelf. *Renewable Energy*, 50:676–691.
- Markit, I. (2016). IHS petrodata offshore rig day rate trends. <https://www.ihs.com/products/oil-gas-drilling-rigs-offshore-day-rates>. Acessado em 22 de março de 2016.
- Neves, T. A. e Ochi, L. S. (2007). Grasp com memoria adaptativa aplicado ao problema de roteamento e scheduling de sondas de manutenção. *Anais da XXVII do Congresso da Sociedade Brasileira de Computação*, p. 1–10.
- Osmundsen, P., Roll, K. H., e Tveterås, R. (2010). Exploration drilling productivity at the norwegian shelf. *Journal of Petroleum Science and Engineering*, 73(1-2):122–128.
- Pacheco, A. V. F., Ribeiro, G. M., e Mauri, G. R. (2012). *A GRASP with path-relinking for the workover rig scheduling problem*. IGI Global.
- Paiva, R., Schiozer, D., e Bordalo, S. (2000). Optimizing the itinerary of workover rigs. In *16th World Petroleum Congress*. World Petroleum Congress.
- Pochet, Y. e Wolsey, L. A. (2006). *Production planning by mixed integer programming*. Springer Science & Business Media.
- Ribeiro, G. M., Desaulniers, G., e Desrosiers, J. (2012aa). A branch-price-and-cut algorithm for the workover rig routing problem. *Computers & Operations Research*, 39(12):3305–3315.
- Ribeiro, G. M., Desaulniers, G., Desrosiers, J., Vidal, T., e Vieira, B. S. (2014). Efficient heuristics for the workover rig routing problem with a heterogeneous fleet and a finite horizon. *Journal of Heuristics*, 20(6):677–708.
- Ribeiro, G. M., Laporte, G., e Mauri, G. R. (2012b). A comparison of three metaheuristics for the workover rig routing problem. *European Journal of Operational Research*, 220(1):28–36.

- Ribeiro, G. M., Mauri, G. R., e Lorena, L. A. N. (2011). A simple and robust simulated annealing algorithm for scheduling workover rigs on onshore oil fields. *Computers & Industrial Engineering*, 60(4):519–526.
- Riedler, M., Jatschka, T., Maschler, J., e Raidl, G. R. (2017). An iterative time-bucket refinement algorithm for a high-resolution resource-constrained project scheduling problem. *International Transactions in Operational Research*.
- Serra, T., Nishioka, G., e Marcellino, F. J. (2012). Integrated project selection and resource scheduling of offshore oil well developments: An evaluation of cp models and heuristic assumptions.
- Serra, T., Nishioka, G., e Marcellino, F. J. (2012b). *The offshore resources scheduling problem: detailing a constraint programming approach*.