

HEURÍSTICA LAGRANGEANA APLICADA A UM PROBLEMA DE PROGRAMAÇÃO DE FROTA DE APOIO MARÍTIMO OFFSHORE

Victor Cavinato Moura

Departamento de Engenharia Naval e Oceânica, USP
victorc85@gmail.com

André Bergsten Mendes

Departamento de Engenharia Naval e Oceânica, USP
andbergs@usp.br

Resumo

O presente trabalho aborda o problema de programação de uma frota de navios de lançamento de dutos, que atua no apoio marítimo *offshore*. O problema foi enquadrado em um problema de máquinas paralelas, para o qual foi elaborado um método heurístico de solução. Este combina uma heurística lagrangeana que permite estimar o valor ótimo da função objetivo, bem como fornece sementes para a metaheurística VNS. O método proposto foi testado em um conjunto de problemas adaptados da literatura, e mostrou ser capaz de gerar soluções ótimas e soluções com gaps médios inferiores a 1%.

Palavras-Chaves: PLSV, apoio marítimo offshore, programação de máquinas paralelas, relaxação lagrangeana, VNS

Abstract

This work focuses on a pipe layer support vessel scheduling problem, found in offshore operations. The problem was treated as a parallel machine scheduling problem and solved heuristically by a combination of the lagrangean relaxation method and a VNS metaheuristic. The former was used for generating lower bounds and also to provide seeds for the VNS. This method was tested for a set of benchmark problems adapted from the literature and proved to be effective inasmuch optimal and near optimal solution were found.

Keywords: PLSV; offshore support vessels; parallel machine scheduling; lagrangean relaxation; VNS

1. INTRODUÇÃO

O segmento de transporte marítimo é comumente compreendido segundo os seus modos de operação. Christiansen et al. [1] os classificam em linhas regulares, linhas não-regulares e frota industrial. O primeiro modo é característico de operações em que há uma rota definida e um serviço regular associado, como o caso do mercado de contêineres. A segunda forma é típica de armadores que saem à procura de carga, segundo o aquecimento do mercado, como no transporte de grãos. Já a frota industrial é encontrada em operações de grandes grupos industriais que gerenciam uma dada frota, seja ela própria ou afretada, alocando-a de forma a atender a sua demanda e otimizando uma figura de mérito como, por exemplo, a minimização do custo total da operação.

É neste contexto em que o presente trabalho se enquadra, em que uma empresa de petróleo necessita otimizar a alocação da sua frota de PLSVs (*pipe layer support vessels*). Estas embarcações de apoio marítimo *offshore* são responsáveis pelo lançamento de dutos e a interligação destes à infraestrutura submarina, possibilitando o transporte de óleo entre os poços dos quais são extraídos a uma unidade estacionária de produção.

O problema em questão foi enquadrado em um problema de programação de máquinas paralelas, com o objetivo regular de minimização do atraso total ponderado das

tarefas de interligação. Este problema será resolvido de maneira aproximada por uma heurística que combina a técnica da relaxação lagrangeana [2] com a metaheurística VNS [3]. A relaxação lagrangeana terá duas funções importantes, a saber, a de estimar um *lower bound* para a solução ótima do problema, e a de prover uma solução inicial para o VNS. Neste caso, primeiramente, será feita uma conversão da solução dual para o espaço de solução primal.

O artigo apresentará o problema em maiores detalhes na seção 2, seguido da formulação matemática na seção 3. A seção 4 introduz as heurísticas desenvolvidas. Os resultados são apresentados na seção 5. A seção 6 traz as principais conclusões e a bibliografia consultada consta na seção 7.

2. DESCRIÇÃO DO PROBLEMA

Diversas são as etapas de preparação de um campo de petróleo para que o mesmo entre em operação. As embarcações de lançamento de dutos atuam na última etapa, a qual precede o início da produção de um poço.

Uma empresa de petróleo, em seu planejamento anual, fixa um horizonte de planejamento (por exemplo, um período de 4 ou 6 meses), e necessita determinar a sequência de tarefas de lançamento de dutos e interligação que irá realizar, levando em consideração a demanda, a frota disponível, o local e a duração das tarefas (que podem durar até 60 dias), o potencial de produção associado a cada tarefa, e o instante a partir do qual a tarefa poderia ser iniciada.

Sabe-se que a conclusão de cada tarefa gera um acréscimo nos níveis de produção da empresa, de forma que, a postergação do início de um serviço de interligação traz como consequência uma perda financeira pelo adiamento do retorno sobre o investimento realizado. Busca-se, desta maneira, priorizar as tarefas que proporcionam maiores acréscimo nos níveis de produção, respeitando as restrições operacionais.

As tarefas de instalação e interligação são usualmente compostas de um conjunto de atividades que devem ser feitas sequencialmente. É imprescindível que haja, em primeiro lugar, o carregamento das linhas no porto, juntamente com todos os materiais pertinentes, como conectores, ganchos, manilhas, etc. Em seguida a embarcação se dirige para o local previsto e inicia o processo de preparação, conexão e disposição da linha no leito marítimo, unindo um poço a um *manifold* submarino (isto é, uma caixa de conexão), ou unindo um duto rígido a um duto flexível. O processo de ida ao porto pode ser repetido diversas vezes em função do comprimento da linha a ser lançada. Por último, os materiais não utilizados são devolvidos a uma base operacional.

A programação das atividades que compõem uma tarefa de interligação não é o escopo desse trabalho. É assumido que o setor de engenharia submarina da empresa petrolífera definiu previamente a melhor forma de instalar um duto ou interligar um poço, e especificou a duração mínima possível da interligação.

O tempo que cada embarcação que compõe a frota de PLSVs da empresa leva para executar uma interligação varia de tarefa para tarefa. É assumido que qualquer embarcação da frota seja capaz de realizar qualquer uma das tarefas previstas, observando que cada embarcação deverá executar uma única tarefa por vez. Além disso, se uma tarefa for iniciada, a mesma deverá ser finalizada sem a possibilidade de interromper o serviço para retomá-lo mais tarde. Visto que uma embarcação PLSV, durante a execução de uma tarefa, se desloca constantemente até o porto e lá permanece até que os materiais sejam carregados, será admitido que o abastecimento da embarcação e a troca da tripulação sejam realizados nesses momentos em que a embarcação está atracada no porto. Além disso, a duração da execução de cada tarefa incluirá os tempos médios de porto, bem como os tempos médios de navegação entre o porto e o local da execução da tarefa, além do tempo necessário para execução do serviço. Tendo em vista que uma tarefa sempre inicia e termina no porto, o tempo de *setup* ou deslocamento entre duas tarefas consecutivas é zero.

Outro aspecto de fundamental importância é a data mais cedo em que a tarefa poderá ser iniciada. Esta data varia de acordo com as datas de disponibilização de material nos portos, e conforme as concessões de licenças emitidas por setores internos e externos à empresa.

Considerando os aspectos acima apontados, o problema a ser resolvido passa a ser a determinação da sequência das tarefas a serem realizadas pelas embarcações, de forma que a penalização econômico-financeira global seja minimizada. Impõe-se ainda que toda a demanda seja atendida dentro do horizonte de planejamento estipulado. A sequência das tarefas pode ser indicada pelo instante de início de cada tarefa em uma única embarcação.

O problema descrito é de interesse prático por ser um problema real que afeta o desempenho e a continuidade das atividades de preparação dos poços para etapa produtiva, e de interesse acadêmico pela necessidade de resolução de problemas de escala real.

3. MODELAGEM MATEMÁTICA

Com base na descrição apresentada, o problema em questão será resolvido tendo como base as seguintes premissas: i) A demanda é totalmente conhecida *a priori*; ii) Cada tarefa possui um tempo específico de execução; iii) Cada tarefa possui uma data de liberação, antes da qual não é permitido o seu início; iv) Cada tarefa tem uma data de entrega prevista; v) Tarefas finalizadas após a data de entrega prevista estarão sujeitas a uma penalização que incide para cada dia de atraso, acumulando de forma linear na função objetivo; vi) Uma tarefa, após ser iniciada, não é interrompida; vii) Cada tarefa é realizada por apenas uma única embarcação; viii) Uma embarcação não pode executar mais de uma tarefa ao mesmo tempo; ix) As tarefas consideradas neste trabalho são independentes entre si, não existindo quaisquer restrições de precedência entre as mesmas; x) A frota é homogênea, e qualquer embarcação está apta para realizar todas as tarefas; xi) O tempo de deslocamento de uma embarcação entre tarefas é nulo, uma vez que a tarefa inicia e termina em um porto.

A formulação adotada para representar o problema é indexada ao tempo, adaptada de van den Akker et al. [4], na qual o tempo é dividido em períodos, de forma que o período t inicia no instante $t-1$ e termina no instante t , e compreende T períodos.

Os seguintes parâmetros foram usados: N - conjunto de tarefas; E - conjunto de embarcações; T - horizonte de planejamento; m - número de embarcações; n - número de tarefas; c_j^t - custo de oportunidade incorrido quando a tarefa $j \in N$ é iniciada na data t ; p_j - duração da tarefa $j \in N$; r_j - data a partir da qual a tarefa $j \in N$ pode ser iniciada. A variável de decisão é x_j^t , a qual será 1 se a tarefa j iniciar na data t , e 0, em caso contrário.

O modelo matemático fica definido nos seguintes termos:

$$\min C = \sum_{j=1}^n \sum_{t=r_j}^{T-p_j+1} c_j^t x_j^t \quad (1)$$

Sujeito a:

$$\sum_{t=r_j}^{T-p_j+1} x_j^t = 1 \quad j \in N \quad (2)$$

$$\sum_{j=1}^n \sum_{s=\max(r_j, t-p_j+1)}^{\min(t, T-p_j+1)} x_j^s \leq m \quad \forall t \quad (3)$$

$$x_j^t \in \{0,1\} \quad j \in N \quad (4)$$

A função objetivo (1) representa o custo total de oportunidade incorrido devido ao início das tarefas nos respectivos instantes, de acordo com as variáveis de decisão x_j^t e a

matriz de custos de oportunidade c_j^f . A restrição (2) garante que todas as tarefas sejam iniciadas uma única vez. A restrição (3) garante que o número de tarefas sendo executadas em um dado momento não ultrapasse o número de embarcações disponível. Por fim, a restrição (4) indica que as variáveis de decisão x_j^f são binárias.

4. MÉTODO DE SOLUÇÃO

O método de solução proposto consiste, em primeiro lugar, gerar uma solução inicial com o suporte de uma heurística construtiva. Esta heurística se baseia em uma abordagem intuitiva, visando explorar de forma apropriada as características do problema. A heurística implementada atribui prioridades às tarefas demandadas, de forma que as mesmas possam ser ordenadas em uma fila. Uma a uma, as tarefas que têm maior prioridade são alocadas na embarcação que está disponível mais cedo, até que todas as tarefas estejam alocadas.

A solução gerada por este procedimento não só constitui uma solução factível para o problema, como o respectivo valor da função objetivo é um parâmetro de entrada importante no método da relaxação lagrangeana, a qual executa duas funções importantes dentro do método de solução proposto, a saber, o de prover um *lower bound* para o valor da função objetivo, e o de fornecer vizinhanças novas, a cada iteração, para alimentar a heurística de busca local.

A heurística de busca local foi estruturada a partir de uma adaptação do método VNS (*variable neighborhood search*), descrito em [2]. A geração de vizinhanças é feita a partir de uma heurística que transforma as soluções inviáveis, geradas pelo método da relaxação lagrangeana, em soluções viáveis. Os métodos de busca local são então, aplicados sobre essas soluções viáveis “adaptadas”, fornecidas a cada iteração.

4.1. HEURÍSTICA CONSTRUTIVA

A heurística construtiva proposta é composta de uma heurística gulosa que funciona de forma recursiva, aliada a um método de busca local que visa aprimorar a solução construída antes de iniciar a etapa da relaxação lagrangeana.

Na heurística gulosa, a cada iteração uma tarefa é selecionada e alocada a uma embarcação. Para definir qual tarefa será alocada a qual embarcação em uma determinada iteração, são utilizados alguns critérios que dependem das características da tarefa (data de liberação, data de entrega, penalidade por atraso, duração da tarefa).

O motivo de a heurística ser caracterizada como gulosa é que uma vez que uma tarefa foi escolhida e inserida em uma embarcação, a mesma não poderá ser transferida de embarcação ou de posição, independentemente das características das tarefas restantes que ainda não foram inseridas. Cada alocação em uma embarcação é feita de forma que a tarefa inserida seja executada depois das tarefas alocadas anteriormente na mesma embarcação.

Para definir qual tarefa será alocada a uma embarcação, a cada iteração são calculados os seguintes parâmetros: i) **Penalização Marginal**: Calcula-se qual o custo em que se incorre caso o início da tarefa seja postergado em 1 (um) intervalo de tempo em relação ao instante mais cedo disponível entre todas as embarcações. O instante disponível das embarcações é o instante seguinte à conclusão de todas as tarefas associadas àquela embarcação. Tarefas que ainda não foram liberadas para início, assumem penalização marginal = 0, para efeito de cálculo de prioridade; ii) **Folga**: Calcula-se a quantidade de instantes de tempo que separam o hipotético final da tarefa, caso ela seja alocada à embarcação que está disponível mais cedo, e a data de entrega desta tarefa. Tarefas que ainda não foram liberadas para início, assumem folga infinita para efeito de cálculo de prioridade.

Após o cálculo dos parâmetros acima explicitados, a próxima tarefa a ser alocada em uma dada iteração é definida de acordo com os critérios de maior penalização marginal, menor folga, maior peso por atraso, menor duração da tarefa. A escolha é feita de forma que, caso tenha-se empate no primeiro critério, a escolha é feita pelo segundo critério. Caso o

empate persista, a tarefa é escolhida de acordo com o próximo critério e assim sucessivamente até o final de todos os critérios. Caso as tarefas tenham todos os critérios iguais, a escolha é feita pela primeira tarefa avaliada. Quanto à escolha da embarcação, será priorizada aquela que possuir o instante mais cedo de disponibilidade.

O procedimento descrito acima é repetido até que todas as tarefas demandadas estejam alocadas às embarcações. Com todas as tarefas alocadas, gera-se a primeira solução para o problema, que pode ser viável ou não, já que na heurística gulosa é possível que a programação proposta não esteja respeitando o horizonte de planejamento do problema.

Nos casos em que não for possível gerar uma solução viável, é aplicado um segundo procedimento tentativo, visando a geração de soluções que respeitem o horizonte de planejamento. Para isso foram utilizadas idéias intuitivas, comumente aplicadas em problemas de minimização de tempo total (“makespan”), para processamento de uma lista de tarefas. O procedimento consiste em listar as tarefas de acordo com a ordem decrescente dos tempos de processamento e em seguida alocá-las uma a uma na embarcação que estiver disponível mais cedo. Conforme explicado em De e Morton [5], o uso desta técnica balanceia o tempo de processamento nas embarcações, e minimiza o efeito das últimas alocações já que estas têm menor duração.

Na fase final da heurística construtiva, é aplicada uma busca local à solução fornecida pela heurística gulosa visando garantir que esta solução, além de viável, tenha uma boa qualidade. Este valor será o limitante superior da relaxação lagrangeana. O procedimento de busca local é o mesmo empregado na busca da solução gerada no âmbito da relaxação lagrangeana, e será detalhado na seção 4.3.

4.2. RELAXAÇÃO LAGRANGEANA

Quanto ao problema em questão, representado pelo modelo descrito na seção 3, pode-se observar que a solução do mesmo é facilitada quando a restrição de capacidade (3) é relaxada. Na ausência de tal restrição, todas as tarefas podem ser programadas sem incorrer em atraso, sendo simples a proposição de uma solução ótima para o problema.

Considerando, contudo, que é de interesse prático uma solução que respeite a disponibilidade de embarcações, a violação da capacidade será penalizada na função objetivo por meio dos multiplicadores de lagrange. Um novo problema passará a ser a determinação do valor ótimo desses multiplicadores.

Para resolver o problema relaxado será proposta uma estratégia de solução, na qual o problema é decomposto em subproblemas menores, de fácil solução, sendo um subproblema por tarefa. Esta abordagem de decomposição é favorecida pelo caráter aditivo da função objetivo e da restrição de capacidade.

A atualização dos multiplicadores de lagrange foi feita por meio do algoritmo do subgradiente, adaptado de Luh et al. [6].

4.2.1. Problema Relaxado (R)

Relaxando a restrição de capacidade (3) e penalizando-a na função objetivo com o uso dos multiplicadores de lagrange π^t , o problema relaxado R adquire a seguinte estrutura:

$$\begin{aligned}
 R(\pi) &= \sum_{j=1}^n \sum_{t=r_j}^{T-p_j+1} c_j^t x_j^t + \sum_{t=1}^T \pi^t \left(\sum_{\substack{j=1: s=\max(r_j, t-p_j+1) \\ t \geq r_j}}^n \sum_{s=\max(r_j, t-p_j+1)}^{\min(t, T-p_j+1)} x_j^s - m \right) = \\
 &= - \sum_{t=1}^T \pi^t \cdot m + \sum_{j=1}^n \sum_{t=r_j}^{T-p_j+1} c_j^t x_j^t + \min \left(\sum_{t=1}^T \sum_{\substack{j=1: s=\max(r_j, t-p_j+1) \\ t \geq r_j}}^n \sum_{s=\max(r_j, t-p_j+1)}^{\min(t, T-p_j+1)} \pi^t \cdot x_j^s \right) \quad (5)
 \end{aligned}$$

Sujeito a: (2), (4) e

$$\pi^t \geq 0 \quad \forall t \quad (6)$$

4.2.2. Problema Dual (D)

Considerando que a penalização introduzida à função objetivo é multiplicada por um fator maior ou igual a zero, o valor da função objetivo do problema relaxado sempre será menor ou igual a função objetivo do problema original (1). Dado que um limitante inferior para a função objetivo é dado por $R(\pi)$, a função dual está então maximizada, no espaço de solução das variáveis π^t ($t: 1, \dots, T$), dando origem ao problema dual D .

$$D = \max_{\pi \geq 0} R(\pi) = \max_{\pi \geq 0} \left\{ \min \left(- \sum_{t=1}^T \pi^t m + \sum_{j=1}^n \sum_{t=r_j}^{T-p_j+1} c_j^t x_j^t + \sum_{t=1}^T \sum_{\substack{j=1: \\ t \geq r_j}}^n \sum_{s=\max(r_j, t-p_j+1)}^{\min(t, T-p_j+1)} \pi^t x_j^s \right) \right\} \quad (7)$$

Sujeito a: (2) e (4)

4.2.3. Programação das Tarefas

Em primeiro lugar, a expressão (7) será reescrita, decompondo a função objetivo, em termos independentes, por tarefa.

$$D = \max_{\pi} \left\{ \sum_{j=1}^N \min D(j, \pi) - \sum_{t=1}^T \pi^t m \right\} \quad (8)$$

Em que:

$$D(j, \pi) = \sum_{t=r_j}^{T-p_j+1} c_j^t x_j^t + \sum_{t=r_j}^T \sum_{s=\max(r_j, t-p_j+1)}^{\min(t, T-p_j+1)} x_j^s \pi^t \quad (9)$$

Para um dado valor de π^t ($t: 1, \dots, T$), a resolução de (8) pode ser feita resolvendo-se um subproblema (9) por tarefa. Cabe ressaltar que o segundo termo de (8), a saber: $-\sum_{t=1}^T \pi^t m$, é constante. Dado que a função $D(j, \pi)$ deve ser minimizada, para que o instante de início de uma tarefa seja determinado, é necessário avaliar, dentre as datas de início $r_j \leq t \leq T - p_j + 1$, aquela que minimiza a função $D(j, \pi)$, o que pode ser feito por comparação explícita de todas as alternativas.

4.2.4. Resolução do Problema Dual

Para resolver o problema dual, no espaço de solução das variáveis π , será aplicado o método do subgradiente, o qual atualizará o valor de π , de acordo com a expressão (10).

$$\pi^{n+1} = [\pi^n + \alpha^n g^n(\pi^n)]^+ \quad (10)$$

Em que:

n – número da iteração

g^n – subgradiente de D para um dado π , cujo t -ésimo componente é dado por:

$$\left(\sum_{\substack{j=1: \\ t \geq r_j}}^n \sum_{s=\max(r_j, t-p_j+1)}^{\min(t, T-p_j+1)} x_j^s \right) - m$$

α^n passo na interação n , dado por:

$$\alpha^n = \lambda \frac{\bar{L} - L^n}{g^n(\pi^n)^T g^n(\pi^n)} \quad 0 < \lambda < 2$$

\bar{L} é o limitante superior da função objetivo do problema original, calculado pela solução heurística no início do algoritmo.

L^n é o limitante inferior da função objetivo do problema original, na iteração n , cujo valor é dado por

$$\min \left(- \sum_{t=1}^T \pi^t m + \sum_{j=1}^n D(j, \pi) \right)$$

λ é um fator de ajuste do tamanho do passo, que será reduzido pela metade cada vez que a iteração não melhorar o limitante inferior por um número consecutivo de vezes. Esse número é um parâmetro que influi no critério de parada e, portanto deverá ser calibrado apropriadamente. Pode ser interessante, em termos de desempenho do algoritmo, que esse número varie de acordo com o tamanho do passo ou outro fator relevante. Por último, o operador $[\phi]^+$ irá retornar $\max(0, \phi)$.

4.3. GERAÇÃO DE SOLUÇÕES VIÁVEIS & BUSCA LOCAL

O método de solução desenvolvido nesta pesquisa tem como proposta explorar a vizinhança das soluções duais, geradas pela heurística lagrangeana, visando encontrar soluções de boa qualidade no espaço de solução do problema original. Tendo em vista que grande parte das soluções duais é inviável, foi proposto um algoritmo que transforma a solução inviável em viável e, em seguida, a busca local é realizada.

No caso de uma solução viável estar localizada em uma região distante da solução ótima, o algoritmo de busca poderá gastar muito tempo de processamento para encontrar regiões mais promissoras. Em função disso, foi proposto um critério mediante o qual a solução dual será ou não convertida em solução primal e, posteriormente, submetida à busca local. Este consiste comparar o valor da função objetivo da solução dual que acabou de ser gerada com o valor da função objetivo da solução ótima estimada até o momento (máximo valor da função objetivo de todas as soluções duais geradas até o momento). Caso valor da função objetivo da solução dual, dividido pelo valor da função objetivo da estimativa do ótimo seja maior do que α , a busca local é executada. Onde o parâmetro α , um valor que varia de 0 a 1, pode ser calibrado de forma a evitar grande esforço computacional com soluções duais muito distantes da região viável.

4.3.1. Geração de Solução Viável

O algoritmo usado na geração de soluções viáveis foi criado pensando na estrutura do problema relaxado e nas características das soluções obtidas. O problema original, resolvido pelo método da relaxação lagrangeana, tinha apenas um tipo de restrição violada, a saber, a restrição de capacidade. Assim sendo, as soluções geradas pela resolução do problema relaxado atendiam a restrição de atendimento integral da demanda sem necessariamente respeitar a capacidade das embarcações ao longo do horizonte de planejamento. A heurística que transforma uma possível solução inviável em viável identifica os instantes em que a restrição de capacidade é violada, e posterga o início de uma ou mais tarefas realizadas nestes instantes. O deslocamento das tarefas é feito percorrendo a linha do tempo até que em nenhum momento a restrição de capacidade do problema original fosse violada.

Os critérios utilizados para definir quais tarefas terão seu início postergado são dois. O primeiro é a data de início da tarefa sugerido na resolução do problema relaxado, e o segundo é baseado no custo associado a postergar o início da tarefa em um dia. Caso as tarefas tenham a mesma data de início sugerida, a tarefa que manterá o início da execução será aquela que tiver a maior penalidade associada a atrasar em um dia o seu início.

É importante observar que as soluções do problema relaxado podem ser inviáveis ao não respeitar a restrição de capacidade das embarcações. Por isso, não é necessário verificar se as tarefas estão sendo iniciadas antes da data de liberação das mesmas, fazendo com que este parâmetro de liberação não seja levado em conta no algoritmo que gera soluções viáveis a partir das soluções do problema relaxado.

4.3.2. Busca Local

A busca local implementada consistiu em uma adaptação do “variable neighborhood search” (VNS). O VNS é uma meta-heurística que se baseia na exploração de diferentes estruturas de vizinhanças a partir de uma solução inicial, visando encontrar soluções de melhor qualidade até atingir o ponto ótimo. O VNS empregado nesta pesquisa utiliza diferentes estruturas de vizinhança, o que não restringe a busca a uma região específica da vizinhança que se pretende explorar. Quanto maior for o número de soluções que se consiga gerar e avaliar, a partir das diferentes estruturas de vizinhanças, menor é a probabilidade do algoritmo ficar preso a um possível ótimo local encontrado durante a execução dos procedimentos de busca.

Nesta pesquisa optou-se pela construção de três estruturas de vizinhança distintas. A primeira delas se refere ao reposicionamento de uma única tarefa, determinada por sorteio, e alocada na posição que minimiza a função objetivo. A segunda estrutura de busca, mais complexa do ponto de vista computacional, baseia-se no reposicionamento simultâneo de duas tarefas sorteadas aleatoriamente, de forma a melhorar a função objetivo. A última busca compara a troca de posição entre todos os pares possíveis de tarefas avaliando qual a melhor troca.

Para aumentar a eficiência do ponto de vista computacional, é necessário iniciar pelas buscas mais simples, a fim de encontrar soluções de boa qualidade sem grande necessidade de processamento. A partir do momento em que as buscas simples não surtam mais efeito, parte-se para o uso das buscas mais complexas que verificam a qualidade de soluções mais afastadas da solução que está sendo examinada.

A primeira estrutura de vizinhança, a busca local 1 (BL1), escolhe uma tarefa qualquer do problema, de forma aleatória, e retira essa tarefa da embarcação em que ela está alocada. Em seguida inicia-se a rotina de colocar a tarefa em todas as posições possíveis de todas as embarcações disponíveis, a fim de determinar a configuração que implica em menor valor da função objetivo. Caso o custo dessa configuração seja menor do que a solução que está sendo analisada, é considerado que a busca local teve sucesso.

A segunda estrutura de vizinhança proposta, a busca local 2 (BL2), seleciona duas tarefas de maneira aleatória, observando a condição de que estas tarefas devam estar alocadas, originalmente, em embarcações diferentes. As duas tarefas sorteadas são retiradas das embarcações em que se encontravam, gerando uma configuração denominada de alocação inicial (a menos das 2 tarefas excluídas). Para definir qual a melhor forma de alocar as tarefas, parte-se da alocação inicial procurando qual a melhor posição para recolocar a primeira tarefa que foi sorteada (s1). Definida a posição de inserção da tarefa s1 que minimiza a função objetivo, faz-se a inserção da mesma. Em seguida, verifica-se qual a melhor posição para inserir a segunda tarefa sorteada (s2). Ao encontrar a melhor posição para inserir a tarefa s2, obtém-se uma solução viável para o problema. Caso a solução viável encontrada forneça uma função objetivo menor que a melhor função objetivo encontrada até o momento, atualiza-se a melhor solução viável com a solução encontrada e considera-se que a busca local teve sucesso.

A Busca Local de recolocação de duas tarefas, contudo, ainda não é considerada concluída. Partindo-se da alocação original, inicia-se agora a alocação das duas tarefas sorteadas pela tarefa s2 para, depois de definida a sua melhor posição de inserção avaliar a melhor posição para inserção da tarefa s1. Obtém-se, ao final dessas duas alocações otimizadas, uma solução viável que, caso tenha o valor da função objetivo menor que a solução corrente, caracteriza a busca local como um sucesso, atualizando a melhor solução encontrada até o momento.

A última estrutura de busca local implementada, a busca local 3 (BL3), embora de fácil compreensão, implica em grande esforço computacional. Nesta busca, analisa-se o valor da função objetivo para todas as trocas de pares de tarefas, independentemente de estarem na

mesma embarcação ou não, ou seja, inverte-se a posição do par de tarefas escolhido.

4.4. RESUMO DO PROCEDIMENTO

Para facilitar a compreensão da metodologia apresentada, foi proposto o fluxograma mostrado na Figura 1. Pode ser visto, neste fluxograma, como estão relacionadas a heurística construtiva, a relaxação lagrangeana, o método de busca local, além de parâmetros como o multiplicador do passo (utilizado na relaxação lagrangeana), o contador de iterações, usado como um dos critérios de parada e a atualização da melhor solução viável e da estimativa da solução viável para o problema.

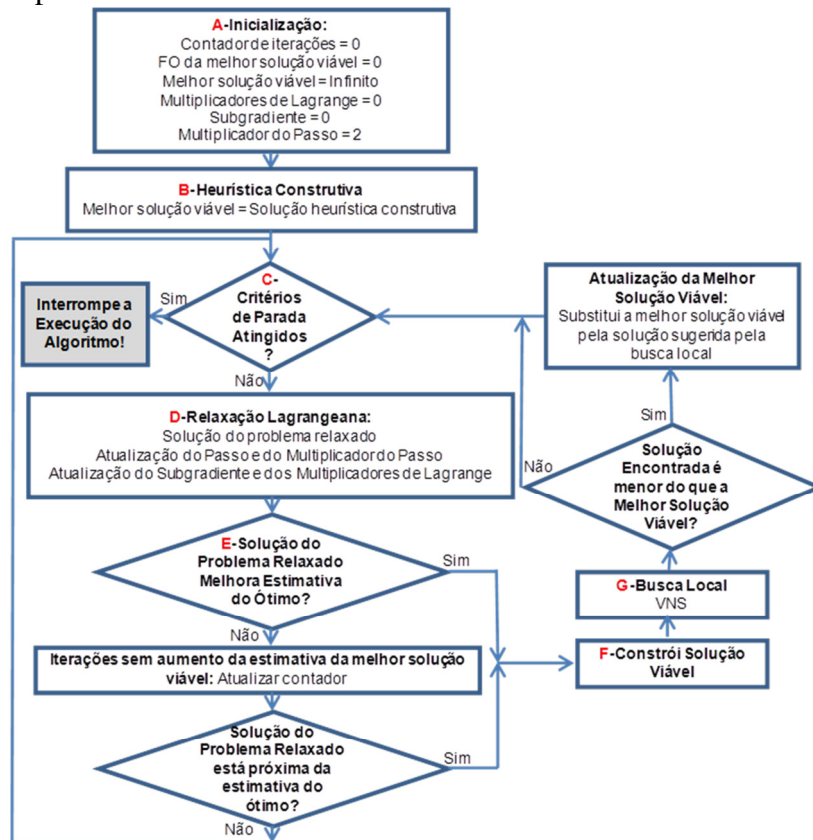


Figura 1 – Procedimento de Solução

5. TESTES COMPUTACIONAIS

Foram geradas instâncias de testes para avaliar o desempenho do método de solução proposto. A metodologia utilizada para geração das instâncias foi adaptada do trabalho de Jouglet e Savourey [7]. O número de tarefas (n) variou da seguinte forma: n = 20, 30, 50 e 100. Além de variar o número de tarefas nos problemas, variou-se a quantidade de embarcações (m): m = 4, 5, 6, 8, 10.

Os parâmetros associados às tarefas, em cada cenário, foram gerados de forma aleatória, em função dos fatores α (parâmetro de liberação de tarefas no instante 0), β (parâmetro de folga, que afeta a quantidade de datas disponíveis para início da tarefa sem incorrer em penalidade), da seguinte forma: **Duração máxima** (D_m): Este parâmetro é calculado em função do número de embarcações, do número de tarefas e do horizonte de planejamento, e é utilizado para estimar a máxima duração das tarefas de forma a evitar a geração de instâncias inviáveis. Adotou-se a seguinte fórmula: $D_m = \frac{m \cdot \alpha \cdot \beta}{n}$; **Duração das tarefas** (p_i): número aleatório inteiro no intervalo que varia de 1 à duração máxima (D_m) da instância; **Penalidade por atraso na finalização da tarefa** (w_i): Distribuição uniforme variando de 1 a 10; **Data de liberação das tarefas** (r_i) é calculado, tarefa a tarefa, da seguinte

forma: o parâmetro α define a probabilidade da tarefa ser liberada no instante 0. Caso a tarefa não seja liberada no instante 0 ela será liberada em um instante definido por uma número aleatório inteiro entre os datas 0 e $h_p - p_i$. **Data de entrega das tarefas (D_i):** Soma da data de liberação (r_i), com a duração da tarefa (p_i), com um número aleatório inteiro entre 0 e a duração da tarefa (p_i) multiplicado pelo fator de folga β , ou seja: $D_i = r_i + p_i + aleat(0; p_i \times \beta)$.

A quantidade de instâncias testadas, variando os parâmetros supracitados, foi de 240. Os resultados obtidos não serão apresentados individualmente, mas agrupados de acordo com os parâmetros que definiram a estrutura de cada um, e indicados na tabela 1.

Tabela 1 - Resultado Médio dos Cenários

Cenário	Número de Instâncias	Tarefas	Embarcações	Fator de Folga	Fator de Liberação	GAP Médio	GAP + Médio	GAP - Médio	GAP 0 Médio	Número de Iterações Médio	Tempo de Processamento Médio (s)
1	10	20	4	0%	90%	0,02%	0,01%	0,01%	2,55%	339	11
2	10	20	4	50%	90%	0,05%	0,01%	0,04%	6,66%	383	13
3	10	20	4	100%	90%	0,49%	0,22%	0,27%	7,04%	610	19
4	10	20	5	0%	90%	0,01%	0,00%	0,01%	2,82%	361	12
5	10	20	5	50%	90%	0,11%	0,02%	0,09%	6,78%	603	18
6	10	20	5	100%	90%	0,30%	0,02%	0,28%	16,01%	567	17
7	10	30	5	0%	90%	0,07%	0,07%	0,01%	3,50%	666	41
8	10	30	5	50%	90%	0,01%	0,01%	0,00%	2,80%	588	31
9	10	30	5	100%	90%	0,05%	0,01%	0,04%	4,31%	509	30
10	10	30	6	0%	90%	0,01%	0,00%	0,00%	2,87%	513	27
11	10	30	6	50%	90%	0,00%	0,00%	0,00%	5,58%	449	24
12	10	30	6	100%	90%	0,61%	0,30%	0,31%	10,04%	768	44
13	10	50	6	0%	90%	0,02%	0,02%	0,00%	2,21%	765	797
14	10	50	6	50%	90%	0,05%	0,04%	0,01%	4,24%	804	895
15	10	50	6	100%	90%	0,09%	0,07%	0,03%	4,21%	1.008	1.064
16	10	50	8	0%	90%	0,02%	0,02%	0,00%	3,16%	763	771
17	10	50	8	50%	90%	0,06%	0,05%	0,01%	3,95%	769	775
18	10	50	8	100%	90%	0,17%	0,14%	0,03%	7,74%	901	890
19	10	100	8	0%	90%	0,06%	0,06%	0,00%	1,82%	1.073	597
20	10	100	8	50%	90%	0,06%	0,06%	0,00%	2,36%	1.078	920
21	10	100	8	100%	90%	0,06%	0,06%	0,00%	1,75%	1.061	3.025
22	10	100	10	0%	90%	0,06%	0,06%	0,00%	2,10%	1.090	2.980
23	10	100	10	50%	90%	0,06%	0,06%	0,00%	2,29%	1.072	2.823
24	10	100	10	100%	90%	0,12%	0,12%	0,01%	3,26%	1.054	2.905

Foram avaliados os GAPs, agrupando os cenários de acordo com o seu porte, ou seja, a quantidade de tarefas e a quantidade de embarcações disponíveis, como podem ser vistos na tabela 2. Como era de se esperar, o aumento do número de tarefas dificulta a resolução do problema. Isso pode ser constatado pelo aumento tanto do número de iterações média como do tempo médio para resolução das instâncias conforme aumenta o número de tarefas. Em relação à qualidade dos limitantes, aparentemente não há relação com o número de tarefas das instâncias. Independentemente do porte do cenário, o valor médio do GAP variou de 0,0% a 0,2%.

Observando os resultados das tabelas 1 e 2, pode-se notar que os indicadores propostos variam bastante dentro de cenário de mesmo porte (igual número de embarcações e tarefas). Neste caso, a avaliação exclusiva dos cenários agrupados por número de tarefas e número de embarcações não leva em conta todos os aspectos importantes para determinar a dificuldade dos problemas resolvidos. Para analisar o motivo das diferenças entre cenários de mesmo porte, serão agrupados os resultados de acordo com o parâmetro de dispersão β , lembrando que esse parâmetro representa a dispersão da quantidade de datas disponível para início das tarefas sem que as mesmas incorram em penalidade.

Tabela 2 - Resultados agrupados por porte do cenário

Cenários	Número de Instâncias	Tarefas	Embarcações	GAP Médio	GAP Máximo	GAP Mínimo	Número de Iterações Médio	Tempo de Processamento Médio (s)
1 a 3	30	20	4	0,19%	2,17%	0,00%	446	14
3 a 6	30	20	5	0,14%	2,52%	0,00%	507	16
7 a 9	30	30	5	0,04%	0,48%	0,00%	585	34
10 a 12	30	30	6	0,21%	3,41%	0,00%	577	32
13 a 15	30	50	6	0,06%	0,19%	0,00%	859	919
16 a 18	30	50	8	0,08%	0,63%	0,00%	811	812
19 a 21	30	100	8	0,06%	0,25%	0,00%	1.071	1.514
22 a 24	30	100	10	0,08%	0,30%	0,00%	1.072	2.903

A tabela 3 agrupa os resultados de acordo com o parâmetro β . Em relação ao parâmetro α , que impacta na dispersão das datas de liberação, o mesmo foi mantido com o valor constante de 90% para todas as instâncias das tabelas de resultados aqui publicadas. Isto foi feito visando representar problemas reais de programação de frota de embarcações de lançamentos de dutos.

Tabela 3 - Resultados agrupados pelo parâmetro (β)

Número de Instâncias	Fator de Folga	GAP Médio	GAP Máximo	GAP Mínimo	Número de Iterações Médio	Tempo de Processamento Médio (s)
80	0%	0,03%	0,48%	0,00%	697	662
80	50%	0,05%	0,47%	0,00%	724	705
80	100%	0,24%	3,41%	0,00%	810	999

Conforme pode ser observado na tabela acima, o aumento do fator de folga, e o consequente aumento das datas disponíveis para início das tarefas sem que se incorra em penalização é um fator que dificulta a resolução do problema. Os indicadores GAP Médio, GAP Máximo, Número de Iterações e Tempo aumentam conforme o aumento do parâmetro β , ou seja, é esperado que a distância dos limitantes inferior e superior aumente, assim como o esforço computacional necessário, com o aumento de β .

6. CONCLUSÕES

Foi estudado o problema de programação de embarcações PLSVs, utilizadas na operação de apoio marítimo de lançamento de dutos submarinos. Embora o problema tenha sido enquadrado em uma classe de problemas conhecido, o presente trabalho contribuiu ao desenvolver um procedimento de solução que envolve a exploração de soluções duais obtidas por meio da relaxação lagrangeana, e integrando diferentes métodos, de forma a prover estimadores da solução ótima (*lower bounds*) e soluções primais (*upper bounds*) concomitantemente.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] CHRISTIANSEN, M.; FAGERHOLT, K.; RONEN, D. Ship routing and scheduling: Status and Perspectives. **Transportation Science**, 38:1, p.1-18, 2004.
- [2] HANSEN, P., & MLADENOVIC, N. A tutorial on variable neighbourhood search. *Le cahiers du GERAD G-2003-46*, 2003.
- [3] WOLSEY, L. **Integer programming**. John Wiley & Sons, Inc., 266p.,1998.
- [4] AKKER J. M. Time Indexed Formulation for Machine Scheduling Problems: Column Generation, **Journal on Computing**, 12:2, 2000.

- [5] DE, P.; MORTON, T. E. Scheduling to Minimize Makespan on Unequal Parallel Processors. **Decision Science**,11, p.586-602, 1980.
- [6] LUH, P.B.; HOITOMT, D.J.; MAX E.; PATTIPATI K.R. Schedule Generation and Reconfiguration for Parallel Machines. **IEEE Transactions on Robotics and Automation**, 6:6, p.687-696, 1990.
- [7] JOUGLET, A.; SAVOUREY, D. Dominance rules for the parallel machine total weighted tardiness scheduling problem with release dates. **Computers & Operations Research**, 38, p.1259-1266, 2011.