UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE
DEPARTAMENTO DE COMPUTAÇÃO

HARRISON WILSON LOPES DE OLIVEIRA

CLUSTERING SEARCH PARA O PROBLEMA DE ESCALONAMENTO DE SONDAS DE MANUTENÇÃO EM POÇOS DE PETRÓLEO

ALEGRE - ES
2020

HARRISON WILSON LOPES DE OLIVEIRA

CLUSTERING SEARCH PARA O PROBLEMA DE ESCALONAMENTO DE SONDAS DE MANUTENÇÃO EM POÇOS DE PETRÓLEO

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Geraldo Regis Mauri.

ALEGRE - ES

2020

HARRISON WILSON LOPES DE OLIVEIRA

CLUSTERING SEARCH PARA O PROBLEMA DE ESCALONAMENTO DE SONDAS DE MANUTENÇÃO EM POÇOS DE PETRÓLEO

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em ____ de dezembro de 2020.

BANCADA EXAMINADORA

_____

Prof. Dr. Geraldo Regis Mauri
Universidade Federal do Espírito Santo

_____

Prof. Dr. Dayan de Castro Bissoli
Universidade Federal do Espírito Santo

_____

Prof. Dr. Antonio Almeida de Barros Junior
Universidade Federal do Espírito Santo

“Perseverança é o trabalho duro que você faz depois de ter se cansado de fazer o trabalho duro que você já fez”.

Newt Gingrich

AGRADECIMENTOS

Agradeço primeiramente a minha família, a qual me forneceu todo amor, incentivo e suporte para que eu conseguisse chegar até aqui.

À Lavínia, por todo amor, companheirismo, incentivo e pela força nos momentos difíceis.

Ao meu orientador Geraldo, pelo direcionamento, esclarecimento das dúvidas e pelo apoio no decorrer deste trabalho.

Aos meus amigos, por toda força e suporte necessário, especialmente ao meu amigo Kevin que esteve comigo durante toda a graduação.

Aos colegas e amigos de graduação, principalmente àqueles dos cursos de Sistemas de Informação e Ciência da Computação, que me acompanharam nessa jornada.

À Universidade Federal do Espírito Santo, pela oportunidade de crescimento pessoal e profissional.

A todos, o meu muito obrigado.

RESUMO

O processo de extração do petróleo se dá por meio da utilização de um equipamento que bombeia o petróleo presente nos poços para a superfície. Porém, esse equipamento pode apresentar falhas ao longo do tempo, fazendo com que seja necessária a intervenção nestes poços para realização de serviços de manutenção. Essas intervenções são realizadas por meio de equipamentos caros e limitados chamados sondas de manutenção. Entretanto, o número de sondas disponíveis é pequeno comparado ao de poços requerendo manutenção. Dessa forma, faz-se necessária a criação de um cronograma eficiente de atendimento a esses poços. A criação desse cronograma de atendimento a partir da associação entre sondas e poços é conhecida como Problema de Sondas de Manutenção (PSM). Neste trabalho, foi utilizada a meta-heurística CS como alternativa para resolução de uma das duas principais abordagens do PSM encontradas na literatura, o Problema de Escalonamento de Sondas de Manutenção (PESM). O PESM consiste em escalonar o atendimento dos poços com o objetivo minimizar a perda total de produção causada por poços inativos. Os testes computacionais foram executados em instâncias baseadas em casos reais do sudeste do Brasil. Os resultados computacionais provam a eficiência do método proposto, que foi capaz de encontrar novas melhores soluções quando comparado a outros trabalhos da literatura que também utilizaram meta-heurísticas para resolução do PESM.

Palavras-chave: Sondas de Manutenção; *Clustering Search; Oil production*.

ABSTRACT

The oil extracting process occurs through the utilization of equipment that pumps the oil present in the wells to the surface. However, this equipment may fail during the process, making intervention in these wells necessary for maintenance services. These interventions are performed using expensive and limited equipment called workover rigs. Nonetheless, few workover rigs are available when compared to the number of wells that requires maintenance. Therefore, it is necessary to create an efficient service schedule for these wells. The creation of this schedule based on the association between rigs and wells is known as the Workover Rigs Problem (WRP). In this paper, the Clustering Search metaheuristic was used as an alternative to solve one of the two main approaches to WRP found in the literature, the Workover Rigs Scheduling Problem (WRSP). The WRSP consists of scheduling the service of the rigs to the wells in order to minimize the total loss of production caused by inactive wells. Computational tests were performed in instances based on real cases in southeastern Brazil. The computational results prove the efficiency of the proposed method, which was able to find new better solutions when compared to other studies that also used metaheuristics to solve the WRSP.

Keywords: Workover Rigs; Clustering Search; Oil production.

LISTA DE FIGURAS

Figura 1 - Equipamento realizando um serviço de manutenção.....12
Figura 2 - Principais abordagens do PSM existentes na literatura.....13
Figura 3 – Fluxograma do CS.....25
Figura 4 – Pseudocódigo da heurística construtora.....26
Figura 5 – Movimento de realocação.....27
Figura 6 – Movimento de permutação entre dois poços.....28
Figura 7 – Representação da distância de *Hamming* entre duas soluções s_1 e s_2.....29
Figura 8 – Pseudocódigo do CS utilizado.....32

LISTA DE TABELAS

Tabela 1 – Resultados da busca na plataforma *Science Direct* utilizando as palavras-chaves em inglês.....16
Tabela 2 – Resultados da busca na plataforma *Science Direct* utilizando as palavras-chaves em português.....17
Tabela 3 – Resultados da busca na plataforma Google Scholar utilizando as palavras-chaves em inglês.....18
Tabela 4 – Resultados da busca na plataforma *Google Scholar* utilizando as palavras-chaves em português.....18
Tabela 5 – Intervalo de valores possíveis para cada parâmetro durante a calibração.....34
Tabela 6 – Valores dos parâmetros do SA e do CS após a calibração.....34
Tabela 7 – Resumo dos resultados computacionais.....35
Tabela 8 – Comparação dos resultados com outras abordagens heurísticas da literatura.....37
Tabela 9 – Comparação dos tempos computacionais com outras abordagens heurísticas da literatura.....38
Tabela 10 – Comparação dos resultados com o método exato.....39

SUMÁRIO

1. INTRODUÇÃO.....11
1.1. O PROBLEMA E SUA IMPORTÂNCIA.....12
1.2. OBJETIVOS.....14
1.2.1. Objetivo Geral.....14
1.2.2. Objetivos Específicos.....14
1.3. ESTRUTURA DO TRABALHO.....14
2. REVISÃO DA LITERATURA.....16
2.1. TRABALHOS CORRELATOS AO PESM.....19
2.2. ESTADO DA ARTE DO PRSM.....22
2.3. CONSIDERAÇÕES.....22
3. METODOLOGIA.....24
3.1. PRINCIPAIS COMPONENTES DO CS.....24
3.1.1 Meta-heurística geradora de soluções.....25
3.1.1.1 Solução inicial.....26
3.1.1.2 Geração de vizinhança.....27
3.1.2 Processo de agrupamento iterativo.....28
3.1.4 Heurística de busca local.....29
3.1.4 Pseudocódigo do CS.....30
4. TESTES COMPUTACIONAIS.....33
4.1. CALIBRAÇÃO DOS PARÂMETROS.....33
5. RESULTADOS OBTIDOS.....34
5.1. RESULTADOS COMPUTACIONAIS.....35
5.2. COMPARAÇÃO COM A LITERATURA.....37
6. CONCLUSÕES.....41
REFERÊNCIAS BIBLIOGRÁFICAS.....43

1. INTRODUÇÃO

O petróleo movimenta atualmente milhões de dólares diariamente e constitui a base da economia mundial. Além de ser um recurso vital para setores da economia como o transporte e a produção de eletricidade, por ser a maior fonte de energia da nossa época, é também a base de centenas produtos industrializados como tecidos, plásticos, produtos farmacêuticos, cosméticos, entre outros. Dessa forma, o país que detém esse recurso natural consegue ter grande participação no comércio internacional por meio da sua exportação e de seus produtos derivados (BRASIL, 2011).

O Brasil vive em constante crescimento no setor petrolífero e, dados do boletim mensal da Agência Nacional de Petróleo, Gás Natural e Biocombustíveis (ANP, 2020) apontam que a produção de petróleo diária no país é de 3,16 milhões de barris. Além disso, de acordo com o Ministério de Minas e Energia (2019), o petróleo e seus derivados representaram, no ano de 2018, 34,4% da Oferta Interna de Energia (OIE), que é a energia necessária para movimentar a economia.

De acordo com Bissoli (2014), o mercado petrolífero é muito competitivo e pelo fato do petróleo ser um recurso não-renovável, a sua exploração deve ser feita de maneira eficiente, para que as jazidas sejam aproveitadas ao máximo. Entretanto, um dos grandes desafios enfrentados pelas empresas responsáveis pela extração do petróleo é encontrar uma maneira de reduzir os custos relacionados a sua exploração.

Neste trabalho, é proposta a utilização da meta-heurística *Clustering Search* (CS) para resolução de um problema comum encontrado na exploração de petróleo, conhecido como Problema de Escalonamento de Sondas de Manutenção – PESM, descrito a seguir. A escolha do CS para resolução do PESM se baseou nos seus bons resultados para outros problemas de otimização encontrados na literatura, como Problema *Job Shop* Flexível (ALTOÉ et al., 2018), Problema de Gerenciamento Florestal (MELO; BARROS JUNIOR; MAURI, 2015) e Problema de Rotação de Culturas (ARAÚJO; MAURI, 2013), além do fato de o mesmo ainda não ter sido aplicado para resolução do PESM.

1.1. O PROBLEMA E SUA IMPORTÂNCIA

A exploração do petróleo é uma atividade de alto custo realizada por meio da utilização de um equipamento que bombeia o petróleo presente nos poços para a superfície. Porém, existem alguns desafios a serem considerados, já que esse equipamento pode apresentar falhas ao longo do tempo, fazendo com que seja necessária a intervenção nesses poços. Essas intervenções são realizadas por meio de equipamentos chamados de “sondas de manutenção” ou “sondas de intervenção” (*workover rigs*), que realizam os serviços de: limpeza, reintegração ou substituição de equipamentos de produção, estimulação de reservatório, e outros (BISSOLI; CHAVES; RIBEIRO, 2016; MONEMI et al., 2018).

Figura 1 - Equipamento realizando um serviço de manutenção.
Fonte: Aloise et al. (2006).

Esses equipamentos ilustrados na Figura 1 são limitados e tem um custo de aluguel elevado. Dessa forma, é inviável que uma empresa petrolífera mantenha um número grande desses equipamentos. Ou seja, em um cenário real existem poucas sondas disponíveis em comparação a quantidade de poços que demandam intervenção. Portanto, é necessária a criação de escalas de atendimento aos poços para evitar perdas de produção causadas por poços inativos. Desse modo, a associação entre

poços e sondas deve ser planejada de maneira minuciosa, com o objetivo de estabelecer um cronograma eficiente (ALOISE et al., 2006; PÉREZ, 2017).

A criação desse cronograma de atendimento a partir da associação entre sondas e poços é conhecida como Problema de Sondas de Manutenção – PSM (GOUVÊA; GOLDBARG; COSTA, 2002).

Segundo Bissoli, Chaves e Ribeiro (2016), existem duas abordagens principais na literatura relacionadas ao PSM: o escalonamento, abordagem tratada nesse trabalho, e o roteamento. Na abordagem de escalonamento é considerada a utilização de frota de sondas homogênea e a distância entre os poços é desconsiderada. Todos os poços devem ser atendidos dentro de um horizonte de processamento, onde cada poço tem um prazo para receber atendimento. Em relação à abordagem de roteamento, as frotas de sondas podem ser homogêneas ou heterogêneas e a distância entre poços é considerada. Além disso, nesta abordagem não é obrigatório o atendimento de todos os poços até o fim do horizonte de processamento. Caso todos os poços não sejam atendidos, os poços remanescentes são considerados na próxima sessão de planejamento. A Figura 2 resume a classificação encontrada na literatura para o PSM.

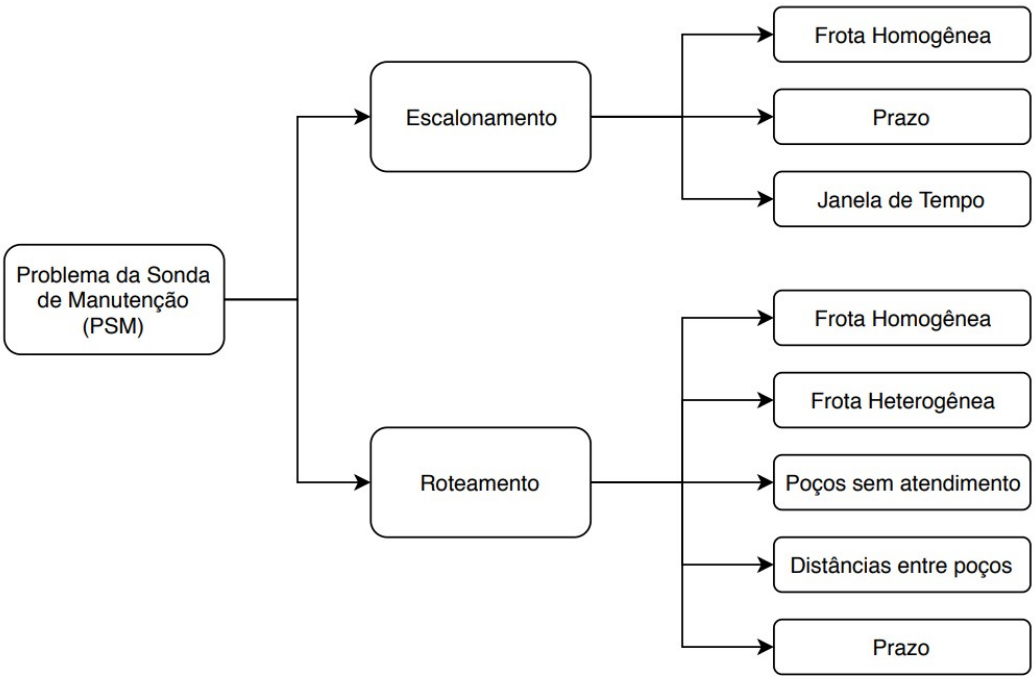

Figura 2 - Principais abordagens do PSM existentes na literatura.

Fonte: Bissoli (2014).

A partir de um conjunto de poços que requerem manutenção e uma frota de sondas homogênea, o Problema de Escalonamento de Sondas de Manutenção – PESM consiste em programar o atendimento a esses poços com o objetivo minimizar a perda total de produção causada por poços inativos, garantindo que todos os poços sejam atendidos dentro de um horizonte de processamento finito e assegurando que cada poço seja atendido por somente uma única sonda. Ou seja, o PESM pode ser visto como um problema de agendamento de trabalhos utilizando máquinas paralelas (PÉREZ; OLIVEIRA; HAMACHER, 2018).

De acordo com Aloise et al. (2006), fornecer soluções ótimas para o PESM pode fazer com que uma empresa petrolífera economize milhares de dólares devido à diminuição da perda de produção de petróleo causada por poços inativos.

1.2. OBJETIVOS

A seguir são apresentados os objetivos gerais e específicos relacionados a este trabalho.

1.2.1. Objetivo Geral

Este trabalho tem como objetivo geral aplicar a meta-heurística *Clustering Search* (CS) para resolução do Problema de Escalonamento de Sondas de Manutenção – PESM.

1.2.2. Objetivos Específicos

a) Implementar o CS para resolver o PESM;
b) Realizar de experimentos computacionais utilizando as instâncias propostas por Costa (2005);
c) Avaliar o desempenho do CS comparando com os resultados apresentados na literatura.

1.3. ESTRUTURA DO TRABALHO

A seguir, é apresentada uma revisão da literatura no Capítulo 2, contendo uma descrição detalhada dos trabalhos com abordagem similar ao PESM e uma descrição do atual estado-da-arte sobre o roteamento de sondas. A metodologia

utilizada é descrita no Capítulo 3, no qual é retratada a meta-heurística *Clustering Search* e os seus principais componentes. No Capítulo 4 são apresentados os testes computacionais. No Capítulo 5 são relatados os resultados obtidos, e as conclusões são descritas no Capítulo 6, seguidas pelas referências bibliográficas.

2. REVISÃO DA LITERATURA

A busca de trabalhos correlatos ao Problema de Escalonamento de Sondas de Manutenção (PESM) foi realizada nas plataformas *Google Scholar*[1] e *Science Direct*[2] utilizando como parâmetros de busca o filtro de ano de publicação (2014 – 2019, últimos 5 anos) e as palavras-chaves: "*Workover Rigs Scheduling"*, "*Workover Metaheuristic"*, *"Workover Simulated Annealing"* e *"Workover Rig Problem"*; e em português: "Escalonamento sonda de manutenção", "Meta-heurística sonda de manutenção", "Sonda de manutenção *Simulated Annealing*" e "Problema de sonda de manutenção". A busca de cada palavra-chave foi realizada de maneira independente, de modo a atrelar a palavra-chave ao seu respectivo número de resultados.

Para a seleção dos trabalhos, utilizou-se uma análise baseada inicialmente na leitura do título de cada um dos resultados da busca, verificando a existência de relação entre o mesmo e o assunto tratado nesta pesquisa. Para os trabalhos selecionados, foi realizada a leitura do seu resumo.

As Tabelas 1 e 2 apresentam o resultado da análise das buscas realizadas na plataforma *Science Direct*.

Tabela 1 – Resultados da busca na plataforma *Science Direct* utilizando as palavras-chaves em inglês.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Workover Rigs Scheduling	64	2
Workover Metaheuristic	6	1
Workover Simulated Annealing	13	2
Workover Rig Problem	190	2
Total	273	7

Fonte: o autor.

Dos 273 resultados apresentados na Tabela 1, apenas 27 estavam relacionados à otimização, e o restante deles eram referentes especificamente à literatura do

1 https://scholar.google.com.br/
2 https://www.sciencedirect.com/

petróleo. Entretanto, destes poucos trabalhos da área de otimização, apenas 7 foram selecionados, pelo fato de estarem relacionados ao PESM. Dentre estes 7 selecionados, 5 foram repetidos, ou seja, os mesmos trabalhos foram obtidos por diferentes palavras-chave. Assim, dessa busca, restaram 2 trabalhos distintos.

Como pode ser observado na Tabela 2, nenhum dos 29 resultados obtidos estava relacionado com o tema tratado nessa pesquisa, pois ao realizar a leitura dos títulos destes trabalhos, percebeu-se que nenhum deles tinha relação com o PESM.

Tabela 2 – Resultados da busca na plataforma *Science Direct* utilizando as palavras-chaves em português.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Escalonamento sonda de manutenção	2	0
Meta-heurística sonda de manutenção	0	0
Sonda de manutenção *Simulated Annealing*	0	0
Problema da sonda de manutenção	27	0
Total	29	0

Fonte: o autor.

As Tabelas 3 e 4 apresentam o resultado da análise das buscas realizadas na plataforma *Google Scholar*. Vale ressaltar que esta plataforma limita o número páginas em 100 (mostrando 10 *links* por página) ou 50 (mostrando 20 *links* por página), o que reduz o número total de resultados acessíveis em 1000 por pesquisa. Logo, não foi possível analisar todos os artigos das palavras-chaves que tiveram o resultado maior do que 1000.

A Tabela 3 apresenta uma quantidade elevada de resultados, totalizando 7.398. Na primeira palavra-chave foram selecionados 5 trabalhos, porém 2 deles já haviam sido identificados na pesquisa realizada na plataforma *Science Direct*. Na segunda foram selecionados 3 trabalhos, mas nenhum deles foi escolhido, devido ao fato de que estes já haviam sido identificados na primeira palavra-chave. Na terceira, todos os 3 trabalhos selecionados já tinham sido identificados anteriormente. Na quarta,

apenas 1 foi selecionado, pois os 4 restantes também já haviam sido identificados. Por fim, 7 trabalhos foram selecionados, sendo: 4 referentes ao escalonamento, 1 referente à revisão da literatura do PSM e os 2 restantes foram desconsiderados, pois não podiam ser acessados mesmo pela rede interna da UFES.

Tabela 3 – Resultados da busca na plataforma Google Scholar utilizando as palavras-chaves em inglês.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Workover Rigs Scheduling	1430	5
Workover Metaheuristic	447	3
Workover Simulated Annealing	981	3
Workover Rig Problem	4540	5
Total	7398	16

Fonte: o autor.

A Tabela 4 também apresenta uma quantidade grande de resultados, totalizando 12.343. Porém, nenhum foi escolhido, devido ao fato de que todos esses trabalhos selecionados já tinham sido identificados nas buscas anteriores.

Tabela 4 – Resultados da busca na plataforma *Google Scholar* utilizando as palavras-chaves em português.

Palavra-chave	**Resultados**	**Número de trabalhos selecionados**
Escalonamento sonda de manutenção	438	3
Meta-heurística sonda de manutenção	39	5
Sonda de manutenção *Simulated Annealing*	66	5
Problema da sonda de manutenção	11800	8
Total	12343	21

Fonte: o autor.

Após a análise de todos os resultados, 5 trabalhos relacionados ao tema tratado nessa pesquisa foram selecionados, sendo: 4 relacionados ao PESM e 1 de revisão da literatura do PSM. Além disso, outros 5 trabalhos foram identificados por meio da referência cruzada. Estes não tinham sido revelados na busca anterior devido ao filtro baseado no ano de publicação (2014 – 2019), pois os mesmos foram publicados em anos anteriores. Todos os trabalhos selecionados são apresentados em detalhes a seguir.

2.1. TRABALHOS CORRELATOS AO PESM

Costa (2005) apresentou instâncias para o PESM geradas a partir do tratamento de dados estatísticos de casos reais e implementou uma meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) para resolvê-las. Essas instâncias são formadas por cinco grupos identificados pela quantidade de poços que necessitam de intervenção, sendo estes: grupo de 25 poços, grupo de 50 poços, grupo de 75 poços, grupo de 100 poços e grupo de 125 poços. Além disso, cada um desses grupos se distingue também no número de sondas, que pode variar entre 2, 4, 6, 8 ou 10 sondas. Portanto, considerando as combinações entre poços e sondas, tem-se um total 25 instâncias. Ademais, essas instâncias consideram que cada poço tem uma perda diária constante e desconsidera o custo de transporte pelo fato dos poços serem próximos um dos outros.

Aloise et al. (2006) propuseram um *Variable Neighborhood Search Heuristic* (VNS) para resolver instâncias de casos reais fornecidas pela Petrobras. Por se tratarem de dados confidenciais da empresa, os autores não apresentam todas as informações dessas instâncias. Desse modo, é apresentado somente que essas instâncias são formadas por 130 poços requerendo manutenção e 9 sondas heterogêneas. Por fim, com os resultados alcançados com a utilização do VNS, os autores forneceram uma economia de 107 mil dólares em relação à antiga solução encontrada pelos técnicos da empresa.

Douro e Lorenzoni (2009) apresentaram um Algoritmo Genético-2opt (AG) para a resolução do problema. Primeiramente, para gerar uma solução, os poços são ordenados de maneira decrescente de acordo com a sua perda de produção e posteriormente cada poço é alocado sequencialmente para cada sonda. Para as

operações genéticas, os autores utilizaram operador *single-point* para o cruzamento e a heurística 2opt para a mutação. Os testes foram realizados nas instâncias de Costa (2005) e o algoritmo proposto apresentou melhores soluções em relação ao método proposto por Costa (2005).

Pacheco (2011) apresentou um *Greed Randomized Adaptive Search Procedure* (GRASP) e um Algoritmo Memético (AM) como alternativas para solucionar as instâncias de Costa (2005). Para o GRASP apresentado, o autor considera o uso de soluções com violações, ou seja, considera soluções inviáveis como forma de diversificar as soluções e explorar mais o espaço de busca. Além disso, na busca local, o autor considerou dois movimentos diferentes para realocação dos poços: Remove e Adiciona e Troca de Poços de Diferentes Sondas. Um *path-relinking* foi utilizado para intensificar a busca por melhores soluções. A geração da solução inicial do AM utilizou uma abordagem similar à apresentada por Douro e Lorenzoni (2009). Para as operações genéticas no AM, foi utilizado o *Cycle CrossOver* e uma mutação aleatória. Além disso, o autor acrescentou uma etapa chamada de pós-otimização no AM para tentar diversificar e intensificar as soluções. Os resultados apresentados mostraram que o *path-relinking* melhorou as soluções encontradas pelo GRASP, porém o AM obteve melhores resultados para todas as instâncias. Além disso, o AM superou o AG proposto por Douro e Lorenzini (2009).

Ribeiro, Mauri e Lorena (2011) propuseram uma meta-heurística *Simulated Annealing* (SA) para resolução do PESM. A solução inicial é gerada por meio de uma heurística de distribuição na qual os poços são ordenados de maneira decrescente de acordo com a sua perda de produção e posteriormente cada poço é alocado sequencialmente para cada sonda. Para evitar ótimos locais, os autores utilizaram três diferentes métodos para compor a estrutura de vizinhança: *Re-order*, *Re-allocate* e *Swap wells*, sendo que, a cada iteração do algoritmo, um destes métodos era escolhido de maneira aleatória. Nos resultados computacionais obtidos com as instâncias de Costa (2005), os autores mostraram que algoritmo proposto foi capaz de apresentar melhores resultados em relação aos trabalhos anteriores.

Santos (2014) propôs um algoritmo baseado no *Variable Neighborhood Search* (VNS) que utiliza uma busca reativa para selecionar qual das diferentes buscas locais iria ser utilizada em uma determinada iteração. Dessa forma, a decisão de

utilizar uma busca local em detrimento de outra durante a execução era dada por meio do uso de um algoritmo de aprendizagem por reforço chamado Q-*learning*. Essa decisão era tomada com base em uma matriz *Q* que era atualizada de acordo com os resultados obtidos pelas iterações anteriores. Essa matriz é uma espécie de memória adaptativa utilizada como alternativa para tentar diversificar e intensificar as soluções. A fim de validar o método proposto e compará-lo ao VNS puro (sem a utilização da matriz *Q*), o autor apresentou os resultados para instâncias aleatórias criadas a partir de dados fornecidos por Aloise et al. (2006). O algoritmo proposto obteve melhores resultados quando comparado ao VNS puro (também implementado pelo autor).

Pérez, Oliveira e Hamacher (2016) apresentaram o *Reformulated Mathematical Model* (RMM), que é uma reformulação do modelo matemático original (OMM) de Costa e Ferreira Filho (2004). A partir dessa reformulação, os autores propuseram um novo modelo com menos variáveis e restrições, com o objetivo de permitir resolver grandes instâncias, chamado *Decomposed Mathematical Model* (DMM). Para avaliar o desempenho do modelo proposto (DMM) e compará-lo com o OMM, foram realizados testes computacionais no CPLEX utilizando as instâncias de Costa (2005). Os autores também compararam os resultados obtidos pelo método exato utilizando o OMM e DMM com os melhores resultados disponíveis na literatura utilizando as meta-heurísticas *Simulated Annealing* (RIBEIRO; MAURI; LORENA, 2011), *Genetic Algorithm* (DOURO; LORENZONI, 2009), e o *Memetic Algorithm* (PACHECO, 2011). Por fim, o DMM apresentou melhores resultados quando comparado aos outros métodos, sendo capaz de resolver todas as instâncias em menor tempo e encontrar soluções ótimas, sendo algumas destas não conhecidas até então.

Bissoli, Chaves e Ribeiro (2016) apresentam uma revisão da literatura dividida entre os dois principais grupos: *scheduling* (PESM) e *routing* (PRSM). Com base nessa revisão, os autores identificaram semelhanças entre o PRSM e o *Vehicle Routing Problem* (VRP) e indicaram novas áreas para futuras pesquisas, como a consideração de novos objetivos: minimização da quantidade de sondas, balanceamento da frota de sondas, priorização de poços, entre outros.

Pérez, Oliveira e Hamacher (2018) propuseram uma extensão dos modelos matemáticos apresentados por Pérez, Oliveira e Hamacher (2016) e Costa (2005), formulando dois novos modelos matemáticos para o *Workover Rig Fleet Sizing and Scheduling* (WRFSSP), onde o dimensionamento do tamanho da frota heterogênea é considerado por meio de dois modelos: determinístico (DTMM) e estocástico (STMM). A fim de comparação entre o DTMM e o modelo original (OMM) apresentado em Pérez, Oliveira e Hamacher (2016), foram apresentados resultados computacionais utilizando o CPLEX com instâncias de 50, 100, 150 e 200 poços, sendo que o modelo DTMM apresentou resolução em menor tempo para todas as instâncias. Os métodos *Monte Carlo, Scenario Reduction* e o *Quasi-Monte Carlo* foram utilizados para geração de cenários para o STMM. Dentre esses três métodos, o *Quasi-Monte Carlo* foi o mais adequado, pois apresentou melhor convergência, se mostrando estável até mesmo em cenários maiores, com 150 e 200 poços.

2.2. ESTADO DA ARTE DO PRSM

O PRSM recebeu atenção dos pesquisadores da área de otimização nos últimos 5 anos. Variadas meta-heurísticas foram apresentadas para resolução do problema, tais como: os 4 métodos de Ribeiro et al. (2014), *Variable Neighborhood Search* (VNS), *Adaptive Large Neighborhood Search* (ALNS), *Branch, Price and Cut* (BPC) e *Hybrid Genetic Algorithm* (HGA); ALNS de Bissoli (2014); HGA de Tozzo (2017) e BPC e a *Hyper-Heuristic Approach* de Monemi et al. (2018). Além disso, Silva (2016) e Silva e Silva (2018) propuseram modelos matemáticos dinâmicos para o problema e apresentaram resultados computacionais utilizando o CPLEX. Dessas meta-heurísticas, o HGA de Ribeiro et al. (2014) apresentou os melhores resultados conhecidos até o momento, seguidos pelo ALNS de Ribeiro et al. (2014) e o ALNS de Bissoli (2014).

2.3. CONSIDERAÇÕES

Considerando o exposto, foi proposta neste trabalho a utilização da meta-heurística *Clustering Search* (CS) para resolução do PESM. A escolha desta meta-heurística em detrimento das demais baseou-se nos seus bons resultados para outros problemas de otimização, como apresentado em Altoé et al. (2018), Melo, Barros Junior e Mauri (2015), Araújo e Mauri (2013). Além disso, uma lacuna na literatura

pôde ser identificada, já que não foram encontrados trabalhos que apresentam soluções para o PESM utilizando essa meta-heurística.

Este trabalho utilizou o grupo de instâncias geradas por Costa (2005) para realização dos testes computacionais. Diferente das instâncias utilizadas por Aloise et al. (2006), as instâncias geradas por Costa (2005) estão disponíveis na literatura e, como mencionado anteriormente, foram criadas a partir de dados estatísticos de problemas reais e então não são confidenciais. Por este motivo, elas têm sido utilizadas em diversos trabalhos encontrados na literatura, como o de Douro e Lorenzoni (2009), Pacheco (2011), Ribeiro, Mauri e Lorena (2011) e Pérez, Oliveira e Hamacher (2016). Dessa forma, foi possível realizar uma comparação entre os resultados obtidos e os disponibilizados na literatura.

3. METODOLOGIA

A metodologia deste trabalho consiste na utilização da meta-heurística *Clustering Search* (CS) para resolução do Problema de Escalonamento de Sondas de Manutenção (PESM).

Segundo Oliveira, Chaves e Lorena (2013), o CS é uma meta-heurística híbrida que permite a combinação de meta-heurísticas de busca com agrupamentos com o objetivo de encontrar regiões promissoras e, a partir disso, aplicar procedimentos de buscas locais nessas regiões.

Essas regiões promissoras são detectadas por meio do armazenamento em *clusters.* Cada um desses *clusters* possui um centro c_i, que representam a sua localização no espaço de soluções, um volume v_i, que é dado pela quantidade de soluções associadas a ele e um índice de ineficácia r_i, que serve como um contador para informar a quantidade de vezes que o centro do *cluster* não tem obtido melhoras pela busca local (OLIVEIRA; CHAVES; LORENA, 2013).

3.1. PRINCIPAIS COMPONENTES DO CS

Ainda segundo Oliveira, Chaves e Lorena (2013), o CS é um algoritmo iterativo que pode ser dividido em 4 partes independentes:

1. Meta-heurística geradora de soluções;
2. Processo de agrupamento iterativo;
3. Analisador de agrupamentos;
4. Heurística de busca local.

A Figura 3 apresenta o fluxograma conceitual do CS.

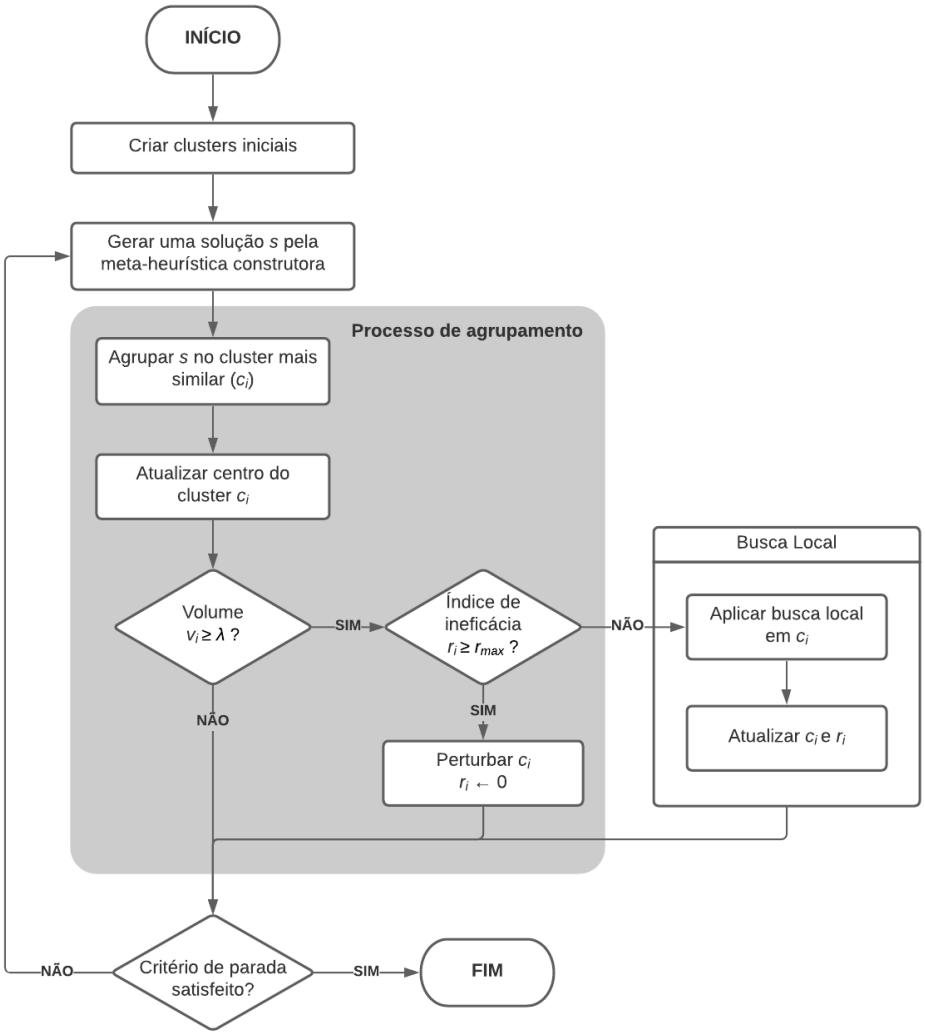

Figura 3 – Fluxograma do CS.
Fonte: Adaptado de Oliveira, Chaves e Lorena (2013).

3.1.1 Meta-heurística geradora de soluções

Utiliza um algoritmo gerador de soluções que funciona de maneira integral, gerando uma solução no decorrer de cada uma das iterações, que é encaminhada para o Analisador de Agrupamentos. Qualquer meta-heurística pode ser utilizada para realizar este trabalho. Entretanto, recomendam-se aquelas que gerem soluções diversificadas do espaço de busca (OLIVEIRA; CHAVES; LORENA, 2013).

A meta-heurística geradora de soluções utilizada nesse trabalho foi o *Simulated Annealing* (SA), baseado no CS proposto nos trabalhos de Araújo e Mauri (2013), Altoé et al. (2018) e Melo, Barros Junior e Mauri (2015).

O SA é um algoritmo proposto por Kirkpatrick, Gelatt e Vecchi (1983) que teve sua origem no algoritmo Metropolis (utilizado na estatística mecânica) e foi apresentado como uma alternativa para solucionar problemas de natureza combinatória. Foi desenvolvido baseando-se em um processo termodinâmico de recozimento (*annealing*) empregado na geração de cristais e vidros. Esse processo consiste no aquecimento desses materiais a altas temperaturas e posteriormente no seu resfriamento, até que sua estrutura seja congelada (ZBIGNIEW; FOGEL, 2000).

A ideia principal do algoritmo se baseia na simulação do processo de *annealing* (recozimento) que serve como mecanismo para escapar de ótimos locais, pois permite a realização de movimentos que resultem em soluções piores do que a solução atual. A probabilidade de aceitar esses movimentos que levam a soluções piores decresce durante as iterações do algoritmo (GOLDBARG; GOLDBARG; LUNA, 2016).

3.1.1.1 Solução inicial

Para geração da solução inicial, foi utilizada a mesma estratégia apresentada por Ribeiro, Mauri e Lorena (2011).

```
1.  poço ← 1      poço_max ← número de poços      sonda ← 1      sonda_max ← número de
    sondas
2.  listaPoços ← criarListaPoços()    listaPoçosOrdenados ← ordenar(listaPoços )
3.  ENQUANTO  poço <= poço_max  FAÇA
4.        associar(listaPoçosOrdenados[poço], sonda)
5.        SE sonda = sonda_max  ENTÃO
6.              sonda ← 1
7.        SENÃO
8.              sonda ← sonda + 1
9.        FIM-SE
10.  FIM-ENQUANTO
```

Figura 4 – Pseudocódigo da heurística construtora.
Fonte: Adaptado de Ribeiro, Mauri e Lorena (2011).

Essa estratégia é representada na Figura 4 e se baseia inicialmente na ordenação dos poços de acordo com o fator $P_i/\Delta t_i$, (linha 2) onde P_i representa a perda de produção do poço por unidade de tempo e Δt_i representa o tempo de manutenção necessário para o seu conserto. Após a ordenação, os poços são distribuídos sequencialmente para cada sonda durante as iterações (linha 4). É importante notar que sempre que um poço for distribuído para a última sonda, o próximo poço será

distribuído para a primeira sonda na iteração seguinte (linhas 5 e 6). Esse processo se repete até que todos os poços estejam sendo atendidos. Conforme os autores explicaram, essa estratégia garante que cada poço esteja sendo atendido por somente por uma sonda, porém não garante que as restrições de janela de tempo e de horizonte de processamento estejam sendo respeitadas.

3.1.1.2 Geração de vizinhança

Neste trabalho, foram utilizadas duas estratégias para geração de vizinho: realocação e a permutação entre dois poços.

Realocação: Consiste na troca da sonda que irá realizar atendimento a um determinado poço. Primeiramente, um poço é escolhido aleatoriamente (Figura 5a), então é escolhida de forma aleatória uma sonda para passar a atender esse poço (Figura 5b), desde que esta seja diferente da sonda que anteriormente o atendia.

Permutação entre dois poços: Consiste na troca das sondas que realizarão atendimento entre dois poços. Inicialmente, dois poços que estão sendo atendidos por sondas distintas são selecionados forma aleatória (Figura 6a). Então, as sondas que atendem esses poços são trocadas entre si (Figura 6b).

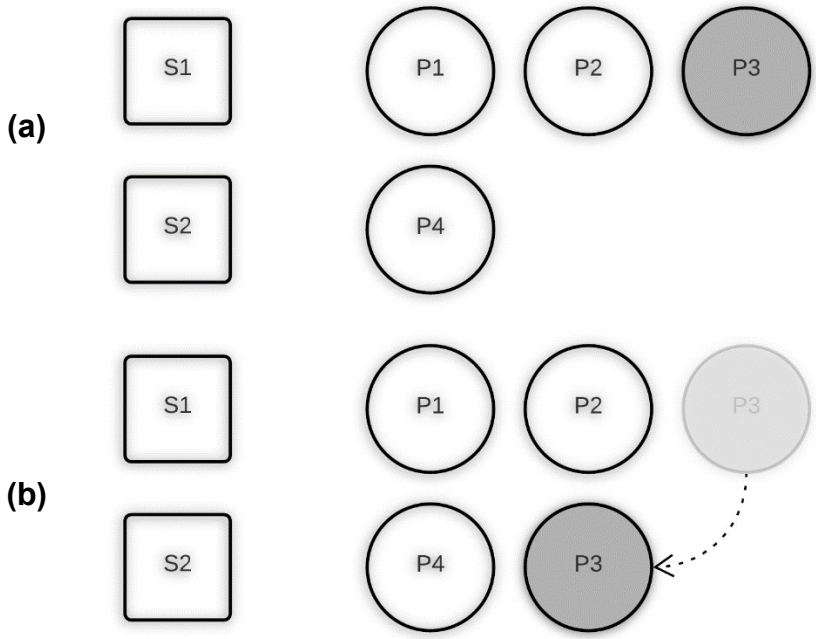

Figura 5 – Movimento de realocação.

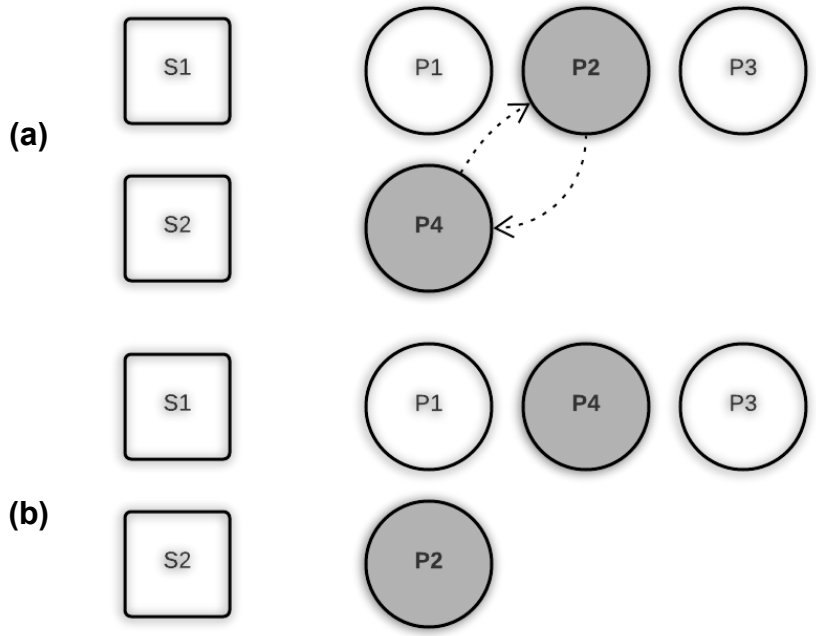

Figura 6 – Movimento de permutação entre dois poços.

3.1.2 Processo de agrupamento iterativo

Tem como objetivo identificar qual *cluster* possui maior similaridade com a solução atual. Essa similaridade é calculada através da distância de *Hamming* (HAMMING, 1950), que contabiliza o número de diferenças entre a solução atual e a solução encontrada no centro de cada *cluster*. Ao final desta etapa, a solução será associada ao *cluster* que apresentar o menor número de diferenças, ou seja, a menor distância, e o volume do *cluster* será incrementado. Caso a solução *s* seja melhor que a solução encontrada no centro do *cluster* ao qual foi associada, ou seja, minimize a função objetivo, ela passará então a representar o novo centro do *cluster* (OLIVEIRA; CHAVES; LORENA, 2013).

A Figura 7 apresenta a distância de *Hamming* entre duas soluções distintas s_1 e s_2. A diferença entre as soluções se encontra somente na sonda que atende o segundo poço.

s_1

Poço	1	2	
Sonda	1	**1**	

s_2

Poço	1	2	
Sonda	1	**2**	

Figura 7 – Representação da distância de *Hamming* entre duas soluções s_1 e s_2.
Fonte: Adaptado de Altoé et al. (2018).

3.1.3 Analisador de agrupamentos

Após a atualização do volume do *cluster* c_i na etapa anterior, é analisado se o volume v_i atingiu o limitante previamente definido v_{max}. Caso tenha atingido, significa que esse *cluster* pode estar indicando uma possível região promissora e vale a pena ser analisado mais a fundo. Então, é aplicada uma busca local no centro desse *cluster.* Caso a busca local não consiga melhorar a solução atualmente associada ao centro do *cluster*, o índice de ineficácia r_j do *cluster* é incrementado e caso tal índice atinja o seu limitante r_{max}, significa que a busca local foi executada $r_{\max}$ vezes naquele *cluster* e não obteve melhora. Neste caso, é aplicada uma perturbação no centro do *cluster*, com o objetivo de diversificar a solução e escapar de ótimos locais (OLIVEIRA; CHAVES; LORENA, 2013).

3.1.4 Heurística de busca local

É uma heurística responsável por realizar a busca local em uma possível área promissora, enquadrada em um *cluster*. Dado um conjunto de soluções, a busca local percorre a vizinhança de cada uma dessas soluções a procura de um vizinho que tenha uma solução melhor. Caso a solução vizinha encontrada seja melhor, essa será retornada e substituirá a solução anterior. Essa busca local pode ser realizada utilizando diferentes meta-heurísticas (OLIVEIRA; CHAVES; LORENA, 2013).

A busca local utilizada neste trabalho foi o 2-opt, que segundo Goldbarg e Luna (2000) é uma estratégia de melhoria que parte de um ciclo hamiltoniano e que apresenta bom desempenho computacional.

Para melhor exemplificar o 2-opt, considere como exemplo o Problema do Caixeiro Viajante (PCV), cujo objetivo é determinar a menor rota (roteiro) para percorrer um conjunto de cidades, visitando uma única vez cada uma delas, e retornar à cidade de origem ao final (CUNHA; BONASSER; ABRAHÃO, 2002). Nesse contexto de roteiro de visita do PCV, o algoritmo 2-opt se baseia na efetuação de trocas sucessivas entre duas arestas (onde cada vértice é uma cidade) com o objetivo de diminuir a distância total. De forma mais detalhada, o algoritmo seleciona duas arestas distintas e as remove, quebrando o circuito em dois caminhos, e os reconecta da outra maneira possível se e somente se essa modificação diminuir a distância total do roteiro. Essas trocas são efetuadas até que o algoritmo alcance um ótimo local, ou seja, o critério de parada do 2-opt é alcançado quando nenhuma troca possível pode resultar melhoria na solução (ENGLERT; RÖGLIN; VÖCKING, 2017).

Trazendo a explicação do algoritmo para o contexto deste trabalho, as arestas representam a ordem de atendimentos das sondas aos poços, e os poços por sua vez representam os vértices. Dessa maneira, o objetivo do algoritmo é realizar troca das sondas que atendem os poços e também na ordem de atendimento de forma a minimizar a perda de produção.

3.1.4 Pseudocódigo do CS

O pseudocódigo do algoritmo implementado neste trabalho é apresentado na Figura 8.

Nas linhas 1 e 2, é realizada a inicialização das variáveis utilizadas no decorrer do algoritmo. T_0 é um parâmetro que representa a temperatura inicial do SA e T é a variável que representa a temperatura atual, a qual é decrementada de acordo no decorrer das iterações do SA. A variável s representa a solução atual, enquanto s^* representa a melhor solução registrada até o momento. *Tempo* representa o tempo de execução do algoritmo e $Tempo_{max}$ corresponde ao tempo máximo de sua execução. A constante γ representa o número de *clusters*, onde cada *cluster* $i \in$

{1,...,γ} possui um centro c_i, um volume v_i e um índice de ineficácia r_i, que representam respectivamente, a melhor solução associada ao *cluster i*, o número de soluções associadas ao *cluster i* e a quantidade de vezes que a busca local foi aplicada a um *cluster i* sem obter melhoria na solução.

Um algoritmo SA clássico é apresentando nas linhas 4 a 17, onde T_f representa a temperatura final, *iteração* e SA_{max} representam respectivamente a iteração corrente e o número máximo de iterações, *s'* representa uma solução vizinha obtida através de um dos movimentos de geração de vizinhança escolhido aleatoriamente (linha 7) e a taxa de resfriamento é denotada como α.

A solução gerada pelo SA é enviada ao CS a partir da linha 18, onde essa solução é associada ao *cluster i* mais similar, de acordo com a distância de *Hamming* (representada na Figura 7), e o volume do *cluster i* é incrementado. Um *cluster i* é considerado promissor quando o seu volume v_i alcança o limitador λ, e quando isso acontece, é aplicado uma busca local no seu centro c_i (linha 21). Se a busca local conseguir melhorar a solução, esta passará então a representar o centro do *cluster i* (linhas 21 a 23), caso contrário, o seu índice de ineficácia r_i é incrementado e uma perturbação é aplicada se r_i atingir o limitante r_{max} (linhas 24 a 30).

Na linha 22, a melhor solução s^* recebe a menor solução entre o c_i *e* s^* e na linha 33, a variável *Tempo* é atualizada, recebendo a diferença entre o $Tempo_{atual}$ e o $Tempo_{inicial}$. O algoritmo é encerrado quando o tempo de execução alcançar o limitador $Tempo_{max}$.

```
1.  T ← T0        s ← solucao inicial        s* ← s        Tempo ← 0
2.  vi ← 0 , ri ← 0, ci ← solução inicial {∀i ∈ N: 1 < i < γ}
3.  ENQUANTO  Tempo < Tempomax  FAÇA
4.      ENQUANTO  T > Tf  FAÇA
5.          Iteração ← 0
6.          ENQUANTO  Iteração < SAmax  FAÇA
7.              s′ ← vizinho^aleatório[0,1](s)
8.              SE  f(s′) < f(s)  ENTÃO
9.                  s ← s′
10.                 SE  f(s′) < f(s*)  ENTÃO
11.                     s* ← s′
12.                 FIM-SE
13.             SENÃO
14.                 s ← s′, com probabilidade e^(−(f(s′)−f(s))/T)
15.             FIM-SE
16.         FIM-ENQUANTO
17.         T ← T × α
18          i ← arg min_{∀ i∈ N:1<i<γ} {Hi}        vi ← vi + 1        ci ← melhor(ci, s)
19.         SE  vi = λ  ENTÃO
20              vi ← 0
.
21.             s ← busca local( ci)
22              SE  f(s) < f(ci)  ENTÃO
.
23                  ri ← 0
.
24              SENÃO
.
25                 ri ← ri + 1
.
26                  SE  ri = rmax  ENTÃO
.
27                     ri ← 0
.
28                     ci ← vizinho^aleatório[0,1](ci)
.
29                  FIM-SE
.
30             FIM-SE
.
31.         FIM-SE
32          s* ← mínimo(s*, ci)
.
33         Tempo ← Tempoatual − Tempoinicial
.
34      FIM-ENQUANTO
.
35  FIM-ENQUANTO
.
36  Retornar s*
```

.

Figura 8 – Pseudocódigo do CS utilizado.
Fonte: Adaptado de Araújo e Mauri (2013).

4. TESTES COMPUTACIONAIS

Para validar a eficácia do CS na resolução do PESM, testes computacionais foram realizados nas instâncias de Costa (2005). Conforme mencionado na revisão da literatura, essas amostras são compostas por um conjunto de 25 instâncias que possuem 25, 50, 75,100 e 125 poços, variando entre 2, 4, 6, 8 e 10 sondas.

O CS foi implementado na linguagem C/C++ e compilado pelo GNU GCC *Compiler* versão 7.0.3. O algoritmo foi executado em um computador equipado com um processador Intel Core i3 7100 de 3.9 GHz, 16 GB de memória RAM e sistema operacional Windows 10.

4.1. CALIBRAÇÃO DOS PARÂMETROS

A estratégia para calibração de parâmetro foi inspirada na de Ribeiro, Mauri e Lorena (2011) e foi dividida em duas etapas: inicialização dos parâmetros e aprimoramento.

1. Inicialização dos parâmetros: definição dos valores iniciais dos parâmetros. Nesta etapa, os valores escolhidos inicialmente foram os mesmos utilizados na calibração de CS de Altoé et al. (2018).
2. Aprimoramento: realização de alterações nos parâmetros com o objetivo de melhorar a eficácia do algoritmo. Nesta parte, um parâmetro era escolhido e então passava a assumir outros valores dentro do intervalo especificado na Tabela 5, enquanto o restante dos parâmetros permanecia fixo. Para cada alteração, o conjunto de testes era executado e a calibração que levava ao melhor resultado era mantida. Caso não houvesse melhora durante três mudanças consecutivas, este parâmetro era fixado e então um próximo parâmetro era definido e essa etapa era executada novamente.

Dessa maneira, a segunda etapa foi executada várias vezes para todos os parâmetros, onde os valores eram definidos empiricamente em cada iteração e ajustados de acordo com os resultados obtidos nos testes. A condição de parada era atendida quando não havia melhora na solução com três mudanças consecutivas para todos os parâmetros durante a mesma iteração.

Tabela 5 – Intervalo de valores possíveis para cada parâmetro durante a calibração.

Parâmetro	Significado	Intervalo de valores
T_0	Temperatura inicial	{300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, 1000}
T_f	Temperatura de congelamento	{0.001, 0.0018, 0.005, 0.0075, 0,01}
SA_{max}	Número máximo de iterações	{3000, 3500, 4000, 4500, 5000, 5500, 6000, 6500, 7000, 7500}
α	Taxa de resfriamento	{0.965, 0.970, 0.975}
γ	Número de *clusters*	[4, 12]
λ	Volume máximo dos *clusters*	[3, 10]
r_{max}	Limite do índice de ineficácia dos *clusters*	[3, 8]

O conjunto de teste é composto por seis instâncias: 25-2, 25-6, 25-10, 125-2, 125-6 e 125-10. Onde o primeiro valor representa o número de poços e o segundo, o número de sondas disponíveis. Após a finalização da calibração, os parâmetros foram definidos conforme apresentado na Tabela 6.

Tabela 6 – Valores dos parâmetros do SA e do CS após a calibração.

Meta-heurística	Parâmetro	Significado	Valor
SA	T_0	Temperatura inicial	400
	T_f	Temperatura de congelamento	0,0018
	SA_{max}	Número máximo de iterações	7500
	α	Taxa de resfriamento	0,975
CS	γ	Número de *clusters*	10
	λ	Volume máximo dos *clusters*	7
	r_{max}	Limite do índice de ineficácia dos *clusters*	4

5. RESULTADOS OBTIDOS

Esta seção apresenta os resultados computacionais do CS aplicado ao PESM e tem três objetivos principais:

1. Analisar a robustez do CS;
2. Comparar o CS com outras abordagens heurísticas encontradas na literatura;
3. Comparar o CS com o método exato de Pérez, Oliveira e Hamacher (2016).

5.1. RESULTADOS COMPUTACIONAIS

A Tabela 7 apresenta o resumo dos resultados computacionais obtidos após executar o CS cinco vezes para cada instância durante 30 segundos. Nessa tabela, é apresentado para cada instância a melhor função objetivo encontrada durante todas as execuções, a média aritmética (entre as 5 execuções), o desvio (calculado por $Desvio(\%) = \frac{Média - MelhorFO}{MelhorFO} \times 100$) e o tempo médio (em segundos) requerido para encontrar a melhor função obtivo durante as execuções.

Tabela 7 – Resumo dos resultados computacionais.

Inst.	*Clustering Search*			
	Melhor FO	Média	Desvio (%)	Tempo médio (s)
25-2	16.329	16.329	0,000	0,08
25-4	10.312	10.312	0,000	0,55
25-6	8.497	8.497	0,000	0,71
25-8	7.733	7.733	0,000	0,71
25-10	7.322	7.322	0,000	0,71
50-2	66.904	66.904	0,000	0,65
50-4	37.891	37.891	0,000	3,27
50-6	28.346	28.347	0,002	6,26
50-8	23.788	23.788	0,000	5,40
50-10	21.348	21.348	0,000	4,57
75-2	187.236	187.236	0,000	2,71
75-4	103.200	103.202	0,002	14,63
75-6	75.498	75.499	0,001	14,10
75-8	61.882	61.883	0,001	11,84
75-10	53.876	53.877	0,001	10,34
100-2	299.040	299.040	0,000	11,52
100-4	159.941	159.946	0,003	12,42
100-6	114.270	114.271	0,001	18,38
100-8	91.751	91.754	0,003	12,50
100-10	78.369	78.372	0,004	18,87
125-2	380.507	380.510	0,001	11,87
125-4	200.296	200.299	0,001	10,92
125-6	140.499	140.501	0,001	15,88
125-8	110.782	110.784	0,002	22,04
125-10	93.041	93.044	0,003	18,78

Como pode ser observado na Tabela 7, o CS apresentou um desvio baixo, com o máximo de 0,004% no pior caso, para a instância 100-10, e um desvio médio de 0,0011%. Demonstrou também bons resultados no que se refere ao tempo computacional, apresentando um tempo máximo 22,04 segundos no pior caso, na instância 125-8, e um tempo médio durante as execuções de 9,19 segundos. Logo, nota-se que o CS se mostrou como um algoritmo rápido e robusto para resolução do PESM.

5.2. COMPARAÇÃO COM A LITERATURA

A Tabela 8 apresenta a comparação dos resultados obtidos com o CS com outras abordagens encontradas na literatura, porém, considerando somente os trabalhos que utilizaram meta-heurísticas. Tais resultados são apresentados em ordem cronológica.

Tabela 8 – Comparação dos resultados com outras abordagens heurísticas da literatura.

Inst.	AG[3]	AM[4]	SA[5]	CS
25-2	**16.329***	**16.329***	**16.329***	**16.329***
25-4	**10.312***	**10.312***	**10.312***	**10.312***
25-6	8.499	**8.497***	**8.497***	**8.497***
25-8	7.736	**7.733***	**7.733***	**7.733***
25-10	7.325	**7.322***	**7.322***	**7.322***
50-2	66.907	**66.904***	**66.904***	**66.904***
50-4	37.896	**37.891***	**37.891***	**37.891***
50-6	28.353	**28.346***	**28.346***	**28.346***
50-8	23.788*	**23.788***	**23.788***	**23.788***
50-10	21.351	**21.348***	**21.348***	**21.348***
75-2	187.240	**187.236***	**187.236***	**187.236***
75-4	103.218	103.204	103.202	**103.200**
75-6	75.524	75.499	75.499	**75.498**
75-8	61.916	61.884	**61.882***	**61.882***
75-10	53.889	53.881	**53.876***	**53.876***
100-2	299.051	299.041	299.041	**299.040**
100-4	159.983	159.945	159.948	**159.941**
100-6	114.275	114.273	114.274	**114.270**
100-8	91.769	91.765	91.758	**91.751**
100-10	78.402	78.391	78.376	**78.369**
125-2	380.523	380.511	380.511	**380.507**
125-4	200.368	200.297	200.302	**200.296**
125-6	140.550	140.513	140.503	**140.499**
125-8	110.844	**110.782***	110.787	**110.782***
125-10	93.078	93.049	93.045	**93.041**
Média	95.165,04	95.149,64	95.148,40	95.146,32

3 Algoritmo Genético (AG) com busca local 2-opt (DOURO; LORENZONI, 2009). Computador utilizado: Processador Pentium IV 2.0 GHz com 480 MB de memória RAM.

4 Algoritmo Memético (AM) (PACHECO, 2011). Computador utilizado: Processador Pentium Core i3 2.13 GHz com 4 GB de memória RAM.

5 *Simulated Annealing* (CS) (RIBEIRO; MAURI; LORENA, 2011). Computador utilizado: Processador AMD Athlon 64 3500 2.2 GHz com 1 GB de memória RAM.

Os melhores resultados para cada instância se encontram em negrito. O sufixo asterisco (*) representa um empate, enquanto os resultados em negrito e sublinhados representam uma vitória exclusiva, ou seja, o melhor resultado conhecido.

Analisando as instâncias menores, nota-se que o MA, SA e o CS alcançaram 100% dos resultados iguais para as 10 (40%) instâncias de 25 e 50 poços.

Levando em consideração as 15 instâncias restantes, ou seja, as instâncias com 75 poços ou mais, percebe-se que o CS superou as outras abordagens encontradas na literatura, pois foi capaz de encontrar novas melhores soluções para 11 instâncias (73,3%) e empatar nas 4 instâncias (26,7%) restantes.

Tabela 9 – Comparação dos tempos computacionais com outras abordagens heurísticas da literatura.

Inst.	AG[3]	AM[4]	SA[5]	CS
25-2	4,00	2,40	4,20	0,08
25-4	5,00	3,50	4,60	0,55
25-6	2,00	3,90	5,70	0,71
25-8	1,00	5,10	7,00	0,71
25-10	1,00	6,90	8,30	0,71
50-2	4,00	4,10	6,70	0,65
50-4	5,00	6,80	6,70	3,27
50-6	2,00	7,90	7,70	6,26
50-8	1,00	10,90	8,50	5,40
50-10	1,00	14,00	10,40	4,57
75-2	36,00	5,90	9,20	2,71
75-4	27,00	7,50	8,40	14,63
75-6	25,00	12,60	9,60	14,10
75-8	18,00	16,80	10,60	11,84
75-10	29,00	22,60	11,90	10,34
100-2	61,00	17,60	11,90	11,52
100-4	48,00	10,40	10,30	12,42
100-6	136,00	15,10	11,10	18,38
100-8	141,00	20,70	12,90	12,50
100-10	58,00	28,20	13,80	18,87
125-2	146,00	10,50	14,60	11,87
125-4	60,00	12,20	12,10	10,92
125-6	84,00	19,40	12,70	15,88
125-8	66,00	26,00	14,20	22,04
125-10	164,00	34,50	15,70	18,78
Média	45,00	13,02	9,95	9,19

A Tabela 9 apresenta os tempos computacionais reportados pelas meta-heurísticas AG, AM, SA e CS.

Como mostrado no resumo dos resultados computacionais, o CS apresentou no pior caso um tempo de 22,04 segundos na instância 125-8 e um tempo médio de 9,19 segundos, que é inferior ao tempo médio obtido pelas outras abordagens. Entretanto, é difícil realizar uma comparação do tempo computacional de maneira justa devido à diferença de *hardware* dos computadores utilizados para execução dos testes computacionais.

Tabela 10 – Comparação dos resultados com o método exato.

Inst.	Método Exato[6]		CS		
	FO	Tempo (s)	FO	Tempo (s)	Diferença (%)
25-2	**16.329***	0,00	**16.329***	0,08	0,000
25-4	**10.312***	0,00	**10.312***	0,55	0,000
25-6	**8.497***	0,00	**8.497***	0,71	0,000
25-8	**7.733***	0,00	**7.733***	0,71	0,000
25-10	**7.322***	0,00	**7.322***	0,71	0,000
50-2	**66.904***	0,30	**66.904***	0,65	0,000
50-4	**37.891***	0,20	**37.891***	3,27	0,000
50-6	**28.346***	0,00	**28.346***	6,26	0,000
50-8	**23.788***	0,00	**23.788***	5,40	0,000
50-10	**21.348***	0,00	**21.348***	4,57	0,000
75-2	**187.236***	1,40	**187.236***	2,71	0,000
75-4	**103.199**	0,40	103.200	14,63	-0,001
75-6	**75.498***	0,20	**75.498***	14,10	0,000
75-8	**61.881**	0,10	61.882	11,84	-0,002
75-10	**53.876***	0,10	**53.876***	10,34	0,000
100-2	**299.040***	3,80	**299.040***	11,52	0,000
100-4	**159.939**	1,50	159.941	12,42	-0,001
100-6	**114.264**	0,50	114.270	18,38	-0,005
100-8	**91.746**	0,50	91.751	12,50	-0,005
100-10	**78.364**	0,30	78.369	18,87	-0,006
125-2	**380.506**	6,20	380.507	11,87	0,000
125-4	**200.291**	3,30	200.296	10,92	-0,002
125-6	**140.493**	0,90	140.499	15,88	-0,004
125-8	**110.775**	0,90	110.782	22,04	-0,006
125-10	**93.040**	0,50	93.041	18,78	-0,001
Média	95.144,72	0,84	95.146,32	9,19	-0,001

6 Método Exato (Modelo Matemático Decomposto - DMM) (PÉREZ; OLIVEIRA; HAMACHER, 2016). Computador utilizado: Processador Intel Core i7-3960X com 3.3 GHz e 64 GB de RAM.

A Tabela 10 apresenta a comparação dos resultados do CS com os resultados ótimos encontrados pelo método exato de Pérez, Oliveira e Hamacher (2016). A coluna Diferença representa a diferença entre a solução obtida pelo CS e solução ótima conhecida, este valor é obtido através do cálculo $Diferença(\%) = \frac{FO\,obtida - FO\,ótima}{FO\,ótima} \times 100$.

Analisando os resultados apresentados na Tabela 10, percebe-se que CS foi capaz de encontrar 14 soluções ótimas (56%), do total de 25 instâncias. Além disso, provou novamente sua robustez ao encontrar soluções próximas do ótimo nas 11 instâncias restantes, apresentando uma diferença máxima de 0,006% na instância 125-8 e uma diferença média de 0,001%.

Do ponto de vista de tempo computacional, pode-se observar que o método exato superou o CS em todas as instâncias, apresentando um tempo médio de 0,84 segundos contra 9,19 segundos do CS. Entretanto, como apontado por Pérez, Oliveira e Hamacher (2016), os tempos computacionais não podem ser comparados diretamente pelo fato das máquinas utilizadas na execução do algoritmo apresentarem características distintas.

6. CONCLUSÕES

Este trabalho apresentou uma nova alternativa para resolução de um dos Problemas de Sondas de Manutenção (PSM), conhecido como Problema de Escalonamento de Sondas de Manutenção (PESM). O PESM é um dos problemas mais desafiadores da indústria de óleo e gás por conta das altas taxas de perda de produção causadas por poços inativos aguardando manutenção e pelo custo alto de utilização das sondas.

Para a resolução do PESM, foi proposta a utilização da meta-heurística *Clustering Search* (CS), por ter apresentado bons resultados para outros problemas combinatoriais e também pelo fato dessa meta-heurística ainda não ter sido utilizada para resolução do PESM.

Para validar a meta-heurística para resolução do PESM, testes computacionais foram realizados em instâncias criadas a partir de dados estatísticos de problemas reais no Sudeste Brasil. O CS apresentou bons resultados nesses testes, com um desvio baixo e um tempo computacional médio de aproximadamente 9 segundos. Além disso, os resultados mostraram que o CS foi capaz de encontrar novas melhores soluções quando comparado a outros trabalhos que também utilizaram meta-heurísticas para resolução do PESM, e se mostrou eficiente mesmo quando comparado com a resolução exata de Pérez, Oliveira e Hamacher (2016), sendo capaz de encontrar 56% das soluções ótimas e com uma diferença média de 0,003% das melhores soluções restantes, porém, com um tempo computacional superior.

Os resultados obtidos mostram a eficiência da meta-heurística proposta, dado que as soluções apresentadas foram melhores que as reportadas por outras meta-heurísticas na literatura. Deste modo, pode-se afirmar que o CS se mostrou uma alternativa para resolução do PESM.

Em trabalhos futuros, seria interessante verificar o desempenho do método proposto para a outra modalidade do PSM, o Problema de Roteamento de Sondas de Manutenção. Além disso, analisar a possibilidade de utilizar abordagens estocásticas para lidar com eventos inesperados, como apresentado por Bissoli, Chaves e Ribeiro (2016).

Por fim, é importante salientar a necessidade da geração de novas instâncias para o PESM, já que o único grupo de instâncias não confidencial disponível na literatura é o de Costa (2005).

REFERÊNCIAS BIBLIOGRÁFICAS

ALOISE, D. J. et al. Scheduling workover rigs for onshore oil production. **Discrete Applied Mathematics**, n. 154, p. 695–702, 2006.

ALTOÉ, W.A.S. et al. A clustering search metaheuristic for the bi-objective flexible job shop scheduling problem. XLIV CLEI - Latin American Computing Conference, São Paulo, 2018.

ANP, Agência Nacional de Petróleo, Gás Natural e Biocombustíveis. **Boletim da Produção de Petróleo e Gás Natural**. n. 113, 36p, jan. 2020.

ARAÚJO, D. L.; MAURI, G. R. Clustering search para resolução de um problema de rotação de culturas com restrições de adjacências. Anais do XLV SBPO - Simpósio Brasileiro de Pesquisa Operacional, Natal, 2013.

BISSOLI, D. C. **Uma abordagem heurística para o problema de roteamento de sondas de intervenção bi-objetivo**. 2014. 102 f. Dissertação (Pós-graduação em Energia), Universidade Federal do Espírito Santo, São Mateus, ES, 2014.

BISSOLI, D. C.; CHAVES, G. de L. D.; RIBEIRO; G. M. Drivers to the workover rig problem. **Journal of Petroleum Science and Engineering,** v. 139, p. 13-22, 2016.

Brasil; Governo Brasileiro – Brasil. Petróleo e derivados constituem base da economia produtiva mundial. 2011. Disponível em <http://www.brasil.gov.br/noticias/infraestrutura/2011/11/petroleo-e-derivados constituem-base-da-economia-mundial>. Acesso em junho de 2019.

COSTA, L. R. **Soluções para o Problema de Otimização de Itinerário de Sondas**. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil, 2005.

COSTA, L. R.; FERREIRA FILHO, V. J. M F. Uma heurística para o problema do planejamento de itinerários de sondas em intervenções de poços de petróleo. Anais do XXXVI SBPO – Simpósio Brasileiro de Pesquisa Operacional, São João Del Rei, MG, 2004.

CUNHA, C. B.; BONASSER, U. O.; ABRAHÃO, F. T M. Experimentos computacionais com heurísticas de melhorias para o problema do caixeiro viajante. In: XVI Congresso da Anpet. 2002.

DOURO, R. F.; LORENZONI, L. L. Um algoritmo genético-2opt aplicado ao problema de otimização de itinerário de sondas de produção terrestre. XLI SBPO - Simpósio Brasileiro de Pesquisa Operacional, Porto Seguro, BA, 2009.

ENGLERT, M.; RÖGLIN, H.; VÖCKING, B. Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP. In: SODA, p. 1295-1304. 2007.

GOLDBARG, M. C.; GOLDBARG, E. G.; LUNA, H. P. L. **Otimização combinatória e meta-heurísticas**: Algoritmos e aplicações. Rio de Janeiro: Elsevier, 2016. ISBN 978-85-352-7812-5.

GOLDBARG, M. C.; LUNA, H. P. L. Otimização combinatória e programação linear: modelos e algoritmos. Campus, Rio de Janeiro, 2000.

GOUVÊA, E. F.; GOLDBARG, M. C.; COSTA, W. E. Algoritmos evolucionários na solução do problema de otimização do emprego de sondas de produção em poços de petróleo. Anais do XXXVI SBPO - Simpósio Brasileiro de Pesquisa Operacional, Rio de Janeiro, 2002.

HAMMING, R. W. Error detecting and error correcting codes. **Bell System Technical Journal**, v. 26, p. 147-160, 1950.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. Science, v. 220, n. 4598, p. 671- 680, 1983.

MELO, R. S.; BARROS JUNIOR, A. A.; MAURI, G.R. Clustering search com path relinking para resolução de um problema de planejamento florestal. XLVII SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2015.

Ministério de Minas e Energia; Empresa de Pesquisa Energética – Brasil. Resenha Energética Brasileira. Brasília: 2019. Disponível em <http://www.mme.gov.br/documents/1138787/1732840/Resenha+Energ %C3%A9tica+Brasileira+-+edi%C3%A7%C3%A3o+2019+v2.pdf/66a837a8-4164-4b37-be4a-59a5ad270c50?version=1.0>. Acesso em junho de 2019.

MONEMI, R. N. et al. Solution methods for scheduling of heterogeneous parallel machines applied to the workover rig problem. **Expert Systems with Applications**, v. 42, p. 4493-4505, 2018.

OLIVEIRA, A. C. M.; CHAVES, A. A.; LORENA, L. A. N. Clustering search. **Pesquisa Operacional**, Rio de Janeiro , v. 33, n. 1, p. 105-121, abr. 2013.

PACHECO, A. V. F. **Métodos de solução para o problema da alocação de sondas a poços de petróleo**. 2011. 68 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Produção), Universidade Federal do Espírito Santo, São Mateus, ES, 2011.

PÉREZ, M.; OLIVEIRA, F.; HAMACHER, S. A new mathematical model for the workover rig scheduling problem. **Pesquisa Operacional**, v. 36, n. 2, p. 241-257, 2016.

PÉREZ, M.; OLIVEIRA, F.; HAMACHER, S. **Modelos de programação matemática para o problema de intervenção em poços terrestres de petróleo**. 2017. 93 f. Tese (Doutorado em Engenharia de Produção), Pontífica Universidade Católica, Rio de Janeiro, RJ, 2017.

PÉREZ, M.; OLIVEIRA, F.; HAMACHER, S. Optimizing Workover Rig Fleet Sizing and Scheduling Using Deterministic and Stochastic Programming Models. **Industrial & Engineering Chemistry Research**, v. 57, n. 22, p. 7544-7554, 2018.

RIBEIRO, G. M. et al. Efficient heuristics for the workover rig routing problem with a heterogeneous fleet and a finite horizon. **Journal of Heuristics**. v. 20, n.6, p. 677-708, 2014.

RIBEIRO, G. M.; MAURI, G. R.; LORENA, L. A. M. A simple and robust simulated annealing algorithm for scheduling workover rigs on onshore oil fields. **Computers & Industrial Engineering**, v. 60, p. 519–526, 2011.

SANTOS, J. P. Q. **Estratégias de Busca Reativa Utilizando Aprendizagem por Reforço e Algoritmos de Busca Local**. 2014. 115 f. Tese (Doutorado em Engenharia Elétrica e Computação), Universidade Federal do Rio Grande do Norte, Natal, RN, 2014.

SILVA, F. T. **Proposta de modelo matemático para o problema de roteamento de sondas de intervenção a poços de petróleo terrestres revelados dinamicamente com o período de atendimento viável**. 2016. 72 f. Dissertação (Pós-graduação em Energia), Universidade Federal do Espírito Santo, São Mateus, ES, 2016.

SILVA, F. T.; SILVA, R. P. Roteamento dinâmico de sondas de intervenção para otimização da produção de poços de petróleo: Um modelo matemático para o PRSI Dinâmico. **Brazilian Journal of Production Engineering**, São Mateus, v. 4, n. 3, p. 169-183, 2018.

TOZZO, E. **A hybrid multi-objective genetic algorithm for scheduling heterogeneous workover rigs on onshore oil fields**. 2017. 90 f. Tese (Pós-graduação em Engenharia de Produção), Universidade Federal de Pernambuco, Recife, PE, 2017.

ZBIGNIEW, M.; FOGEL, D.B. How to Solve it: Modern Heuristics. 2 ed. Springer-Verlag, 2000.